

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

DIPLOMOVÁ PRÁCE

Využití metod zpracování obrazu pro
určování mechanických vlastností polymerů

Plzeň, 2012

Jan Vlk

PROHLÁŠENÍ:

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni. Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 18. května 2012

.....

Jan Vlk

Poděkování

Rád bych tímto poděkoval panu Ing. Miroslavu Jiříkovi za vedení mé diplomové práce a své rodině za pochopení a podporu při práci.

Abstrakt

Předložená práce se zabývá hodnocením metod extrakce příznaků z obrazových dat a dále testováním klasifikačních algoritmů. Testovaná data jsou svým charakterem textury. Původní motivací napsání této práce byla potřeba klasifikovat jednotlivé typy mikroskopických struktur polymerových skel a následné určení jejich fyzikálních vlastností. Problém texturní klasifikace polymerů můžeme samozřejmě zobecnit na různé typy obrazových vstupů.

Klíčová slova:

polymer, textura, local binary patterns, matice sousednosti, GLRL, geometrické momenty Huovy momenty, Bayesův klasifikátor, support vector machine, klasifikátor dle k-nejbližšího souseda

Abstract

The goal of the thesis is evaluation of feature extraction and classification algorithms for texture recognition. First motivation for the thesis writing was physical feature recognition of polymer glasses through image processing. This problem can be generalized about simple texture recognition for different types of images not only for pictures of polymers.

Keywords:

polymer, texture, local binary patterns, coocurance matrix, GLRL, geometric moments, Hu's moments, Bayesian classifier, support vector machine, nearest neighbor classifier

Obsah

1 Úvod	5
1.1 Motivace a cíle práce	6
2 Metody extrakce příznaků	7
2.1 Statistické metody	7
2.1.1 Statistiky prvního řádu	7
2.1.2 Statistiky druhého řádu	9
2.1.3 Gray level run length	12
2.1.4 Geometrické momenty	13
2.1.5 Huovy momenty	14
2.2 Local binary patterns	15
2.2.1 Rotační invariance a Uniform patterns	16
2.2.2 Modifikace metody LBP	16
3 Klasifikační algoritmy	18
3.1 Bayesův klasifikátor	18
3.1.1 Odvození diskriminační funkce	18
3.1.2 Trénování Bayesova klasifikátoru	21
3.2 Klasifikace dle k-nejbližšího souseda	22
3.3 Support vector machine	22
3.3.1 Lineární SVM	22
3.3.2 Nelineární SVM	25
3.3.3 Klasifikace do více tříd	26
3.4 Metody evaluace výkonu klasifikačních algoritmů	27

3.4.1	Procentuální úspěšnost a doba klasifikace	27
3.4.2	Confusion matrix	27
4	Implementace testovacího systému	30
4.1	Předzpracování dat	30
4.2	Struktura a konfigurace testovaných metod	31
5	Výsledky a jejich interpretace	34
5.1	Testy mikroskopických snímků polymerů	34
5.2	Testy na texturách z galerie Brodatz	40
5.3	Shrnutí dosažených výsledků	45
6	Závěr	47
A	Přílohy	49

Seznam obrázků

2.1	Ukázka šedotónového obrázku a jeho histogramu	8
2.2	Textury se stejným histogramem	9
2.3	Výpočet jednoho pixelu LBP-obrazu	15
2.4	Původní obraz a LBP-obraz	15
2.5	Různé typy LBP filtrů [6]	17
3.1	Schéma Bayesova klasifikátoru	21
3.2	SVM klasifikátor s lineární jádrovou funkcí	23
3.3	Ukázka mapování z 1D prostoru do 2D prostoru	25
4.1	Ukázka původního obrazu a jeho výřezu	31
5.1	Mikroskopické snímky polymerů	34
5.2	4.třída polymerů	36
5.3	6.třída polymerů	36
5.4	Vliv počtu nejbližších sousedů na přesnost klasifikace (polymery)	38
5.5	Vliv parametru σ na přesnost klasifikace (polymery)	39
5.6	Textury z galerie Brodatz	40
5.7	rákos Brodatz	42
5.8	ratan Brodatz	42
5.9	cihly Brodatz	43
5.10	ratan Brodatz	43
5.11	Vliv počtu nejbližších sousedů na přesnost klasifikace (Brodatz)	44
5.12	Vliv parametru σ na přesnost klasifikace (Brodatz)	45

Seznam tabulek

2.1	Ukázka dvou různých matic sousednosti	10
3.1	Matematická vsuvka	20
3.2	Confusion matrix	28
5.1	Vliv LBP-filtru na úspěšnost klasifikace[%]	35
5.2	Vliv nastavení offsetu matice sousednosti na úspěšnost klasifikace (polymery)	37
5.3	Celkový přehled úspěšnosti a časů klasifikace (polymery)	39
5.4	Vliv LBP-filtru na úspěšnost klasifikace (Brodatz)	40
5.5	Vliv nastavení offsetu matice sousednosti na úspěšnost klasifikace	42
5.6	Celkový přehled úspěšnosti a časů klasifikace (Brodatz)	45

1 Úvod

Zkoumání povrchové struktury různých materiálů pomocí metod počítačového vidění se může stát rychlým a levným nástrojem jejich analýzy. Důvodů pro velmi časté nasazování inteligentních snímacích zařízení v průmyslu je hned několik. Např. eliminace lidského faktoru z výroby kvůli zajištění bezpečnosti, či kvůli snížení nákladů na pracovní sílu. Dnes nás díky novým technologiím neomezuje velikost zpracovávaných dat potřebných k získání informace z obrazu, jako tomu bylo v minulosti. Výhodou texturní analýzy obrazu (příznakové či strukturální popisy textur, klasifikace, segmentace) je, že zkoumá právě povrchovou strukturu objektů pozorování, která vypovídá např. o vlastnostech objektu.

Aplikaci tohoto odvětví počítačového vidění můžeme realizovat v biologii, strojírenství, materiálovém inženýrství, medicíně , atd.

Textura, jak bylo naznačeno výše, vypovídá o vlastnostech povrchu objektů, či jejich struktuře. Pod tímto pojmem si můžeme také představit vizuální vzor v obraze nebo opakování určitých základních prvků, jež nazýváme primitiva. V závislosti na velikosti těchto primitiv pak můžeme dělit textury na jemné (malá primitiva), hrubé (velká primitiva), zrnité (středně velká primitiva), atd. Tento popis textur je v zásadě velmi intuitivní a pro použití v algoritmech rozpoznávání textur nepoužitelný. Proto existuje, a také stále vzniká mnoho matematických nástrojů, které mají za úkol popsat textury způsobem“ použitelným pro algoritmy počítačového vidění[1].

1.1 Motivace a cíle práce

Motivací k napsání této práce bylo zkoumání fyzikálních vlastností (tvrdost, světelná propustnost, odrazivost, atd.) polymerových skel do automobilových světlometů z hlediska jejich mikroskopické struktury. V současnosti existuje mnoho typů mechanických či chemických testů, které dávají informaci o různých fyzikálních vlastnostech. Tyto testy jsou však náročné na čas i na použitý materiál. Představa rozpoznání fyzikálních vlastností ve zlomcích sekundy a pouze s použitím mikroskopu s fotoaparátem je v současnosti neřešený problém. K dispozici budeme mít několik mikroskopických snímků různých typů polymerů. Tyto obrazy vykazují charakter textur a tudíž přistoupíme k jejich analýze pomocí metod texturního popisu obrazů. Určení fyzikálních vlastností se v podstatě rovná zařazování obrazů do tříd, tudíž budeme tento problém řešit, jako úlohu klasifikace obrazů a v předložené práci budeme testovat několik metod extrakce příznaků v kombinaci s algoritmy strojového učení.

Cílem této práce je návrh a implementace testovacího systému pro evaluaci metod extrakce příznaků a klasifikace mikroskopických snímků polymerů a jeho zobecnění na texturní obrazy. Dalším cílem je vyhodnocení algoritmů texturní analýzy na dostupném obrazovém materiálu a nalezení optimálních kombinací pro různé typy textur.

2 Metody extrakce příznaků

V oblasti počítačového vidění je extrakce příznaků speciální disciplína, jež zajišťuje redukci dat, která by byla příliš velká pro algoritmy zpracování obrazu. Díky extrakci příznaků, se též můžeme zbavit nepotřebných dat a naopak zviditelnit data důležitá.

V této kapitole popíšeme použité metody extrakce příznaků, specializované na obrazová data typu textury. Nejprve vysvětlíme problematiku statistického popisu textur (metody prvního řádu, metody využívající matice sousednosti či metody využívající počet opakování jasů). Dále se zaměříme na metodu Local binary pattern a její modifikace (uniform pattern, různé podoby filtru).

2.1 Statistické metody

Statistické metody popisují textury ve formě, která je vhodná, pro statistické rozpoznávání. Výstupem těchto algoritmů je příznakový vektor vypovídající o vlastnostech jednotlivých textur[1]. Vstupem budou obrazy převedené do šedotónové reprezentace. V této části popíšeme statistiky prvního a druhého řádu, statistiky využívající počet opakování jasů (GLRL), a momentové metody (geometrické momenty, Huovy momenty).

2.1.1 Statistiky prvního řádu

Metody prvního řádu jsou nejjednodušším způsobem získání příznaků z obrazových dat na základě analýzy jasu. Výpočet se provádí pomocí relativního histogramu $H_R(p)$, který je funkcí jasu a vyjadřuje relativní četnost bodů s daným

jasem.

- Absolutní histogram

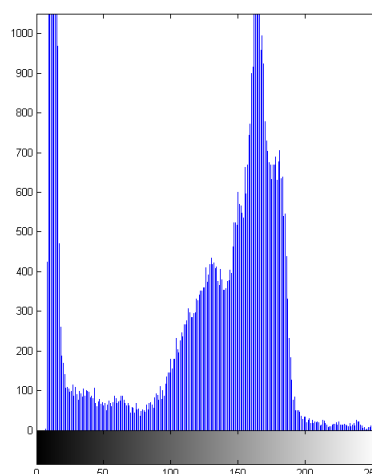
$$H(p) = \sum_{i,j} h(i, j, p) \quad (2.1.1)$$

$$h(i, j, p) = \left\langle \begin{array}{l} 1 \text{ pro } f(i, j) = p \\ 0 \text{ pro } f(i, j) \neq p \end{array} \right\rangle \quad (2.1.2)$$

- Relativní histogram

$$H_R(p) = \frac{H(p)}{\sum_p H(p)} \quad (2.1.3)$$

$$\sum_p H_R(p) = 1 \quad (2.1.4)$$



Obrázek 2.1: Ukázka šedotónového obrázku a jeho histogramu

Jako příznaky jsme použili první dva centrální momenty (M_1, M_2) a také koeficienty šikmosti a špičatosti histogramu, které z centrálních momentů vycházejí. Tyto příznaky jsme zvolili z důvodu jednoduché a rychlé vypočitatelnosti.

- k-tý Centrální moment

$$M_k = \frac{1}{P} \sum_{p=1}^P (H_R(p) - \mu)^k \quad (2.1.5)$$

- Koeficient šikmosti

$$\gamma_1 = \frac{M_3}{\sqrt{M_2^3}} \quad (2.1.6)$$

- Koeficient špičatosti

$$\gamma_2 = \frac{M_4}{M_2^2} \quad (2.1.7)$$

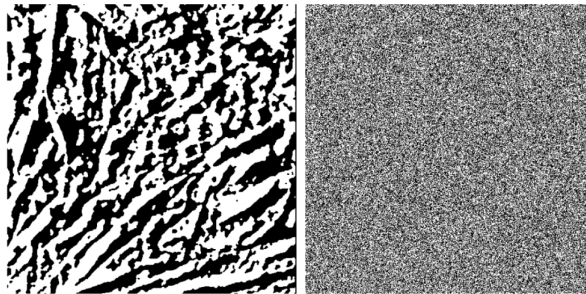
Pozn. μ je odhad střední hodnoty daný vztahem:

$$\mu = \frac{1}{P} \sum_{p=1}^P H_R(p) \quad (2.1.8)$$

[2]

2.1.2 Statistiky druhého řádu

Problémem statistik prvního řádu, zmíněných výše je fakt, že vypovídají pouze o množstevním zasoupení jasů v textuře, nikoliv o jejich prostorovém uspořádání. Nebude tedy náročné představit si textury se stejným histogramem, avšak strukturou jasů naprosto odlišné, viz. Obrázek 2.2



Obrázek 2.2: Textury se stejným histogramem

Tento nedostatek řeší matice susednosti (co-occurrence matrix), jež popisuje počty opakujících se jasů v určitých směrech. Tuto matici je tedy možné konstruovat v závislosti na směru ($0^\circ, 45^\circ, 90^\circ, 135^\circ$) a počtu pixelů se stejným jasnem. Díky tomu můžeme velmi dobře rozlišit jemné textury od hrubých. Rovnice (2.1.9-2.1.12) popisují výpočet matice susednosti $P_{\phi,d}(a,b)$ pro různé směry opakování jasů [1]:

$$\begin{aligned}
P_{0^\circ,d}(a,b) &= |[[(k,l), (m,n)] \in D : \\
& k - m = 0, |l - n| = d, f(k,l) = a, f(m,n) = b| \quad (2.1.9)
\end{aligned}$$

$$\begin{aligned}
P_{45^\circ,d}(a,b) &= |[[(k,l), (m,n)] \in D : \\
& (k - m = d, l - n = -d) \vee (k - m = -d, l - n = d), \\
& f(k,l) = a, f(m,n) = b| \quad (2.1.10)
\end{aligned}$$

$$\begin{aligned}
P_{90^\circ,d}(a,b) &= |[[(k,l), (m,n)] \in D : \\
& k - m = d, |l - n| = 0, f(k,l) = a, f(m,n) = b| \quad (2.1.11)
\end{aligned}$$

$$\begin{aligned}
P_{135^\circ,d}(a,b) &= |[[(k,l), (m,n)] \in D : \\
& (k - m = d, l - n = d) \vee (k - m = -d, l - n = -d), \\
& f(k,l) = a, f(m,n) = b| \quad (2.1.12)
\end{aligned}$$

Nyní uvedeme příklad výpočtu matice sousednosti. Mějme obraz I , který bude obsahovat 4 úrovně jasů, viz.(2.1.13)

$$I = \begin{pmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 3 & 3 & 3 \\ 3 & 3 & 4 & 4 \end{pmatrix} \quad (2.1.13)$$

Matice $P_{0^\circ,1}$ je konstruována následovně. Prvek matice $P_{0^\circ,1}(1,1)$ udává počet pixelů o jasech 1 a 1, jež jsou oddělené vzdáleností 1 ve směru 0° symetricky, tj. 180° je stejný směr jako 0° atd. Pro tento směr bude tedy hodnota matice sousednosti $P_{0^\circ,1}(1,1) = 4$, jelikož hodnoty 1, 1 spolu sousedí 4 krát na vzdálenost 1 pixel a ve směru 0° , resp. 180° .

$$P_{0^\circ,1} = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix} \quad P_{135^\circ,1} = \begin{pmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

Tabulka 2.1: Ukázka dvou různých matic sousednosti

Z matice sousednosti vypočteme dle vztahů(2.1.14-2.1.16) příznaky (kontrast, korelace, energie, homogenita)[1].

- **Kontrast** (typicky $\kappa = 2, \lambda = 1$)

Míra lokálních změn jasu v obraze. Pro obraz s konstantním jasnem by byla hodnota kontrastu 0.

$$\sum_{a,b} |a - b|^\kappa P_{\phi,d}^\lambda(a, b) \quad (2.1.14)$$

- **Energie**

Míra stejnorodosti obrazu. Čím více je obraz stejnorodý, tím vyšší je hodnota energie.

$$\sum_{a,b} P_{\phi,d}^2(a, b) \quad (2.1.15)$$

- **Homogenita**

$$\sum_{a,b} \frac{P_{\phi,d}(a, b)}{1 - |a - b|} \quad (2.1.16)$$

- **Korelace**

Míra linearitity obrazu, objekty lineárně uspořádané ve směru ϕ , způsobí vysoké hodnoty korelace v tomto směru.

$$\frac{\sum_{a,b} [(ab)P_{\phi,d}(a, b)] - \mu_x \mu_y}{\sigma_x, \sigma_y} \quad (2.1.17)$$

Veličiny $\mu_x, \mu_y, \sigma_x, \sigma_y$ se vypočítají dle následujících vztahů.

$$\mu_x = \sum_a a \sum_b P_{\phi,d}(a, b) \quad (2.1.18)$$

$$\mu_y = \sum_b b \sum_a P_{\phi,d}(a, b) \quad (2.1.19)$$

$$\sigma_x = \sum_a (a - \mu_x)^2 \sum_b P_{\phi,d}(a, b) \quad (2.1.20)$$

$$\sigma_y = \sum_b (b - \mu_y)^2 \sum_a P_{\phi,d}(a, b) \quad (2.1.21)$$

2.1.3 Gray level run length

Abychom mohli definovat tuto metodu popisu textury, musíme nejdříve uvést pojem běh (run). Run je počet opakování pixelů se stejným jasem v řadě, v určitém směru. Nechť $B(a, r)$ udává počet všech běhů v obraze o délce r a s jasem a . B je pak matice počtů opakování jasů (Gray level run length). Jako příznaky jsme zvolili Short run emphasis, Long run emphasis, Gray level nonuniformity a Run length nonuniformity a Run percentage, které vychází právě z matice počtu opakování jasů. Konstanta K vyjadřuje počet všech opakování jasů. Metody opakování jasů se mohou stát užitečným nástrojem pro popis textur, které jsou směrově nepravidelné, jelikož uvažují především počty primitiv (resp. oblastí se stejným jasem) a ne jejich směry [10].

$$K = \sum_{a=1}^L \sum_{r=1}^{N_r} B(a, r) \quad (2.1.22)$$

- **Short run emphasis**

Zvýrazňuje opakování krátkých běhů, vysoká hodnota vypovídá o jemnosti textury.

$$\frac{1}{K} \sum_{a=1}^L \sum_{r=1}^{N_r} \frac{B(a, r)}{r^2} \quad (2.1.23)$$

- **Long run emphasis**

Zvýrazňuje naopak dlouhé běhy, tedy rozsáhlé plochy o stejném jasu. Tato hodnota vypovídá o tom, zda je textura hrubá.

$$\frac{1}{K} \sum_{a=1}^L \sum_{r=1}^{N_r} B(a, r) r^2 \quad (2.1.24)$$

- **Gray level nonuniformity**

Nerovnoměrnost jasů v obraze je vyjádřena vysokou hodnotou tohoto příznaku.

$$\frac{1}{K} \sum_{a=1}^L \left[\sum_{r=1}^{N_r} B(a, r) \right]^2 \quad (2.1.25)$$

- **Run length uniformity**

Míra rovnoměrnosti délek opakování jasů. Nabývá vysokých hodnot jestliže jsou délky opakování jasu v obraze rovnoměrně zastoupeny.

$$\frac{1}{K} \sum_{r=1}^{N_r} \left[\sum_{a=1}^L B(a, r) \right]^2 \quad (2.1.26)$$

- **Run percentage**

Vyjadřuje celkový počet opakování běhů v obraze.

$$\frac{K}{\sum_{a=1}^L \sum_{r=1}^{N_r} B(a, r)r} = \frac{K}{MN} \quad (2.1.27)$$

2.1.4 Geometrické momenty

Geometrické momenty jsou často nasazovány v oblastech zpracování obrazu jako jeho alternativní reprezentace. Ze všech geometrických momentů je možné zpětně zrekonstruovat původní obraz. Pro spojitý obraz $I(x,y)$ je obecný geometrický moment definován dle vztahu(2.1.28)

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q I(x, y) dx dy \quad (2.1.28)$$

Pro naše potřeby si převedeme tento vztah do diskretní podoby.

$$m_{pq} = \sum_i \sum_j I(i, j) x_i^p y_j^q \quad (2.1.29)$$

Centrované momenty vycházejí ze vztahu (2.1.28) a jsou invariantní vůči translaci.

$$\mu_{pq} = \sum_i \sum_j I(i, j) (x_i - x_c)^p (y_j - y_c)^q \quad (2.1.30)$$

kde

$$x_c = \frac{m_{10}}{m_{00}}, y_c = \frac{m_{01}}{m_{00}} \quad (2.1.31)$$

Dále ještě nadefinujeme normované centrované momenty, které použijeme k vytvoření příznakových vektorů. Jejich výhodou je, že jsou invariantní vůči translaci a změně měřítka, což je v současnosti u popisu obrazu často žádanou vlastností.

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{\gamma}{2}}}, \gamma = \frac{p+q+2}{2} \quad (2.1.32)$$

2.1.5 Huovy momenty

V oblasti digitálního zpracování obrazu se Huovy momenty velmi často používají, právě pro popis obrazů. Vychází z geometrických momentů popsaných výše. Jejich výpočet popisují vztahy(2.1.34-2.1.42)

$$\Phi_1 = \eta_{20} + \eta_{02} \quad (2.1.33)$$

$$\Phi_2 = (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2 \quad (2.1.34)$$

$$\Phi_3 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \quad (2.1.35)$$

$$\Phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2 \quad (2.1.36)$$

$$\Phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{03} + \eta_{21})^2] \quad (2.1.37)$$

$$+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (2.1.38)$$

$$\Phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (2.1.39)$$

$$+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (2.1.40)$$

$$\Phi_7 = (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{21})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \quad (2.1.41)$$

$$+ (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{03} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (2.1.42)$$

Jejich velikou výhodou a důvodem, proč jsme je zvolili pro naše testování je jejich invariance vůči rotaci, translaci a změně měřítka, prvních 6 momentů je dokonce invariantní vůči zrcadlení[5].

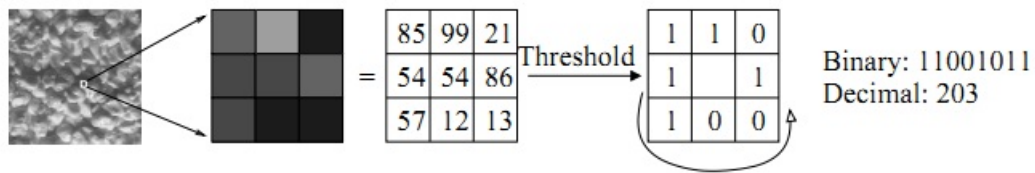
2.2 Local binary patterns

LBP je jednoduchá metoda pro extrakci texturních příznaků. Obraz je nejprve převeden do šedotónové reprezentace, pro každý pixel (x_c, y_c) obrazu se zpracovává jeho okolí a to podle rovnice (2.2.1), kde P je počet pixelů v okolí centrálního pixelu a R je poloměr neboli euklidovská vzdálenost centrálního a sousedního pixelu, g_p, g_c jsou hodnoty jejich jasů. Pro naše potřeby použijeme operátor $LBP_{1,8}$, tudíž uvažujeme tzv. osmiokolí sledovaného pixelu.

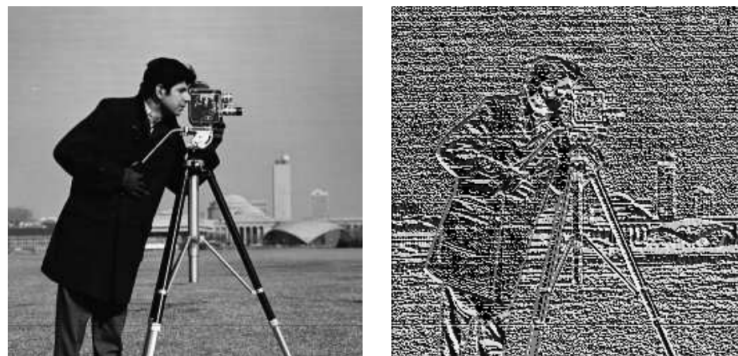
$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p; \quad s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (2.2.1)$$

Z vytvořeného LBP obrazu, jehož rozměry jsou shodné s původním obrazem, vypočteme histogram, jenž je zároveň vektorem příznaků [4].

Nyní naznačíme příklad výpočtu LBP hodnoty pro jeden pixel šedotónového obrazu (viz. Obrázek 2.3).



Obrázek 2.3: Výpočet jednoho pixelu LBP-obrazu



Obrázek 2.4: Původní obraz a LBP-obraz

Výsledný LBP-obraz ilustruje Obrázek 2.4. Jak jsme již uvedli, pro naše potřeby extrakce příznaků je původní obraz šedotónový. Nabízí se samozřejmě i možnost aplikace algoritmu LBP na barevné obrazy. V tomto případě bychom museli počítat LBP-obraz pro každou barevnou vrstvu zvlášť (R,G,B) a vznikly by tak 3 různé příznakové vektory pro 1 barevný obraz. Otázka, zda volit výpočet LBP-obrazu z barevného či šedotónového obrazu záleží na charakteru zpracovávaných dat. Pro naše potřeby je dostačující šedotónová varianta.

2.2.1 Rotační invariance a Uniform patterns

Jestliže uvažujeme LBP-filtr s osmiokolím, pak výsledný LBP-obraz může obsahovat až 256 úrovní jasu (0-255). Jeho histogramem je vektor obsahující 256 čísel. Pro naše potřeby je tento počet příznaků příliš vysoký. Jedním z algoritmů, které redukují počet příznaků a navíc zajišťují invarianci vůči natočení obrazu je tzv. minimalizace. Pomocí cyklické bitové rotace nalezneme minimální hodnotu příznaku v binárním tvaru. Na konci vybereme ze všech příznaků pouze ty, které jsou minimální. Hledání minima binárního čísla 01001000 cyklickou bitovou rotací ukazuje vztah (2.2.2)

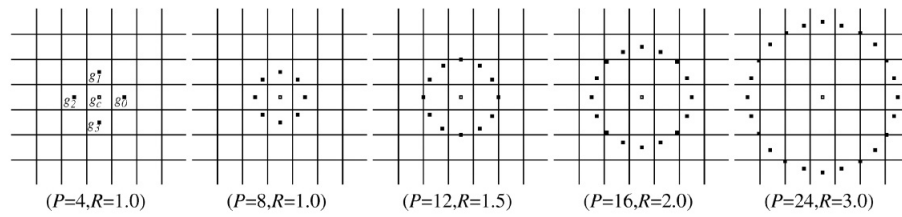
$$\begin{aligned} 01001000 &\rightarrow 00100100 \rightarrow 00010010 \rightarrow 00001001 \rightarrow & (2.2.2) \\ 10000100 &\rightarrow 01000010 \rightarrow 00100001 \rightarrow 10010000 \end{aligned}$$

z těchto 8 binárních čísel vybereme minimum, tedy 00001001.

Tyto příznaky můžeme ještě zredukovat pomocí algoritmu *uniform patterns*, jenž vybere pouze ta binární čísla, která mají počet přechodů 0/1 nebo 1/0 nižší nebo roven 2, jelikož bylo dokázáno, že takovéto příznaky mají v texturách nejvyšší zastoupení [6]. Takovouto dvojitou redukcí získáme z 256 původních 9 příznaků.

2.2.2 Modifikace metody LBP

Nabízí se otázka, zdali je dobré, omezit se pouze na filtr, tvořící osmiokolí pixelu. Různé poloměry filtrů a různé počty pixelů demonstruje Obrázek 2.2.2, jenž ukazuje 5 různých kruhově symetrických filtrů. Jestliže uvažujeme souřadnice centrálního



Obrázek 2.5: Různé typy LBP filtrů [6]

pixelu g_c $[0,0]$ pak souřadnice sousedních pixelů g_p se vypočte dle vztahu [6]

$$g_c = [R\sin(\frac{2\pi p}{P}), R\cos(\frac{2\pi p}{P})] \quad (2.2.3)$$

Variabilita použití filtrů je široká co se týče poloměru nebo počtu pixelů. LBP-filtry s větším poloměrem jsou např. vhodnější pro popis hrubších textur, naopak pro zachycení jemnějších detailů jsou vhodné filtry s menším poloměrem. Při aplikaci filtrů s velkým počtem pixelů musíme ovšem počítat s vyšší výpočetní náročností, jelikož s rostoucím počtem pixelů ve filtru rapidně vzrůstá i počet příznaků. Např. Příznakový vektor LBP-obrazu vytvořeného pomocí filtru se 16-ti pixely může mít v neredukovaném stavu 65536 hodnot. Po redukci se počet příznaků sníží na 16.

3 Klasifikační algoritmy

V této kapitole představíme několik algoritmů strojového učení, které často hrají hlavní roli v úlohách rozpoznávání obrazu. Pro naše testování jsme vybrali Bayesův klasifikátor, klasifikátor dle k -nejbližšího souseda a support vector machine. Při trénování těchto klasifikátorů použijeme metodu učení s učitelem.

3.1 Bayesův klasifikátor

Bayesův klasifikátor nachází svůj základ v klasifikačních algoritmech, řídících se dle kritéria minimální chyby a jeho nespornou výhodou je jednoduchost nastřádání, ovšem za cenu zlosti charakteru dat (např. uvažujeme data s normálním rozložením).

3.1.1 Odvození diskriminační funkce

Nyní odvodíme pravděpodobnostní diskriminační funkci na základě tohoto kritéria.

Mějme obrazový prostor X , jenž obsahuje obrazy z R tříd $\omega_1 \dots \omega_R$. Apriorní pravděpodobnost výskytu obrazů náležících do třídy ω_r označme jako $P(\omega_r)$. Ztrátu, jež vznikla chybným zařazením obrazu z třídy ω_s do třídy ω_r definujme následovně:

$$l(\omega_r|\omega_s) \tag{3.1.1}$$

Nyní můžeme napsat celou matici ztrátových funkcí, kterou použijeme dále při vy-

jadřování podmíněné střední ztráty.

$$\vec{l} = \begin{bmatrix} l(\omega_1, \omega_1); & l(\omega_1, \omega_2); & \dots & l(\omega_1, \omega_R); \\ l(\omega_2, \omega_1); & l(\omega_2, \omega_2); & \dots & l(\omega_2, \omega_R); \\ \vdots & \vdots & \ddots & \vdots \\ l(\omega_R, \omega_1); & l(\omega_R, \omega_2); & \dots & l(\omega_R, \omega_R); \end{bmatrix} \quad (3.1.2)$$

Diskriminační funkce, neboli pravidlo, dle kterého bude klasifikátor zařazovat vstupní obrazy do jednotlivých tříd ω_r bude vypadat takto $\omega_r = d(\vec{x}, \vec{q})$, kde q je vektor parametrů klasifikátoru.

Střední ztráta podmíněná výběrem obrazů ze třídy ω_s bude vypadat následovně:

$$J(\vec{q}|\omega_s) = \int_X l(d(\vec{x}, \vec{q})|\omega_s) p(\vec{x}|\omega_s) d\vec{x} \quad (3.1.3)$$

Celková střední ztráta:

$$J(\vec{q}) = \sum_{s=1}^R J(\vec{q}|\omega_s) P(\omega_s) = \int_X \sum_{s=1}^R l(d(\vec{x}, \vec{q})|\omega_s) p(\vec{x}|\omega_s) P(\omega_s) d\vec{x} \quad (3.1.4)$$

Hledáme parametry klasifikátoru \vec{q}^* , které minimalizují $J(\vec{q})$

$$J(\vec{q}^*) = \min_{\vec{q}} J(\vec{q}) = \min_{\vec{q}} \int_X \sum_{s=1}^R l(d(\vec{x}, \vec{q})|\omega_s) p(\vec{x}|\omega_s) P(\omega_s) d\vec{x} \quad (3.1.5)$$

$$= \int_X \min_{\vec{q}} \sum_{s=1}^R l(d(\vec{x}, \vec{q})|\omega_s) p(\vec{x}|\omega_s) P(\omega_s) d\vec{x} = \int_X \min L_x(\omega_r) dx \quad (3.1.6)$$

Hledáme tedy minimum $L_x(\omega_r)$

$$L_x(\omega_r) = \sum_{s=1}^R l(d(\vec{x}, \vec{q})|\omega_s) p(\vec{x}|\omega_s) P(\omega_s) d\vec{x} \quad (3.1.7)$$

Dále musíme vhodně zvolit tvar ztrátové funkce:

$$l(\omega_r|\omega_s) = 1 - \sigma_{rs} \quad (3.1.8)$$

$$\sigma_{rs} = \begin{cases} 1 & \text{pro } r = s \\ 0 & \text{pro } r \neq s \end{cases} \quad (3.1.9)$$

Nyní můžeme ztrátovou funkci dosadit do minimalizované funkce $L_x(\omega_r)$

$$\begin{aligned} \sum_{s=1}^R P(\omega_s|\vec{x}) &= 1 \\ \sum_{s=1}^R \frac{p(\vec{x}|\omega_s)P(\omega_s)}{p(\vec{x})} &= 1 \\ \sum_{s=1}^R p(\vec{x}|\omega_s)P(\omega_s) &= p(\vec{x}) \end{aligned}$$

Tabulka 3.1: Matematická vsuvka

$$L_x(\omega_r) = \sum_{s=1}^R (1 - \sigma_{rs}) p(\vec{x}|\omega_s) P(\omega_s) \quad (3.1.10)$$

$$= \sum_{s=1}^R p(\vec{x}|\omega_s) P(\omega_s) - p(\vec{x}|\omega_r) P(\omega_r) \quad (3.1.11)$$

Díky známému Bayesovu vztahu 3.1.12

$$P(\omega_s|x) = \frac{p(x|\omega_s)P(\omega_s)}{p(x)} \quad (3.1.12)$$

můžeme funkci $L_x(\omega_r)$ zjednodušit. K tomu bude třeba malá matematická odbočka, viz. Tabulka 3.1:

Po dosazení dostáváme následující vztah:

$$L_x(\omega_r) = p(x) - p(x|\omega_r)P(\omega_r) \quad (3.1.13)$$

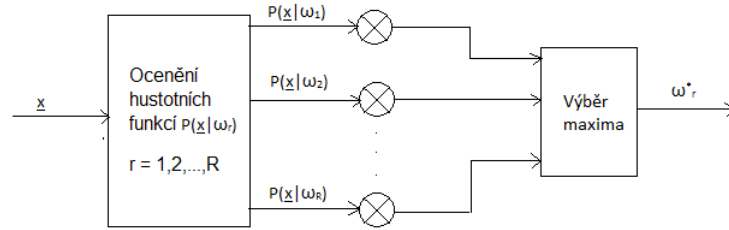
Jestliže požadujeme, aby funkce $L_x(\omega_r)$ byla minimální, musí být člen $p(x|\omega_r)P(\omega_r)$ maximální, tudíž můžeme provést úpravu:

$$L'_x(\omega_r) = p(x|\omega_r)P(\omega_r) \quad (3.1.14)$$

Nyní budeme hledat takovou třídu ω_r pro kterou je funkce $L'_x(\omega_r)$ maximální. Klasifikátor zařadí obraz \vec{x} do takové třídy ω_r pro kterou platí

$$\omega_r^* = \underset{r}{\operatorname{argmax}} p(x|\omega_r)P(\omega_r) \quad (3.1.15)$$

Jestliže mají třídy shodnou velikost (např. počet obrazů v jednotlivých třídách), pak je $P(\omega_r) = 1$, čímž dojde k dalšímu zjednodušení diskriminační funkce V praxi se velmi často používá její přirozený logaritmus $L'_x(\omega_r)$ [7].



Obrázek 3.1: Schéma Bayesova klasifikátoru

3.1.2 Trénování Bayesova klasifikátoru

Přepokádejme, že hustotní funkce pravděpodobnosti (resp. obrazy v jednotlivých třídách) $p(\vec{x}|\omega_r)$ bude mít normální rozložení. Pak ji můžeme vyjádřit dle následujícího vztahu:

$$p(\vec{x}|\omega_r) = \frac{1}{\sqrt{2\pi^n \det C_r}} e^{-0.5(\vec{x}-\vec{\mu}_r)^T C_r^{-1}(\vec{x}-\vec{\mu}_r)} \quad (3.1.16)$$

kde μ_r je vektor středních hodnot obrazů r-té třídy a C_r jejich kovarianční matice.

K natrénování Bayesova klasifikátoru je třeba znát apriorní pravděpodobnost $P(\omega_r)$ a spočítat z trénovacích dat střední hodnotu a kovarianční matici pro každou třídu [7].

$$\vec{\mu}_r = E[\vec{x}]_{\vec{x} \in \omega_r} \quad (3.1.17)$$

$$C_r = E[(\vec{x} - \vec{\mu}_r)(\vec{x} - \vec{\mu}_r)^T]_{\vec{x} \in \omega_r} \quad (3.1.18)$$

3.2 Klasifikace dle k-nejbližšího souseda

Tento klasifikátor patří do kategorie klasifikátorů dle minimální vzdálenosti. Pro každou třídu ω_r , $r = 1 \dots R$ vyjádříme vzdálenosti $\|\vec{x} - \mu_{r,s}\|$ ke vzorovým obrazům $\mu_{r,s}$, kde $s = 1 \dots S_r$. Poté je pro každou třídu ω_r uspořádáme tak, aby platilo:

$$\|\vec{x} - \mu_{r[1]}\| \leq \|\vec{x} - \mu_{r[2]}\| \leq \dots \leq \|\vec{x} - \mu_{r[S_r]}\| \quad (3.2.1)$$

Neznámý obraz x se pak zařadí do třídy dle minima průměrné vzdálenosti ke k nejbližším sousedům [7].

$$\omega_{r^*} = \operatorname{argmin}_r \frac{1}{k} \sum_{i=1}^k \|\vec{x} - \mu_{r[i]}\| \quad (3.2.2)$$

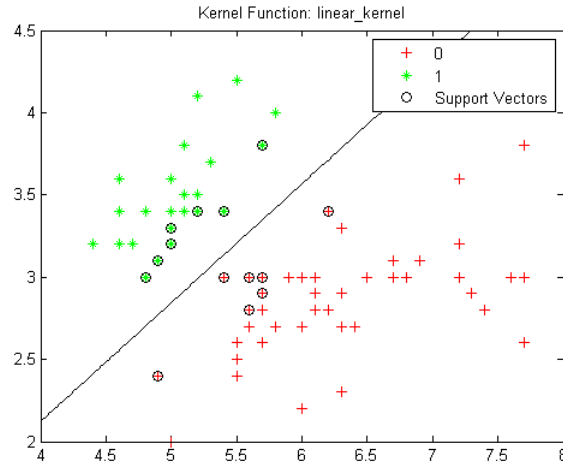
Klasifikátor dle k-nejbližšího souseda jsme zařadili do našeho testování z důvodu jeho jednoduché implementace. Použití tohoto klasifikátoru je vhodné i v případech, kdy není zabezpečená jeho optimální funkce dle kritéria minimální chyby. Např. jestliže není známé pravděpodobnostní rozložení obrazů v jednotlivých třídách, nebo u normálních rozložení nejsou známy disperzní matice atd. [7]

3.3 Support vector machine

Support vector machine (dále SVM) je metoda strojového učení hledající v obrazovém prostoru takovou nadrovinu, která optimálně rozděluje obrazy do dvou tříd. Základní podmínka pro konstrukci nadroviny je, aby její vzdálenost od nejbližších obrazů obou tříd byla co největší. Tyto nejbližší obrazy nazýváme podpůrné vektory (resp. support vectors). Tuto problematiku ve dvourozměrném prostoru ilustruje Obrázek 3.3

3.3.1 Lineární SVM

Mějme množinu trénovacích dat $\{\vec{x}_i, y_i\}$, $i = 1, 2, \dots, N$, kde $x_i \in \mathbb{R}^n$ jsou trénovací obrazy a $y_i \in \{-1, 1\}$ je informace o zařazení obrazu do první či druhé třídy. Obrazy \vec{x}_i ležící na hyperploše oddělující jednotlivé třídy splňují rovnici



Obrázek 3.2: SVM klasifikátor s lineární jádrovou funkcí

$$\vec{w}\vec{x} + b = 0 \quad (3.3.1)$$

\vec{w} je normálový vektor oddělující hyperplochy.

Označme vzdálenosti nejbližších obrazů k jednotlivým třídám d_+ , d_- , pak šířka hranice mezi třídami bude $d_+ + d_-$. Naším cílem bude nalézt takové nastavení parametrů hyperplochy, které maximalizuje šířku hranice.

Pro všechny obrazy \vec{x} platí

$$\vec{w}\vec{x} + b \geq +1 \quad (3.3.2)$$

$$\vec{w}\vec{x} + b \leq -1 \quad (3.3.3)$$

Obrazy pozitivní třídy splňují rovnici (3.3.2) a pro obrazy z negativní třídy platí rovnice (3.3.3). Tyto rovnice můžeme spojit v jednu

$$y_i(\vec{w}\vec{x}_i + b) \geq +1 \quad (3.3.4)$$

Šířka hranice je tím větší čím menší je euklidovská norma normálového vektoru oddělující hyperplochy $\|\vec{w}\|$. Bude tedy vhodné minimalizovat $\frac{1}{2}\|\vec{w}\|^2$. Omezující podmínku vyjadřuje rovnice (3.3.4).

$$L_p(\vec{w}, b, \alpha) = \frac{1}{2} \|\vec{v}\|^2 - \sum_i \alpha_i (y_i (\vec{w} \vec{x}_i + b) - 1) \quad (3.3.5)$$

Řešení tohoto problému se provádí pomocí metody Lagrangeových multiplikátorů a principu duality.

$$\vec{w} = \sum_i \alpha_i y_i \vec{x}_i \quad (3.3.6)$$

$$0 = \sum_i y_i \vec{x}_i \quad (3.3.7)$$

$$0 = \alpha_i (y_i (\vec{w} \vec{x}_i + b) - 1) \dots \forall i \quad (3.3.8)$$

Dosazením rovnice (3.3.6) do rovnice (3.3.5) dostáváme výraz pro duální lagrangian

$$L_d(\alpha) = \sum_i -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i \vec{x}_j \quad (3.3.9)$$

tento výraz již není závislý na parametrech rozdělující hyperplochy \vec{w} , b . Nyní se budeme snažit nalézt takové Lagrangeovy multiplikátory α_i , které budou splňovat podmínky:

$$\alpha_i \geq 0 \quad (3.3.10)$$

$$\sum_i \alpha_i y_i = 0 \quad (3.3.11)$$

Po výpočtu Lagrangeových multiplikátorů α_i lze snadno z rovnice (3.3.6) vyjádřit normálový vektor \vec{w} a z rovnice (3.3.8) lze vypočítat posunutí b . Klasifikátor bude zařazovat příchozí obrazy x do tříd dle následující diskriminační funkce [8]:

$$f(\vec{x}) = \vec{w} \vec{x} + b \quad (3.3.12)$$

kde

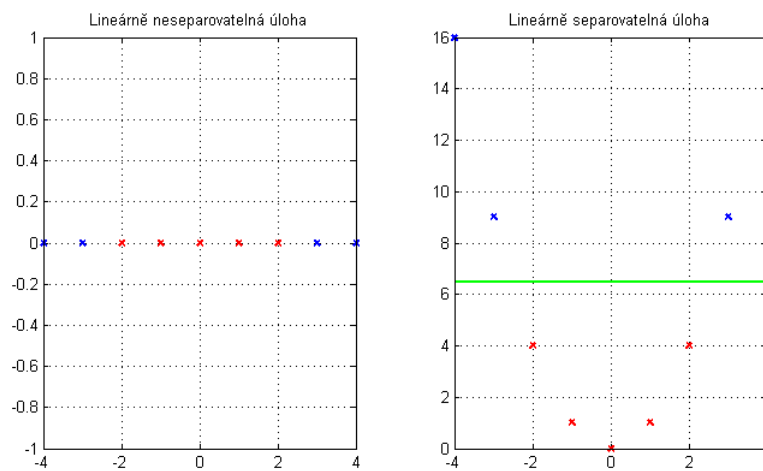
$$y_{\vec{x}} = +1 \dots f(\vec{x}) \geq 0 \quad (3.3.13)$$

$$y_{\vec{x}} = -1 \dots f(\vec{x}) < 0 \quad (3.3.14)$$

$$(3.3.15)$$

3.3.2 Nelineární SVM

Existuje mnoho úloh, které nejsou lineárně separabilní. Řešení tohoto problému se provádí pomocí mapování původních obrazů do alternativního prostoru často vyšší dimenze, v němž je úloha opět lineárně separovatelná. Ukázkou takového transformace je Obrázek 3.3.2. Z prvního obrázku je patrné, že třídy v jednorozměrném prostoru nejsou lineárně separabilní. Pomocí jádrové transformace (polynomiální jádrová funkce) jsme tyto třídy převedli do dvourozměrného prostoru. Tento převod nám zajistil lineární separabilitu tříd, ovšem za cenu vyšší dimenze a výpočetní náročnosti.



Obrázek 3.3: Ukázka mapování z 1D prostoru do 2D prostoru

Mapování se provádí pomocí jádrových funkcí. Pro naše testování jsme vybrali tři často používané: polynomiální funkci, RBF a funkci reprezentující dvouvrstvou neuronovou síť se sigmoidálními aktivačními funkcemi [8].

- Polynomiální jádrová funkce

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \vec{x}_j + c)^p \quad (3.3.16)$$

- Gaussovská radiální basická funkce

$$K(\vec{x}_i, \vec{x}_j) = e^{(\vec{x}_i \vec{x}_j + c)^p} \quad (3.3.17)$$

- Funkce reprezentující dvouvrstvou sigmoidální neuronovou síť

$$K(\vec{x}_i, \vec{x}_j) = \tanh(\kappa x_i x_j - \delta) \quad (3.3.18)$$

3.3.3 Klasifikace do více tříd

SVM je ze své podstaty binární klasifikátor. Diskriminační funkce vždy rozhodne o příslušnosti obrazu do první či druhé třídy. Řešením problému klasifikace do více tříd je jeho rozdělení na několik binárních úloh.

- **Každý s každým** Pro každé dvě třídy spočítáme diskriminační funkci. Jestliže počet tříd bude N , pak počet diskriminačních funkcí bude N_d . Tento počet můžeme vyjádřit pomocí kombinačního čísla, viz rovnice (3.3.19)

$$N_d = \binom{N}{2} = \frac{N!}{(N-2)!2!} \quad (3.3.19)$$

Klasifikaci obrazu počítáme pro každé dvě třídy s využitím již vypočtené diskriminační funkce. Výsledná třída je ta, do níž se klasifikovaný obraz umístil nejvícekrát.

- **One against all algoritmus**

Pro každou třídu spočteme jednu diskriminační funkcií oproti všem ostatním třídám, tj. tyto třídy sloučíme do jedné. Pro N tříd tak vznikne N diskriminačních funkcí. Poté klasifikovaný obraz dosadíme do těchto diskriminačních funkcí a spočteme jejich hodnoty. Obraz posléze zařadíme do třídy s nejvyšší hodnotou diskriminační funkce. Algoritmus One against all je velmi rychlý a nenáročný na výpočet, ale nepracuje dobře jestliže třídy nejsou relativně separabilní [11].

3.4 Metody evaluace výkonu klasifikačních algoritmů

K tomu, abychom mohli dobře vyhodnotit vhodnost použití algoritmů (extrakce příznaků resp. klasifikace) pro různé typy textur potřebujeme nástroje pro evaluaci výsledků použitých metod. Za důležité ukazatele považujeme procentuální úspěšnost klasifikace obrazů a dobu klasifikace, která je velmi důležitým faktorem při nasazování těchto algoritmů do procesů probíhajících v reálném čase. Pro podrobnější informaci o obrazech zařazených do jednotlivých tříd použijeme tzv. matici záměn (confusion matrix)

3.4.1 Procentuální úspěšnost a doba klasifikace

Procentuální úspěšnost zařazování obrazů vypočteme podle vztahu

$$\frac{n_c}{n} \cdot 100\% \quad (3.4.1)$$

kde n_c je počet správně klasifikovaných obrazů a n je počet všech klasifikovaných obrazů. Dobu klasifikace uvažujeme jako čas, jenž klasifikátor potřebuje k zařazení jednoho obrazu do třídy.

3.4.2 Confusion matrix

Matice záměn je jedním z nejdůležitějších, a také nejpoužívanějších algoritmů v oblasti evaluace algoritmů strojového učení, jelikož nám dává přesnou informaci

o počtech správně i špatně zařazených obrazů. Číslo na pozici (i,j) v matici záměn udává počet obrazů j -té třídy zařazených klasifikátorem do i -té třídy. Uvažujme např. klasifikátor natrénovaný na rozpoznávání tří barev červená (R), zelená (G), modrá (B). Nyní předložíme klasifikátoru 9 červených, 7 zelených a 10 modrých vzorků. Tabulka 3.4.2 ukazuje, že klasifikátor označil 4 červené vzorky jako červené, 2 červené jako zelené a tři červené jako modré vzorky.

		Skutečná třída		
		R	G	B
Třída určená klasifikátorem	R	4	1	0
	G	2	5	2
	B	3	1	8

Tabulka 3.2: Confusion matrix

Matrice záměn je také základem pro výpočet mnoha ukazatelů kvality klasifikace, např. celkové procentuální úspěšnosti klasifikace (correct rate-CR), kterou jsme zmínili v úvodu kapitoly. Ta se z matice záměn spočítá dle rovnice(3.4.2)

$$CR = \frac{\sum_{i=1}^N (T_i)}{\sum_{i=1}^N (T_i + F_i)} \quad (3.4.2)$$

kde T_i (True) je počet správně zařazených obrazů do třídy i a F_i (False) je počet chybně zařazených obrazů do třídy i . Dalším ukazatelem může být celková chyba (Error rate), která se spočítá jednoduše $ER = 1 - CR$. Existují také ukazatelé popisující výsledky získané maticí záměn podrobněji než celková úspěšnost či chyba, nebo jsou tyto ukazatelé zaměřené na specifické situace, např. nerovnoměrnost testovacích dat. K takovým ukazatelům patří úspěšnost klasifikace pro jednotlivé třídy, přesnost a úplnost, nebo sensitivita a specificita.

- Úspěšnost klasifikace pro jednotlivé třídy

$$CR_i = \frac{T_i}{T_i + F_i} \quad (3.4.3)$$

- Přesnost a úplnost

Pojmy z oblasti vyhledávání informací.

$$P = \frac{T_i}{T_i + F_i} \quad (3.4.4)$$

$$U = \frac{T_i}{T_i + \sum_{j=1}^{i-1}(F_j) + \sum_{j=i+1}^N(F_j)} \quad (3.4.5)$$

Přesnost je v podstatě to samé co úspěšnost klasifikace pro jednotlivé třídy.

- **Sensitivita a specificita**

Tyto pojmy jsou používané především v oblasti medicíny.

$$SE = \frac{T_i}{T_i + \sum_{j=1}^{i-1}(F_j) + \sum_{j=i+1}^N(F_j)} \quad (3.4.6)$$

$$SP = \frac{\sum_{j=1}^{i-1}(F_j) + \sum_{j=i+1}^N(F_j)}{\sum_{j=1}^{i-1}(F_j) + \sum_{j=i+1}^N(F_j) + F_i} \quad (3.4.7)$$

Opět můžeme vidět podobnost mezi sensitivitou a úplností, což je způsobeno používáním pojmů v různých oblastech.

Další možností, jak modifikovat klasickou matici záměn je opatření jednotlivých chybných zařazení různými cenami, tj. záleží nám i na typu chyby. Matice záměn pak bývá doplněna zároveň o matici cen. Její prvky specifikují cenu vykonaných zařazení do tříd [9].

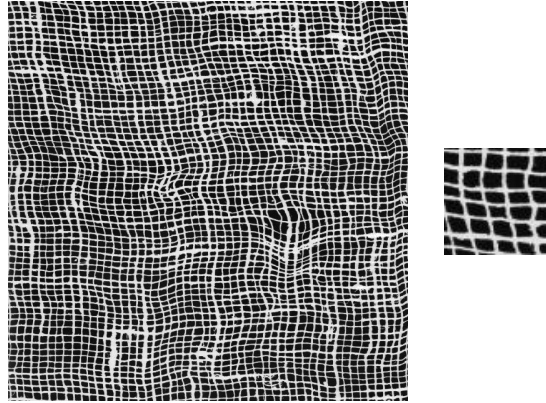
4 Implementace testovacího systému

Tato kapitola se bude věnovat návrhu, struktuře a implementaci systému pro testování algoritmů extrakce texturních příznaků a texturní klasifikace. Pro vytvoření testovacího systému, jenž obsahuje jednotlivé metody extrakce příznaků, klasifikátory a nástroje pro evaluaci výsledků, jsme použili programovací jazyk Matlab. Jeho výhodou je, že část námi používaných algoritmů je v tomto jazyce již implementována. Konfigurace testovacího prostředí je napsána v jazyce YAML a ke komunikaci s jazykem Matlab slouží balík funkcí *YAMLMatlab_0.4.1*. Důraz je kladen na snadnou rozšiřitelnost testovacího systému o nové funkce, které ovšem musí dodržet strukturu popsanou v 4.2.

4.1 Předzpracování dat

Jelikož pracujeme s obrazovými daty, je nutné připravit si tato data pro testování. Jak jsme uvedli v 2 popis textur námi používaných metod vychází z šedotónové reprezentace obrazu, do níž data převede funkce *loadTrainingImages.m* při načítání. O další zpracování obrazu se stará funkce *preprocessing.m*. Uvažujme např. nedostatek dat k natrénování klasifikátoru, každá třída je např. reprezentována jedním velkým obrazem. Funkce *preprocessing.m* rozdělí každý obraz na určený počet menších pod obrázků o náhodné pozici za účelem zvýšení počtu dat potřebných k natrénování klasifikátoru. Díky náhodnému výběru pozice menších obrazů se tyto obrazy mohou i překrývat a nevzniká tak třída s unikátními obrazy, což zvyšuje vypovídací hodnotu jednotlivých tříd. Tyto obrazy jsou uloženy do složek podle

tříd, do nichž patří. Obrázek 4.1 ukazuje původní obraz z galerie Brodatz a jeden z výřezů (zvětšený) provedený funkcí *preprocessing.m*.



Obrázek 4.1: Ukázka původního obrazu a jeho výřezu

4.2 Struktura a konfigurace testovaných metod

Společným rysem funkcí pro extrakci příznaků je, že vracejí vektor příznaků. Prvním vstupním parametrem je vždy zpracováváný obraz. Dále mohou následovat další vstupní parametry, u funkce *LBPvector.m* jsou to např. parametry LBP-filtru (poměr, počet pixelů).

Funkce pro klasifikaci zase shodně vracejí vektor s informací o zařazení testovacích obrazů do tříd. Shodnými vstupními parametry jsou matice trénovacích a testovacích dat, vektor, jenž určuje zařazení trénovacích dat do tříd, a také údaj o počtu tříd. Vstupními parametry specifickými pro jednotlivé funkce jsou např. počet nejbližších sousedů ve funkci *k_NN.m*, či parametr jádrové RBF funkce sigma ve funkci *SVM.m*. Podrobnější informace o těchto funkcích lze nalézt v jejich komentovaných částech.

Konfigurace použitých funkcí, zejména jejich specifické parametry, je zapsána v souboru *configuration.yml*. Dále v tomto souboru nalezneme cesty ke zdrojovým kódům a k obrazovým materiálům. Ve druhém konfiguračním souboru *run_conf.yml* se nachází výčet algoritmů extrakce příznaků a klasifikace, které budou testovány. Tyto konfigurační soubory slouží uživateli ke komunikaci s testovacím systémem a v budoucnu by mohly být nahrazené grafickým uživatelským rozhraním.

Ukázka části konfiguračního souboru *configuration.yml*.

```
cesty:
test_data: ../../sample_data/polymery
sources:
- ../classification/k_NN
- ../classification/SVM
- ../extrakce_priznaku/LBP
konfigurace:
preprocessing:
- 100 #number of subimages, which is cut from one big image
- 10 #number of classes
- 0.1 #number of testing images, expressed by percentage(0-1)
klasifikator:
-
name: k_NN
in:
- 10 # number of nearest neighbors
-
name: SVM
in:
- 1 #RBF_Sigma parameter value
priznaky:
-
name: LBPvector
in:
- 8 # number of pixels in LBP filter
- 1 # radius of LBP filter
```

Při implementaci testovacího systému a při testech obecně, jsme použili několik funkcí, které jsou součástí toolboxů v Matlab. Např. funkce pro extrakci příznaků založené na použití matice sousednosti, *graycomatrix.m*, *graycoprops*, či funkce pro SVM klasifikaci *svmtrain.m*, *svmclassify*, k nimž jsme museli naprogramovat algoritmus pro klasifikaci do více tříd, jelikož tyto funkce jsou primárně použitelné pouze pro binární klasifikaci. Dále bylo nutné implementovat další funkce pro extrakci příznaků, jako *LBPvector.m*, či pro klasifikaci, např. *bayes_class.m* či *k_NN.m*. Pomyslným předělem mezi konfiguračním souborem a jednotlivými funkcemi pro klasifikaci a extrakci příznaků jsou funkce *make_features.m* (uspořádává jednotlivé příznakové vektory do matice příznaků) a *class_eval.m* (vyhodnocuje výkon klasifikačních algoritmů).

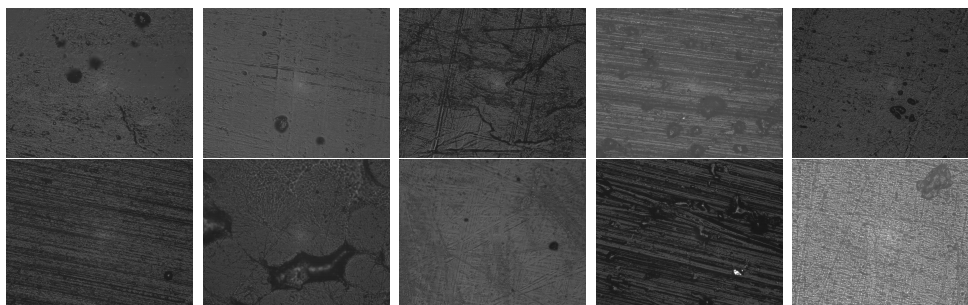
Testovací systém se spouští skriptem *test.m*. Uživatel nastaví výčet testovaných metod extrakce příznaků a klasifikace v souboru *run_conf.yml*. Dále určí cestu k testovaným datům a volitelné parametry u některých metod v souboru *configuration.yml*. Systém pak spočte a vytiskne informaci o tom, jaké metody byly testovány, confusion matrix, procentuální úspěšnost klasifikátoru a čas klasifikace pro

5 Výsledky a jejich interpretace

V této kapitole vyhodnotíme výsledky testů na dvou skupinách vstupních dat. Nejprve se budeme věnovat klasifikaci mikroskopických snímků polymerů. Tento problém pak ve druhé části kapitoly zobecníme na rozpoznávání textur. Data pro druhou část testů jsme si vypůjčili ze známé internetové galerie Brodatz.

5.1 Testy mikroskopických snímků polymerů

První sadou obrazů pro testování texturních příznaků budou mikroskopické snímky polymerů, viz. Obrázek 5.1. Rozpoznávání struktury (resp. texturní klasifikace) za účelem určení fyzikálních vlastností polymerů bylo původní motivací k vytvoření této diplomové práce.



Obrázek 5.1: Mikroskopické snímky polymerů

Častým problémem při rozpoznávání obrazu je nedostatek dat k natrénování klasifikačních algoritmů. Při nastavování klasifikátorů se setkáváme se dvěma negativními jevy. Při nedostatku trénovacích dat dochází k nedotrénování klasifikátoru, naopak při příliš vysokém objemu dat může docházet k jeho přetrénování. Ani

v jednom z těchto případů klasifikátor neplní svou úlohu správně. Proto je volba počtu trénovacích dat velmi důležitou úlohou při nastavování klasifikačního algoritmu. Pro potřeby testování jsme rozdělili každý snímek na 100 menších podsímků, tak vznikla data k natrénování klasifikátorů (70% snímků z každé třídy) a k jejich následnému testování (30% snímků). Menší podsímků nemohou být příliš malé, jinak by nebylo možné je dobře popisovat statistickými metodami, tím by ztratily svou vypovídací hodnotu, stejně tak musí být vyřezávány z náhodných míst původního obrazu, abychom zajistili nestrannost výsledků, jelikož předem určená pozice těchto výřezů by mohla negativně ovlivnit trénování i testování klasifikátoru.

Jelikož některé z metod extrakce příznaků i klasifikace mají nastavitelné parametry (např. tvar LBP-filtru, offset matice sousednosti, počet nejbližších sousedů či parametr sigma radiální bazické funkce klasifikátoru SVM), testovali jsme také vliv změny těchto parametrů na úspěšnost klasifikace.

	Bayes.klas.	k-NN	SVM
LBP(8,1)	79.33	75.00	77.33
LBP(16,2)	67.33	68.67	53.67

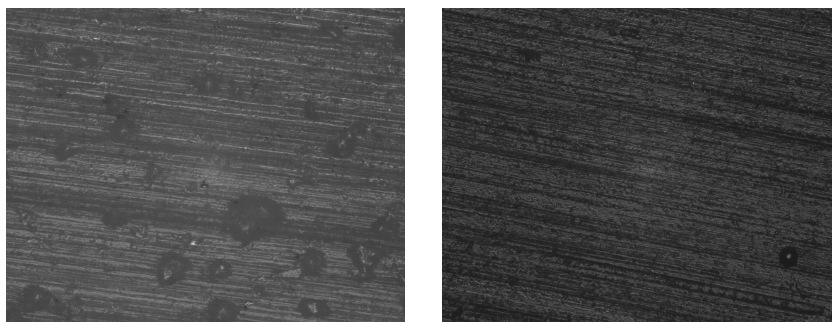
Tabulka 5.1: Vliv LBP-filtru na úspěšnost klasifikace[%]

Tabulka 5.1 ukazuje, jak velikost LBP-filtru ovlivní úspěšnost klasifikace polymerů. Nejlepších výsledků dosahuje filtr LBP(8,1). Textury v této sadě obrazů mají relativně dobře rozlišitelnou strukturu, tudíž můžeme uvažovat tento filtr, jako vhodný pro takové typy textur. Snížená úspěšnost klasifikace pro filtr LBP(16,2) v kombinaci s klasifikátorem SVM je způsobena vyšší dimenzí vektorů příznaků, kterou metoda LBP se zmíněným filtrem produkuje.

Nyní se podívejme blíže na přesnost Bayessova klasifikátoru v kombinaci s metodou LBP(8,1). Mějme matici záměn CM vyhodnocenou pro tuto kombinaci metod. Z (5.1.1) je patrné, že nejlépe se klasifikovaly obrazy z 2. a 10. třídy polymerů, nejhůře obrazy z 6. třídy. Při klasifikaci do 6. třídy docházelo nejčastěji k záměně se 4. třídou.

$$CM = \begin{pmatrix} 22 & 0 & 2 & 4 & 1 & 7 & 2 & 1 & 4 & 0 \\ 0 & 28 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 19 & 0 & 3 & 0 & 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 23 & 0 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 & 22 & 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 2 & 2 & 1 & 16 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 & 0 & 24 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 27 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 23 & 0 \\ 1 & 0 & 2 & 1 & 0 & 0 & 1 & 1 & 0 & 28 \end{pmatrix} \quad (5.1.1)$$

Z Obrázků 5.2 a 5.3 je patrná podobná struktura těchto textur, jež byla příčinou tak častých záměn. Ze získaných výsledků, viz. (5.1.1) lze také usuzovat dobrou odolnost metody vůči šumu, který se objevuje ve snímcích např. v podobě náhodně rozmístěných teček.



Obrázek 5.2: 4.třída polymerů Obrázek 5.3: 6.třída polymerů

Další zkoumanou metodou z hlediska změny parametrů bude extrakce příznaků založená na výpočtu matice sousednosti. Při těchto testech jsme měnili offset (d, ϕ) , tedy vzdálenost mezi pixely ($d = 1$ a $d = 5$) a úhel natočení ($\phi = 0^\circ, 45^\circ, 90^\circ, 135^\circ$).

Závislost úspěšnosti klasifikace na nastavení parametrů matice sousednosti ukazuje Tabulka 5.2. Vyšší přesnosti klasifikace dosahují příznaky obdržené při použití vzdálenosti ($d = 1$). Získané výsledky také ukazují, že charakter příznaků není vhodný pro klasifikaci dle k-nejbližšího souseda. K podrobnější představě o charakteru dat vhodných pro popis pomocí matice sousednosti (zvolený offset $(5, 0^\circ)$) slouží matice záměn (5.1.2). Je evidentní, že jemné textury (např. 1. či 8.

	Bayes.klas.	k-NN	SVM
(1, 0°)	68.67	47.67	72.67
(1, 45°)	70.67	41.00	81.00
(1, 90°)	74.00	39.67	75.00
(1, 135°)	70.00	38.00	75.33
(5, 0°)	66.00	41.33	70.33
(5, 45°)	63.33	42.67	62.67
(5, 90°)	66.67	43.33	64.00
(5, 135°)	59.67	39.00	60.33

Tabulka 5.2: Vliv nastavení offsetu matice sousednosti na úspěšnost klasifikace (polymery)

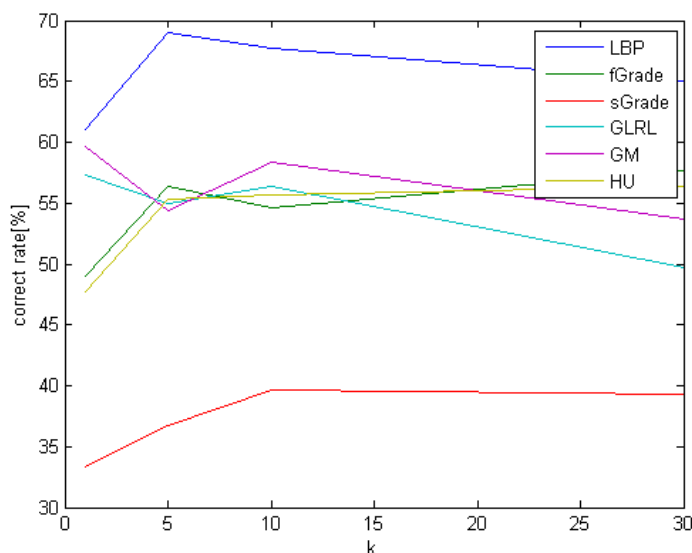
třída) jsou klasifikovány špatně narozdíl od textur s výrazným vzorem (4., 9. třída)

$$CM = \begin{pmatrix} 13 & 0 & 4 & 0 & 0 & 5 & 2 & 0 & 0 & 1 \\ 0 & 29 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 18 & 0 & 0 & 0 & 0 & 5 & 1 & 0 \\ 2 & 0 & 0 & 27 & 0 & 5 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 22 & 0 & 0 & 1 & 0 & 5 \\ 5 & 0 & 0 & 1 & 0 & 18 & 0 & 1 & 0 & 0 \\ 2 & 0 & 2 & 1 & 2 & 2 & 25 & 1 & 0 & 4 \\ 1 & 0 & 4 & 0 & 0 & 0 & 0 & 13 & 3 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 0 & 8 & 26 & 0 \\ 1 & 1 & 0 & 1 & 6 & 0 & 1 & 1 & 0 & 20 \end{pmatrix} \quad (5.1.2)$$

Některé z testovaných klasifikačních algoritmů mají také nastavitelné parametry (k-NN, SVM). Podívejme se tedy, jakou úlohu budou hrát změny těchto parameterů v úloze klasifikace polymerových snímků.

Obrázek 5.4 vyjadřuje závislost úspěšnosti klasifikace dle k-nejbližšího souseda na počtu nejbližších sousedů pro jednotlivé metody extrakce příznaků. Z grafu je patrné, že nejvhodnější je kombinace s metodou LBP a počtem 5 nejbližších sousedů, naopak příznaky založené na výpočtu matice sousednosti jsou pro třídu klasifikátorů dle

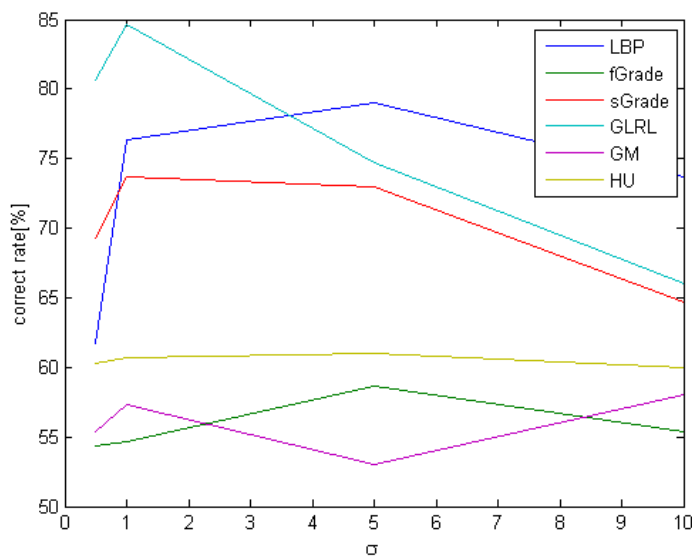
nejmenší vzdálenosti naprosto nevhodné. Optimální počet nejbližších sousedů se zdá být v intervalu (5,20), kdy klasifikátor dosahuje zajímavých výsledků. S rostoucím počtem nejbližších sousedů (30 a více) úspěšnost klasifikace klesá. Důvodem tohoto jevu může být např. přetrénování klasifikátoru.



Obrázek 5.4: Vliv počtu nejbližších sousedů na přesnost klasifikace (polymery)

Při klasifikaci algoritmem SVM hraje důležitou roli parametr radiální bazické funkce σ , pro naše potřeby jsme zvolili hodnoty parametru $\sigma = 0.5, 1, 5, 10$. Obrázek 5.5 ukazuje úspěšnost klasifikace v závislosti na změně tohoto parametru. Nejlepších výsledků klasifikace dosahují příznaky vytvořené algoritmy LBP, matice sousednosti a GLRL. Charakter těchto příznaků a jejich rozložení je evidentně vhodné použít v kombinaci s klasifikátorem SVM, na ostatní metody extrakce příznaků nemá změna parametru σ větší vliv. Optimální volba zkoumaného parametru bude z intervalu (1,3). Za ideální kombinaci algoritmů a jejich nastavení pro rozpoznávání polymerových snímků lze považovat GLRL a SVM($\sigma = 1$).

Na závěr provedeme celkové vyhodnocení použitých algoritmů v základním nastavení. Z Tabulky 5.3 je patrné, že nejvhodnější kombinací algoritmů pro rozpoznávání mikroskopických snímků polymerů jsou metody extrakce příznaků LBP, GLRL spolu s Bayesovým klasifikátorem a SVM, které dosahují v průměru 80% úspěšně zaklasifikovaných obrazů

Obrázek 5.5: Vliv parametru σ na přesnost klasifikace (polymery)

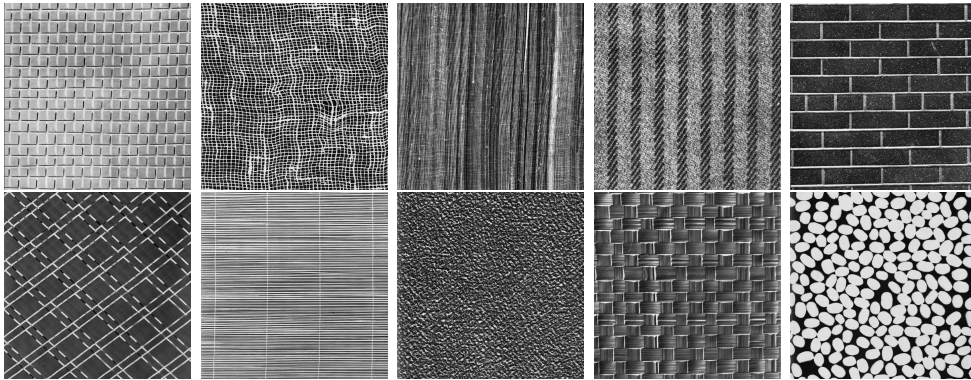
	Bayes.klas.	k-NN	SVM
LBP	82.00	74.67	77.33
První řád	57.33	55.00	60.00
Matice sousednosti	69.00	44.33	73.67
GLRL	81.67	56.33	82.67
Geometrické momenty	48.33	54.00	57.33
Huovy momenty	48.67	52.33	64.67
čas klasifikace[s]	4.04e-05	7.12e-04	0.03

Tabulka 5.3: Celkový přehled úspěšnosti a časů klasifikace (polymery)

Důležitou součástí hodnocení klasifikátorů pro nasazení do online režimu je doba, za kterou klasifikátor dokáže zařadit obraz do třídy. Informaci o časech klasifikace udává Tabulka 5.3, z té je patrné, že nejlepších výsledků dosahuje Bayesův klasifikátor, naopak klasifikátor SVM klasifikuje pomaleji, což se ovšem projeví pouze při klasifikaci mnoha obrazů

5.2 Testy na texturách z galerie Brodatz

V druhé části této kapitoly budeme testovat obrazy ze známé databáze textur Brodatz, která je přímo určená pro evaluaci algoritmů pro rozpoznávání textur. Obsahuje 111 různých vzorů, z nichž jsme pro potřeby testování vybrali 10. Účelem testování algoritmů na této sadě obrazů je zobecnění problému rozpoznávání polymerů na rozpoznávání textur. Výsledkem těchto testů by měla být optimalizace volby metod pro extrakci příznaků a klasifikace pro různé typy textur.



Obrázek 5.6: Textury z galerie Brodatz

Podobně jako v první části jsme rozdělili každý snímek na 100 menších podsímků za účelem zvětšení objemu dat k trénování a testování klasifikačních algoritmů. Také v této části budeme pozorovat vliv parametrů jednotlivých metod na úspěšnost klasifikace.

	Bayes.klas.	k-NN	SVM
LBP(8,1)	99.67	97.67	95.33
LBP(16,2)	88.33	94.67	69.67

Tabulka 5.4: Vliv LBP-filtru na úspěšnost klasifikace (Brodatz)

Tabulka 5.2 ukazuje, jak velikost LBP-filtru ovlivní úspěšnost klasifikace textur. Oproti testům provedeným na obrazech polymerů pozorujeme zvýšenou přesnost klasifikovaných obrazů, což je dáno větší rozmanitostí obrazů z různých tříd a také nepřítomností šumu.

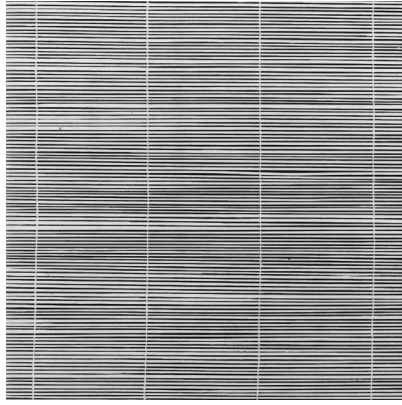
Nejllepších výsledků dosahuje filtr LBP(8,1). Textury v této sadě obrazů jsou svým charakterem spíše jemné než hrubé, tudíž můžeme uvažovat tento filtr, jako vhodný pro takové typy textur. Snížená úspěšnost klasifikace pro filtr LBP(16,2) a klasifikátor SVM je způsobena vyšší dimenzí vektorů příznaků, kterou metoda LBP se zmíněným filtrem produkuje.

Nyní prozkoumáme přesnost Bayesova klasifikátoru v kombinaci s metodou LBP(16,2). Mějme matici záměn CM vyhodnocenou pro tuto kombinaci metod. Z (5.2.1) je patrné, že klasifikace textur z galerie Brodatz je velmi přesná.

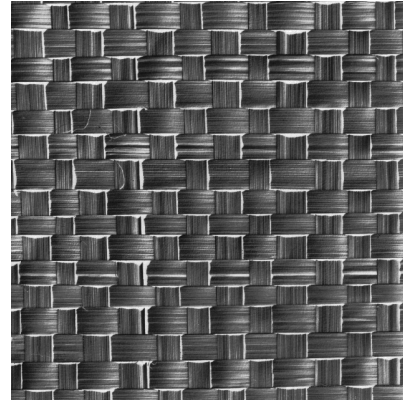
$$CM = \begin{pmatrix} 30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 30 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 30 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 29 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 30 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 23 & 0 & 29 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 30 \end{pmatrix} \quad (5.2.1)$$

Při klasifikaci do 7. třídy (rákos) docházelo nejčastěji k záměně s 9. třídou (ratan). Důvodem je neschopnost fitru o poloměru 2 pixely rozlišit velmi jemnou strukturu obrazu ze 7. třídy. Obrazy z ostatních tříd, které jsou svým charakterem spíše hrubé a zrnité, popisuje algoritmus LBP(16,2) velmi dobře, viz (5.2.1). Algoritmy LBP mají velmi dobré výsledky při popisu textur s výraznou a pravidelnou strukturou, ovšem musíme si uvědomit, že obrazy z galerie Brodatz jsou v jistém smyslu ideální (žádný šum, rovnoměrné nasvícení, vysoká rozmanitost). V praxi se s takovými podmínkami bohužel často nesetkáme.

Další zkoumanou metodou z hlediska změny parametrů bude extrakce příznaků založená na výpočtu matice sousednosti. Při těchto testech jsme měnili offset (d, ϕ) matice, tedy vzdálenost mezi pixely ($d = 1$ a $d = 5$) a úhel natočení ($\phi = 0^\circ, 45^\circ, 90^\circ, 135^\circ$). Toto nastavení jsme zvolili s ohledem na charakter textur. Jelikož v testovacích datech nemáme textury s velmi velkými primitivy, je použití



Obrázek 5.7: rákos Brodatz



Obrázek 5.8: ratan Brodatz

vzdálenosti d větší než 5 pixelů zbytečné. Nastavení úhlu ϕ testujeme v celém rozsahu, který nám dovoluje funkce *graycomatrix.m*, jelikož textury z galerie Brodatz mají primitiva natočená do různých směrů .

	Bayes.klas.	k-NN	SVM
(1, 0°)	98.33	64.00	98.33
(1, 45°)	99.67	54.00	99.00
(1, 90°)	97.33	57.00	96.67
(1, 135°)	99.00	58.00	99.67
(5, 0°)	92.33	53.33	94.00
(5, 45°)	96.67	44.00	98.33
(5, 90°)	98.33	50.00	98.67
(5, 135°)	96.67	45.67	97.00

Tabulka 5.5: Vliv nastavení offsetu matice sousednosti na úspěšnost klasifikace

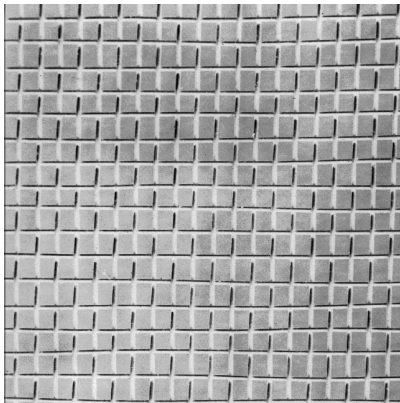
Závislost úspěšnosti klasifikace na nastavení parametrů matice sousednosti ukazuje Tabulka 5.5. Vyšší přesnosti klasifikace dosahují příznaky obdržené při použití vzdálenosti ($d = 1$). Získané výsledky také ukazují, že charakter příznaků není vhodný pro klasifikaci dle k-nejbližšího souseda.

Bližší pohled na výsledky klasifikace pro kombinaci Bayesův klasifikátor-matice sousednosti dává matice záměn (5.2.2). Vybrali jsme offset $(5, 0^\circ)$, jelikož v rámci

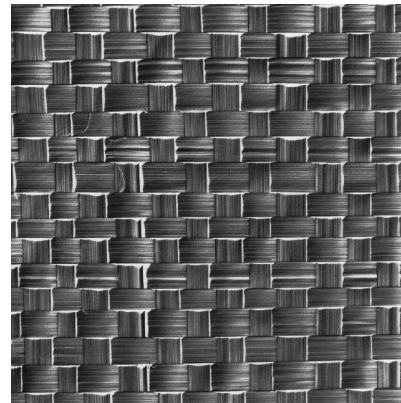
testování Bayesova klasifikátoru dosahuje nejnižší přesnosti.

$$CM = \begin{pmatrix} 17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 25 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 29 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 30 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 29 & 0 & 0 \\ 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 27 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 30 \end{pmatrix} \quad (5.2.2)$$

Důvodem klasifikačních nepřesností mezi 1. a 9. třídou (cihly, ratan) při použití offsetu $(5, 0^\circ)$ může být podobnost horizontálních prvků, konkrétně světlých čar, v obou třídách, viz Obrázky 5.9 a 5.10. Zkoumaný offset klade důraz právě na horizontální prvky textury, vše je umocněno použitou vzdáleností 5 pixelů. I přes tento problém byla více než polovina testovaných vzorků z první třídy klasifikována správně, což poukazuje na vysokou kvalitu popisu textur pro metody druhého řádu.



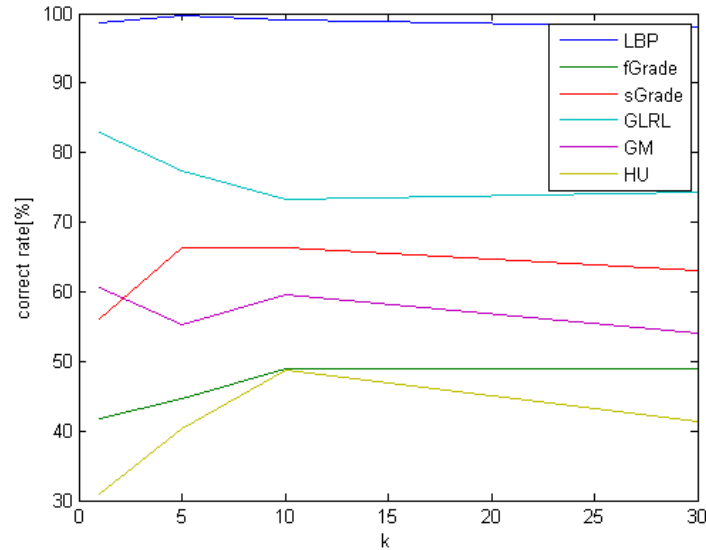
Obrázek 5.9: cihly Brodatz



Obrázek 5.10: ratan Brodatz

Při testování vlivu počtu nejbližších sousedů na přesnost klasifikace dosahuje metoda LBP opět nejlepších výsledků. Signifikantní změnou oproti předchozím testům na polymerech je zvýšená přesnost klasifikace pro metody založené na výpočtu matice sousednosti. Můžeme usuzovat, že tyto metody jsou vhodné pro popis členitých textur. Obrázek 5.11 vyjadřuje závislost úspěšnosti klasifikace dle k-nejbližšího

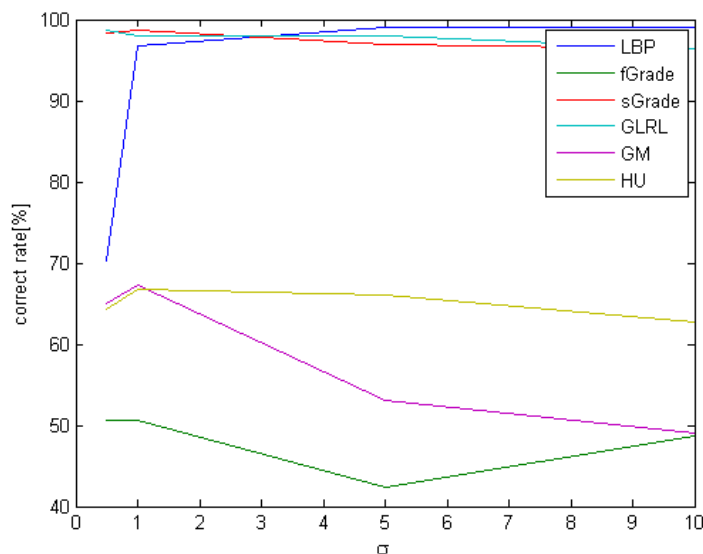
sousedů na počtu nejbližších sousedů pro jednotlivé metody extrakce příznaků. Nejvhodnějším algoritmem popisu textur pro zmíněný tento klasifikátor je LBP. Na rozdíl od testů provedených na obrazech polymerů, počet k-nejbližších sousedů nemá příliš vliv na přesnost klasifikace.



Obrázek 5.11: Vliv počtu nejbližších sousedů na přesnost klasifikace (Brodatz)

Obrázek 5.12 ukazuje úspěšnost klasifikace v závislosti na změně parametru $\sigma = 0.5, 1, 5, 10$. Úspěšnost klasifikace při použití LBP se zvyšuje s rostoucím σ , naopak při aplikaci geometrických momentů klesá. Velmi dobrých výsledků dosahují algoritmy druhého řádu a GLRL, které změna parametru σ nijak neovlivní.

Na závěr provedeme opět celkové vyhodnocení použitých algoritmů v základním nastavení. Z Tabulky 5.6 je patrné, že klasifikace textur z galerie Brodatz je celkově přesnější oproti obrazům polymerů, a to kvůli jejich strukturální rozmanitosti, a také kvůli nepřítomnosti šumu. Nejlepších výsledků pro tento typ dat dosahují algoritmy založené na výpočtu matice sousednosti, LBP a GLRL, naopak metody prvního řádu se zdají být pro tuto úlohu naprosto nevhodné. K tabulce s přehledem úspěšností jsme přidali informaci o průměrné době zařazení jednoho obrazu.

Obrázek 5.12: Vliv parametru σ na přesnost klasifikace (Brodatz)

	Bayes.klas.	k-NN	SVM
LBP	99.33	98.33	95.33
První řád	44.67	48.00	45.67
Matice sousednosti	98.00	66.00	98.33
GLRL	99.00	73.67	98.67
Geometrické momenty	65.67	62.33	68.00
Huovy momenty	65.00	44.67	71.33
čas klasifikace[s]	0.0001	0.0007	0.0249

Tabulka 5.6: Celkový přehled úspěšnosti a časů klasifikace (Brodatz)

5.3 Shrnutí dosažených výsledků

V této kapitole jsme testovali různé kombinace metod pro extrakci texturních příznaků, klasifikačních algoritmů a vlivy jejich nastavení na přesnost klasifikace. Všechny testy jsme prováděli na dvou skupinách obrazů (mikroskopické snímky polymerů, textury z galerie Brodatz). Nejvhodnějšími algoritmy pro klasifikaci polymerů jsou dle výsledků metody LBP a GLRL spolu s Bayesovým klasifikátorem či SVM. SVM má sice oproti Bayesovu klasifikátoru vyšší čas klasifikace, nicméně hodnoty kolem setin sekundy se projeví pouze při spouštění rozsáhlých testů. Při

klasifikaci jednotlivých obrazů, např. při průmyslovém provozu by byl nějaký rozdíl stěží patrný. Testy na texturách z galerie Brodatz potvrdily velmi dobrou popisovací schopnost metod LBP a GLRL. K algoritmům s nejlepšími výsledky na těchto datech patří také příznaky získané na základě výpočtu matice sousednosti.

6 Závěr

V této práci jsme se zabývali analýzou a testováním metod texturní analýzy. Práce přináší přehled běžně používaných algoritmů extrakce texturních příznaků, klasifikačních algoritmů a nástrojů pro evaluaci těchto metod. Dále jsme implementovali systém pro testování zmíněných algoritmů. Důležitou částí této práce je vyhodnocení testů pro parametrizaci a klasifikaci mikroskopických snímků polymerů za účelem rozpoznávání jejich fyzikálních vlastností na základě struktury. Tento problém jsme dále zobecnili na úlohu klasifikace textur. Pro obě skupiny dat jsme testovali vliv změny parametrů metod extrakce příznaků a klasifikačních algoritmů na úspěšnost klasifikace. Z testů jsme také vyvodili vhodnost použití těchto algoritmů pro specifické typy textur. Nejlepších výsledků ve všech testech dosahovala kombinace algoritmů LBP a Bayesova klasifikátoru. Tuto kombinaci můžeme označit také jako univerzální z hlediska použití pro různé typy textur. Očekávané dobré výsledky pro statistiky druhého řádu se potvrdily pouze na velmi různorodých texturách z galerie Brodatz, a také pouze v kombinaci s některými klasifikátory (např. SVM). Překvapivě dobrých výsledků dosáhly metody počtu opakování jasů (GLRL), naopak algoritmy popisu textur založené na geometrických momentech (geometrické momenty, Huovy momenty) v testech selhaly. Dosažené výsledky mohou sloužit při optimalizaci algoritmů texturní analýzy pro specifické typy textur. Do testovacího systému je samozřejmě možné přidat další algoritmy extrakce příznaků či klasifikace a testovat je na libovolných obrazech.

Do budoucna by bylo vhodné zaměřit se na propojení znalostí fyziky polymerů s algoritmy zpracování obrazu a na implementaci systému pro rozpoznávání fyzi-

kálních vlastností polymerů v praxi. Dále bychom mohli rozšířit testovací systém o další algoritmy pro popis textur a klasifikace, či provést testy na skupinách textur specifických pro různé oblasti (např. medicína, mikrobiologie, stavebnictví atp.)

Texturní analýza rozhodně není uzavřenou oblastí. Vývoj nových metod pro popis textur je jen otázkou času. Stále se objevují nové problémy řešitelné pomocí algoritmů zpracování obrazu (viz. rozpoznávání polymerů nebo texturní segmentace patologických jevů v medicíně) a popis objektů pomocí jejich struktury bude mít vždy jednu z největších vypovídacích hodnot.

A Přílohy

Přílohou této diplomové práce je CD, které obsahuje následující strukturu adresářů:

- **results** - výsledky testů na obrazech polymerů a obrazech z galerie Brodatz
 - polymery
 - brodatz
- **sample_data** - obrazová data použitá k testování
 - polymery
 - brodatz
- **scripts** - skripty v jazyce Matlab
 - classification - funkce pro klasifikaci
 - configuration - konfigurační soubory a funkce pro evaluaci
 - feature_extraction - funkce pro extrakci příznaků
 - * LBP
 - * statistical_methods
 - files - funkce pro práci se soubory
- **text** - obsahuje text diplomové práce
- **YAMLMatlab_0.4.1** - obsahuje nástroje pro práci s yaml soubory v Matlab

Literatura

- [1] Šonka, M.-Hlaváč, V.-Boyle, R.: *Image Processing, Analysis and Machine Vision*. Thomson 2008
- [2] Müller, L.: *Diagnostika a rozhodování(Učební text)*., Katedra kybernetiky, Fakulta aplikovaných věd, ZČU Plzeň 2010.
- [3] Železný, M.: *Zpracování digitalizovaného obrazu(Učební text)*., Katedra kybernetiky, Fakulta aplikovaných věd, ZČU Plzeň 2010.
- [4] MÄENPÄÄ, T.: *The Local Binary Pattern Approach to Texture Analysis - Extensions and Applications* University of Oulu 2003
- [5] Munoz-Rodriguez, J.-Asundi, A.-Rodriguez-Vera, R.: *Recognition of a light line pattern by Hu moments for 3-D reconstruction of a rotated object*. Elsevier 2005
- [6] Ojala, T.-Pietikäinen, M.Mäenpää, T.: *Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns*. IEEE 2002
- [7] Psutka, J.: *Učící se systémy a klasifikátory(Učební text)*., Katedra kybernetiky, Fakulta aplikovaných věd, ZČU Plzeň 2010.

-
- [8] Krontorád, J.: *Implementace algoritmu SVM v FPGA*. VUT Brno 2009.
- [9] Berka, P.: *Dobývání znalostí z databází (Učební text)*., Katedra informačního a znalostního inženýrství, Fakulta informatiky a statistiky, VŠE Praha.
- [10] Pons, M.-Belaroui, K-Simer, N.: *Characterisation of grinding media wear by visual texture.*, Elsevier 2004.
- [11] Arun Kumar, M.-Gopal, M.: *Reduced one-against-all method for multiclass SVM classification.*, Elsevier 2011.