

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

Bakalářská práce

Rozpoznávání obličejů pomocí metod založených na lokálních binárních vzorech

Originální zadání

Prohlášení

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

V Plzni dne 27. dubna 2017

Antonín Vrba

Poděkování

Děkuji vedoucímu bakalářské práce Doc. Ing. Pavlu Královi, Ph.D. za jeho podporu a čas strávený na přínosných konzultacích.

Abstract

This bachelor thesis deals with testing and enhancing methods based on Local Binary Patterns (LBP) in the face recognition task. The thesis theoretically describes the individual phases of recognition and the functioning principle of the LBP image descriptor. In addition, based on the LBP there are suggested and implemented two new methods for obtaining binary patterns from image data. An important part of the thesis is an application created for experimental testing of the proposed methods with the use of OpenCV library. For the testing part there was used the new face database Czech News Agency (ČTK) together with the FERET database. The subject of the experiments is the verification of the functionality of the methods and defining suitable parameters for high recognition success rate. The experiment results are evaluated and compared to results of other reference methods. The newly proposed methods outperformed the LBP reference method in both tested databases and new maximum success rate has been obtained for the database ČTK. The recognition methods are tested in the task of texture recognition.

Abstrakt

Tato bakalářská práce se zabývá testováním a vylepšením metod na bázi lokálních binárních vzorů (LBP) v úloze rozpoznávání obličejů. Práce nejdříve teoreticky popisuje jednotlivé fáze procesu rozpoznávání a také princip fungování obrazového deskriptoru LBP. Na jeho základě jsou dále navrženy a implementovány dva nové způsoby získávání binárních vzorů z obrazových dat. Součástí práce je vytvořena aplikace pro experimentální testování navržených metod s využitím knihovny OpenCV. Pro testování je převážně využívána obličejová databáze České tiskové kanceláře (ČTK) společně s databází FERET. Předmětem experimentů je ověření funkčnosti metod a zároveň určení vhodných parametrů pro co nejlepší úspěšnost rozpoznávání. Výsledky experimentů jsou zhodnoceny a porovnány s výsledky jiných referenčních metod. Nově navržené metody překonaly referenční metodu LBP na obou testovaných databázích a pro databázi ČTK byla dosažena nová maximální úspěšnost. Metody rozpoznávání jsou dále testovány v úloze rozpoznávání textur.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 1 |
| 2 | Rozpoznávání obličejů | 2 |
| 2.1 | Detekce obličejů | 2 |
| 2.2 | Proces rozpoznávání | 3 |
| 2.3 | Databáze obličejů | 4 |
| 2.3.1 | Nová databáze obličejů ČTK | 4 |
| 2.4 | Rozpoznávací metody | 5 |
| 2.4.1 | Metoda Eigenfaces | 6 |
| 2.4.2 | Lokální binární vzory | 7 |
| 2.4.2.1 | Vlastnosti | 7 |
| 2.4.2.2 | Obecná varianta LBP | 8 |
| 2.4.2.3 | Uniformní rozšíření LBP | 8 |
| 2.4.3 | POEM | 9 |
| 2.4.4 | Histogram jako příznakový vektor | 10 |
| 2.4.5 | Gáborovy filtry | 11 |
| 2.4.6 | Porovnání obličejů | 12 |
| 2.4.7 | Klasifikace | 13 |
| 3 | Knihovna OpenCV | 14 |
| 3.1 | Moduly knihovny | 14 |
| 3.2 | Načtení obrazových dat | 15 |
| 3.3 | Práce s obrazovými daty | 16 |
| 4 | Nově vytvořené deskriptory | 18 |
| 4.1 | Deskriptor E-LBP | 18 |
| 4.2 | Deskriptor S-LBP | 19 |
| 5 | Implementace aplikace | 21 |
| 5.1 | Požadavky na aplikaci | 21 |
| 5.2 | Struktura aplikace | 21 |
| 5.3 | Konfigurační soubor | 22 |

| | | |
|----------|--|-----------|
| 5.4 | Implementace Gáborových filtrů | 24 |
| 5.4.1 | Nalezení vhodných parametrů | 25 |
| 5.5 | Analýza časových nároků | 27 |
| 5.6 | Kompilace a spuštění | 28 |
| 6 | Experimenty | 30 |
| 6.1 | Metodika | 30 |
| 6.2 | Výsledky experimentů | 30 |
| 6.3 | Pravidelná mřížka podhistogramů | 31 |
| 6.3.1 | Rozměr mřížky a metody porovnání | 31 |
| 6.3.2 | Vzdálenost sousedních prvků | 32 |
| 6.3.3 | Počet sousedních elementů a jejich tvar | 33 |
| 6.3.4 | Generované konfigurace | 34 |
| 6.3.5 | Aplikace deskriptorů na databázi FERET | 35 |
| 6.4 | Gáborovy filtry | 37 |
| 6.4.1 | Počet nalezených bodů | 37 |
| 6.4.2 | Velikost histogramu | 38 |
| 6.4.3 | Velikost ohraničení nalezených bodů | 38 |
| 6.4.4 | Generované konfigurace | 39 |
| 6.4.5 | Aplikace Gáborových filtrů na databázi FERET | 40 |
| 6.5 | Rozpoznávání textur | 41 |
| 6.5.1 | Databáze DTD | 41 |
| 6.5.2 | Testovací protokol | 41 |
| 6.5.3 | Naměřené výsledky | 42 |
| 6.5.4 | Diskuze výsledků | 42 |
| 6.6 | Porovnání výsledků s existujícími pracemi | 44 |
| 7 | Návrh dalšího rozšíření | 46 |
| 8 | Závěr | 47 |
| | Literatura | 48 |
| | Přílohy | 49 |
| A | Obsah příloženého DVD | 50 |

1 Úvod

V posledních letech je vynaloženo stále více úsilí na vývoj nových metod pro rozpoznávání obličejů. Vyvíjené metody jsou testovány na různě obtížných obličejových databázích. Při dodržení testovacích protokolů jednotlivých databází lze rozpoznávací metody porovnat. Nejlepší metody musí pracovat s co nejnižší časovou náročností a vysokou úspěšností.

V rámci této práce se seznámíme s nově vzniklou obličejovou databází ČTK, dále budou prozkoumány metody rozpoznávání obličejů využívající lokální binární vzory. Na základě principu fungování již existujících obrazových deskriptorů budou navrženy dvě nové metody pro získávání binárních vzorů. S využitím knihovny počítačového vidění OpenCV a programovacího jazyka C++ bude vyvinuta aplikace, kde budou navržené deskriptory implementovány. Aplikace umožní testování úspěšnosti navržených metod rozpoznávání. Při experimentech na databázi ČTK budou nalezeny vhodné parametry v celém procesu rozpoznávání za účelem získání maximální možné úspěšnosti. Metody budou s nalezenými parametry aplikovány i na databázi FERET. Nad rámec zadání této práce budou metody otestovány v úloze rozpoznávání textur. Výsledky experimentů budou vyhodnoceny a porovnány s výsledky jiných prací.

Hlavním přínosem této práce je návrh nových metod rozpoznávání a také způsob nalezení jejich nejvíce vyhovujících parametrů. Jelikož pro obličejovou databázi ČTK existuje zatím jen málo výsledků napříč všemi druhy rozpoznávacích algoritmů, je možné tuto práci považovat za přínos i v této oblasti.

2 Rozpoznávání obličejů

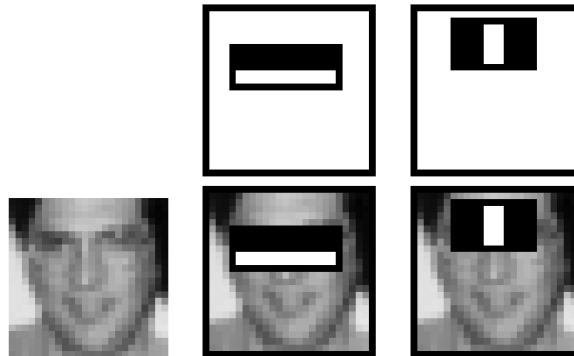
Identitu člověka lze určit na základě různých biometrických faktorů. Existuje mnoho používaných metod jako jsou například: otisk prstu, oční sítnice nebo dynamika chůze. Jedním z nejžádanějších způsobů identifikace poslední doby je rozpoznávání osob podle obličeje. Snažíme se počítač naučit rozpoznávat obličeje stejně dobře nebo i lépe než dokážeme my rozpoznávat osoby z našeho okolí. V posledních letech se také tato disciplína stala velmi atraktivní i pro výrobce kamerových a bezpečnostních systémů. S nárůstem výpočetního výkonu a razantním zlepšením kvality záznamu se na trhu objevují první „smart“ bezpečnostní systémy s podporou rozpoznávání osob. Avšak i po 45 letech výzkumu v této oblasti jsou rozpoznávací výsledky v obtížných podmínkách stále značně nespolehlivé. Je zde tedy prostor pro další výzkum a návrh nových metod rozpoznávání.

2.1 Detekce obličeje

Počátečním problémem rozpoznávání je nalezení obličeje ve vstupních obrazových datech. Často užívanou metodou pro detekci libovolných objektů je metoda Haarových kaskád známá též jako Viola-Jones algoritmus, který byl představen v roce 2001 Viola – Jones [13]. Princip metody spočívá v natrénování klasifikátoru, který v trénovacích datech (obličejích) nachází Haarovy vzory - primitiva reprezentující sumu jasových složek pod ním viz Obrázek 2.1. Tato primitiva a jejich pozice se nacházejí uvnitř trénovacího rámce o velikosti např. 24×24 px. Tento rámec se poté posouvá po vstupním obraze a testuje se umístění sady naučených primitiv. Vstupní obraz se může předzpracovat na tzv. integrální tvar, který obsahuje v pravidelných bodech informaci o lokálním součtu hodnot pixelů. Vzory lze aplikovat na určnou oblast v kaskádách podle jejich významu, nejdříve jsou tedy testována velká primitiva a až po jejich úspěšném nalezení potenciálního obličeje je použita další rozšířená sada primitiv, která může případně úspěšnou detekci více zpřesnit nebo vyvrátit. Při použití pouhých 200 vzorů byla na frontálních obličejových datech naměřena úspěšnost 95%. Pro obrázek s rozlišením 384×288 px trvala detekce 0.7 sekundy.

Práce z roku 2011 Oro et al. [10] poukazuje na výpočetní náročnost ob-

ličejové detekce v reálném čase a také je pro nás ukazatelem desetiletého pokroku od uvedení algoritmu [13]. Byl zpracováván video stream s rozlišením 1920×1080 px při 25 snímcích za sekundu, datový tok činil 3520 Kb/s. Implementovaný systém dokázal analyzovat kontinuálně každý snímek videa, na který byl aplikován obličejový detektor na bázi Haarových kaskád s časem výpočtu menším než 40ms. Paralelní výpočty prováděla grafická karta s podporou technologie CUDA¹.



Obrázek 2.1: Základní vzory reprezentující rozdílný odstín očí proti tvářím nebo nosu, převzato z [13].

2.2 Proces rozpoznávání

Rozpoznávání obličeje je proces, který na základě vstupních dat určí identitu hledané osoby. Určení identity může a nemusí být správné. Úpravou jednotlivých funkčních částí procesu a opakováním experimentu je usilováno o nejvyšší možnou správnost výsledku. Hlavní funkční celky procesu rozpoznávání jsou rozděleny na následující části:

1. **Načtení vstupních dat z databáze obličejů**
2. **Předzpracování obrazu** - Obličej je detekován, provede se normalizace a oříznutí. Často se obrázek převádí na jednonábový barevný model (šedotónový).
3. **Extrakce příznaků** - Z obrazu je extrahován příznakový vektor pomocí zvoleného deskriptoru.

¹Architektura pro spouštění programů v C/C++ na grafickém HW.

4. **Klasifikace** - Testovaným obličejům je přiřazena identita zvoleným klasifikátorem.

5. **Vyhodnocení úspěšnosti**

2.3 Databáze obličejů

Pro testování a ladění metod rozpoznávání obličejů vznikla v průběhu let celá řada databází. Součástí databáze jsou trénovací a testovací data. K jedné rozpoznávané osobě většinou patří více než jeden obrázek z dat trénovacích. Obličejové sady se liší obtížností i svým rozsahem. V práci Lenc – Král [7] jsou uvedeny známé a stále používané databáze:

- **AT&T** - Databáze též známá jako databáze ORL. Obrázky byly pořízeny během let 1992 až 1994. Obsahuje snímky 40 lidí, 10 pro každou osobu s černým pozadím. Fotografie zachycují různá natočení a výrazy v obličejí. Rozlišení obrázků je 92×112 px.
- **FERET** - Obsahuje 14 051 snímků 1 199 osob pořízených během let 1993 až 1996 v rozlišení 256×384 px. Snímky jsou rozděleny do jednotlivých kategorií podle natočení obličejí, pro hlavní kategorii čelních snímků se rozlišuje trénovací skupina *fa* (1196 osob) a jednotlivé testovací skupiny *fc*, *fb*, *dup1* a *dup2*. Použití rozpoznávacích algoritmů na všechny testovací skupiny poté v úspěšnosti odráží i schopnost rozpoznávání natočených obličejů.
- **AR** - Čítá více než 4 000 barevných snímků 126 osob z roku 1998. Rozlišení snímků je 768×576 px. Fotografie jsou pořízeny v různých světelných podmínkách a s různým výrazem v obličejí. Jsou zde zastoupeny i osoby s brýlemi nebo se zahalenými částmi obličejí.

2.3.1 Nová databáze obličejů ČTK

Práce Lenc – Král [7] představuje nově vzniklou obličejovou databázi (2015). Je tvořena reálnými fotografiemi z fotoarchivu ČTK². Databáze má dvě části:

²Česká tisková kancelář

1. Unconstrained Facial Images (UFI) - Obsahuje obrázky 530 osob o velikosti 384×384 px. Ke každé osobě je přiřazeno v průměru 8.2 fotografií. Obličeje jsou na libovolných místech fotografie viz Obrázek 2.2 v proměnlivých velikostech. Pro zpracování takové databáze je nejdříve nutné obličeje detekovat a vhodně obrázky předzpracovat.
2. Cropped images - Obsahuje fotografie 605 osob v rozlišení 128×128 px. Každá osoba má v trénovací sadě průměrně 7.1 fotografií. Tvorba této sady zahrnovala detekování obličeje Viola-Jones algoritmem a nalezení pozice očí. Předzpracování spočívalo v rotaci obličeje podle horizontální linie očí a následné škálování na požadovanou velikost 128×128 px viz Obrázek 2.3. Používání této sady dat nevyžaduje vlastní implementaci předzpracování, které by výrazně ovlivnilo výslednou úspěšnost.



Obrázek 2.2: Ukázka UFI obrázku, převzatu z [8].



Obrázek 2.3: Cropped obrázek, převzato z [8].

ČTK databáze je označena jako obtížná a to hlavně kvůli použitým reálným fotografiím nižší kvality. Článek [6] srovnává výsledky základních a rozšířených rozpoznávacích metod na databázích: AT&T, FERET, AR a ČTK. Výsledky podle neupravené „baseline“ verze algoritmu [2] z roku 2004 jsou v pořadí vyjmenovaných databází: 56.17 %, 93.89 %, 87.71 %, 39.81 %. Zde je vidět markantní rozdíl úspěšnosti na nově vytvořené databázi proti stávajícím. Databáze se tak stává velkou výzvou pro nově vzniklé metody rozpoznávání.

2.4 Rozpoznávací metody

Dobrá reprezentace objektu, v našem případě obraz obličeje, je klíčová pro dobrý systém rozpoznávání. Setkáváme se s problémy, které části obrazu jsou

podstatné pro danou úlohu a jak je správně extrahovat. Efektivní deskriptor by měl spolehlivě reprezentovat svůj zdrojový obraz, aby bylo co nejnadhodnější určit do jaké třídy patří. Dobrý deskriptor si lze představit tak, že v n -rozměrném prostoru bude obličej dvou a více osob separovat do vlastních shluků (tříd). Mělo by docházet k co nejmenšímu počtu záměn mezi třídami. Výhodou deskriptoru je také jednoduchost jeho výpočtu. Jedna z prvních metod, která se ukázala v tomto smyslu jako úspěšná se nazývá Eigenfaces a byla představena již v roce 1991. O několik let později byla představena metoda lokálních binárních vzorů (LBP) a překonáním řady dříve známých metod v oblasti rozpoznávání obličejů ukázala směr dalšího vývoje deskriptorů. V článku Zhang et al. [15] z roku 2010 je kromě výčtu dříve používaných deskriptorů představen nový deskriptor LDP. Ve stejném roce byl také představen pokročilý deskriptor POEM v práci Vu – Caplier [14] viz sekce 2.4.3 jenž v práci Košar [5] dosahuje lepších výsledků než LBP i LDP.

2.4.1 Metoda Eigenfaces

V práci Turk – Pentland [12] byla poprvé představena metoda Eigenfaces. Vstupem metody je N obličejů z trénovací sady. Pro každý obličej s velikostí $K \times K$ pixelů je vytvořen vektor o K^2 dimenzích a obsahuje všechny hodnoty z obrázku (po řádcích). Vektory jsou umístěny do matice o rozměru $K^2 \times N$, kde každý obličej reprezentuje jeden sloupec matice. Na tuto matici je aplikována metoda zvaná PCA (analýza hlavních komponent), rozměr matice je redukován na požadované m tedy $K^2 \times m$. Sloupce redukované matice jsou tzv. eigen-vektory, jež lze zpět vizualizovat viz Obrázek 2.4.



Obrázek 2.4: Vizualizace eigen-vektorů, převzato z [12].

Převědeme-li takto všechny sloupce na obrazová data, dostáváme sadu obrazů reprezentující změny proti průměrnému obličejí ze vstupních dat. Nově rozpoznávaný obličej poté vyjádříme (projektujeme) jako součet průměrného obličejí a lineární kombinace eigen-vektorů viz Obrázek 2.5. Z kombinace

eigen-vektorů a jejich zastoupením v reprezentaci neznámého obrazu lze získat procentuální vyjádření o podobnosti k osobě ze vstupních (trénovacích) dat. Velkou nevýhodou této metody je citlivost na změnu světelných podmínek. Další nevýhoda je přidávání obličejů do naučených eigen-vektorů, je pak nutné přepočítání celé matice.



Obrázek 2.5: Projekce rozpoznávaného obrázku (nalevo) do prostoru eigen-vektorů, převzato z [12].

2.4.2 Lokální binární vzory

2.4.2.1 Vlastnosti

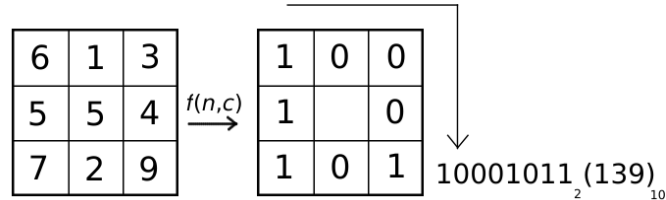
Deskriptor obrazu využívající LBP (lokální binární vzory - Local Binary Patterns) byl představen v roce 1996 v práci Ahonen et al. [2]. Nevyužívá žádných matematických metod pro snížení počtu dimenzí, ale vytváří pro každý trénovací i testovací obrázek vhodný popis - deskriptor, který lze následně porovnávat s ostatními. Operátor LBP popisuje jednotlivé pixely v obrazu porovnáním hodnot centrálního pixelu a jeho okolí v matici 3×3 px. Každému pixelu je přiřazena dekadická hodnota, která se poté stává členem histogramů, jenž obrázek popisují.

Pro určení binární hodnoty v rámci jednoho porovnání centrálního a sousedního pixelu je použit vztah 2.1, parametry c a n jsou hodnoty pixelů.

$$f(n, c) = \begin{cases} 1 & n \geq c \\ 0 & n < c \end{cases} \quad (2.1)$$

Na Obrázku 2.6 je znázorněn výpočet binární hodnoty popisující centrální pixel. Významnou výhodou LBP operátoru je invariace vůči světelné intenzitě. Pokud rovnoměrně osvětlíme zkoumanou strukturu, výsledné LBP vzory

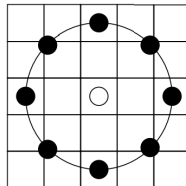
budou mít stejnou hodnotu. Dalšími výhodami tohoto deskriptoru je velice jednoduchý výpočet a možnost deskriptor dále modifikovat. Podle binární sekvence kolem středového pixelu lze určit, zda se středový pixel nachází u hrany přechodu barev nebo je součástí stejnobarevné plochy.



Obrázek 2.6: Základní LBP operátor

2.4.2.2 Obecná varianta LBP

Základní LBP operátor lze zobecnit přidáním dvou parametrů (P, R) . Kde P označuje počet bodů a R poloměr kružnice, na které se body nacházejí. Operátor poté označujeme jako $LBP_{P,R}$ viz Obrázek 2.7. Před aplikací operátoru se provádí interpolace bodů na kružnici do tabulky pozic, která se poté používá pro výběry hodnot pixelů na konkrétní pozici.



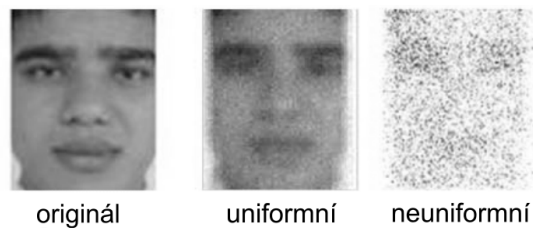
Obrázek 2.7: Obecný $LBP_{8,2}$ operátor, převzato z [2].

2.4.2.3 Uniformní rozšíření LBP

Pokud číslo P udává počet sousedních pixelů v okolí R , potom histogram četností, pokud $(P = 8)$ musí nutně mít 2^P pozic. Můžeme však tento počet výrazně zmenšit a uvažovat jen taková 8-bitová čísla, která mají žádný nebo maximálně 2 přechody z hodnoty 0 na 1 a obráceně, pokud se na číslo díváme jako na cyklický buffer. Příkladem takových uniformních čísel jsou: 00000000, 11110000, 10000001. Víme, že taková čísla reprezentují podstatné informace z obrázku (hrany, rohy, linie nebo neměnnou plochu). Histogram

lze díky této vlastnosti zmenšit na pouhých 58 pozic, vzory které nesplňují uniformní podmínky jsou umístěny na přidanou pozici 59. Počet uniformních vzorů lze vypočítat pomocí vzorce $P(P - 1) + 2$. Pro 16 sousedních pixelů je tedy redukován počet binů³ z 65536 na 242, což představuje obrovskou paměťovou úsporu za cenu malé ztráty obrazových dat viz Obrázek 2.8, na kterém jsou vykresleny pouze pixely, pro které byla hodnota uniformní nebo neuniformní.

Ahonen et al. [2] ukázali v experimentech s texturami, že 90 % vzorů je uniformních při použití základního LBP operátoru $LBP_{8,1}$ a okolo 70 % při $LBP_{16,2}$.



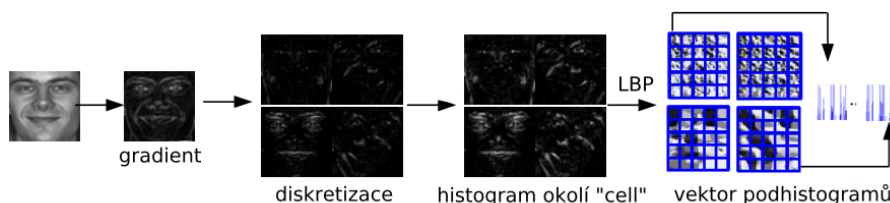
Obrázek 2.8: Vykreslení pouze uniformních a neuniformních pixelů, převzato z [11].

2.4.3 POEM

Tento deskriptor s názvem Patterns of Oriented Edge Magnitudes je robustní deskriptor využívající LBP, je schopný získat bohatší informace o vstupním obrazu. V práci [14] je popsán způsob výpočtu deskriptoru a také výsledky experimentů. Obraz je nejdříve převeden na gradientní podobu a každý pixel reprezentuje 2D vektor, jehož velikost odpovídá intenzitě pixelu v gradientním obraze. Dále je obraz diskretizován do podoby, kdy uvažujeme jen určitý počet p dovolených směrů vypočítaných vektorů viz 2.9. Pro každý pixel je přiřazen vektor reprezentující lokální histogram, který má p položek (binů) a obsahuje četnosti směru okolních vektorů, toto okolí se nazývá „cell“. Hodnoty jsou dále zakódovány pomocí již známé metody $LBP_{P,R}$, kde R označuje poloměr oblasti známé též jako „block“ a uloženy do dílčích histogramů viz sekce 2.4.4. Pro každý pixel není zakódována jen lokální hodnota popisující tvar nebo přechod jako je tomu při použití samnotného LBP deskriptoru, ale je zde také vztah informací mezi okolními regiony v obrázku.

³určitý interval histogramu

Díky této schopnosti se dokáže POEM velice dobře vyrovnat s natočením obličejům, změnám osvětlení a různým výrazům v obličeji. Z experimentů [14] je patrné, že pro 3 diskretizační směry je vhodná velikost 7×7 px pro „cell“ a 10×10 px pro „block“, na obličejové databázi FERET bylo tímto deskriptorem dosaženo až o 45 % lepších výsledků než s použitím klasické metody LBP (obličejový set fc s proměnlivým osvětlením).



Obrázek 2.9: Výpočet POEM deskriptoru, převzato z [14].

2.4.4 Histogram jako příznakový vektor

Po výpočtu LBP vzorů pro celý obrázek lze vytvořit histogram četností výsledných vzorů. Tento přístup se používá při popisu jednoduté struktury, ale pro obličejová data není vhodný. Obličej nás zajímá z pohledu prostorového rozložení různých elementů, které dokážeme popsat lokálními četnostmi vzorů. Rozložením reprezentace obličeje na vhodný počet podhistogramů libovolné velikosti lze dosáhnout dobrého popisu obličeje. Složení podhistogramů za sebe poté utváří příznakový vektor. Práce Košář [5] se zabývá srovnáním různých obrazových deskriptorů s použitím pravidelné mřížky podhistogramů. Z výsledků lze usoudit, že existuje optimální velikost pravidelné $n \times n$ mřížky podhistogramů, která na konkrétní databázi vykazuje nejvyšší úspěšnost.

Zpřesnění popisu obličejových dat je dále možné pomocí váhování jednotlivých podhistogramů. Důležitým oblastem na obrázku lze přiřadit různou váhu např. oblast očí, úst atd. V práci [5] se díky experimentálnímu genetickému algoritmu podařilo nalézt masky vah, které dokáží zvýšit úspěšnost rozpoznávání až o 9 %. V práci López [9] je představena maska, která upřednostňuje obličejové prvky a vynechává oblasti s malou rozpoznávací hodnotou viz Obrázek 2.10. Přiřazení vah koresponduje s Obrázkem 2.11.

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 4 | 4 | 1 | 4 | 4 | 4 |
| 4 | 4 | 4 | 2 | 4 | 4 | 4 |
| 1 | 1 | 2 | 2 | 2 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 3 | 3 | 3 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |

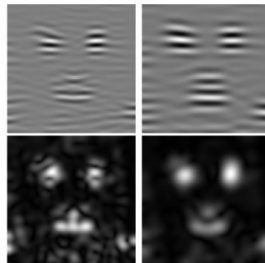
Obrázek 2.10: Váhy pro jednotlivé podhistogramy, převzato z [9].



Obrázek 2.11: Mřížka podhistogramů 7×7 na obličejí po aplikaci $LBP_{8,2}$, převzato z [9].

2.4.5 Gáborovy filtry

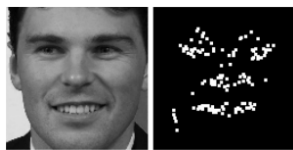
Práce Lenc – Král [6] se zabývá nahrazením pravidelné podhistogramové mřížky soustavou úmyslně pozicovaných podhistogramů. Podhistogramy jsou umístovány na „zajímavá“ místa v obrázku na základě použití Gáborových filtrů. Gáborovy filtry se používají v obrazové analýze kvůli své schopnosti zachytit důležité informace např. detekovat hrany v obrázku a tím nám mohou po vhodném nastavení parametrů zvýraznit oblasti obličeje jako jsou oči, ústa nebo nos. Parametry se skládají z vlnové délky, orientace, poměru stran a propustnosti. Na Obrázku 2.12 jsou v horní části zobrazeny samotné Gáborovy funkce a poté i jejich aplikace na obrázek obličeje.



Obrázek 2.12: Příklady obrazu filtrovaného pomocí Gáborových filtrů, převzato z [6].

Nalezení výchozích bodů pro umístění podhistogramů lze provést metodou shlukování. Po aplikaci Gáborových filtrů je provedeno prahování výsledného obrazu a na pozice bodů s vysokou intenzitou je aplikován shlukovací algoritmus K-means, který redukuje počet těchto bodů na požadovanou hodnotu (viz Obrázek 2.13). Jelikož jsou Gáborovy filtry náročné na výpočet lze pro všechny obličeje vytvořit globální masku výchozích bodů na základě náhodně vybraného vzorků obličejů z trénovacích dat. Z výsledků práce [6] je vidět signifikantní nárůst úspěšnosti při experimentech na obličejové data-

bázi ČTK, kde bylo dosaženo o 20 % větší úspěšnosti než s použitím metody LBP s pravidelnou mřížkou podhistogramů.



Obrázek 2.13: Originální obrázek a jemu přiřazených 64 bodů, převzato z [6].

2.4.6 Porovnání obličejů

Uvážíme-li metodu extrakce uniformních příznaků $LBP_{8,1}$ a pravidelnou mřížku podhistogramů o rozměru 10×10 , výsledný příznakový vektor bude mít 100×59 položek. Vzdálenost mezi příznakovými vektory určujeme součtem vzdáleností jejich jednotlivých podhistogramů. K příznakovému vektoru (histogramu) přistupujeme stejně jako k dvourozměrnému poli $A[i][j]$, kde i označuje index podhistogramu a j index hodnoty konkrétního podhistogramu. Dále jsou uvedeny dvě používané metody pro porovnání histogramů A a B :

1. **Euklidovská vzdálenost (ED)** je jednou z neznámějších metod k určení vzdálenosti v n -rozměrném prostoru, která je mezi dvěma histogramy H_1 a H_2 dána vztahem:

$$d(H_1, H_2) = \sqrt{\sum_{j=0}^n (H_1[j] - H_2[j])^2} \quad (2.2)$$

Celková vzdálenost je poté určena součtem k histogramů:

$$ED(A, B) = \sum_{i=0}^k d(A[i], B[i]) \quad (2.3)$$

2. **Intersekcce histogramů (HI)** se dříve využívala při vyhledávání podobných obrázků v rozsáhlých databázích (image retrieval) je dána předpisem:

$$HI(A, B) = \sum_{i=0}^n \left[\sum_{j=0}^k A[i][j] - \sum_{j=0}^k \min(A[i][j], B[i][j]) \right] \quad (2.4)$$

2.4.7 Klasifikace

Běžně používaným algoritmem pro klasifikaci je 1NN - Nejbližší soused (Nearest-Neighbor) klasifikátor. 1NN klasifikace obličejů spočívá v porovnání a nalezení minima vzdálenosti se všemi obličejů v databázi. Pokud databáze obsahuje více záznamů hledaného obličejů, lze klasifikátor upravit na K -Nearest-Neighbor, přiřazení identity se následně vyhodnocuje z N nejbližších výsledků a je přiřazena třída (identita) s nejvyšší četností. Na konci procesu rozpoznávání se provede validace výsledků a výstupem je číslo p reprezentující úspěšnost rozpoznávání viz vztah 2.5, kde c je počet dobře identifikovaných osob a b počet chybných.

$$p = \frac{c}{c + b} \quad p \in [0, 1] \quad (2.5)$$

3 Knihovna OpenCV

Knihovna Open Source Computer Vision Library (OpenCV)¹ je volně dostupná knihovna určená pro úlohy počítačového vidění psaná nativně v jazyce C++. Obsahuje řadu optimalizovaných algoritmů a datových struktur pro práci s obrazovými daty. Algoritmy mají uplatnění v detekci a rozpoznávání obličejů, sledování objektů, identifikace osob, sledování pohybu očí a dokonce pro práci s virtuální realitou. Knihovna je neustále vyvíjena, poslední verze 3.2 je dostupná na všechny větší platformy: Windows, Linux, Mac, Android a IOS. Knihovna podporuje programovací jazyky: C++, C, Python, Java a MATLAB [1]. OpenCV se díky své velké komunitě a podrobné dokumentaci stává nejčastější volbou pro experimentální i komerční vývoj aplikací v oblasti počítačového vidění od roku 1999, kdy byla uvolněna první alpha verze. V dalších sekcích jsou popsány hlavní moduly a základní práce s knihovnou.

3.1 Moduly knihovny

OpenCV má modulární strukturu složenou ze sdílených² a statických³ knihoven [4]. Pro rozpoznávání obličejů lze využít následující moduly:

- **Core functionality** - datové struktury, vícerozměrná pole `cv::Mat`, základní funkcionality pro ostatní moduly
- **Image processing** - lineární a nelineární filtry, transformace obrazu, převody barevných modelů, histogramy
- **video** - analýza videa, sledování objektů
- **features2d** - deskriptory a hledání jejich společných vlastností
- **objdetect** - detekce objektů (obličejů, očí, osoby, auta ...)
- **highgui** - základní prvky grafického uživatelského rozhraní (okna, posuvníky ...)

¹<http://opencv.org/>

² *.so soubory (Linux) pro referencování kódu za běhu programů

³ *.a soubory (Linux) přímo linkované do programu během kompilace

- **Video I/O** - záznam videa a kodeky
- **gpu** - akcelerace výpočtů přes grafický HW (CUDA, OpenCL⁴)

3.2 Načtení obrazových dat

Pro načítání obrazových dat do paměti je v OpenCV k dispozici datová struktura *Mat*, kterou lze chápat jako n -rozměrné pole numerických datových typů. V případě dvourozměrného pole jsou v paměti uloženy data po řádkách. Pro reprezentaci obrazových dat je jako prvek matice zvolen vektor splňující svou velikostí požadavky na barevný model např. RGB. Výpis 3.1 obsahuje ukázkou C++ kódu s inicializací matice M o rozměru 100×100 a následným zapsáním číselné hodnoty na zvolenou pozici. Barevný model *CV_8U* podporuje bezznaménkové hodnoty v intervalu $\langle 0, 255 \rangle$.

```
cv::Mat M(100, 100, CV_8U);
M.at<unsigned char>(50, 50) = 255;
```

Výpis 3.1: Inicializace matice a zápis dat.

OpenCV podporuje načítání řady obrazových formátů např. BMP, PGM nebo JPEG. Funkce *imread* načte obrazová data ze souboru. Konstantním parametrem funkce jako je *CV_LOAD_IMAGE_GRAYSCALE* bude zajištěna barevná konverze do černobílého modelu. Výpis 3.2 obsahuje ukázkou kódu s načtením obrázku a následným zobrazením pomocí grafického rozhraní, které knihovna nabízí.

```
cv::Mat image = cv::imread("image.pgm",
    CV_LOAD_IMAGE_GRAYSCALE);
cv::namedWindow("Window", cv::WINDOW_AUTOSIZE);
cv::imshow("Window", image);
cv::waitKey(0);
```

Výpis 3.2: Načtení a zobrazení obrazových dat.

Funkce *namedWindow* registruje podle identifikátoru okno, které je následující funkcí *imshow* zobrazeno s určeným obsahem. Funkce *waitKey* pouze čeká na stisknutí libovolné klávesy pro zavření zobrazeného okna.

⁴paralelní programování heterogenních počítačových systémů

3.3 Práce s obrazovými daty

Výhodou použití knihovny OpenCV je vyřešená správa paměti pro všechny použitelné objekty. Programování aplikací tak nemusí obsahovat dodatečnou dealokaci paměťových zdrojů, která by byla nutná v případě vlastní implementace. Při práci s obličejovými daty lze při předzpracování obrazu využít funkcí jako je oříznutí, škálování nebo aplikace řady filtrů.

Výpis 3.3 obsahuje ukázkový kód, ve kterém jsou z načteného obrázku *image* pomocí dostupných funkcí odvozeny další 4 objekty *Mat*. Oříznutý obrázek *cropped* je vytvořen předáním obdelníkového tvaru *Rect* se souřadnicemi levého horního rohu a velikostí. Při výpočtu obrazových deskriptorů je nutné obrázek rozšířit o několik pixelů v závislosti na použitém deskriptoru. Funkce *copyMakeBorder* zvětší vstupní obrázek o 10 px na každé straně pomocí techniky zrcadlení, určené konstantou *BORDER_REFLECT*. Pro prahování obrázku je připravena funkce *threshold*, která hodnotí hodnotu každého pixelu podle parametrů a nastaví příslušnou binární hodnotu. Jako poslední je uvedena funkce *blur*, která provádí rozmazání obrázku podle velikosti zadané v parametru, tento filtr lze použít pro redukci šumu obrázku.

```
cv::Mat image, cropped, bordered, thresholded, blurred;

cropped = image(cv::Rect(15, 30, 100, 120));
cv::copyMakeBorder(image, bordered, 10, 10, 10, 10, cv::
    BORDER_REFLECT);
cv::threshold(image, thresholded, 66, 255, 0);
cv::blur(image, blurred, cv::Size(5,5));
```

Výpis 3.3: Aplikace funkcí na načtený obrázek *image*.

Pro zobrazení vytvořených obrázků je nutné spojit všechny do výsledné matice *target* viz následující Výpis 3.4. Aby je bylo možné spojit, je potřeba změnit velikost obrázků *bordered* a *cropped* pomocí funkce *resize* na původní hodnotu. Po vytvoření vektoru matic je využita funkce *hconcat* pro horizontální sjednocení obrázků. Poslední funkce *putText* demonstruje vložení textu. Na Obrázku 3.1 je zobrazena matice *target* a tedy výsledek všech uvedených funkcí.


```

cv::resize(bordered, bordered, image.size());
cv::resize(cropped, cropped, image.size());

std::vector<cv::Mat> matrices = {
    image, cropped, bordered,
    tresholded, blurred
};

cv::Mat target;
hconcat(matrices, target);
cv::putText(target, "ORIGINAL", cvPoint(35, image.size().
    height - 7), CV_FONT_VECTOR0, 0.4, 255, 1, CV_AA);

```

Výpis 3.4: Vytvoření výsledné matice *target* z matic odvozených.



Obrázek 3.1: Výsledek ukázkových kódů pro práci s obrazovými daty. Zleva: *image*, *cropped*, *bordered*, *tresholded* a *blurred*.

4 Nově vytvořené deskriptory

4.1 Deskriptor E-LBP

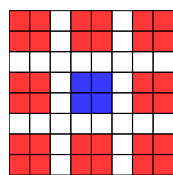
Deskriptor E-LBP (rozšířené - Enhanced LBP) je navržený tak, aby lokální vzory reprezentovaly větší část svého okolí. Na rozdíl od kruhového okolí v LBP má okolí operátoru E-LBP čtvercový tvar. Toto je dosaženo zvětšením elementů pro výpočet binární hodnoty. Středové a sousední elementy jsou tvořeny z čtvercových matic o rozměru 2×2 , 3×3 pixelů. Pro každou čtvercovou oblast je vypočítána průměrná hodnota. V původním návrhu metody byl zvažován ještě parametr pro volbu způsobu vyhodnocení hodnoty oblasti (průměr, maximum a minimum), ale bez průměrné hodnoty není možné určit chování intenzity v okolí bodu, a proto byl tento parametr vyřazen. Výpočet binárního popisu je totožný s metodou LBP, ale výsledná reprezentace je zapsána do všech pozic centrálního elementu. V případě rozměru 2×2 jsou zapsány 4 stejné hodnoty a vizuální reprezentaci deskriptoru tak lze znázornit s použitím čtvrtiny pixelů vstupního šedotónového obrazu.

Deskriptor $E - LBP(C, N, r)$ má následující parametry:

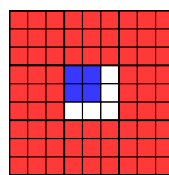
- Počet bodů centrální a sousední oblasti - C, N
- Vzdálenost mezi centrálním elementem a okolím - r
- Uniformita binárních vzorů - explicitně určena

Na Obrázku 4.1 je znázorněna konfigurace E-LBP operátoru s velikostí čtverců 2×2 pixelů. Vzdálenost okolních prvků je nastavena na hodnotu 3. Z výchozí pozice x, y levého horního rohu centrálního elementu jsou určeny výchozí souřadnice pro všech 8 sousedních čtverců. Obrázek 4.2 se liší pouze velikostí sousedních matic. Změna velikostí elementů a jejich vzdálenost má vliv na celkovou úspěšnost rozpoznávací metody, pro nalezení vhodných parametrů je nutné provést experimenty.

Lze očekávat, že větší jádro E-LBP bude zvýrazňovat dominantní části v obrazu. Na Obrázku 4.3 je vidět, že originální metoda LBP zanechává ze své



Obrázek 4.1: $E - LBP(4, 4, 3)$



Obrázek 4.2: $E - LBP(4, 9, 3)$

podstaty velké množství informačního šumu, který je dobře patrný i v histogramu pod obrázkem. Vizualizace E-LBP deskriptoru potvrzuje, že dochází k odstranění informačního šumu a zvýraznění dominantních částí obrázku.



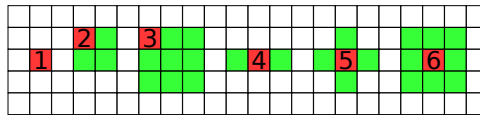
Obrázek 4.3: Vizualizace uniformních deskriptorů a jejich histogramů. Zleva: originál, $LBP_{8,1}$, $E - LBP(4, 4, 2)$, $E - LBP(9, 9, 3)$.

4.2 Deskriptor S-LBP

Dalším z nově navržených deskriptorů je S-LBP (rozšíření tvaru okolí - Shaped LBP). Tento deskriptor se snaží do jednoho pixelu vložit co nejvíce informací o svém rozšířeném kruhovém okolí. S-LBP přidává do obecné metody LBP možnost zvolení tvaru sousedních elementů, ze kterých se vypočítá průměrná hodnota pro porovnání. Čísla jednotlivých tvarů se udávají v parametrech metody viz Obrázek 4.4.

Deskriptor $S - LBP(N, r, S)$ má následující parametry:

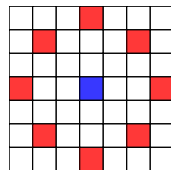
- Počet sousedních prvků - N
- Vzdálenost sousedních bodů od středu kružnice - r



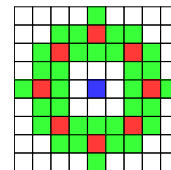
Obrázek 4.4: Volitelné tvary sousedních prvků metody S-LBP.

- Tvar sousedních prvků - S
- Uniformita binárních vzorů - explicitně určena

Parametry metody S-LBP dovolují vytvořit podstatně více validních konfigurací než je možné vytvořit u předchozí metody E-LBP. Na Obrázku 4.5 je znázorněna podoba operátoru LBP s osmi sousedními prvky a vzdáleností 3. Na Obrázku 4.6 je poté vyobrazen operátor S-LBP doplněný o tvar číslo 5. Okolí je tím rozšířeno o 32 pixelů, které se podílí na určení středové binární hodnoty. Snížením vzdálenosti nebo zvýšením počtu interpolovaných sousedů se mohou tvary začít překrývat a ubývá tak přidané plochy operátoru.



Obrázek 4.5: $LBP_{8,3}$



Obrázek 4.6: $S - LBP(8, 3, 5)$

Na Obrázku 4.7 jsou vizualizovány deskriptory s využitím operátorů z Obrázků 4.5 a 4.6. Obrázek ukazuje, že jsou si oba operátory velmi blízké, nicméně ne zcela totožné viz rozdíl obou operátorů. Podle charakteristiky rozdílu je možné soudit, že rozšířené okolí S-LBP operátoru vkládá do deskriptoru méně informací o plochách v obrázku s podobným odstínem šedé. Dochází tak k odstraňování šumu jako v metodě E-LBP, s tím rozdílem, že šum je odstraněn jen z určitých míst obrazu.



Obrázek 4.7: Vizualizace uniformních deskriptorů. Zleva: originál, $LBP_{8,3}$, $S - LBP(8, 3, 5)$ a jejich rozdíl.

5 Implementace aplikace

5.1 Požadavky na aplikaci

Součástí zadání této práce bylo vytvořit aplikaci pro experimentální testování úspěšnosti nově navržených metod rozpoznávání. Na aplikaci byly kladeny následující požadavky:

- Aplikace bude vyvíjena s využitím programovacího jazyka C++ a otevřenou knihovnou OpenCV na cílovou platformu Linux.
- K nastavování všech parametrů bude existovat konfigurační soubor.
- Spuštění bude možné z terminálu se zadáním konfigurace a cílového souboru pro průběžné logování událostí a výsledků.
- Aplikace musí být dostatečně robustní na to, aby mohl být každý experiment reprodukován se stejnými výsledky na všech verzích aplikace.

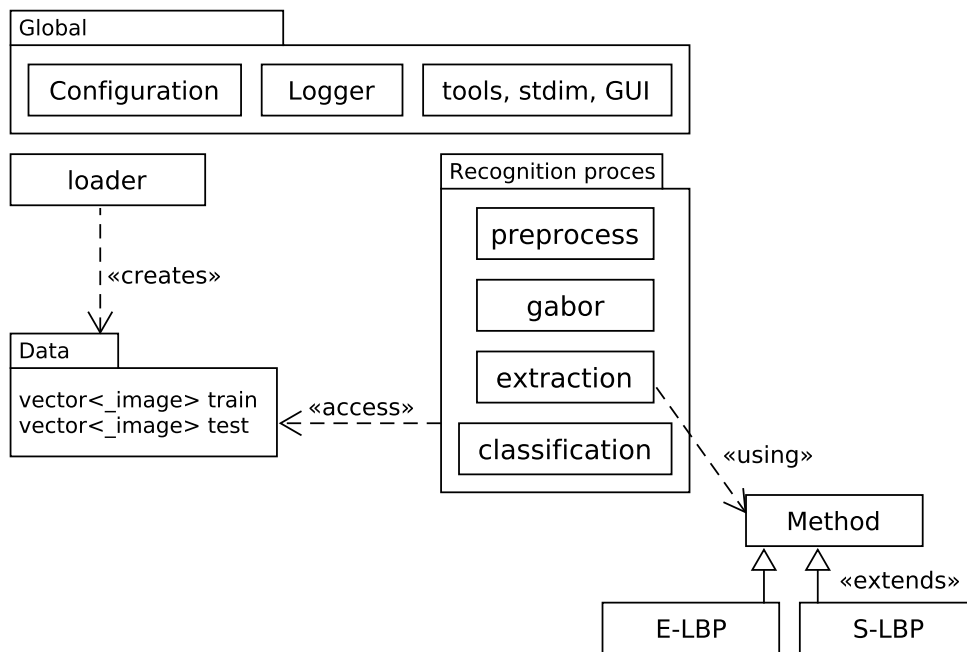
Reálné využívání aplikace je takové, že na serveru je inicializováno spuštění programu se seznamem konfigurací, které se testují na konkrétní databázi. Podle takto definovaného procesu rozpoznávání jsou konfigurace postupně zpracovávány a jejich výsledky ukládány. Výpočet jedné konfigurace může na databázi UFI trvat i desítky minut v závislosti na obtížnosti deskriptoru a volbě metody pro určení příznaku.

5.2 Struktura aplikace

Obrázek 5.1 znázorňuje nejdůležitější logické uskupení a strukturu programu. Třídy *Configuration* a *Logger* využívají návrhového vzoru Singleton¹ a poskytují globální podporu v celém programu. Dalšími pomocnými globálními moduly jsou: *tools*, *stdim* a *GUI*. Dále je v diagramu nastíněn datový tok aplikace, modul *loader* načte data z obličejové databáze a vytvoří dvě datové struktury pro uchování testovacích a trénovacích dat. K těmto datům

¹sdílení jedné instance třídy bez nutnosti předávání v parametrech

přístupují jednotlivé části rozpoznávacího procesu. Třídy deskriptorů *E-LBP* a *S-LBP* dědí vlastnosti z rodičovské třídy *Method*, kde jsou implementovány všechny metody získávání příznaků z obrazu. Právě dědičnosti deskriptorů využívá modul *extraction*. Všechna data jsou během procesu rozpoznávání zapisována do struktury *_image*, která uchovává informace, jako jsou: identifikace osoby, zdrojový obrázek nebo příznakový vektor. Výsledek procesu rozpoznávání se získává v modulu *classification*.



Obrázek 5.1: Diagram struktury programu.

OpenCV nabízí základní prvky grafického uživatelského rozhraní, ty jsou implementovány v modulu *GUI*. Nachází se zde grafické rozhraní pro editování vlastností Gáborových filtrů a také rozhraní pro grafické zobrazení výsledků programu. Dále jsou zde implementovány podpůrné metody pro vizualizaci při průběžné tvorbě deskriptoru.

5.3 Konfigurační soubor

Ovládání programu je prováděno primárně přes konfigurační soubor, který je požadován při spuštění. Čtení souboru ignoruje kromě prázdných řádek

také řádky začínající znakem `#` - komentář. Standardní konfigurace je vždy zapsána ve formátu *parametr:hodnota*. Rozšířením konfigurace jsou řádky začínající řetězcem *job*, za tento řetězec je umístěný identifikátor úkolu a sekvence parametrů. Rozšířená konfigurace se používá při dávkovém zpracování úkolů. Konfigurační soubor obsahuje následující sekce a položky:

- Umístění databáze obličejů.
 - data_path** - kořenová složka databáze
 - train_folder** - složka s trénovacími daty
 - test_folder** - testovací data
- Parametry předzpracování obrazu všech obrázků.
 - equalize_hist** - ekvalizace histogramů pro zvýšení kontrastu [0, 1]
 - gaussian_blur** - Gaussovské rozostření obrazu [0, 1]
- Režimy programu.
 - comparison_method** - volba metody pro porovnání histogramů [0 - Euklidovská vzdálenost, 1 - Intersekcce histogramů]
 - classification_threads** - počet výpočetních vláken klasifikace
 - job_mode** - Přepnutí mezi normálním a dávkovým režimem. Oba režimy akceptují všechny výše uvedené parametry, ale dávkový režim využívá pro konfiguraci rozpoznávacích metod jednořádkových zápisů, které jsou uvozené řetězcem *job*. [0, 1]
 - result_mode** - režim prohlížení výsledků [0, 1]
 - gabor_editor_mode** - režim editace parametrů pro Gáborovy filtry [0, 1]
 - extraction_method** - výběr deskriptoru [0 - S-LBP, 1 - E-LBP]
- Parametry metody Gáborových filtrů viz sekce 5.4.
 - gabor_setting** - zapnutí nebo vypnutí histogramů pozicovaných na základě Gáborových filtrů, při vypnutí se využívá pravidelné podhistogramové mřížky [0, 1]
 - gabor_points** - počet extrahovaných bodů
 - gabor_histogram_size** - velikost čtvercového histogramu
 - gabor_border_size** - ohraničení validních bodů od hran obrázku
- Společné parametry metod E-LBP a S-LBP.
 - grid_size** - velikost mřížky podhistogramů
 - uniform** - uniformita binárních vzorů [0, 1]

range - vzdálenost sousedních prvků operátoru [0, 1]

- Parametry metody E-LBP

elbp_center_size - rozměr centrální části operátoru

elbp_neighbour_shape - rozměr sousedních elementů

- Parametry metody S-LBP.

slbp_neighbours - počet sousedních prvků operátoru

slbp_neighbour_shape - tvar sousedních elementů

- Dávkové konfigurace (úkoly).

job - Uvození libovolného počtu dávkových konfigurací [0 - S-LBP, 1 - E-LBP, 2 - nové nastavení parametrů Gáborových filtrů]. Typ úkolu je následován již známými parametry metod ve stejném výše zmíněném pořadí např. uniformní metoda $S-LBP(8, 3, 5)$ s použitím pravidelné podhistogramové mřížky s rozměrem 12×12 má následující zápis: **job:0 - 12, 1, 3, 8, 5**.

5.4 Implementace Gáborových filtrů

V předchozí části 2.4.5 byly vysvětleny základní principy použití Gáborových filtrů, které nahrazují pravidelnou podhistogramovou mřížku pro získání příznakového vektoru. Vlastní implementace této techniky využívá knihovní funkce OpenCV pro aplikaci filtru. Velkým rozdílem je zejména hledání bodů v již filtrovaném obraze. Výpis 5.1 obsahuje ukázkou kódu, kde je aplikován filtr s konvoluční maticí *kernel* na obrázek *gabor_src*.

```
Mat kernel = getGaborKernel(KernelSize, Sigma, Theta, Lambda,
    Gamma, Psi);
filter2D(gabor_src, dest, CV_8U, kernel);
```

Výpis 5.1: Aplikace filtru s využitím funkcí OpenCV.

Konvoluční matice též známá jako jádro filtru se postupně přesouvá přes celý vstupní obraz a suma násobku překrývajících se položek vytváří nový pixel filtrovaného obrázku. OpenCV nabízí funkci *getGaborKernel*, jenž na základě vstupních parametrů navrací konvoluční matici. Funkce *filter2D* poté aplikuje matici *kernel* na vstupní obrázek *gabor_src* a filtrovaný obrázek je umístěn v proměnné *dest*.

K získání konvoluční matice jsou potřeba následující parametry:

KernelSize - velikost čtvercové matice

Sigma - směrodatná odchylka Gaussovského okénka (svazku)

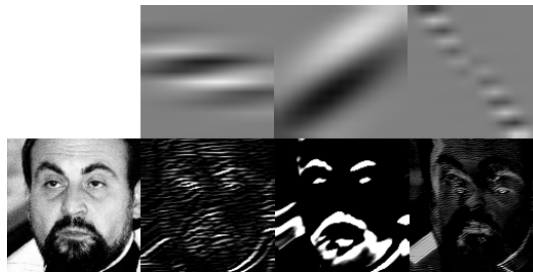
Theta - orientace normály paralelních pruhů filtru

Lambda - vlnová délka

Gamma - prostorový poměr stran

Psi - fázový posun

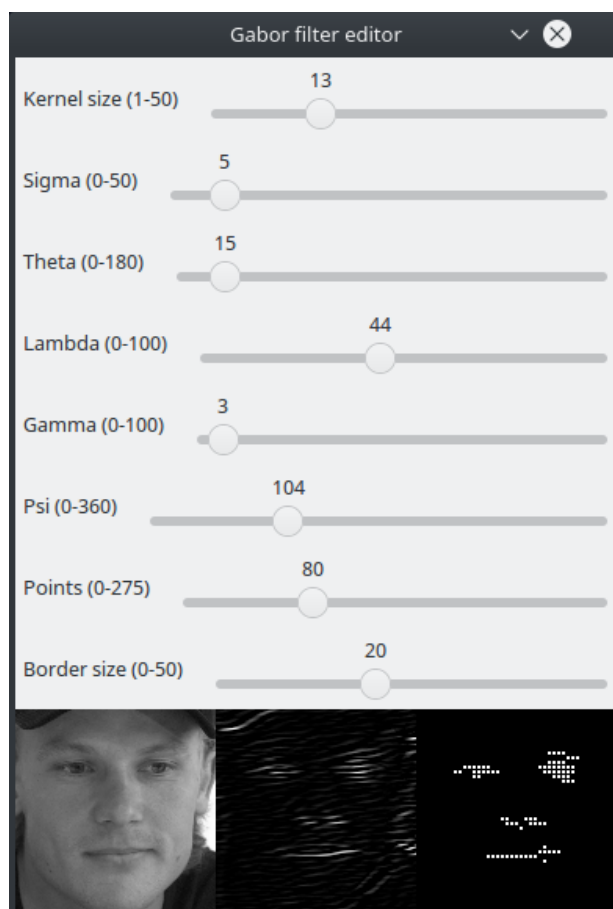
Na Obrázku 5.2 jsou v horní části vykresleny tři různé konvoluční matice Gáborových filtrů s různými parametry. Je zřejmé, že po aplikaci filtrů vznikají naprosto odlišné výstupy. Je důležité poznamenat, že na obrázku jsou konvoluční matice zvětšeny na velikost vstupních obličejů. Ve skutečnosti byly použity konvoluční matice o rozměru 13×13 px na obrázku z databáze ČTK s rozměrem 128×128 px.



Obrázek 5.2: Ukázka tří různých konvolučních matic (horní část) a jejich aplikace na obličej (dolní část).

5.4.1 Nalezení vhodných parametrů

Bohužel není možné předem stanovit vhodné parametry Gaborových filtrů pro úlohu rozpoznávání obličejů. K tomuto účelu vznikl nástroj pro editaci parametrů a následné zobrazení výsledků Gaborových filtrů viz Obrázek 5.3. Aktivace nástroje se provádí nastavením hodnoty *gabor_editor_mode* na 1 v konfiguračním souboru. Po stisku libovolného tlačítka je možné se přesunout na další osobu právě načtené databáze. Nástroj umožňuje okamžitou vizuální zpětnou vazbu. Posuvníky lze filtrovaný obrázek „vyladit“ k požadovaným vlastnostem. Parametry samotného filtru jsou poté v programu staticky nastaveny a nejsou součástí konfiguračního souboru. Je tak učiněno hlavně kvůli jejich velkému počtu a rozsahu.

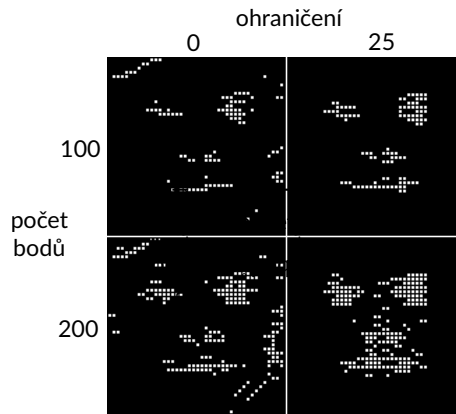


Obrázek 5.3: Editor parametrů pro Gáborovy filtry.

Součástí nástroje jsou dále dva nově přidané parametry, které již upravují způsob, jakým jsou definovány body z filtrovaného obrázku. Parametry jsou uloženy v konfiguračním souboru jako: *gabor_points* a *gabor_border_size*. Nejdříve je celý filtrovaný obrázek analyzován postupným posouváním matice o velikosti 3×3 px. Součty hodnot pixelů nacházející se pod maticemi jsou poté přidávány do seznamu i s pozicemi středů matic. Seznam je seřazen podle sumy pixelů a parametr *gabor_points* vybere n souřadnic s nejvyšší hodnotou. Parametr *gabor_border_size* vytváří ohraničení z každé strany obrazu a nedovoluje tak vzniknout bodům za těmito hranicemi.

Na Obrázku 5.4 je dobře vidět vliv obou výše zmíněných parametrů. Zdrojový obličej je stejný jako na Obrázku 5.3. V pravém sloupci je nastaveno ohraničení na hodnotu 25px, tímto jednoduchým krokem je odstraněna informace o pokrývce hlavy a okolí, což může mít vliv na úspěšnost rozpoznávání. Lze předpokládat, že existuje optimální počet bodů, s kterým bude na kon-

krétní databázi dosaženo nejvyšší úspěšnosti. Na Obrázku 5.5 je znázorněna oblast, kterou pokrývají čtvercové histogramy. Větší velikost histogramů při stejném počtu bodů dokáže velice dobře pokrýt většinu důležitých rysů v obličeji.



Obrázek 5.4: Vliv parametrů pro získávání bodů z obrázku.



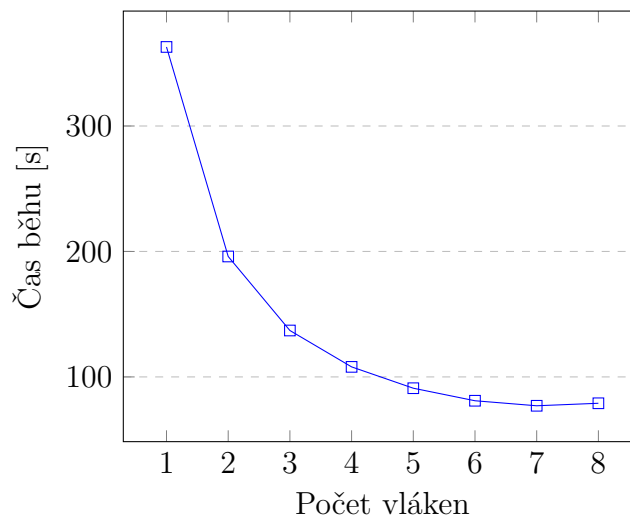
Obrázek 5.5: Plocha kterou zaujímá 100 histogramů o velikosti 10×10 a 20×20 px.

Porovnávání dvou obličejů se provádí podobně jako s pravidelnou podhistogramovou mřížkou. Jediným rozdílem je přidání vyhledávání dvojic histogramů. Ke každému bodu je přiřazený histogram, ten se porovnává s histogramem nejbližšího bodu v druhém obrázku podle již známých metod např. intersekcí. Suma vzdáleností všech histogramů tvoří celkovou vzdálenost mezi dvěma obličejmi.

5.5 Analýza časových nároků

Pro testování časových nároků byla zvolena uniformní „baseline“ metoda LBP s pravidelnou mřížkou podhistogramů 8×8 na databázi ČTK. Všechna měření a výsledky v této práci jsou prováděny na serveru s šestijádrovým procesorem Intel Core i7-4930K s 64 GB operační pamětí.

S výše uvedenými parametry bylo rozpoznávání dokončeno za 363 sekund. Z toho 95 % času je vyhrazeno pouze pro klasifikační část procesu rozpoznávání. Doba extrakce samotných příznaků je v tomto případě naprosto zanedbatelná. Pro zvýšení efektivity v následujících experimentech byla implementována vícevláknová klasifikace obličejů. Počet vláken lze nastavit parametrem *classification_threads* v konfiguračním souboru. Na Obrázku 5.6 je vykreslen časový přínos optimalizace. Při použití osmi klasifikačních vláken byl čas běhu téměř 4,6× rychlejší než s použitím jednoho výpočetního vlákna, tento počet vláken byl také zvolen jako výchozí pro všechny prováděné experimenty.



Obrázek 5.6: Čas běhu rozpoznávání v závislosti na počtu použitých klasifikačních vláken.

5.6 Kompilace a spuštění

Sestavení programu se provádí pomocí nástroje CMake² ve verzi 3.2.2 a vyšší. Je také nutné mít nainstalovaný kompilátor GCC³ v minimální verzi 4.8.4. V kořenové složce programu se nachází soubor *CMakeLists.txt* jenž obsahuje požadavek na nainstalovanou knihovnu OpenCV, tento soubor je parametrem pro nástroj CMake (lze nahradit za `./`). Program je zpětně kompatibilní pro OpenCV až do verze 2.4.9. Výpis 5.2 obsahuje sekvenci bash příkazů, jenž provede překlad a spuštění.

²CMake je multiplatformní SW pro automatický překlad programu

³GCC je sada překladačů pro různé prog. jazyky včetně C++

```
cd VisualDescriptorAnalyzer/  
cmake ./ && make  
mv play* Debug/ && cd Debug/  
nohup ./play* lbp.log server_config.txt &>/dev/null &
```

Výpis 5.2: Bash příkazy pro překlad a spuštění aplikace.

Po vstupu do kořenové složky programu je provedeno sestavení a kompilace. Spustitelný soubor je přesunutý do složky *Debug* a v ní se program spustí s parametrem názvu logovacího souboru a konfigurace. Příkaz *nohup* zajistí běh programu i když se uživatel odhlásí ze vzdáleného terminálu. Tímto způsobem lze inicializovat dlouhodobé testování na serveru a do souboru *lbp.log* budou postupně zapisovány výsledky.

6 Experimenty

6.1 Metodika

K nalezení optimálních parametrů pro nově představené deskriptory neexistuje žádný ověřený přístup k dosažení nejvyšší možné úspěšnosti na testované databázi. První logickou úvahou je vyzkoušení všech možných validních kombinací. Pro deskriptor E-LBP s použitím pravidelné podhistogramové mřížky lze definovat téměř 10 tisíc konfigurací, které by s průměrným časem běhu 5 minut byly zpracovávány desítky dní. V případě deskriptoru S-LBP a Gáborových filtrů se počet validních konfigurací zvýší na stovky tisíc.

Efektivním způsobem nalezení velkého množství parametrů je použití genetického algoritmu, který byl představen v práci Košar [5]. Byl použitý pro nastavení vah jednotlivých histogramů v podhistogramové mřížce. Tento algoritmus se používá na problémy, kde neznáme žádné deterministické řešení. Výstupem bohužel není žádný poznatek o chování metod a vlivu parametrů.

Pro hledání parametrů byl zvolen experimentální přístup, kdy jsou ručně nastaveny výchozí počáteční hodnoty deskriptorů a v jedné sadě konfigurací se testuje 1 volný parametr. Cílem tohoto přístupu je postupné nalezení závislostí a minimalizace nevhodných parametrů. V konečné fázi již lze použít generování časově zvládnutelného rozsahu konfigurací a na základě n nejlepších výsledků lze označit nejúspěšnější konfigurace. Vybrané parametry se tak opírají o experimenty a průběžně získanou zkušenost.

6.2 Výsledky experimentů

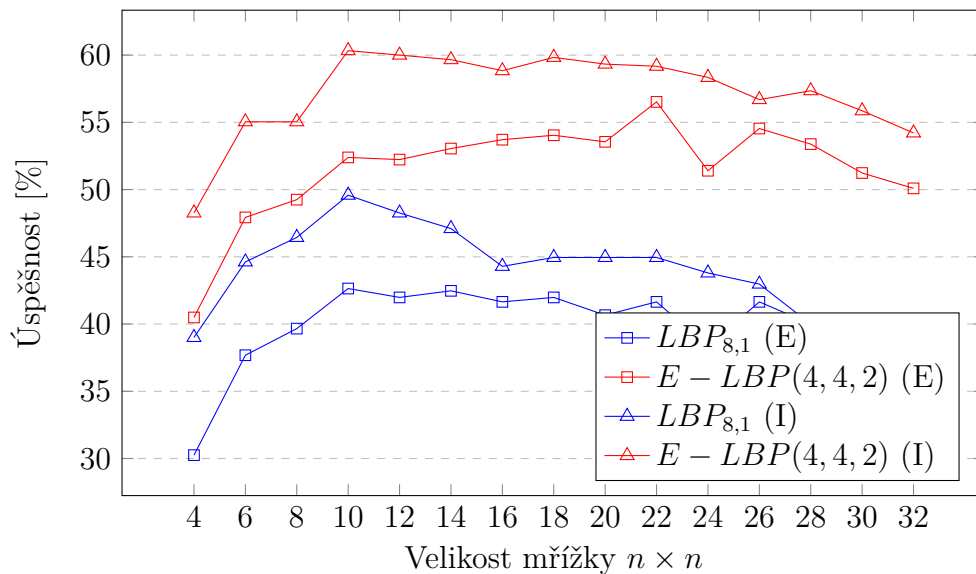
Výsledky jsou rozděleny na dvě hlavní sekce podle způsobu práce s histogramy. První je věnována pravidelné mřížce podhistogramů (PMP) a druhá metodě Gáborových filtrů (GF). V obou sekcích jsou testovány nově představené deskriptory vždy ve srovnání s referenčním deskriptorem LBP. Aby se ve výsledcích nevyskytoval duplicitní obsah s uniformními a neuniformními vzory, je experimentální postup prováděn pouze s uniformními vzory, nicméně neuniformní vzory jsou také testovány a jejich výsledky jsou rovněž

uvedeny ve výsledkových tabulkách viz sekce 6.3.4 a 6.4.4. Experimenty jsou prováděny zejména na databázi ČTK (sada *cropped*) a doplňkově na databázi FERET s rozlišením 130×150 px (sada *fb*). Procentuální porovnání je vždy absolutní vzhledem k výsledkům různých rozpoznávacích metod. V této kapitole se navíc nachází sekce věnující se výsledkům na jiné obrazové doméně (rozpoznávání textur).

6.3 Pravidelná mřížka podhistogramů

6.3.1 Rozměr mřížky a metody porovnání

Na Obrázku 6.1 je znázorněn průběh úspěšnosti základního deskriptoru $LBP_{8,1}$ a nově představeného $E - LBP(4, 4, 2)$ v závislosti na velikosti mřížky. Pro deskriptor E-LBP byly zvoleny počáteční parametry s nejmenšími elementy. Oba deskriptory jsou testovány s porovnávacími metodami představenými v kapitole 2.4.6.



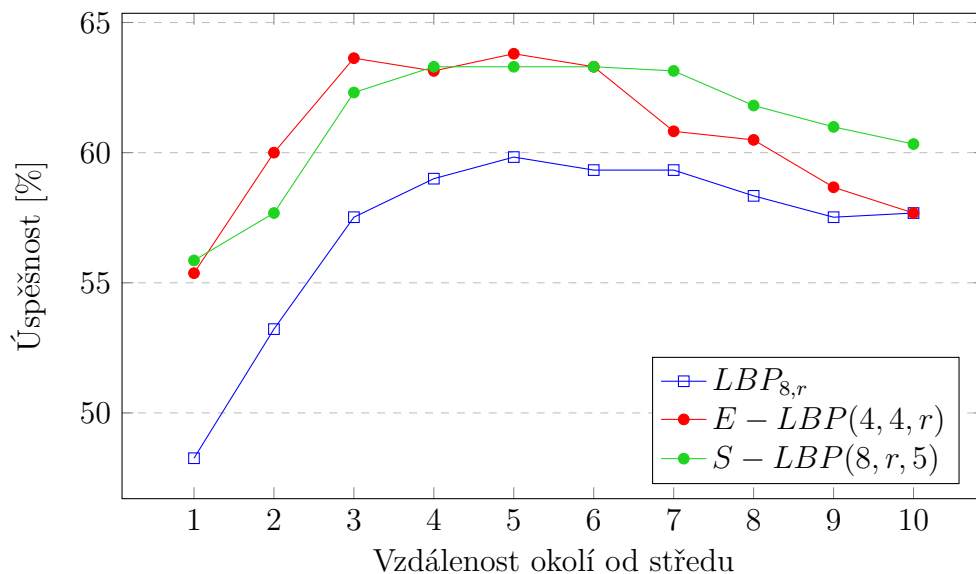
Obrázek 6.1: Výsledky rozpoznávání uniformních obrazových deskriptorů v závislosti na velikosti podhistogramové mřížky a metodě porovnání: Euklidovská vzdálenost (E) a Intersekcce histogramů (I).

Deskriptor E-LBP dosahuje zřetelně lepších výsledků než klasický deskriptor LBP v případě obou porovnávacích metod. E-LBP zde dosahuje maxima

60.33 %, což je o 10,75 % více než maximum LBP. Dále je z grafu zřejmé, že metoda intersekcce histogramů podává nezávisle na deskriptoru lepší výkon než Euklidovská vzdálenost a to průměrně o 4,85 % ve všech měřeních. Intersekcce histogramů proto bude nastavena jako výchozí metoda pro porovnávání příznakových vektorů obličejů ve všech následujících experimentech. Jako nejvhodnější velikost mřížky se jeví hodnota 10, zde oba deskriptory dosahují svého maxima. Proto je v dalších experimentech rozsah mřížky testován v intervalu $\langle 8, 14 \rangle$.

6.3.2 Vzdálenost sousedních prvků

Dalším důležitým parametrem jednotlivých deskriptorů je nastavení vzdálenosti sousedních elementů od středu. Na Obrázku 6.2 jsou znázorněny výsledky tří deskriptorů. Ke dvěma deskriptorům z předchozího experimentu byl přidán deskriptor $S - LBP(8, r, 5)$, kde r představuje volný parametr vzdálenosti a je zvolen tvar okolního elementu s číslem 5. Jako výchozí byl nastaven rozměr mřížky 12 a je již používána pouze metoda intersekcce histogramů.



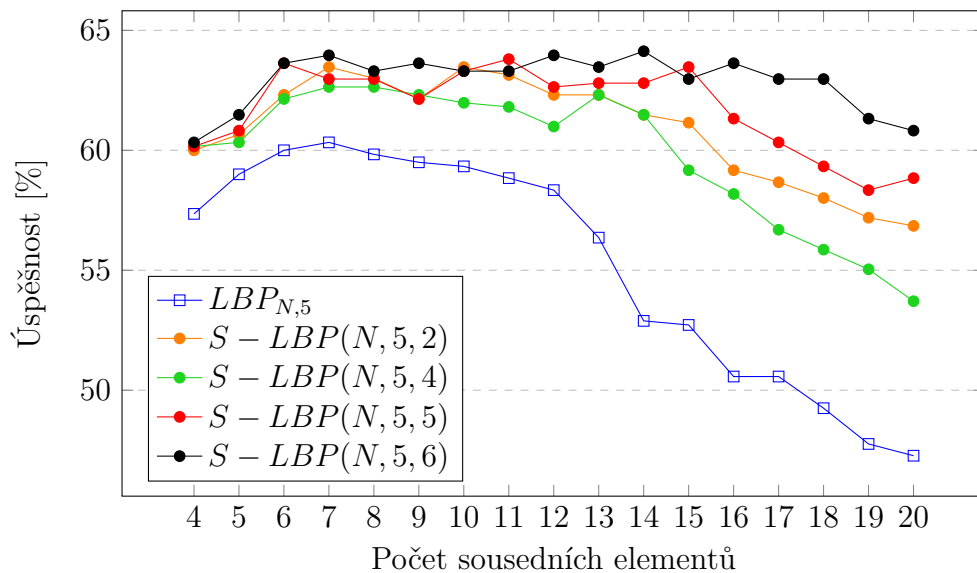
Obrázek 6.2: Výsledky uniformních obrazových deskriptorů v závislosti na vzdálenosti okolí r sousedních prvků od středu operátoru.

Nejvyšší úspěšnosti 63,8 % bylo dosaženo s deskriptorem E-LBP při vzdálenosti od středu 5. Oba nově představené deskriptory dosahují v tomto

případě téměř stejné průměrné úspěšnosti a použití deskriptoru S-LBP zlepšuje průměrnou úspěšnost proti LBP o 4,2 %. Pro další experimenty bude hodnota vzdálenosti od středu redukována na interval $\langle 3, 6 \rangle$.

6.3.3 Počet sousedních elementů a jejich tvar

S-LBP i obecná verze LBP umožňuje zvýšit počet sousedních bodů. Binárním vzorem tak může být více než osmi-bitové číslo. Na Obrázku 6.3 je znázorněna úspěšnost deskriptorů na počtu sousedních elementů ve vzdálenosti 5. Deskriptory se liší tvarem okolí, S-LBP zde zastupují tvary číslo 2, 4, 5 a 6. Tvary byly představeny v sekci 4.2.



Obrázek 6.3: Výsledky uniformních obrazových deskriptorů v závislosti na tvaru a počtu sousedních prvků označených parametrem N .

Všechny varianty deskriptoru S-LBP dosahují lepší úspěšnosti než LBP díky použití rozšířenému tvaru okolí. Průběh grafu ukazuje, že deskriptor s větším obsahem okolí lépe reaguje na zvyšující se počet sousedních elementů. Deskriptor $S - LBP(N, 5, 6)$ dosáhl nového maxima 64,13 % s hodnotou $N = 14$. Interval počtu sousedních elementů pro deskriptory s jednoduchým okolím lze zúžit na $\langle 6, 12 \rangle$, pro deskriptory s rozšířeným okolím lze dále použít interval $\langle 6, 16 \rangle$.

6.3.4 Generované konfigurace

Z předchozích experimentů je díky zúženým intervalům možné provést test generovaných konfigurací pro nalezení nejvyšší úspěšnosti. Bylo provedeno 1656 iterací s celkovou dobou běhu 10 dnů. V Tabulkách 6.1 a 6.2 se nacházejí nejlepší uniformní a neuniformní deskriptory E-LBP.

Tabulka 6.1: Nejlepší **uniformní** deskriptory E-LBP (PMP).

| Deskriptor | Rozměr mřížky | Úspěšnost [%] | Doba běhu [s] |
|--------------------|----------------|---------------|---------------|
| $E - LBP(4, 9, 4)$ | 12×12 | 65,28 | 167 |
| $E - LBP(4, 4, 4)$ | 13×13 | 65,28 | 171 |
| $E - LBP(4, 9, 5)$ | 12×12 | 64,95 | 160 |

Tabulka 6.2: Nejlepší **neuniformní** deskriptory E-LBP (PMP).

| Deskriptor | Rozměr mřížky | Úspěšnost [%] | Doba běhu [s] |
|--------------------|----------------|---------------|---------------|
| $E - LBP(4, 9, 5)$ | 10×10 | 65,28 | 328 |
| $E - LBP(4, 9, 5)$ | 12×12 | 65,12 | 454 |
| $E - LBP(4, 9, 5)$ | 8×8 | 64,95 | 220 |

Nejlepší uniformní a neuniformní E-LBP deskriptory dosahují úspěšnosti 65,28%, ale uniformní varianta je v tomto případě téměř 2× rychlejší. Zajímavou vlastností deskriptoru E-LBP je, že uniformita vzorů nemá žádný vliv na maximální úspěšnost. Rozdíl nalezených parametrů se v rámci uniformity nejvíce liší ve velikosti mřížky a uniformní vzory potřebují pro stejnou úspěšnost větší počet podhistogramů.

V Tabulce 6.3 jsou vypsány nejlepší nalezené uniformní konfigurace pro deskriptor S-LBP. Nejlepší neuniformní konfigurace LBP a S-LBP se nacházejí v Tabulce 6.4.

Tabulka 6.3: Nejlepší **uniformní** deskriptory S-LBP (PMP).

| Deskriptor | Rozměr mřížky | Úspěšnost [%] | Doba běhu [s] |
|---------------------|----------------|---------------|---------------|
| $S - LBP(11, 5, 6)$ | 13×13 | 65,78 | 337 |
| $S - LBP(8, 5, 6)$ | 13×13 | 64,95 | 221 |
| $S - LBP(10, 5, 6)$ | 13×13 | 64,95 | 294 |

Nejvyšší úspěšnosti 65,78 % dosahuje uniformní deskriptor $S - LBP(11, 5, 6)$. Nepatrné zlepšení úspěšnosti je ale proti nejlepší uniformní konfiguraci E-LBP vykoupeno dvojnásobkem výpočetního času. LBP deskriptor byl vždy

testován společně s deskriptorem S-LBP, v Tabulce 6.3 se však nevyskytuje, protože dosahuje značně nižší úspěšnosti a ani nejlepší vygenerovaná konfigurace nedosáhla vyšší úspěšnosti než 61 %. Nejlepším tvarem deskriptoru S-LBP se stal tvar číslo 6, tento tvar má největší plochu (matice 3×3).

Tabulka 6.4: Nejlepší **neuniformní** deskriptory LBP a S-LBP (PMP).

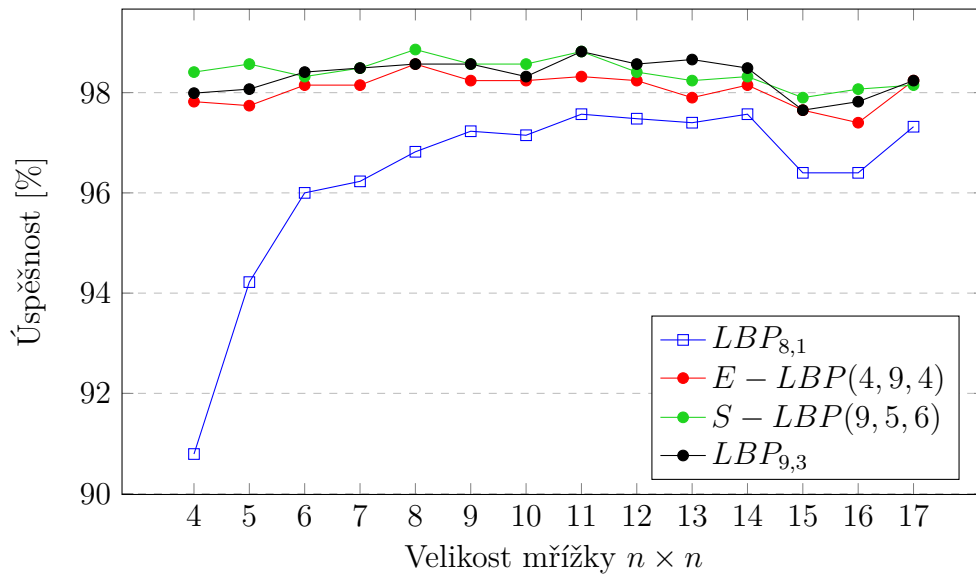
| Deskriptor | Rozměr mřížky | Úspěšnost [%] | Doba běhu [s] |
|---------------------|----------------|---------------|---------------|
| $LBP_{9,3}$ | 13×13 | 65,45 | 1040 |
| $LBP_{10,3}$ | 13×13 | 65,45 | 2004 |
| $S - LBP(10, 3, 4)$ | 12×12 | 65,28 | 1712 |

Nejvyšší úspěšnosti 65,45 % dosahuje obecná varianta LBP s jednoduchým okolím a S-LBP dosahuje maxima s tvarem, který zachycuje nejmenší plochu (číslo 4). Neuniformní vzory tak nevyužívají rozšířené okolí deskriptoru S-LBP. Optimální vzdálenost okolních elementů je snížena na hodnotu 3. Dosaženou úspěšnost ovšem sráží doba běhu neuniformních konfigurací. Z Tabulky 6.4 je patrné, že deskriptor $LBP_{9,3}$ dosahuje stejné úspěšnosti jako $LBP_{10,3}$ při stejném rozměru mřížky, doba běhu je ovšem dvojnásobná. Toto chování je způsobeno přidáním dalšího sousedního prvku a tedy zdvojnásobení velikosti histogramů. Neuniformní deskriptor $LBP_{9,3}$ je ve srovnatelné úspěšnosti více než $6 \times$ pomalejší než nově představený uniformní E-LBP deskriptor.

6.3.5 Aplikace deskriptorů na databázi FERET

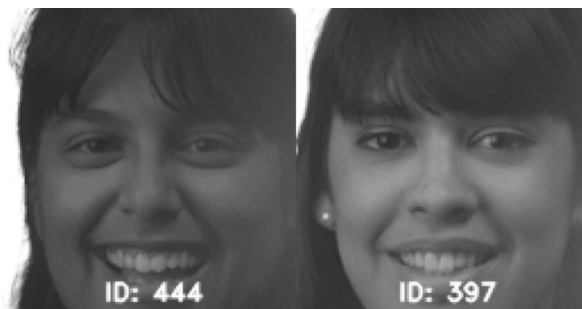
Na Obrázku 6.4 je znázorněna úspěšnost nejlepších metod, které byly experimentálně nalezeny na databázi ČTK. Je zřejmé, že databáze FERET je velice jednoduchá a základní uniformní deskriptor $LBP_{8,1}$ dosahuje 97,57 % úspěšnosti (na databázi ČTK nepřesahuje 50 % viz Obrázek 6.1). Ostatní vybrané deskriptory dosahují velmi stabilních výsledků a maximum 98,86 % s deskriptorem $S - LBP(9, 5, 6)$.

Nezměněné deskriptory z databáze ČTK dosahují slušné úspěšnosti bez nutnosti hledání nových optimálních parametrů viz Obrázek 6.4. I přes tuto skutečnost byly znovu provedeny stejné iterace generovaných konfigurací na databázi FERET, všech 1656 výsledků se nacházelo v rozmezí 98,07 % až 99,07 % úspěšnosti. Získání jednoho procenta úspěšnosti na databázi FERET odpovídá správnému přiřazení dvanácti osob. V poslední setině databáze se nacházejí velice obtížné obličeje viz Obrázek 6.5.



Obrázek 6.4: Úspěšnost vybraných deskriptorů na databázi FERET v závislosti na velikosti mřížky. $LBP_{9,3}$ je jediný neuniformní deskriptor, ostatní jsou uniformní.

Podle výsledků experimentů není vhodné databázi FERET používat jako výchozí pro hledání nejlepších konfigurací deskriptorů, a to hlavně kvůli nízké obtížnosti a malému prostoru pro zlepšení. V této práci byla databáze FERET použita pro ověření funkčnosti nových experimentálních metod na jiné datové sadě a pro dodatečné porovnání úspěšnosti s jinými metodami rozpoznávání.

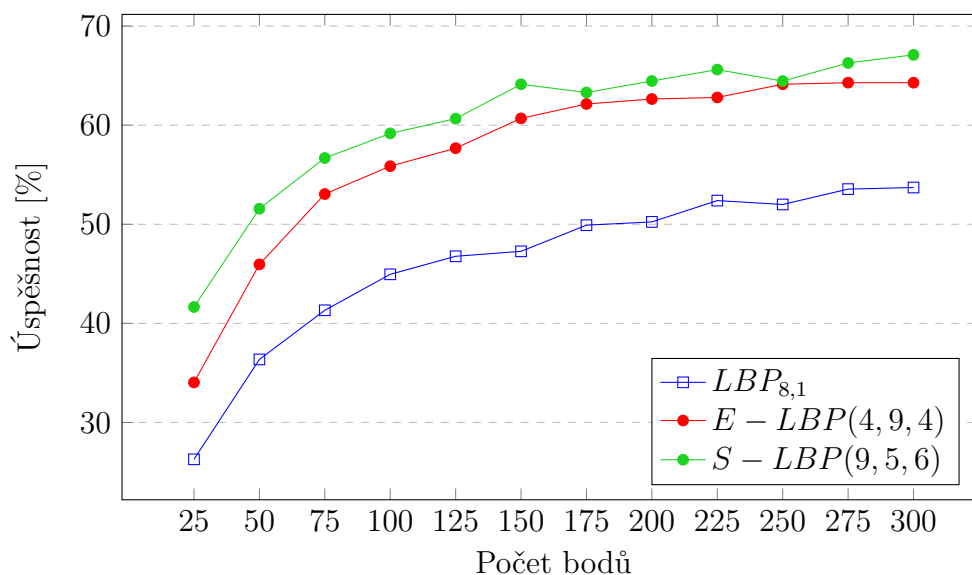


Obrázek 6.5: Ukázka špatného přiřazení osoby z databáze FERET.

6.4 Gáborovy filtry

6.4.1 Počet nalezených bodů

Z experimentů s PMP byla nalezena množina vhodných nastavení parametrů pro nově navržené deskriptory (E-LBP a S-LBP) a tyto konfigurace lze dále využít i v případě extrakce příznaků pomocí Gáborových filtrů. Jedním z parametrů je počet bodů nalezených na základě aplikovaného filtru. Pro stanovení závislosti na počtu nalezených bodů viz Obrázek 6.6 byl nastaven počáteční rozměr podhistogramu na 15×15 px a nebylo uvažováno žádné dodatečné ohraničení.

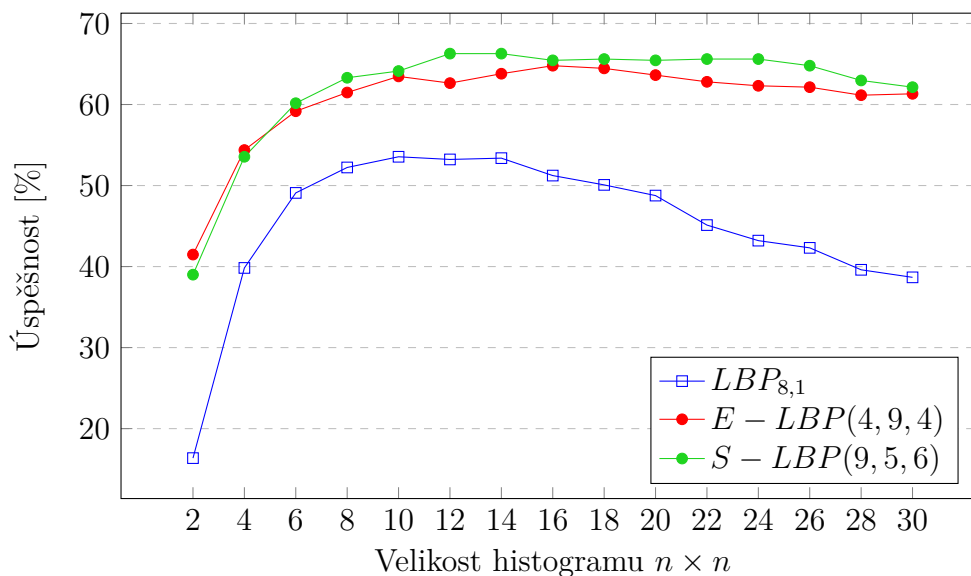


Obrázek 6.6: Úspěšnost uniformních deskriptorů v závislosti na počtu nalezených bodů pomocí Gáborových filtrů.

Z Obrázku 6.6 je patrná silná závislost úspěšnosti na počtu nalezených bodů. Při použití více než 250 bodů je dosaženo stejné úspěšnosti jako při použití PMP. Deskriptory E-LBP a S-LBP zde dosahují průměrně o 16,2 % lepší úspěšnosti než klasický deskriptor LBP. S-LBP zde dosahuje maxima 67,1 % při použití 300 bodů. Zvyšující se počet bodů automaticky přispívá ke složitosti výpočtu. Zatímco výpočet pro 100 bodů deskriptoru S-LBP trvá 241 s, tak pro 300 bodů je to již 1143 s. Metodu Gáborových filtrů tak je tedy časově náročnější, konkrétně u deskriptoru S-LBP je získání stejné úspěšnosti při použití uniformních vzorů 3× náročnější než s použitím PMP.

6.4.2 Velikost histogramu

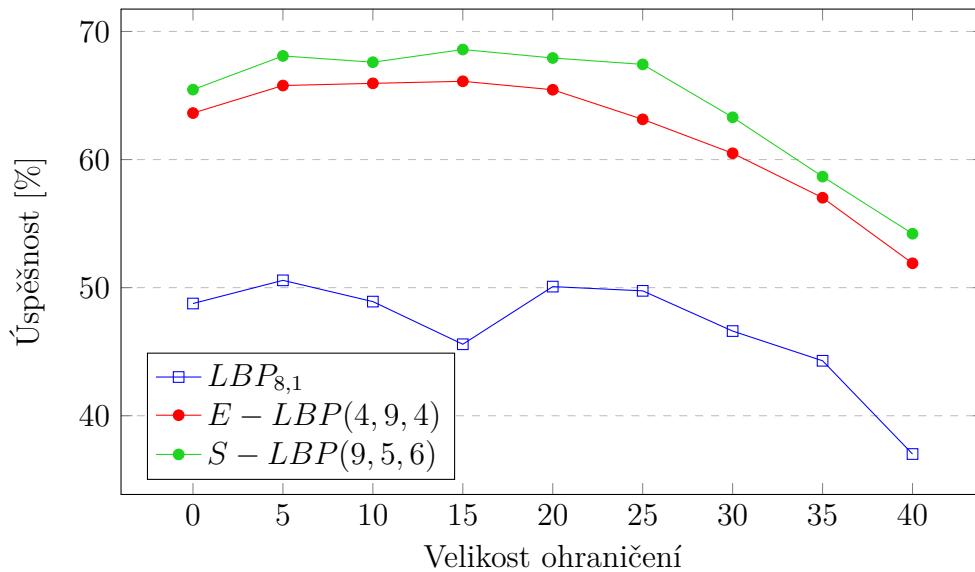
Na základě předchozího experimentu bylo vybráno 250 bodů jako výchozí hodnota pro nalezení závislosti na rozměru lokálních histogramů. Z Obrázku 6.7 se jako optimální rozměr histogramů pro další experimenty jeví hodnoty v intervalu $\langle 10, 22 \rangle$ pro nově navržené deskriptory a $\langle 10, 16 \rangle$ pro LBP. GF dosahují srovnatelné úspěšnosti proti metodě PMP s použitím výrazně větších úmyslně pozicovaných lokálních histogramů.



Obrázek 6.7: Úspěšnost uniformních deskriptorů v závislosti na velikosti lokálních histogramů.

6.4.3 Velikost ohraničení nalezených bodů

Posledním parametrem metody Gáborových filtrů je velikost ohraničení. Předpokladem tohoto parametru bylo odstranění rušivých prvků v krajních oblastech obrázku. Pro velikost histogramu byla na základě předchozího experimentu zvolena hodnota 20. Obrázek 6.8 potvrzuje, že aplikace tohoto parametru je důležitá a dokáže zvýšit úspěšnost. Deskriptor S-LBP přidáním ohraničení o velikosti 15 px zvýšil svou úspěšnost z původních 65,46 % na 68,59 %. Pokud tento parametr přesáhne hranici 25 px, je zaznamenán rychlý pokles úspěšnosti všech deskriptorů. Hodnoty vyšší než 25 px lze vyřadit z následných generovaných konfigurací.



Obrázek 6.8: Úspěšnost uniformních deskriptorů v závislosti na velikosti ohraničení.

6.4.4 Generované konfigurace

Bylo provedeno 1782 iterací s celkovým časem běhu 12 dní. Součástí byly uniformní i neuniformní konfigurace, které byly podpořeny nalezenými závislostmi mezi parametry a omezením intervalů. V Tabulce 6.5 jsou zaznamenány parametry tří nejlepších uniformních deskriptorů.

Tabulka 6.5: Nejlepší **uniformní** deskriptory (GF).

| Deskriptor | Počet bodů | Hist. [px] | Ohraničení [px] | P [%] | T [s] |
|---------------------|------------|----------------|-----------------|--------------|-------|
| $S - LBP(10, 5, 2)$ | 250 | 21×21 | 15 | 69,75 | 812 |
| $S - LBP(11, 5, 6)$ | 275 | 24×24 | 15 | 69,58 | 1003 |
| $S - LBP(10, 5, 2)$ | 225 | 21×21 | 15 | 69,58 | 696 |

Tři nejlepší uniformní deskriptory z Tabulky 6.5 zvyšují průměrnou úspěšnost o 4,41 % v porovnání s Tabulkou 6.3, kde je metoda PMP dosahuje svého maxima. Deskriptor E-LBP na rozdíl od S-LBP nedosahuje více než 66,5 % úspěšnosti a nelze jej tak považovat za vhodný deskriptor v kombinaci s metodou GF. Deskriptor S-LBP v konkrétní konfiguraci $S - LBP(10, 5, 2)$ s počtem bodů 250 dosahuje nového maxima s úspěšností 69,75 % ze všech testovaných uniformních deskriptorů.

V Tabulce 6.6 jsou zaznamenány nejlepší neuniformní deskriptory využíva-

jící GF. Je zřejmé, že neuniformní deskriptory velice dobře reagují na metodu GF a dosahují lepších výsledků než s použitím PMP, kde neuniformita nepřináší žádný nárůst úspěšnosti viz předchozí Tabulky 6.2 a 6.4.

Tabulka 6.6: Nejlepší **neuniformní** deskriptory (GF).

| Deskriptor | Počet bodů | Hist. [px] | Ohraničení [px] | P [%] | T [s] |
|---------------------|------------|----------------|-----------------|--------------|-------|
| $LBP_{9,3}$ | 225 | 18×18 | 20 | 72,06 | 1211 |
| $LBP_{10,3}$ | 250 | 16×16 | 20 | 71,73 | 2153 |
| $LBP_{9,3}$ | 225 | 16×16 | 20 | 71,73 | 1211 |
| $S - LBP(10, 3, 4)$ | 225 | 18×18 | 20 | 71,07 | 1871 |

Tabulka 6.6 ukazuje, že neuniformita favorizuje menší rozměr histogramů a větší ohraničení obrázků. Zajímavým parametrem je počet bodů, který má u neúspěšnější metody hodnotu pouhých 225, i když z Obrázku 6.6 v předchozím experimentu lze usuzovat, že čím více je použito bodů, tím je vyšší úspěšnost. Obecný deskriptor LBP získal nejvyšší úspěšnost v sekci neuniformních deskriptorů. Ke třem nejlepším neuniformním výsledkům z tabulky 6.6 byl připojen výsledek deskriptoru S-LBP, který dosahuje 71,04 % úspěšnosti.

Nejúspěšnější nalezenou konfigurací pro databázi ČTK se stává kombinace Gáborových filtrů a deskriptoru $LBP_{9,3}$ s maximálním neuniformním výsledkem 72,06 %, který je také nejlepším nalezeným ze všech. Gáborovy filtry dokázaly zvýšit maximální úspěšnost o 6,28 % v porovnání s metodou PMP. Tato vysoká úspěšnost je ovšem vykoupena cenou v podobě výpočetních nároků, které jsou u nejlepších výsledků 3,5× větší než u PMP.

6.4.5 Aplikace Gáborových filtrů na databázi FERET

Již od počátku návrhu metody využívající GF nebylo jasné, zda optimální parametry samotného filtru nalezené na databázi ČTK budou použitelné i pro jinou databázi. Z Obrázku 6.9 je však vidět velice dobrá aplikace na obličej z databáze FERET. Nelze však vyloučit, že takto nastavené parametry GF budou muset být upraveny na míru pro jiné databáze, pro dosažení požadovaných výsledků tj. zvýraznění dominantních částí obličeje. Nejlepší deskriptory $S - LBP(10, 5, 2)$ (uniformní) a $LBP_{9,3}$ (neuniformní) dosáhly bez jakékoli úpravy parametrů úspěšnosti 96,73 % a 97,15 %. Hlavním důvodem nižší úspěšnosti na této databázi je vyšší rozlišení, které vyžaduje větší rozměr používaného histogramu. Po naladění parametrů GF přímo

na databázi FERET bylo dosaženo výsledku 98,15 %, který je srovnatelný s metodou PMP. Použití GF tak přináší výhody pouze v databázi reálných fotografií ČTK.



Obrázek 6.9: Aplikace výchozích parametrů GF na obličej z databáze FERET. Zleva: originál, filtrovaný obrázek, vizualizace nalezených bodů.

6.5 Rozpoznávání textur

Tato sekce slouží jako doplnění této kapitoly o další experimentální měření na jiné obrazové doméně. Výsledky z těchto experimentů jsou nad rámec zadání této bakalářské práce s cílem ověření robustnosti navržených deskriptorů.

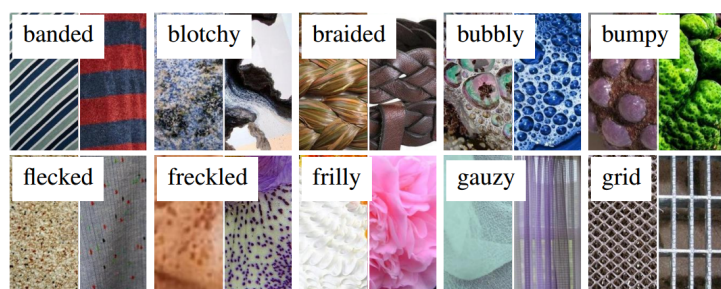
6.5.1 Databáze DTD

Describable Textures Dataset je veřejná¹ volně dostupná databáze textur různého typu, která byla představena v práci z roku 2014 Cimpoi et al. [3]. Obsahuje 5640 obrázků, které jsou organizovány do 47 skupin pojmenovaných podle lidského popisu viz Obrázek 6.10. Každý obrázek obsahuje nejméně 90 % plochy popisovanou texturou. Velikost obrázků je proměnlivá od 300×300 do 640×640 px.

6.5.2 Testovací protokol

Pro získání úspěšnosti rozpoznávání je zapotřebí vypočítat průměr z celkem deseti různých testů. Tyto testy jsou definovány v textových souborech

¹<https://www.robots.ox.ac.uk/~vgg/data/dtd/>



Obrázek 6.10: Slovně popsané skupiny textur, převzato z [3]

jako součást databáze, jednotlivé záznamy v souborech odkazují na testovací nebo trénovací data. Pro splnění těchto specifikací byla implementovaná aplikace upravena speciálně pro tuto činnost. Každému obrázku je přiřazen jeden hlavní atribut a seznam doplňujících atributů, které lze využít při pokročilé práci s databází DTD. V našem případě je testováno jen pomocí hlavních atributů. Řešením pro různé velikosti obrázků je škálování všech na průměrnou velikost 494×450 px.

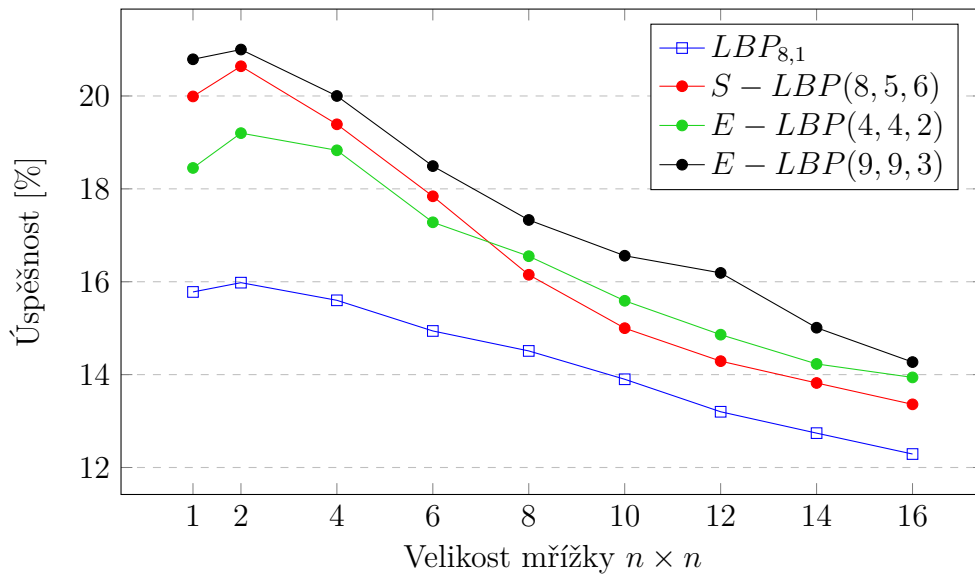
6.5.3 Naměřené výsledky

Pro testování byly zvoleny 4 neuniformní deskriptory s odlišnými vlastnostmi včetně standardní varianty LBP. Pro porovnávání byla zvolena metoda intersekcí histogramů. Na Obrázku 6.11 je znázorněna úspěšnost v závislosti na velikosti PMP. Je zcela evidentní, že zvyšování počtu podhistogramů nepřináší žádné zlepšení v úspěšnosti. Jako nejúspěšnější se ukázala velikost mřížky 2×2 a deskriptor $E - LBP(9, 9, 3)$ s úspěšností 21 %.

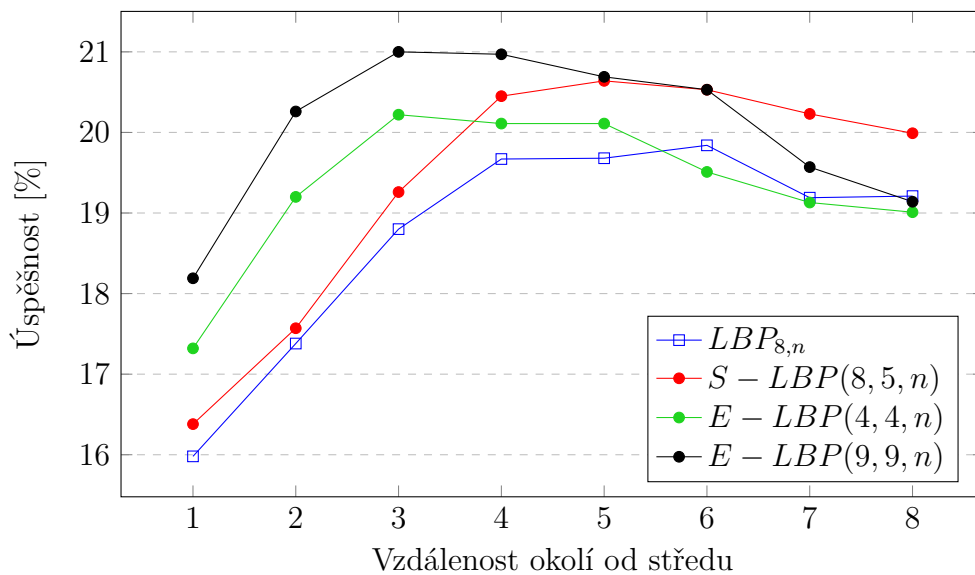
Druhé měření viz Obrázek 6.12 sleduje závislost úspěšnosti deskriptorů na vzdálenosti středu a okolí. Maximum zůstává stejné jako v předchozím případě 21 %. Ani v tomto případě nemá parametr deskriptoru žádný pozitivní vliv na úspěšnost.

6.5.4 Diskuze výsledků

V kontextu práce Cimpoi et al. [3] jsou zde dosažené výsledky rozpoznávání textur podprůměrné. Hlavním důvodem nižší úspěšnosti je použití jednoduššího klasifikátoru $1NN$, který nemůže konkurovat pokročilým klasifikáto-



Obrázek 6.11: Úspěšnost neuniformních deskriptorů v závislosti na velikosti mřížky.



Obrázek 6.12: Úspěšnost neuniformních deskriptorů v závislosti na vzdálenosti n od středu k okolí.

rům na bázi SVM (Support Vector Machines), které používá strojové učení. Výsledek 21 % deskriptoru $E - LBP(9, 9, 3)$ je srovnatelný s deskriptorem LBP-VQ [3] využívající lineární klasifikátor a dosahující úspěšnosti 21.1 %. Aplikací jiného jádra pro klasifikátor SVM je zvýšena úspěšnost deskriptoru

LBP-VQ na 34,7 %. Maximální úspěšnost 63,5 % byla v práci [3] dosažena pomocí metody SIFT IFV.

6.6 Porovnání výsledků s existujícími pracemi

Práce Lenc – Král [7] představuje novou obličejovou databázi ČTK a uvádí první výsledky získané na této databázi. Aby bylo možné přímo porovnat metody rozpoznávání je nutné přesně dodržet uvedený testovací protokol. Implementace aplikace dodržuje pro obličejovou sadu *cropped* všechny potřebné specifiky a výsledky z kapitoly 6.2 tak lze přímo porovnat. V práci [7] jsou použity následující 4 neuniformní rozpoznávací metody:

- Pravidelná mřížka podhistogramů (PMP)
 - LBP_{HS}** - Základní „baseline“ deskriptor $LBP_{8,1}$.
 - LDP_{HS}** - Local Derivative Patterns zachycuje n úrovní charakteristiky obrázku, byla však použita varianta pouze prvního řádu.
 - POEM_{HS}** - Tento pokročilý deskriptor byl popsán v sekci 2.4.3.
- Gáborovy filtry (GF)
 - FS-LBP** - Využívá deskriptor $LBP_{8,2}$ a vlastní implementaci úmyslně pozicovaných histogramů, pro každý obličej je použito individuálních pozic podhistogramů.

V Tabulce 6.7 jsou zobrazeny nejlepší výsledky deskriptorů využívající metodu PMP. Nově představené deskriptory E-LBP a S-LBP markantně převyšují základní deskriptor LBP a také LDP. LDP se jeví jako velice špatný deskriptor pro databáze založené na reálných fotografiích. Oběma novým deskriptorům chybí necelé 2 % úspěšnosti na dorovnání mnohem komplexnějšího deskriptoru POEM, který je považován za jeden z nejvíce robustních deskriptorů využívající lokálních binárních vzorů, toto potvrzuje i práce Košař [5].

V Tabulce 6.8 jsou srovnány deskriptory, které využívají Gáborových filtrů pro nalezení důležitých bodů v obličejí. Uniformní deskriptor S-LBP i neuniformní $LBP_{9,3}$ překonávají dosavadní výsledek FS-LBP o 7,76 % a 8,75 %.

Tabulka 6.7: Srovnání deskriptorů (PMP).

| Deskriptor | Úspěšnost [%] |
|--------------|---------------|
| LPHS [7] | 55,04 |
| LDPHS [7] | 50,25 |
| POEMHS [7] | 67,11 |
| E-LBP | 65,28 |
| S-LBP | 65,78 |

Tabulka 6.8: Srovnání deskriptorů (GF).

| Deskriptor | Úspěšnost [%] |
|--------------------------|---------------|
| FS-LBP [7] | 63,31 |
| S-LBP | 71,07 |
| LBP_{9,3} | 72,06 |

Deskriptor $LBP_{9,3}$ stanovuje nové maximum pro všechny doposud známé výsledky na databázi ČTK.

Z výsledků experimentů a porovnání je zřejmé, že vlastní implementace metody Gáborových filtrů v kombinaci s přesně naladěnými deskriptory dosahuje nejvyšší známé úspěšnosti. Za dobře použitelné lze považovat také oba nově představené deskriptory E-LBP a S-LBP v kombinaci s metodou PMP, které téměř dosahují úspěšnosti deskriptoru POEM a překonávají implementaci Gáborových filtrů z předchozích prací [6] a [7].

7 Návrh dalšího rozšíření

Na základě všech provedených experimentů lze soudit, že byly parametry deskriptorů v kombinaci s pravidelnou mřížkou podhistogramů nastaveny na již hraniční úspěšnost a při jakékoli změně parametrů už nebude docházet k žádnému zlepšení. Hlavní nevýhodou PMP je důvěra v přesnou lokalizaci předzpracovaného obličeje v obrázku. Odchylka pozice obličejů vede k porovnávání lokálních histogramů s daty různých částí obličeje. Řešením tohoto problému je hledání důležitých bodů v obličeji nezávisle na pozici pomocí Gáborových filtrů. Velice přínosným rozšířením této práce by byl návrh a srovnání jiných metod pro získávání důležitých obličejových bodů z obrázku (např. transoframce SIFT). Samotná implementace Gáborových filtrů poskytuje také dostatek prostoru pro vlastní úpravy.

Výsledkem rozpoznávání je vždy pouze jedno číslo, reprezentující úspěšnost. Bylo by vhodné aplikaci rozšířit tak, aby ke každému výsledku byl přiřazen seznam rozpoznávaných osob. Dvě metody rozpoznávání by poté bylo možné porovnat na základě osob, které druhá metoda rozpoznala navíc a osoby, které již nerozpoznala. V ideálním případě by metoda s vyšší úspěšností neměla ztrácet osoby rozpoznané metodou s úspěšností nižší. Přínosem tohoto rozšíření by bylo např. porovnání množiny rozpoznávaných osob metod využívající PMP nebo GF. Tímto způsobem lze také určit, které osoby z databáze dosud nebyly rozpoznány.

Dalším logicky možným rozšířením je aplikování všech uvedených metod rozpoznávání na jiné obličejové databáze společně s dalším srovnáním proti deskriptorům jiného typu. Zde představené deskriptory byly v sekci 6.5 dotečně aplikovány na databázi DTD a hlavní příčinou neuspokojivých výsledků bylo použití příliš jednoduchého klasifikátoru *1NN*. Přínosem by tak byla implementace pokročilejších klasifikátorů v kombinaci s nově představenými deskriptory E-LBP a S-LBP. Současně se nabízí implementace aplikace pro detekci a rozpoznávání obličeje v reálném čase a následné porovnání zde představených deskriptorů z pohledu výkonu a úspěšnosti rozpoznávání v reálném prostředí.

8 Závěr

Na začátku této práce byly vysvětleny základní pojmy z oblasti detekce a rozpoznávání obličejů. Byly představeny existující obličejové databáze společně s nově vzniklou databází ČTK, která je v této práci hlavně využívána. Dále byl popsán proces rozpoznávání využívající obličejových deskriptorů na bázi lokálních binárních vzorů. Společně s deskriptorem LBP byly uvedeny možnosti jeho rozšíření a také dva hlavní způsoby extrakce příznaků: pravidelná mřížka podhistogramů a určení důležitých obličejových bodů pomocí Gáborových filtrů. Byly také popsány dva různé způsoby porovnání příznakových vektorů: Euklidovská vzdálenost a interseke histogramů. Součástí teoretické přípravy bylo také seznámení s knihovnou OpenCV a jejími funkcemi.

Na základě nastudovaného popisu existujících obrazových deskriptorů založených na metodě binárních vzorů byly navrženy dva nové obrazové deskriptory E-LBP a S-LBP, které jsou odvozeny z originálního deskriptoru LBP. Po analýze požadavků v sekci 5.1 byla implementována aplikace v jazyce C++ s využitím knihovny OpenCV pro testování úspěšnosti nově navržených deskriptorů. V kapitole věnující se implementaci je znázorněna struktura aplikace, dostupné parametry a způsob spuštění aplikace. V této práci byla také popsána vlastní implementace určení důležitých obličejových bodů pomocí Gáborových filtrů.

V rámci experimentů byly otestovány nově navržené deskriptory na databázi ČTK a nalezeny jejich nejlepší konfigurace. S těmito konfiguracemi byly deskriptory úspěšně aplikovány také na databázi FERET. Oba nově představené deskriptory se ukázaly velice schopné v kombinaci s uniformními vzory pravidelnou mřížkou podhistogramů. Jejich hlavní výhodou je násobně vyšší rychlost proti neuniformním či jiným pokročilejším deskriptorům. S úspěšností 65 % překonávají všechny dosud známé výsledky s pravidelnou mřížkou podhistogramů na databázi ČTK a dokonce i jiné implementace Gáborových filtrů. Určování důležitých obličejových bodů pomocí Gáborových filtrů se ukázalo jako velmi úspěšné a po nalezení vhodných parametrů bylo dosaženo výsledku 72,06 %, který stanovuje nové maximum na databázi ČTK. Vybrané rozpoznávací metody byly následně testovány na úloze rozpoznávání textur z databáze DTD. Na základě všech dosažených výsledků bylo navrženo další rozšíření.

Literatura

- [1] *Open Source Computer Vision Library* [online]. Itseez, 2016. [cit. 2016/12/29]. Dostupné z: <http://opencv.org/>.
- [2] AHONEN, T. – HADID, A. – PIETIKÄINEN, M. Face Recognition with Local Binary Patterns. Technical report, University of Oulu, Finland, 2004. Dostupné z: http://www.ee.oulu.fi/mvg/files/pdf/pdf_494.pdf. Machine Vision Group, Infotech Oulu.
- [3] CIMPOI, M. et al. Describing Textures in the Wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. Dostupné z: <http://www.robots.ox.ac.uk/~vgg/publications/2014/Cimpoi14/cimpoi14.pdf>.
- [4] *The OpenCV Reference Manual*. Itseez, 3.2.0 edition, December 2016. Dostupné z: <http://docs.opencv.org/3.2.0>.
- [5] KOŠAŘ, I. V. Srovnání deskriptorů pro reprezentaci obrazu. Diplomová práce, Západočeská univerzita v Plzni Fakulta aplikovaných věd Katedra informatiky a výpočetní techniky, 2015.
- [6] LENC, L. – KRÁL, P. Local binary pattern based face recognition with automatically detected fiducial points. *Integrated Computer-Aided Engineering* 23. 2016, s. 129–139. ISSN 1069-2509. doi: 10.3233/ICA-150506. Dostupné z: <http://content.iospress.com/download/integrated-computer-aided-engineering/ica506?id=integrated-computer-aided-engineering%2Fica506>.
- [7] LENC, L. – KRÁL, P. Unconstrained Facial Images: Database for Face Recognition under Real-world Conditions. In *14th Mexican International Conference on Artificial Intelligence (MICAI 2015)*, Cuernavaca, Mexico, 25-31 October 2015 2015. Springer. Dostupné z: https://home.zcu.cz/~pkral/papers/kral_micai15.pdf.
- [8] LENC, L. – KRÁL, P. *Database for Face Recognition under Real-world Conditions* [online]. 2015. [cit. 2016/12/27]. Dostupné z: <http://ufi.kiv.zcu.cz/>.
- [9] LÓPEZ, L. S. Local Binary Patterns applied to Face Detection and Recognition. Technical report, Universitat Politècnica de Catalunya, Barcelona, Spain, November 2010. Dostupné z:

[https://upcommons.upc.edu/bitstream/handle/2099.1/10772/PFC_LauraSanchez_\(LBP_applied_to_FaceDetection&Recognition\).pdf](https://upcommons.upc.edu/bitstream/handle/2099.1/10772/PFC_LauraSanchez_(LBP_applied_to_FaceDetection&Recognition).pdf).
Signal Theory & Communication Department.

- [10] ORO, D. et al. Real-time GPU-based Face Detection in HD Video Sequences. Technical report, Universitat Politècnica de Catalunya, Barcelona, Spain, 2011. Dostupné z: http://www.hertasecurity.com/sites/default/files/publication/files/PUBLICACION_39224000361.pdf. Herta Security Barcelona Spain.
- [11] RAHIM, M. A. et al. Face Recognition using Local Binary Patterns (LBP). *Global Journal of Computer Science and Technology*. 2013, 13, 4. ISSN 0975-4350. Dostupné z: https://globaljournals.org/GJCST_Volume13/1-Face-Recognition-using-Local.pdf.
- [12] TURK, M. – PENTLAND, A. Eigenfaces for Recognition. Technical report, Vision and Modeling Group, The Media Laboratory, Massachusetts Institute of Technology, Massachusetts, 1991. Dostupné z: <http://www.face-rec.org/algorithms/PCA/jcn.pdf>.
- [13] VIOLA, P. – JONES, M. Rapid Object Detection using a Boosted Cascade of Simple Features. Technical report, Mitsubishi Electric, Research Labs Compaq CRL, Cambridge, MA 02139, 2001. Dostupné z: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>.
- [14] VU, N.-S. – CAPLIER, A. Face Recognition with Patterns of Oriented Edge Magnitudes. *Springer-Verlag Berlin Heidelberg*. 2010. Dostupné z: <https://pdfs.semanticscholar.org/d03e/4e938bcb25aa0feb83d8a0830f9cd3eb3ea.pdf>.
- [15] ZHANG, B. et al. Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor. *IEEE TRANSACTIONS ON IMAGE PROCESSING*. 2010, 19, 2. Dostupné z: http://www98.griffith.edu.au/dspace/bitstream/handle/10072/32176/59885_1.pdf?sequence=1.

Přílohy

A Obsah přiloženého DVD

- Dokumentace

- src** - zdrojové soubory \LaTeX a obrázky

- out** - vygenerovaný soubor PDF s textem bakalářské práce

- log** - logovací soubory s výsledky

- Software

- src** - zdrojové kódy aplikace

- out** - spustitelná aplikace

- textures** - upravené zdrojové kódy použité pro rozpoznávání textur

- Obličejová databáze ČTK (*cropped*)