



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

MOBILNÍ APLIKACE PRO AUTOMATICKOU DETEKCI A ROZPOZNÁVÁNÍ TVÁŘÍ

MOBILE APPLICATION FOR AUTOMATIC FACE DETECTION AND RECOGNITION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Kristina Stejskalová

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Branislav Hesko

BRNO 2017

Bakalářská práce

bakalářský studijní obor Biomedicínská technika a bioinformatika
Ústav biomedicínského inženýrství

Studentka: Kristina Stejskalová
Ročník: 3

ID: 173574
Akademický rok: 2016/17

NÁZEV TÉMATU:

Mobilní aplikace pro automatickou detekci a rozpoznávání tváří

POKYNY PRO VYPRACOVÁNÍ:

1) Nastudujte problematiku detekce tváře v obraze. 2) Vytvořte přehled metod používaných k rozpoznávání lidské tváře. 3) Vhodně zvolte libovolnou metodu detekce tváře a implementujte ji v programovacím jazyku JAVA. 4) Vyberte a implementujte klasifikátor pro rozpoznávání tváře. 5) Vytvořte mobilní aplikaci pro systém Android, která bude načítat obrázky z fotoaparátu a nalezené tváře přiřazovat jednotlivým osobám. 6) Vaši výslednou aplikaci otestujte na vzorcích alespoň pěti osob a dosažené výsledky popište.

DOPORUČENÁ LITERATURA:

[1] HJELMAS, Erik a Boon Kee LOW. Face Detection: A Survey. Computer Vision and Image Understanding. 2001, 83(3), 236-274. DOI: 10.1006/cviu.2001.0921. ISSN 10773142.

[2] GONG, Shaogang., Stephen J. MCKENNA a Alexandra. PSARROU. Dynamic vision: from images to face recognition. London: Imperial College Press, 2000. ISBN 1880941818.

Termín zadání: 6.2.2017

Termín odevzdání: 2.6.2017

Vedoucí práce: Ing. Branislav Hesko

Konzultant:

prof. Ing. Ivo Provazník, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Fakulta elektrotechniky a komunikačních technologií, Vysoké učení technické v Brně / Technická 3058/10 / 616 00 / Bmo

Abstrakt

Tato bakalářská práce má za úkol implementovat metodu pro detekci a rozpoznávání tváře v programovacím jazyku Java a následné vytvoření Android aplikace pro detekci a rozpoznávání tváře. Teoretická část popisuje přehled metod určených k detekci a rozpoznávání tváře. V praktické části je v každé kategorii jedna metoda vybrána a implementována, vytvořený kód je pak základem pro vznik mobilní aplikace.

Klíčová slova

detekce tváře, rozpoznávání tváře, Viola-Jones, SURF, Android

Abstract

This bachelor thesis aims to implement face detection and face recognition method in Java and creating an Android application for face detection and recognition. Methods generally used for face detection and recognition are described in theoretical part. Chosen methods for face detection and face recognition are implemented and then used for building mobile application.

Keywords

face detection, face recognition, Viola-Jones, SURF, Android

Bibliografická citace:

STEJSKALOVÁ, K. *Mobilní aplikace pro automatickou detekci a rozpoznávání tváří*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 42 s. Vedoucí práce: Ing. Branislav Hesko.

Prohlášení

„Prohlašuji, že svou závěrečnou práci na téma Mobilní aplikace pro automatickou detekci a rozpoznávání tváří jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Branislavu Heskovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

.....
podpis autora

Obsah

Úvod	8
1 Detekce obličeje	9
1.1 Detekce založená na vědomostech	9
1.2 Detekce založená na vlastnostech.....	10
1.3 Detekce založená na porovnávání šablon	12
1.4 Detekce založená na statistických metodách	13
2 Rozpoznávání tváří	17
2.1 SURF	17
2.1.1 Detekce klíčových bodů	18
2.1.2 Popis klíčových bodů.....	19
2.1.3 Index pro shodu.....	19
2.2 PCA.....	20
2.3 LDA.....	21
2.4 LPP.....	22
3 Implementace algoritmu pro detekci a rozpoznávání tváře	24
3.1 Metoda Viola-Jones.....	24
3.2 Metoda SURF	27
4 Mobilní aplikace pro automatickou detekci a rozpoznávání tváře	30
5 Vyhodnocení	32
5.1 Vyhodnocení metody pro rozpoznání tváře	32
5.2 Vyhodnocení aplikace	33
6 Závěr	36
Seznam použité literatury	37
Seznam příloh.....	40

Seznam obrázků

- Obrázek 1.1 Typický obličej užívaný v top-down metodě
- Obrázek 1.2 Blokové schéma detekce obličeje na základě barvy kůže
- Obrázek 1.3 Rozložení barvy kůže v Y, Cb a Cr kanálu YCbCr modelu
- Obrázek 1.4 Šablona lidského obličeje zepředu
- Obrázek 1.5 Vzor zobrazující vztahy mezi charakteristickými oblastmi
- Obrázek 1.6 Haarovy příznaky
- Obrázek 1.7 Integrovaný obraz
- Obrázek 2.1 Krabicové filtry ukazující $L_{yy}(x,\sigma)$, $L_{xy}(x,\sigma)$, $D_{yy}(x,\sigma)$ a $D_{xy}(x,\sigma)$
- Obrázek 2.2 Demonstrace vzniku deskriptoru
- Obrázek 2.3 Ukázka indexu shody
- Obrázek 2.4 Eigenfaces
- Obrázek 2.5 Fisherfaces
- Obrázek 2.6 Laplacianfaces
- Obrázek 3.1 Ukázka vytvořené databáze
- Obrázek 3.2 Předzpracování pomocí odstínů šedi a pomocí YCbCr
- Obrázek 3.3 Ukázka detekce při přítomnosti vousů
- Obrázek 3.4 Ukázka detekce při nestandardním výrazu
- Obrázek 3.5 Ukázka detekce při tmavé barvě pleti
- Obrázek 3.6 Ukázka výsledku rozpoznávání
- Obrázek 3.7 Ukázka uložení klíčových bodů do obrázku
- Obrázek 3.8 Ukázka rozpoznání tváře
- Obrázek 4.1 Ukázka grafické tvorby aplikace
- Obrázek 5.1 Povolení aplikace ke spuštění fotoaparátu
- Obrázek 5.2 Úvodní obrazovka aplikace
- Obrázek 5.3 Náhled fotoaparátu
- Obrázek 5.4 Zpráva o začátku detekce po stisknutí tlačítka
- Obrázek 5.5 Výsledek rozpoznání

Seznam tabulek

Tabulka 3.1 Výsledky metod pro předzpracování obrazu

Tabulka 5.1 Vyhodnocení rozpoznávání tváří

ÚVOD

Práce se zabývá problematikou detekce a rozpoznání obličeje v obraze. Je to jedna z oblastí, kterou se zabývá počítačové vidění. Detekce obličeje se může využívat např. v kriminalistice ke zjištění totožnosti zájmové osoby. Navíc i co se týče sociologického hlediska, rozpoznávání známých tváří je nejčastější podvědomou činností mozku.

První pokusy o detekci tváře se datují na začátek 70. let, kdy se používaly jednoduché heuristické a antropometrické techniky. Z počátku bylo několik problému, proto zájem o výzkum jaksi uvízl až do 90. let, kdy se praktická detekce obličeje stala skutečností. Během posledního desetiletí proběhlo několik výzkumů, které přinesly důležité poznatky o detekci tváře. Používání statistik a neuronových sítí umožnilo detekovat tváře ze zaplněných scén v různých vzdálenostech od fotoaparátu [3].

Rozpoznávání tváří je pro lidskou bytost velice jednoduchým úkolem. Ovšem pro počítač je poměrně těžký úkol správně obličej přiřadit.

První polo-automatizovaný systém pro rozpoznání tváří byl vytvořen v 60. letech 20. století. Vyžadoval ovšem, aby obsluha lokalizovala na fotografiích rysy (např. oči, uši, nos, pusy) a až poté spočítal vzdálenosti a poměry společného referenčního bodu, který byl potom porovnán s referenčními údaji [10].

Automatické rozpoznávání tváří je poměrně novým konceptem. Prvotní algoritmy pro rozpoznání využívaly jednoduché geometrické modely, ale rozpoznávací proces v dnešní době dospěl do vědy sofistikované matematické reprezentace a odpovídajících procesů. Velké pokroky a iniciativy v posledních deseti až patnácti letech pohánějí technologie rozpoznávání tváří do popředí. Rozpoznávání tváří může být využito jak pro ověření, tak pro identifikaci.

Cílem této bakalářské práce je implementovat metodu pro detekci a rozpoznávání tváře v jazyce Java a následně úspěšné vytvoření mobilní aplikace pro automatickou detekci a rozpoznávání tváří

Samotná práce se skládá celkem z 6 kapitol. První dvě kapitoly přibližují metody užité pro detekci a rozpoznávání tváří. V kapitole 3 je vysvětlen postup implementace vybraných metod, na což navazuje kapitola 4 s popisem vytvořené mobilní aplikace. V kapitole 5 najdeme zhodnocení dosažených výsledků a poslední kapitolou je závěr.

1 DETEKCE OBLIČEJE

Detekce obličeje určuje pozici obličeje v obrázku. Jedná se o rozvíjející se směr rozpoznávání vzorů a klíčovou technologií v získávání informací o lidském obličeji.

Detekci obličeje lze rozdělit do čtyř kategorií: detekce založená na vědomostech, založená na vlastnostech, založená na metodě porovnávání šablon a založená na statistických metodách. Každá z těchto kategorií se dá ještě dál dělit do podkategorií.

Při detekci obličeje existuje několik faktorů, které mohou jeho detekci ztížit [1].

1. Pozice kamery nebo obličeje.

Snímky mohou vzhledem k pozici obličeje pořízeny z několika možných úhlů. Může tak dojít k zakrytí charakteristických rysů obličeje, jako např. při focení z profilu je zakryta celá půlka obličeje.

2. Přítomnost/absence významných prvků obličeje.

Při tomto problému mluvíme hlavně o brýlích, vousech atp. U těchto prvků nemusí být problémem jen jejich přítomnost, ale také to, že mají rozdílné tvary, velikosti, barvy.

3. Výraz v obličeji.

Výrazy obličeje bezesporu ovlivňují náš vzhled. Ať se jedná o křečovitý úsměv, nebo mrknutí při pořízení fotografie.

4. Zakrytí obličeje.

Ať se jedná o stín ze slunce, zakrytí pokrývkou hlavy, vlasy v obličeji, zakrytí obličeje ve skupinové fotografii atd.

5. Světelné podmínky.

Faktory jako intenzita světla, úhel dopadu ovlivňují výsledný obrázek. Může dojít např. ke změně barvy kůže.

1.1 Detekce založená na vědomostech

Tato metoda je nazývána také jako Top-down metoda. Tato metoda využívá obecných vlastností obličeje. Lidský obličej obsahuje dvojice oči, které jsou symetricky uspořádané, a nos, který tvoří středový bod obličeje.

G. Yang a T.S. Huang [7] zveřejnili algoritmus pro detekci obličeje z čelního pohledu. Obličej může být podle nich prezentován jako určitá sada pravidel. Tento obličej je vidět na Obrázek 1.1.



Obrázek 1.1 Typický obličej užívaný v top-down metodě [5]

U těchto metod existují tři úrovně [5]:

Úroveň 1: aplikuje pravidlo, že „středová část obličeje obsahuje 4 buňky s jednotnou intenzitou“ pro hledání tváře,

Úroveň 2: lokální vyrovnání histogramu následováno vyrovnáním hran s následnou detekcí hran,

Úroveň 3: hledání očí a úst pro potvrzení.

U těchto metod je největším problémem, že pokud si pravidla určíme moc obecné, může dojít k několika falešným detekcím, a pokud pravidla uděláme moc podrobná, mnoho obličejů nemusí být vůbec detekovaných [1].

Metody využívající vyhledávání tváří prohledávají jednotlivé části obrazu a porovnávají je se vzory obličejů získaných ze vzorových obrazů. Vzorem může být sada obličejů, která je porovnávána se zkoumanou částí obrazu. Jednodušší a méně náročnou metodou je použití šablony s „univerzálním“ obličejem. Tato šablona pouze určí, jestli se v obrazu jedná o obličej či jiný objekt. Šablona se postupně posunuje obrázkem a hledá místo největší shody. Vlastní detekce probíhá výpočtem korelace mezi vstupním obrazem a šablonou. Pokud je korelace dostatečně velká, vstupní obraz je považován za obličej. Aby bylo vyhledávání dokonalejší, zavádí se další afinní transformace, ovšem tyto operace mohou proces výrazně zpomalit.

Kromě detekce celé hlavy můžeme použít také pouze detekce očí nebo úst. Je to díky tomu, že právě oči a ústa mají pevně dané umístění vzhledem k celému obličejí.

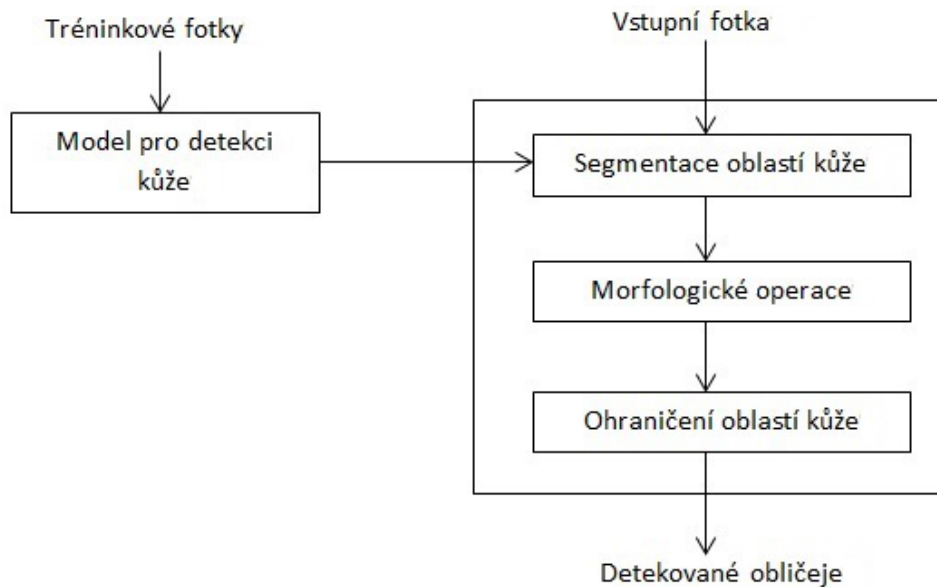
1.2 Detekce založená na vlastnostech

Tyto metody využívají jedinečných rysů tváře a také jejich geometrických vztahů. Těmito rysy mohou být oči, obočí, nos, ústa nebo vousy, které jsou použity pro detekci tváře. Struktura a velikost těchto elementů je jiná oproti neobličejovým objektům.

Je opakem detekce založené na vědomostech. Nejdříve najedeme neměnné rysy obličeje a pak integrujeme prvek, který určuje, zda je rozpoznávaná oblast obličej, nebo ne. Obvykle k extrakci používáme detektor hran a poté vytváříme

statistický model, který popisuje vztah mezi neměnnými rysy obličeje a pozitivní detekcí tváře [6].

Do této kategorie spadá metoda detekce na základě barvy kůže. Metoda funguje podle blokového schématu viz. Obrázek 1.2. Nejdříve se analyzují „tréninkové“ fotky, které jsou manuálně vytvořeny uživatelem. Obsahují pouze výřezy obličejů, nebo jsou místa, které neobsahují kůži na fotce, začerněny. Poté jsou vytvořeny kritéria pro barvu pixelů, které jsou prohlášeny za kůži.



Obrázek 1.2 Blokové schéma detekce obličeje na základě barvy kůže

V počítačové grafice se využívá více barevných modelů a nejvíce je využíván model $YCbCr$. Výhodou tohoto modelu je, že informaci o barvě nesou pouze dvě složky. C_b nese informace o rozdílu mezi modrou složkou a referenční hodnotou, C_r potom nese informace o rozdílu mezi červenou složkou a referenční hodnotou. Písmenko Y nese informaci o tzv. luminaci, což je pojem udávající sílu jasu. Transformace z prostoru RGB do prostoru $YCbCr$ je definována jako [4]:

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \quad 1.1$$

$$C_b = -0,1687 \cdot R - 0,3313 \cdot G + 0,5 \cdot B + 128 \quad 1.2$$

$$C_r = 0,5 \cdot R - 0,4187 \cdot G - 0,0813 \cdot B + 128 \quad 1.3$$

Také můžeme odvodit kritéria pro barvu kůže jak v RGB (pro 8-bitový kanál R , G a B), tak $YCbCr$ modelu (pro 8-bitový kanál C_r a C_b): [2]

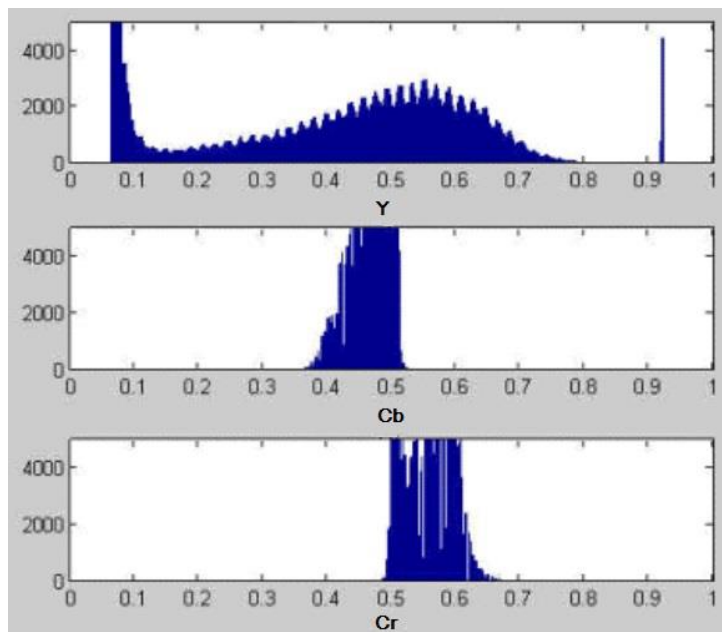
$$(R > 95) \text{ AND } (G > 40) \text{ AND } (B > 20) \text{ AND } (\max\{R, G, B\} - \min\{R, G, B\} > 15) \text{ AND } (|R - G| > 15) \text{ AND } (R > G) \text{ AND } (R > B) \quad 1.4$$

$$100 \leq C_b \leq 130$$

$$130 \leq C_r \leq 160$$

1.5

Po určení kritéria pro barvu kůže může začít samotná detekce. Po segmentaci obrazu na základě barvy kůže nám vzniká binární matice, která nám určí, které části obrazu (pixely) jsou považovány za kůži a které nikoliv.



Obrázek 1.3 Rozložení barvy kůže v Y, Cb a Cr kanálu YCbCr modelu [2]

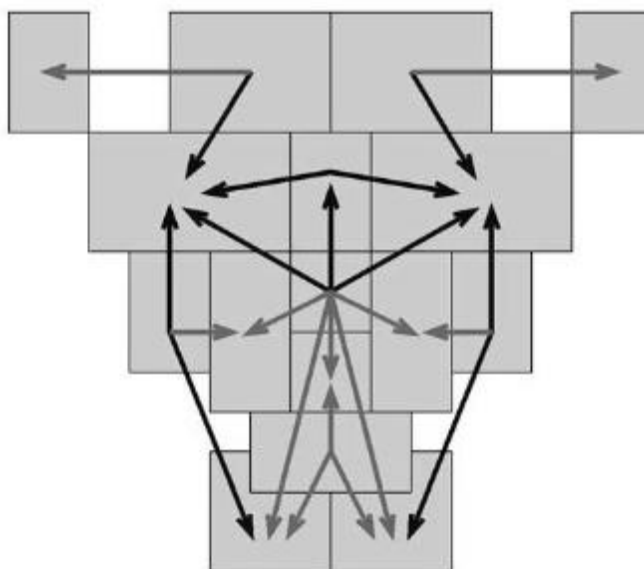
1.3 Detekce založená na porovnávání šablon

Metodu můžeme dál rozdělit do dvou kategorií podle používaných šablon: předem definované šablony a upravovatelné šablony. Tato metoda je využívána, kdy očekáváme velkou pravděpodobnost, že se na obrázku vyskytuje lidská tvář. Šablona je předdefinovaná struktura jednotné velikosti a tvaru, díky čemuž je detekce tváře jednoduchá, jelikož pouze srovnáváme šablonu s objekty na obrázku [5]. Na Obrázek 1.4 vidíme šablonu lidského obličeje zepředu [5].



Obrázek 1.4 Šablona lidského obličeje zepředu [5]

Jedná se o metody založené na předdefinovaných vzorech. Klíčovým prvkem této metody je fakt, že osvětlení mění jas jednotlivých rysů obličeje (oči, tvář apod.), ale relativní změny jasu mezi těmito částmi se více méně nemění [1]. Pokud určíme právě poměry jasů těchto rysů a jejich směry, získáme určitou neměnnost při změnách osvětlení. Závislosti změn jasů jsou zakódovány jako poměry jasů vhodné zvolených oblastí, které odpovídají specifickým charakteristikám obličeje. Obličej je detekován, pokud jsou poměry jasů sousedních oblastí shodné a odpovídá i směr. Na Obrázek 1.5 jsou šipkami vyznačeny vztahy mezi definovanými oblastmi. Základních vztahů je 11 a jsou zobrazeny černými šipkami a šedé šipky, kterých je 12, jsou potvrzovací vztahy.



Obrázek 1.5 Vzor zobrazující vztahy mezi charakteristickými oblastmi [1]

Craw a kol. [18] navrhli metodu určování polohy založenou na tvaru přední plochy šablony (tvaru obličeje). Využívají Sobelův filtr, který extrahuje okraje (hrany obličeje) a tyto okraje jsou uspořádány dohromady, vzhledem k několika omezením užitým k hledání šablony obličeje. Po najetí pozice okraje hlavy je stejný proces opakovaně použit k nalezení polohy očí, obočí a rtů a dalších rysů obličeje v různém měřítku.

Nevýhodou této metody je, že je závislá na velikosti, rotaci a měřítku šablony, protože obličej musí mít tyto parametry stejné jako šablona.

1.4 Detekce založená na statistických metodách

Tyto metody se opírají o statistické analýzy a strojové učení, aby našly obličejové a neobličejové části. Patří mezi ně např. skrytý Markovův model nebo algoritmus AdaBoost.

Nejdřív proběhne předběžné zpracování obrazu, které zahrnuje např. rotaci nebo změnu měřítka. Poté se používají klasifikátory pro rozlišování a označování potřebných údajů. Následně je zvýrazněn detekovaný obličej [5].

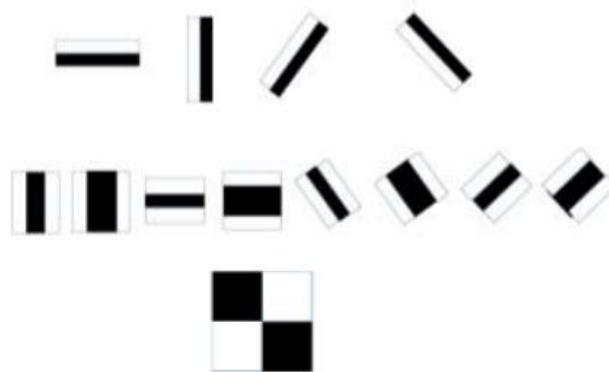
Algoritmus AdaBoost kombinuje slabé a silné klasifikátory a vytváří kaskádový klasifikátor. Obraz obsahující obličej je nejdřív prohledáván slabým klasifikátorem, a pokud překročí určitou hodnotu prahu, projde slabým klasifikátorem do silného klasifikátoru.

Velice rychlým a přesným způsobem detekce obličeje je metoda Viola-Jones [9], zveřejněna roku 2001. Kombinuje následující čtyři kroky [5]:

1. Jednoduché obdélníkové matice, zvané jako Haarovy příznaky
2. Integrální obraz pro rychlou detekci
3. Metodu strojového učení AdaBoost
4. Kaskádový klasifikátor

Haarovy příznaky

Snahou detektoru Viola-Jones je získat velkou řadu jednoduchých příznaků s minimálními výpočetními nároky. Tyto příznaky jsou určeny k detekci variací tmavé a světlé části obrazu. Tento výpočet vytvoří jediný obdélník kolem detekované plochy. Na základě barevného odstínu v blízkosti nosu nebo čela se vytvoří obrys. Některé běžně užívané Haarovy příznaky jsou vidět na Obrázek 1.6.



Obrázek 1.6 Haarovy příznaky [5]

Hodnota Haarového příznaku je rovna jeho odezvě na vstupní obraz. Může být vypočítána pomocí rovnice [19]:

$$f(x) = w_0 r_0 + w_1 r_1, \quad 1.6$$

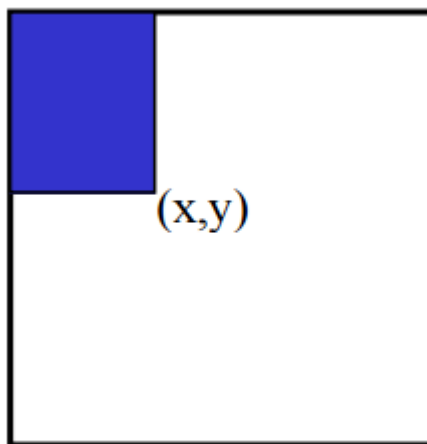
kde $f(x)$ je odezva Haarova příznaku na vstupní snímek x , w_0 je váha bílé obdélníkové oblasti r_0 , w_1 je váha černé obdélníkové oblasti r_1 .

Většinou jsou příznaky generovány pro velikost detekčního okna 24x24 pixelů. Při procesu generování příznaku jsou nejdříve nastaveny jeho nejmenší možné rozměry. Poté je příznak posouván po jednom pixelu horizontálně nebo vertikálně. Při každém posuvu je do seznamu příznaků přidán nový příznak. Jakmile

se příznak posune přes celé okno, zvětší se a znovu se posunuje, dokud jeho velikost není větší než velikost okna.

Integrální obrazy

Jsou známé také jako souhrnné tabulky. Integrální obraz se používá k jednoduššímu a rychlejšímu výpočtu pro jednotlivé příznaky.



Obrázek 1.7 Integrální obraz

Integrální obraz v místě (x,y) obsahuje součet pixelů ve směru nahoru a vlevo od (x,y) včetně, viz. Obrázek 1.7.

$$I(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad 1.7$$

Vlastní výpočet neprobíhá v celém okně, ale je nutné pro výpočet jednotlivých Haarových příznaků vytvořit podokno, které se posouvá ve vyznačeném směru přes celý integrální obraz.

Strojové učení metodou AdaBoost

Využívá kombinace různých klasifikátorů konstruovaných ze stejného datasetu. Kombinací klasifikátorů se zlepšuje nestabilní klasifikátor, což je klasifikátor, kde malá změna v parametrech učení vyvolá velké změny v celém klasifikátoru.

AdaBoost algoritmus pomáhá vybrat drobné rysy obličeje, které usnadňují a urychlují výpočty. Obsahuje následující kroky [5]:

- vstupem je množina snímků $(x_1, y_1) \dots (x_n, y_n)$, kde x_i je snímek a $y_i = \{0, 1\}$ určuje pozitivní nebo negativní třídu
- počet všech snímků je n a $i \in \{1, n\}$
- inicializace vah $w_{1,i} = \left\{ \frac{1}{2m}, \frac{1}{2l} \right\}$ kde m a l jsou čísla pozitivních a negativních snímků

- pro $t = 1, \dots, T$:

1. normalizujeme váhy:

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,i}} \quad 1.8$$

2. pro každý znak j , naučíme klasifikátor h_j , který je omezen pro použití jediného znaku. Chyba je vyhodnocena s ohledem na w_t

$$E_t = \sum_i w_i |h_j(x_i), y_i| \quad 1.9$$

3. vybereme klasifikátor h_j s nejnižší chybou E_t

4. zaktualizujeme váhy

$$w_{t+1,i} = w_{t,i} B_t^{1-e_i}, \quad 1.10$$

kde $e_j = 0$, pokud je x_i klasifikován správně. Jinak $e_j = 1$ a

$$B_t = \frac{e_t}{1-e_t}$$

- výsledný silný klasifikátor je:

$$h(x) = 1, \quad \sum_{t=1}^T a_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T a_t, \quad 1.11$$

$$\text{kde } a_t = \log \frac{1}{B_t}$$

Učební proces AdaBoost je rychlý a poskytuje větší počet požadovaných dat. Tato data pak mohou být zařazena do klasifikátoru, který obsahuje drobné rysy obličeje. Obvykle se používá pro detekci vzorů. Tato metoda je velice přesná a má dobrou detekční rychlost s přibližně 1% falešnou detekcí, ale vyžaduje víc času na trénink [9].

Kaskádový klasifikátor

Detekční algoritmus Viola-Jones eliminuje kandidáty (části obrazu) na lidskou tvář velice rychle pomocí kaskád. Každá kaskáda eliminuje kandidáty tím, že v každé fázi připraví přísnější požadavky a je tak těžší projít vyšší úrovní klasifikátoru. Pokud v jakékoli fázi kaskády kandidát neprojde určitým stupněm klasifikátoru, neprojde do další úrovně. Pouze pokud splní všechny úrovně, je detekován jako obličej.

2 ROZPOZNÁVÁNÍ TVÁŘÍ

Plno rozpoznávacích technik bylo vyvinuto v posledních desítkách let a můžeme je rozdělit do dvou kategorií [12]:

1. Založené na vzhledu, které aplikují holistické prvky textury buď na celé oblasti, nebo na určité oblasti v obraze.
2. Založené na prvcích, které využívají geometrických rysů obličeje a geometrických vztahů.

Do první kategorie patří plno technik, ale asi nejvíce oblíbené pro účel rozpoznávání tváří jsou metody PCA nebo LDA. Metoda SIFT (Scale Invariant Feature Transform), kterou navrhl David G. Lowe [6] je také široce užívaná pro detekci a rozpoznání objektů, např. obličejů. Oproti tradičním algoritmům, metoda SIFT dokáže extrahovat místní specifické příznaky lidského obličeje. Vzhledem k vysokým výpočetním nárokům metody SIFT jsou navrhovány metody k jejímu zrychlení. Byl také navržen nový detektor a deskriptor nazývaný SURF. Celým názvem Speed-Up Robust Features byl navrhnout Herbertem Bayem [5]. SURF je detektor a deskriptor, který nepodléhá měřítku a rotaci a má srovnatelné, ne-li lepší výsledky jako SIFT.

PCA (Principal Component Analysis) je metoda redukce dimenzí, která promítá n dimenzionální data do k dimenzionálního prostoru, kde k je menší jak n , a k dimenzionální prostor je definován hlavními vlastními vektory kovarianční matice dat. V roce 1991 Turk a Pentland dosáhli rozpoznání tváří pomocí promítání tváří do prostoru zvaného „eigenfaces“ právě pomocí metody PCA [11].

LDA (Linear Discriminant Analysis) je naopak kontrolovaný algoritmus učení. Cílem LDA je najít lineární kombinaci znaků, které oddělují různé třídy objektů. Jinými slovy, LDA má za úkol najít prostor, ve kterém jsou objekty jiných tříd od sebe vzdáleny co nejdál, zatímco požadované objekty stejných tříd jsou blízko sebe. LDA bylo využito pro rozpoznávání tváří v roce 1997 Belhumeur, Hespanhem a Kriegmanem [12].

Avšak PCA a LDA nedokáží objevit podkladovou strukturu, která by mohla být nelineárního charakteru [13]. Právě proto byl roku 2005 zveřejněn nový přístup, který posuzuje rozmanitou strukturu pro analýzu tváře pomocí metody LPP (Locality Preserving Projections) [13].

2.1 SURF

SURF je detektor a deskriptor, který nepodléhá měřítku a rotaci. Detektor lokalizuje zajímavé body na obrázku a deskriptor popisuje příznaky těchto zajímavých bodů a konstruuje vektor těchto příznaků [14].

Na rozdíl od metody SIFT, SURF využívá oproti Gaussověmu filtru aproximaci Hessovy matice, která pracuje na integrálním obrazu, aby našla zajímavé body, což výrazně snižuje výpočetní dobu [5].

2.1.1 Detekce klíčových bodů

Jak již bylo řečeno, SURF využívá determinant přibližné Hessovy matice jako základ detektoru [14]. V bodě $x = (x, y)$ v obrazu I je Hessova matice $H(x, \sigma)$ v x v měřítku σ definována jako:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}, \quad 2.1$$

kde $L_{xx}(x, \sigma)$, $L_{xy}(x, \sigma)$ a $L_{yy}(x, \sigma)$ jsou konvolucemi druhé derivace Gaussovy funkce se vstupním obrazem I v bodě x .

K redukci výpočetního času je použita sada 9x9 krabicových filtrů, které jsou jednodušší a rychlejší na výpočet, jelikož aproximují druhé parciální derivace Gaussovy funkce. Označíme je jako $D_{xx}(x, \sigma)$, $D_{yy}(x, \sigma)$ a $D_{xy}(x, \sigma)$. Váhy aplikované na obdélníkové oblasti jsou jednoduché kvůli výpočetní účinnosti. Což vede k:

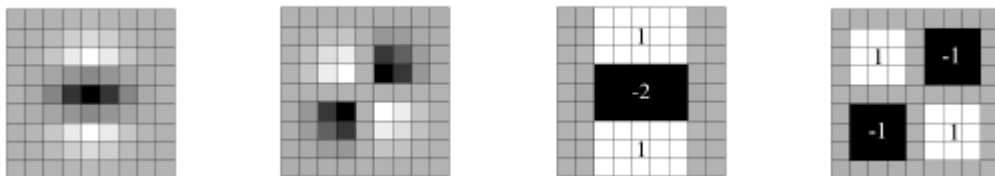
$$\det(H_{approx}) = D_{xx}D_{yy} - (\omega D_{xy})^2, \quad 2.2$$

kde ω je:

$$\omega = \frac{|L_{xy(1.2)}|_F |D_{yy(9)}|_F}{|L_{yy(1.2)}|_F |D_{xy(9)}|_F} = 0.912 \approx 0.9 \quad 2.3$$

$|X|_F$ je Frobeniova norma.

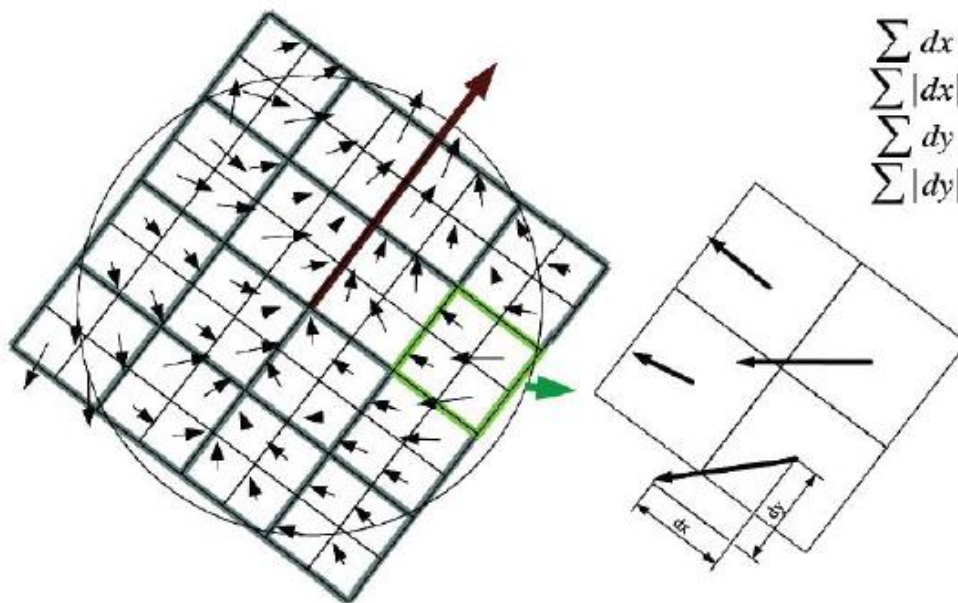
V případě invariantního měřítka SURF vytváří prostor ve tvaru pyramidy a přímo mění měřítko krabicových filtrů, aby implementoval prostor měřítka kvůli krabicovým filtrům a integrálnímu obrazu.



Obrázek 2.1 Krabicové filtry ukazující $L_{yy}(x, \sigma)$, $L_{xx}(x, \sigma)$, $D_{xy}(x, \sigma)$ a $D_{xy}(x, \sigma)$ [14]

2.1.2 Popis klíčových bodů

K popisu příznaku zajímavého bodu využívá SURF sumu odezvy Haarových vlnek. První krok pro extrakci deskriptoru se sestává z konstrukce čtvercového prostoru se středem v zajímavém bodu, podle kterého je také orientován. Tento prostor je rovnoměrně rozdělen do menších 4x4 čtvercových subregionů viz. Obrázek 2.2. Tím se zachovají důležité prostorové orientace. Pro každý subregion vypočteme odezvy Haarových vlnek na 5x5 rovnoměrně rozložené vzorové body. Pro zjednodušení nazýváme d_x odezvu Haarových vlnek v horizontálním směru, d_y odezvu Haarových vlnek ve směru vertikálním [14].

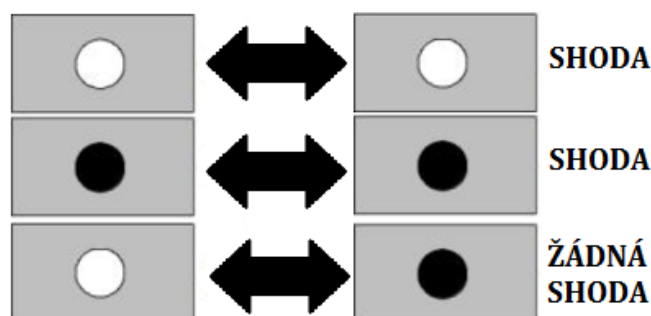


Obrázek 2.2 Demonstrace vzniku deskriptoru [14]

Poté jsou vlnkové odezvy d_x a d_y sečteny nad každým subregionem a vytvoří 2x4 příznaků, což tvoří první sadu vstupních dat ve vektoru příznaků. Dále sečteme absolutních hodnoty odezvy $|d_x|$ a $|d_y|$, což nám vytvoří dalších 2x4 příznaků. Výsledný vektor je tedy tvořen jako $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ o délce 64. Vlnkové odezvy jsou invariantní vůči změnám osvětlení. Přeměněním deskriptorového vektoru na jednotkový vektor se dosáhne toho, že jsou invariantní i vůči měnícímu se kontrastu [5].

2.1.3 Index pro shodu

Pro urychlení nalezení shody se používá znak Laplacián. V porovnávací fázi srovnává pouze funkce, které mají stejný typ kontrastu viz. Obrázek 2.3 [20].



Obrázek 2.3 Ukázka indexu shody

2.2 PCA

Základní algoritmus PCA může být rozdělen do 3 částí [17].

1) Extrakce příznaků

Nechť je $X = \{x_1, x_2, \dots, x_n\}$ maticí obsahující n obrazů tváří. Matice každého obrazu byly převedeny na vektor. Např., $m \times n$ matice je převedena na vektor o $m \times n$ řádcích. Prostor „eigenfaces“ můžeme získat následujícími kroky:

1. Vypočtememe průměr μ

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad 2.4$$

2. Vypočteme kovarianční matici S

$$S = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T \quad 2.5$$

3. Vypočítáme vlastní vektory v_i a vlastní hodnoty λ_i

$$Sv_i = \lambda_i v_i \quad 2.6$$

4. Seřadíme vlastní vektory sestupně podle jejich vlastních hodnot a vybereme k vlastních vektorů odpovídajících k největší vlastní hodnotě

Tyto k vlastní vektory tvoří prostor $W = \{v_1, v_2, \dots, v_n\}^T$, který se nazývá jako „eigenfaces“.



Obrázek 2.4 Eigenfaces [12]

2) Proces trénování

V tomto kroku je trénovací obrázek zobrazen do subprostoru pro výpočet maximální vzdálenosti dvou tváří, kterou nazýváme jako prahovou hodnotu [17]. Proces je následující: každý tréninkový prvek x_i je namapován do subprostoru, abychom získali projekci vektoru Ω_i :

$$\Omega_i = U^T(x_i - \mu) \quad (i = 1, 2, \dots, N) \quad 2.7$$

Potom je prahová vzálenost θ :

$$\theta = \frac{1}{2} \max_{i,j} (\|\Omega_i - \Omega_j\|) (i, j = 1, 2, \dots, N) \quad 2.8$$

3) Proces rozpoznávání

Nejdřív je lidský obličej Γ , určený k rozeznání, zobrazen to subprostoru eigenfaces, abychom získali projektivní vektor Ω_f :

$$\Omega_f = U^T(\Gamma - \mu) \quad 2.9$$

Jako klasifikační standard se používá Euklidovská vzdálenost. Definice Ω_f a každé tréninkové sady pro obličej je:

$$\varepsilon_i^2 = \|\Omega_f - \Omega_i\|^2 (i = 1, 2, \dots, N) \quad 2.10$$

Aby bylo možné rozlišit lidskou a nelidskou tvář, je vzdálenost ε mezi originálním obrazem Γ a obrazem Γ_f získaným z prostoru příznaků spočítána jako:

$$\varepsilon^2 = \|\Gamma - \Gamma_f\|^2, \quad 2.11$$

kde $\Gamma_f = U\Omega_f + \mu$. Pro rozpoznání obrazu pomocí minimální vzdálenosti, jsou pravidla klasifikace následující [17]:

1. pokud je $\varepsilon \geq \theta$, vstupní obraz není obrazem lidského obličeje.
2. pokud je $\varepsilon < \theta, \forall i, \varepsilon_i \geq \theta$, vstupní obraz obsahuje neznámý lidský obličej.
3. pokud je $\varepsilon < \theta, \varepsilon_i = \min \theta_i < \theta$, vstupní obraz je i -tým lidským obličejem v databázi.

2.3 LDA

LDA je kontrolovaný algoritmus učení, což znamená, že označení tříd v tréninkovém setu je použito v procesu učení [12]. Necht' je X matice obsahující tréninkové obrázky obličejů a X_i je matice obsahující obrázky obličejů patřící do třídy i .

$$X = \{X_1, X_2, \dots, X_c\} \quad 2.12$$

$$X_i = \{x_1, x_2, \dots, x_n\} \quad 2.13$$

Musíme vypočítat průměr stejně jako u metody PCA. Pak jsou matice rozptylu mezi třídami S_B a matice rozptylu v rámci jedné třídy S_W vypočítány jako:

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad 2.14$$

$$S_W = \sum_{i=1}^c \sum_{x_j \in X_i} (x_j - \mu_i)(x_j - \mu_i)^T, \quad 2.15$$

kde N_i reprezentuje počet tréninkových vzorků patřících do třídy i , μ_i představuje průměr tréninkových vzorků patřících do třídy i a μ představuje průměr všech vzorků.

Výsledný efekt dostaneme maximalizací poměru determinantu matice S_B a determinantu matice S_W . Matice S_W musí být regulární (čtvercová matice s nenulovým determinanem). Maximalizaci poměru determinantů pak spočítáme ze vzorce:

$$W_{opt} = \underset{w}{\operatorname{argmax}} \frac{|W^T S_B W|}{|W^T S_W W|} \quad 2.16$$

$$S_B v_i = \lambda_i S_W v_i \quad 2.17$$

Potom je $W_{opt} = \{w_1, w_2, \dots, w_m\}$ množinou vlastních vektorů. Sloupcové vektory jsou nazývány jako „fisherfaces“. Tato metoda je odolnější na vlivy osvětlení a změny výrazů tváře [12].



Obrázek 2.5 Fisherfaces [12]

2.4 LPP

Kroky této metody jsou následující:

1. PCA projekce – tréninkový set obrázků je nejdříve promítnut do prostoru PCA tím, že odstraní určité komponenty s malými vlastními hodnotami.
2. Konstrukce grafu nejbližšího souseda – necht' G označuje graf s n uzly. i -tý uzel se vztahuje k obličejí na obrázku x_i . Vytvoří se okraj mezi x_i a x_j , pokud jsou k sobě dostatečně blízko. Zde se využívá k -nejbližší soused, aby zkontroloval, jestli jdou dva uzly blízko u sebe nebo ne.
3. Úprava vah – pokud jsou uzly i a j spojeni, pak se $S_{ij} = 1$, jinak $S_{ij} = 0$.
4. Mapa prostoru – spočítání vlastních vektorů a vlastních hodnot:

$$XLX^T w = \lambda XDX^T w, \quad 2.18$$

kde X je tréninkový set obrázků, D je diagonální matice, jejíž vstupy jsou sloupcové sumy S a $L = D - S$.

Jakmile je vyřešen poslední krok, dostaneme prostor W_{LPP} . Kombinací matice PCA a matice LPP dostaneme:

$$W = W_{PCA} W_{LPP} \quad 2.19$$

Sloupcové vektory prostoru W se nazývají „laplacianfaces“.



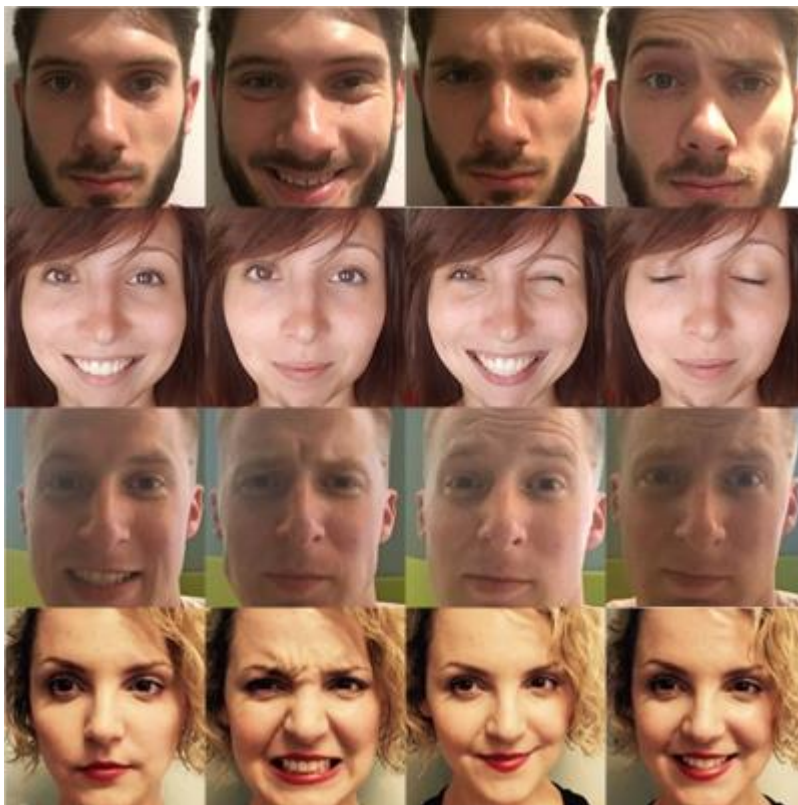
Obrázek 2.6 Laplacianfaces [12]

3 IMPLEMENTACE ALGORITMU PRO DETEKCI A ROZPOZNÁVÁNÍ TVÁŘE

V rámci bakalářské práce byla pro detekci tváří využita metoda Viola-Jones a pro rozpoznávání metoda SURF. Obě metody jsou již popsány v teorii práce.

Implementace obou metod využívala programovacího jazyka Java, ve vývojovém prostředí Eclipse (verze Mars) s využitím „open source“ zdrojů pomocí knihovny OpenCV (verze 2.4.13, volně dostupná z internetu). Díky OpenCV knihovně je možné zpracovávat obrazy ve formě matice. Pro spuštění a správné fungování programu je také nutné mít prostředí pro spuštění JAVA programu, tzv. JRE (také volně dostupné z internetu).

Databáze tvořená pro ověření úspěšnosti metod je tvořená fotkami 6 osob, od každé osoby obsahuje 10 odlišných fotek.



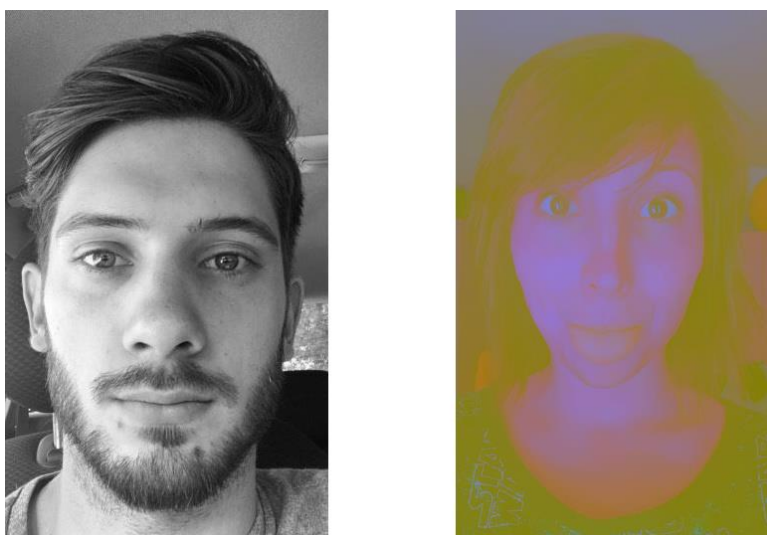
Obrázek 3.1 Ukázka vytvořené databáze

3.1 Metoda Viola-Jones

Jak již bylo psáno v kapitole 1.4, metoda využívá kaskádových klasifikátorů a tzv. Haarových příznaků pro detekci tváře v obraze. Výhodou OpenCV je, že jejím obsahem jsou již naučené kaskádové klasifikátory. Já pro svou práci používám

klasifikátor s názvem *haarcascade_frontalface_alt.xml*. Kód postupně detekuje tváře ve všech souborech obsažených ve vstupní složce (všech 60 obrázků databáze). Následně vypíše počet detekovaných obličejů a kromě toho, že do výstupní složky uloží původní fotky s detekovaným obličejem v zeleném rámečku (s příponou *_FACES.jpg*), také uloží pouze výřez tohoto rámečku (s příponou *_CUT.jpg*).

Stačí tedy nastavit pouze adresář vstupní složky s fotkami v databázi (*INPUT_FOLDER*) a adresář výstupní složky, kam chceme detekované obličeje uložit (*OUTPUT_FOLDER*). Pokud bychom chtěli změnit přípony fotek detekovaných obličejů nebo jejich výřezů, stačí přepsat proměnnou *RESULT_SUFFIX* nebo *CUTOUT_SUFFIX*.



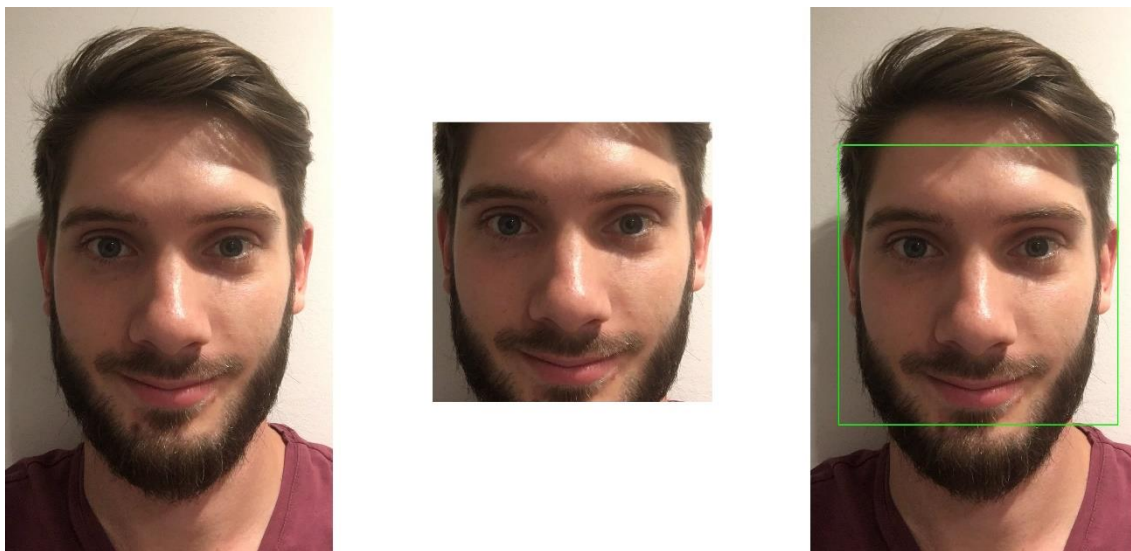
Obrázek 3.2 Předzpracování pomocí odstínů šedi (vlevo) a pomocí YCbCr (vpravo)

Při práci s obrazy je vhodné jejich předzpracování z několika důvodů. Může to být např. redukce dat, protože pokud převedeme barevný obraz do tónů šedi, budeme zpracovávat pouze $\frac{1}{4}$ dat oproti obrazu barevnému. V mé práci jsem všechny objekty z databáze podrobila dvojímu zpracování dat, převodu do tónů šedi a převodu do YCbCr modelu (viz. Obrázek 3.2). Výsledky detekce tváře obou dvou metod můžeme vidět v Tabulka 3.1. Metoda převedení do odstínu šedi byla úspěšnější, a proto jsem si ji vybrala pro další zpracování.

Tabulka 3.1 Výsledky metod pro předzpracování obrazu

Způsob předzpracování	Celkový počet souborů	TP	FN	FP	Úspěšnost [%]
RGB2GRAY	60	59	1	2	95,08
RGB2YCrCb	60	57	3	2	91,93

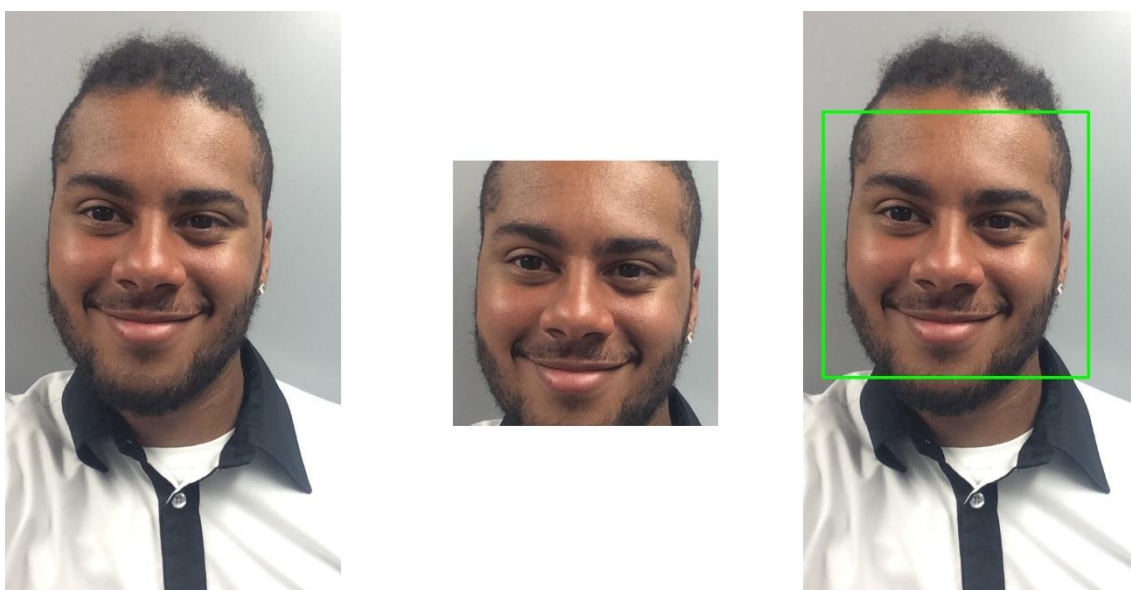
Jelikož je tato metoda využívána i při hledání a porovnání příznaků, je nutné, aby výřezy obličejů, které detekuje na fotkách a ukládá je, měly jednotnou velikost. Ta je nastavená na 260x260 pixelů.



Obrázek 3.3 Ukázka detekce při přítomnosti vousů



Obrázek 3.4 Ukázka detekce při nestandardním výrazu



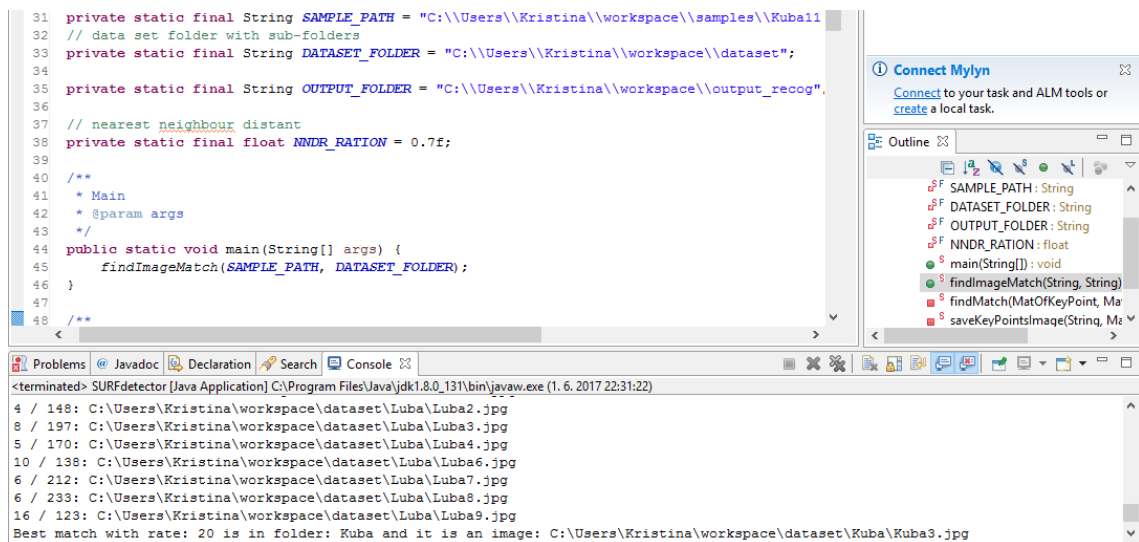
Obrázek 3.5 Ukázka detekce při tmavé barvě pleti

3.2 Metoda SURF

Tuto metodu jsem si vybrala pro výběr příznaků pro rozpoznávání tváří v obraze. Principem této metody je, že extrahuje příznaky ze vstupního obrazu a porovnává je s příznaky obrazů v databázi.

Na začátku kódu definuji několik proměnných, které označují adresář obrázku fotky, kterou chceme rozpoznat – vstupní obrázky (*SAMPLE_PATH*), adresář složky, která obsahuje fotky databáze – může být rozdělena do dalších složek podle názvů osob (*DATASET_FOLDER*). Jako poslední parametr určujeme adresář složky, kam chceme uložit obrázky s detekovanými příznaky (*OUTPUT_FOLDER*).

Výslednými obrázky jsou v této metodě obrázky s vyznačenými příznaky jednotlivých tváří. Po spuštění kódu se uživateli zobrazí proces porovnávání příznaků. U každé fotky z databáze jsou hned na začátku řádku vidět 2 čísla viz. Obrázek 3.6. První číslo udává, kolik příznaků má shodných vstupní obraz a daný obraz v databázi. Druhé číslo udává, kolik příznaků bylo extrahováno z daného obrazu v databázi. Po proběhnutí porovnání shody příznaků se všemi soubory v databázi kód vypíše, s jakým počtem shodných příznaků a v jaké složce se nachází obraz, se kterým našel nejvyšší shodu, tedy která osoba je na obrázku. Pokud uživatel jako vstupní obraz zadá název souboru, který není obsažen ve složce, kterou jsme si určili jako složku se vstupními obrazy, kód vypíše kontrolní hlášku.



```
31 private static final String SAMPLE_PATH = "C:\\Users\\Kristina\\workspace\\samples\\Kubal1
32 // data set folder with sub-folders
33 private static final String DATASET_FOLDER = "C:\\Users\\Kristina\\workspace\\dataset";
34
35 private static final String OUTPUT_FOLDER = "C:\\Users\\Kristina\\workspace\\output_recog".
36
37 // nearest neighbour distant
38 private static final float NNDR_RATIO = 0.7f;
39
40 /**
41  * Main
42  * @param args
43  */
44 public static void main(String[] args) {
45     findImageMatch(SAMPLE_PATH, DATASET_FOLDER);
46 }
47
48 /**
```

Connect Mylyn
Connect to your task and ALM tools or create a local task.

Outline

- SAMPLE_PATH: String
- DATASET_FOLDER: String
- OUTPUT_FOLDER: String
- NNDR_RATIO: float
- main(String[]): void
- findImageMatch(String, String)
- findMatch(MatOfKeyPoint, Ma
- saveKeyPointsImage(String, Me

Problems | Javadoc | Declaration | Search | Console

```
<terminated> SURFdetector [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (1.6.2017.22:31:22)
4 / 148: C:\Users\Kristina\workspace\dataset\Luba\Luba2.jpg
8 / 197: C:\Users\Kristina\workspace\dataset\Luba\Luba3.jpg
5 / 170: C:\Users\Kristina\workspace\dataset\Luba\Luba4.jpg
10 / 138: C:\Users\Kristina\workspace\dataset\Luba\Luba6.jpg
6 / 212: C:\Users\Kristina\workspace\dataset\Luba\Luba7.jpg
6 / 233: C:\Users\Kristina\workspace\dataset\Luba\Luba8.jpg
16 / 123: C:\Users\Kristina\workspace\dataset\Luba\Luba9.jpg
Best match with rate: 20 is in folder: Kuba and it is an image: C:\Users\Kristina\workspace\dataset\Kuba\Kuba3.jpg
```

Obrázek 3.6 Ukázka výsledku rozpoznávání

U této metody využívám již vytvořené metody detekce obličeje a toho, že je schopna oříznout detekované tváře. A pracuji pouze s těmito výřezy – obsahují méně příznaků než celé původní fotografie, ale tyto příznaky jsou pro rozpoznání významnější. Pokud bychom ovšem chtěli detekci vynechat, lze jednoduše tento příkaz v kódu zakomentovat. V tomto případě ovšem nemusí dojít ke správnému rozpoznání, jelikož by fotografie obsahovala spíše příznaky pozadí než tváře.

Základem algoritmu je to, že najde klíčové body ve vstupním obrazu a vypočítá jejich deskriptory. To stejné udělá i pro všechny soubory v databázi a následně porovnává právě vypočítané deskriptory a rozhodne, jestli jsou si objekty podobné nebo ne.

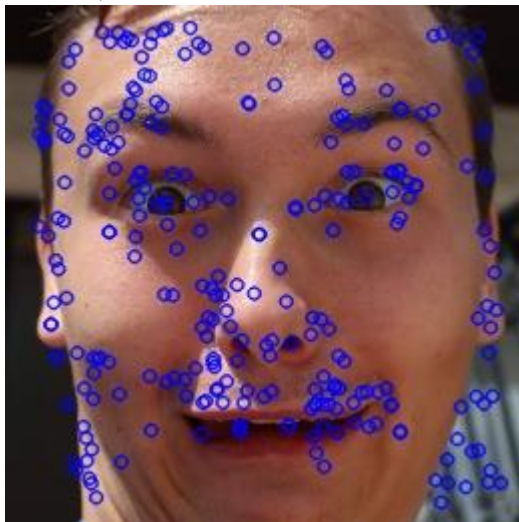
Jádro algoritmu tvoří funkce *findImageMatch*, která obsahuje dvě proměnné, a to cestu ke vstupnímu obrazu, který chceme rozpoznat, a cestu k databázi. Nejdříve načteme vstupní obraz a najdeme v něm klíčové body a následně vypočítáme deskriptory. Obraz s vyznačenými klíčovými body si uložíme do výstupní složky. Poté načteme všechny soubory u databáze, u kterých provedeme stejný postup jako u vstupního obrazu (nalezení klíčových bodů a výpočet deskriptorů). V tomto momentě začíná porovnávání. Knihovna OpenCV obsahuje několik komparátorů, já jsem si pro svůj program zvolila *FlannBasedMatcher*, protože pracoval nejrychleji.

Pro samotné porovnání se využívá metody nejbližšího souseda (*knnMatch*), což je klasifikační algoritmus zabudovaný v OpenCV knihovně. Proces této metody klasifikace zjednodušeně probíhá tak, že vybere soubor z trénovací množiny, který je nejbližší a neznámý prvek zařadí právě do této množiny.

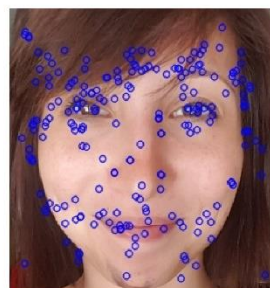
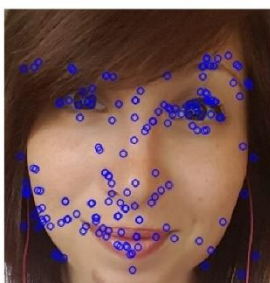
Po nalezení shody musíme odstranit neplatné výsledky. Jinými slovy vyfiltrovat pouze dobré shody a k tomu budeme používat parametr NNDR (nearest neighbor distance ratio), které se vypočítá jako:

$$NNDR = \frac{d_1}{d_2} = \frac{\|D_A - D_B\|}{\|D_A - D_C\|}, \quad 3.1$$

kde d_1 a d_2 jsou vzdálenosti nejbližšího a druhého nejbližšího souseda a D_A , D_B a D_C jsou cílový deskriptor a jeho dva nejbližší sousedé. NNDR neporovnává vzdálenost bodu a nejbližšího souseda pomocí pevně dané prahové hodnoty, ale jako poměr, který závisí na vzdálenosti nejbližšího souseda od druhého nejbližšího.



Obrázek 3.7 Ukázka uložených klíčových bodů do obrázku



Obrázek 3.8 Ukázka rozpoznání tváře

4 MOBILNÍ APLIKACE PRO AUTOMATICKOU DETEKCI A ROZPOZNÁVÁNÍ TVÁŘE

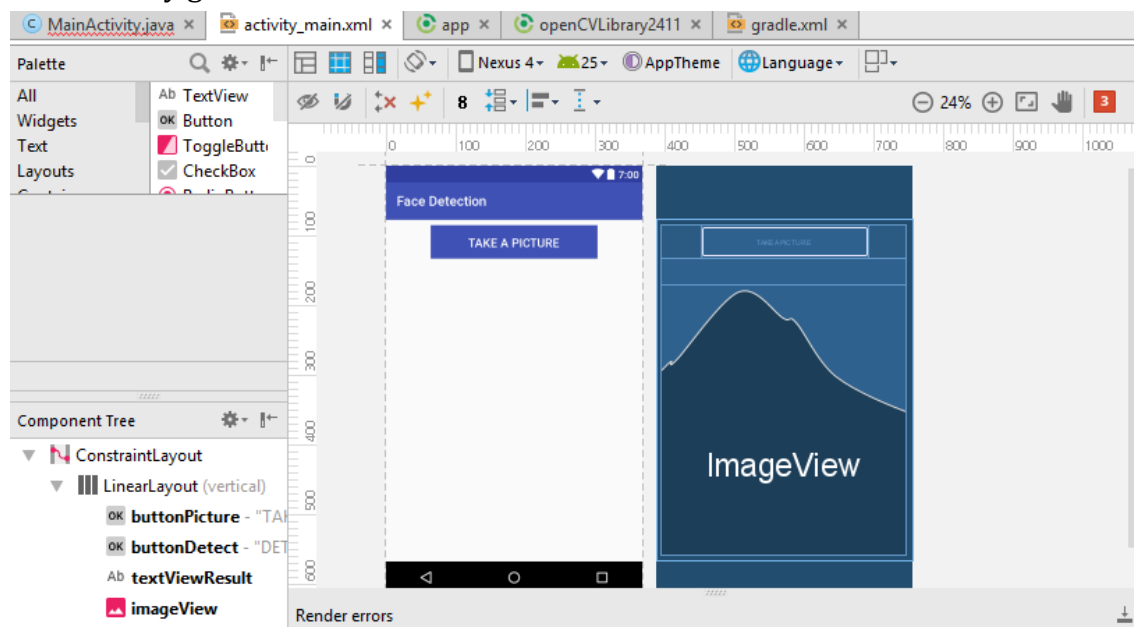
Vývoj aplikace byl prováděn v prostředí Android Studio (volně dostupné z internetu). Kromě rozhraní pro vývoj aplikace je také nutné nainstalovat Android balíček „open source“ knihovny OpenCV (verze 2.4.11, také dostupné z internetu).

Základními stavebními kameny Android aplikace jsou komponenty aktivit, které reprezentují obrazovku aplikace, service umožňující běh akcí na pozadí, content providers poskytující přístup k datům a broadcast receiver reagující na příchozí oznámení.

Aplikace se obecně skládá z několika základních částí:

1. Manifest aplikace.
2. Java třídy.
3. Zdroje, např. grafika.
4. Přidané soubory.

Grafické rozhraní aplikace je definováno v XML souboru. Rozhraní mé aplikace je vidět na Obrázek 4.1. Všechny objekty v grafickém rozhraní mají své specifické ID. Tyto identifikátory jsou v Androidu volány třídou R, která je automaticky generována.



Obrázek 4.1 Ukázka grafické tvorby aplikace

V aplikaci byl využit již vytvořený kód pro detekci a rozpoznání tváří. Ovšem v balíčku Android nejsou dostupné některé funkce knihovny OpenCV, a tak bylo potřeba přepsat určité části kódu (není vůbec dostupný detektor a extraktor *SURF*, byly vyzkoušeny jiné dostupné, a nakonec jako detektor zvolen *FAST* a jako extraktor *BRIEF*).

Další změnou v kódu je, že vstupní obrázek není načítán ze souboru, ale je vyfocen předním fotoaparátem telefonu, uložen a následně využit pro detekci a extrakci klíčových bodů v obrázku. Databáze, se kterou je vstupní obrázek porovnáván je již uložena v aplikaci a není proto nutná její instalace do telefonu.

Aplikace tedy pracuje tak, že po stisknutí tlačítka „TAKE A PICTURE“ se zobrazí přední kamera telefonu, uživatel vyfotí selfie, které potvrdí (nebo může vyfotit znovu). Poté se vrátí na původní obrazovku, kde se vyfocená fotka zobrazí a objeví se nové tlačítko „DETECTION AND RECOGNITION“. Po jeho stisknutí započne samotná detekce a rozpoznávání. Výsledek porovnání se po proběhnutí zobrazí na displeji.







Všechny aktivity spojené s ovládáním tlačítek jsou uloženy v Java třídě nazvané jako *MainActivity*, ze které se postupně volají další dvě, již vytvořené třídy, *SURFdetector* a *FaceDetection*.







Průběh a chování aplikaci si můžeme ověřit buď na zařízení s OS Android nebo na virtuálním zařízení. Tzv. emulátor se dá jednoduše vytvořit při prvním spuštění aplikace, telefon nebo tablet zase stačí pouze připojit přes kabel.







5 VYHODNOCENÍ







5.1 Vyhodnocení metody pro rozpoznání tváře

Tabulka 5.1 Vyhodnocení rozpoznávání tváří

pořadí	1.	2.	3.	4.	5.	6.
fotka						
rozpoznáno	ne	ano	ano	ano	ano	ne

pořadí	7.	8.	9.	10.	11.	12.
fotka						
rozpoznáno	ne	ano	ne	ano	ano	ne

pořadí	13.	14.	15.	16.	17.	18.
fotka						
rozpoznáno	ano	ne	ne	ano	ano	ano

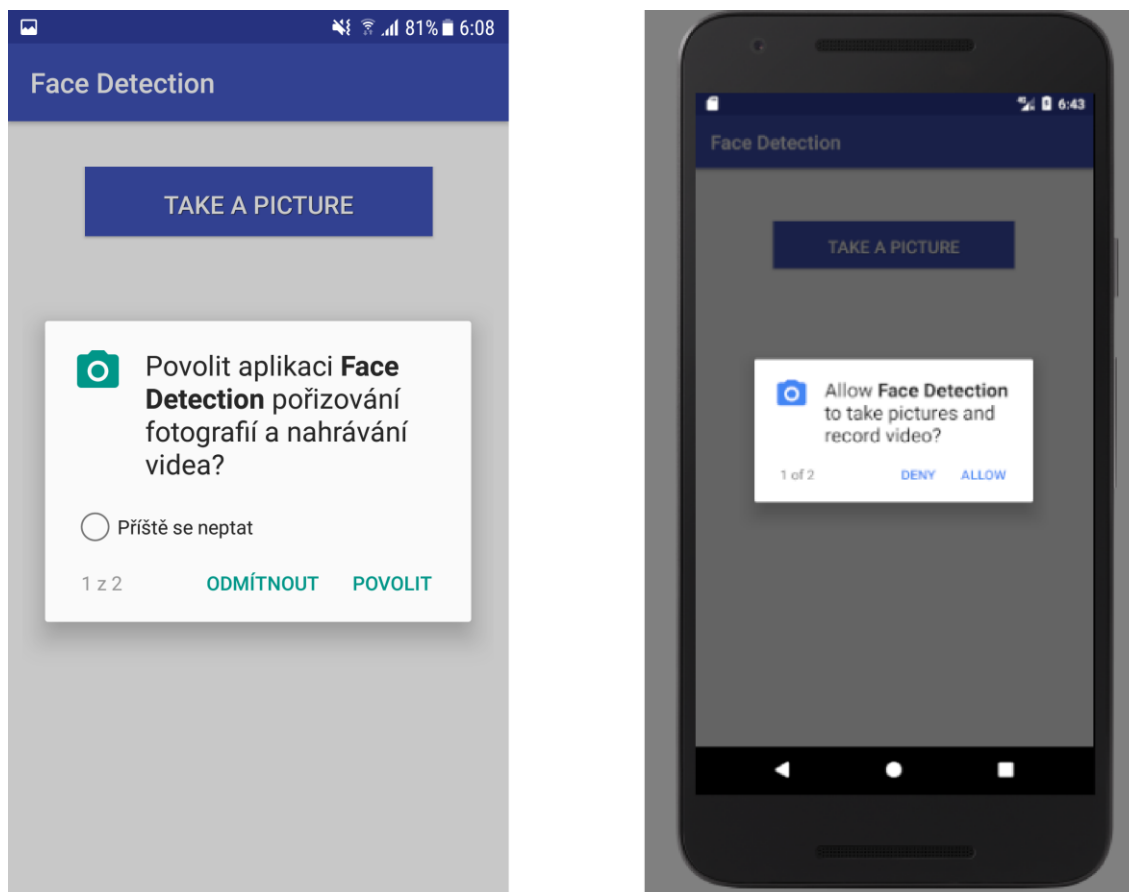
pořadí	19.	20.	21.	22.	23.	24.
fotka						
rozpoznáno	ano	ne	ano	ano	ano	ne

V Tabulka 5.1 je vidět vyhodnocení rozpoznávání tváří pro testovací sadu skládající se z 24 obrázků, obsahující 4 obrázky pro každou z 6 osob. U každého obrázku je napsáno, jestli byla tvář přiřazena správné osobě, či nikoliv. Z 24 tváří bylo správně rozpoznáno 16, což je 66,67 %. I přesto že vytvořená databáze pro ověření úspěšnosti metody obsahuje poměrně malé množství obrázků oproti běžně dostupným testovacím databázím obrázků, dá se implementovaná metoda považovat za úspěšnou.

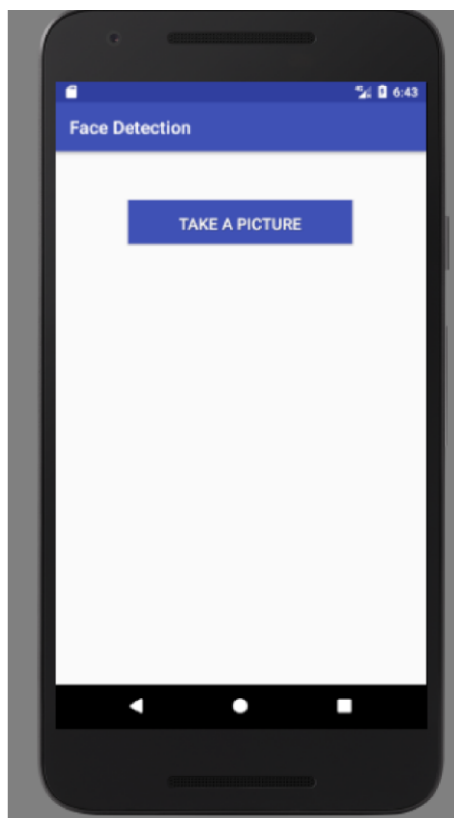
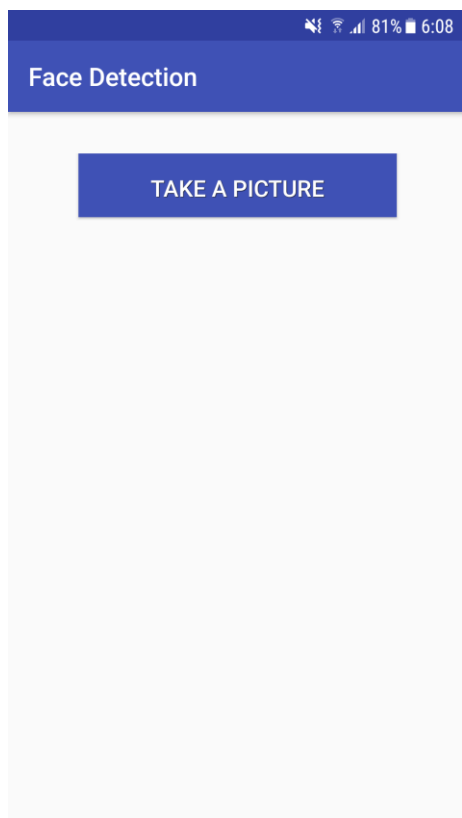
5.2 Vyhodnocení aplikace

Výsledný vzhled aplikace na obrazovce a porovnání průběhů aplikace na telefonu (vždy vlevo) a emulátoru (vždy vpravo) jsou vidět na obrázcích. Nevýhodou aplikace je její časová náročnost, jelikož jsou nejdříve detekovány a extrahovány klíčové body ve vstupním vyfoceném obrázku, a poté jsou klíčové body také lokalizovány ve všech obrázcích databáze.

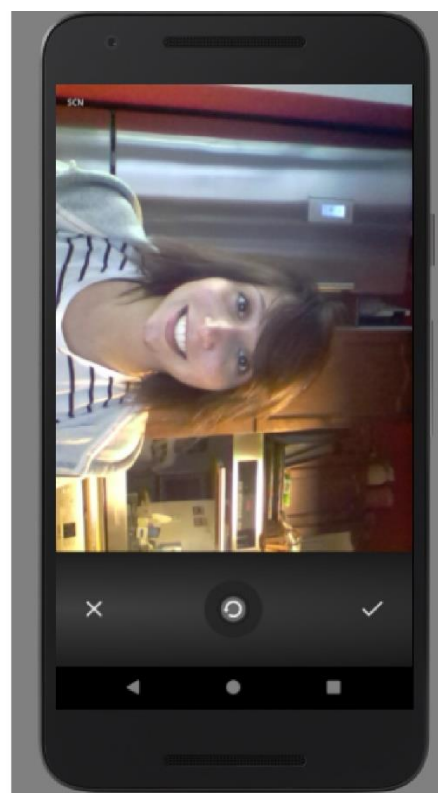
Aplikace bohužel nemá tak dobré výsledky jako samotná metoda SURF implementována v jazyce Java. Z 5 testovaných subjektů správně identifikovala pouze 2, tedy 40 %. Může to být způsobeno právě tím, že při přepisování kódu do Android aplikace, bylo nutné změnit určité parametry kódu, jelikož v balíčku Android nejsou určité funkce z OpenCV knihovny dostupné. Jedná se konkrétně o detektor a extraktor SURF, který byl nahrazen jinými detektory klíčových bodů. Dalším problémem může být, že telefony mají jiné rozměry fotoaparátu než webkamera u počítače (simulující fotoaparát u telefonu) a u některých telefonů docházelo ke špatnému rotování obrázku (viz. Obrázek 5.4).



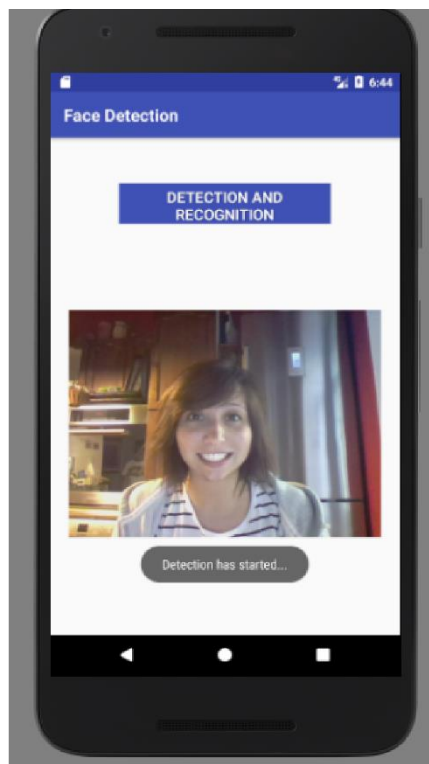
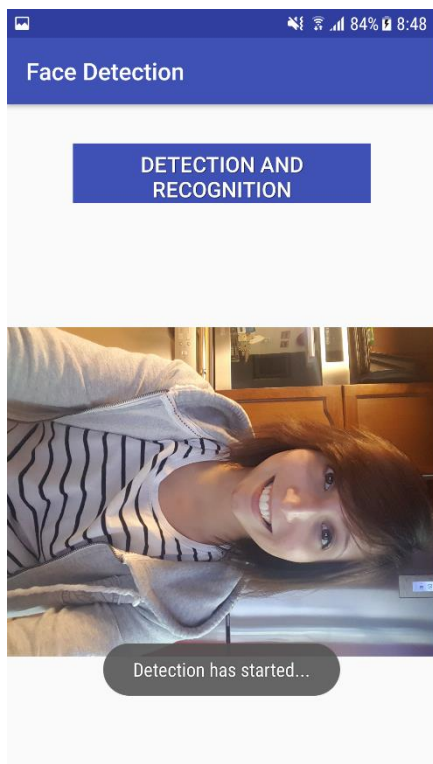
Obrázek 5.1 Povolení aplikace ke spuštění fotoaparátu



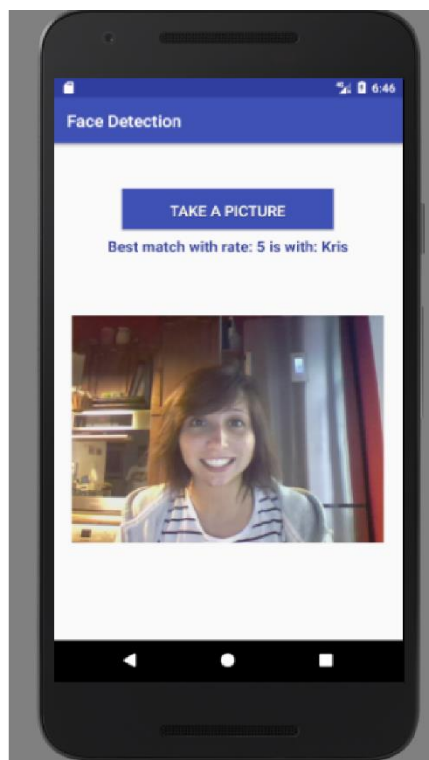
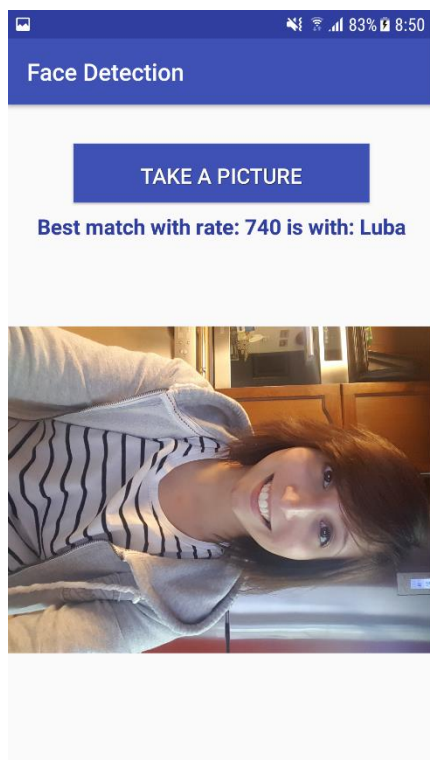
Obrázek 5.2 Úvodní obrazovka aplikace



Obrázek 5.3 Náhled fotoaparátu



Obrázek 5.4 Zpráva o začátku detekce po stisknutí tlačítka



Obrázek 5.5 Výsledek rozpoznání

6 ZÁVĚR

Cílem této bakalářské práce bylo implementovat metodu detekce tváře v obraze a klasifikátor pro rozpoznávání tváře, následně vytvořit mobilní aplikaci, která bude načítat fotky z fotoaparátu a nalezené tváře přiřazovat jednotlivým osobám.

Pro tuto práci byla pro detekci vybrána metoda Viola-Jones a pro rozpoznávání metoda extrakce příznaků SURF. Algoritmus byl úspěšně realizovaný pro obě dvě metody a úspěšnost metody SURF se dá také považovat za dostačující. Úspěšnost navržené aplikace je ovšem nižší, což může být způsobeno úpravami kódu pro Android aplikaci. Časová náročnost aplikace je také omezujícím faktorem.

Do budoucna by se aplikace dala vylepšit tak, aby v databázi byly uloženy fotky s již vyznačenými klíčovými body. Tak by se dala snížit časová náročnost, aby aplikace pokaždé nemusela tyto body ve všech obrázcích databáze hledat. Bylo by také vhodné rozšířit databáze snímků, např. přidáváním fotek do databáze přímo skrz focení fotoaparátem.

Rozšířením by také mohlo být rozpoznání více osob na jednom snímku, nebo možnost aplikace fotit zadním foťákem. Jako další by mohla být zavedena *real time* detekce a rozpoznávání tváře.

SEZNAM POUŽITÉ LITERATURY

[1] YANG, Ming-Hsuan, D.J. KRIEGMAN a N. AHUJA. Detecting faces in images: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence [online]. 2002, vol. 24, issue 1, s. 34-58 [cit. 2016-12-02]. Dostupné z: <http://faculty.ucmerced.edu/mhyang/papers/pami02a.pdf>.

[2] RAHMAN, Nusirwan Anwar bin Abdul, K. Ch. WEI a J. SEE. RGBH- CbCr Skin Colour Model for Human Face Detection. Faculty of Information Technology, Multimedia University [online]. [cit. 2016-12-02]. Dostupné z: http://pesona.mmu.edu.my/~johnsee/research/papers/files/rgbhcbcr_m2usic06.pdf.

[3] HJELMÅS, Erik a LOW KEE, Boon. Face Detection: A Survey. Computer Vision and Image Understanding [online]. [cit. 2016-12-01]. Dostupné z: <http://debut.cis.nctu.edu.tw/~ching/Face/Articles/Biometric%20Identification/Face%20Detection%20Survey.pdf>.

[4] HUKA, Petr. Nová segmentační metoda, principem založená na Gaussově rozdělovací funkci a její aplikace na segmentaci lidské kůže. Elektrorevue [online]. [cit. 2016-12-09]. Dostupné z: <http://www.elektrorevue.cz/clanky/05054/index.html>.

[5] CHAUDHARI, Monali, Shanta SONDUR a Gauresh VANJARE. A review on Face Detection and study of Viola Jones method. International Journal of Computer Trends and Technology [online]. 2015, 25(1), 54-61 [cit. 2017-03-14]. DOI: 10.14445/22312803/IJCTT-V25P110. ISSN 22312803. Dostupné z: <http://www.ijcttjournal.org/archives/ijctt-v25p110>.

[6] ZHU, Xiaobo, Dong REN, Zhenyu JING, Lin YAN a Shuanghui LEI. Comparative research of the common face detection methods. Proceedings of 2012 2nd International Conference on Computer Science and Network Technology [online]. IEEE, 2012, s. 1528-1533 [cit. 2017-04-06]. DOI: 10.1109/ICCSNT.2012.6526210. ISBN 978-1-4673-2964-4. Dostupné z: <http://ieeexplore.ieee.org/document/6526210/>.

[7] G. Yang and T. S. Huang, „Human Face Detection in Complex Background”, Pattern Recognition, vol. 27, no. 1, pp. 53-63, 1994.

- [8] Sharifara, Ali, et al. "A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection." *Biometrics and Security Technologies (ISBAST), 2014 International Symposium on*. IEEE, 2014.
- [9] Paul Viola, Micheal Jones, "Rapid object detection using a Boosted Cascade of Simple features" *CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION*, 2001.
- [10] Choudhury, Tanzeem. "History of Face Recognition." (2000). [cit. 2017-04-06]. Dostupné z: <http://vismod.media.mit.edu/tech-reports/TR-516/node7.html>.
- [11] Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1), 71-86.
- [12] Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7), 711-720.
- [13] He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H. J. (2005). Face recognition using laplacianfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3), 328-340.
- [14] Du, Geng, Fei Su, and Anni Cai. "Face recognition using SURF features." *Sixth International Symposium on Multispectral Image Processing and Pattern Recognition*. International Society for Optics and Photonics, 2009.
- [15] D. G. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, Corfu, Greece, 1150–1157 (1999).
- [16] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "Speeded-up robust features(SURF)," *Comput. Vis. Image Underst.*, 110(3), 346-359 (2008).
- [17] LI, Chunjing, Jinwu LIU, Anning WANG a Kai LI. Matrix Reduction Based on Generalized PCA Method in Face Recognition. In: *2014 5th International Conference on Digital Home* [online]. IEEE, 2014, s. 35-38. DOI: 10.1109/ICDH.2014.14.

[18] Craw, H. Ellis, and J. Lishman, Automatic extraction of face features. *Pattern Recognition Letters*, 1987, 5 (2): 183-187.

[19] Lienhart, R.; Maydt, J. An Extended Set of Haar-like Features for Rapid Object Detection. [online], IEEE ICIP, 2002.

[20] Introduction to SURF (Speeded-Up Robust Features). *OpenCV* [online]. [cit. 2017-06-02]. Dostupné z: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html.

Seznam příloh

Příloha 1. CD

- bakalářská práce ve formátu .pdf
- zdrojové kódy Java
- zdrojové kódy pro aplikaci
- aplikace ve formátu .apk