



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# SYSTEM PRO DETEKOVÁNÍ A IDENTIFIKACI OSOB NA ZÁKLADĚ DIGITÁLNÍHO OBRAZU OBLIČEJE

## Diplomová práce

*Studijní program:* N2612 – Elektrotechnika a informatika  
*Studijní obor:* 1802T007 – Informační technologie  
*Autor práce:* **Bc. Martin Bumba**  
*Vedoucí práce:* doc. Ing. Josef Chaloupka, Ph.D.



## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Martin Bumba**  
Osobní číslo: **M13000175**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Informační technologie**  
Název tématu: **Systém pro detekování a identifikaci osob na základě digitálního obrazu obličeje**  
Zadávací katedra: **Ústav informačních technologií a elektroniky**

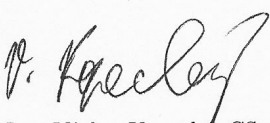
### Z á s a d y p r o v y p r a c o v á n í :

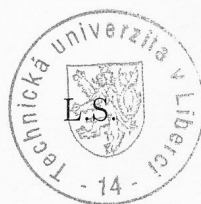
1. Seznamte se s problematikou rozpoznávání obrazu a s knihovnou OpenCV.
2. Navrhněte, případně modifikujte algoritmy vhodné pro detekci a identifikaci osob z obrazu obličeje.
3. Otestujte použité algoritmy na některé z volně dostupných databází obrazů s obličejí.
4. Vytvořte webový systém, ve kterém si bude moci uživatel rozpoznat vlastní obrazy s obličejí osob.


Rozsah grafických prací: Dle potřeby dokumentace  
Rozsah pracovní zprávy: cca 40 až 50 stran  
Forma zpracování diplomové práce: tištěná/elektronická  
Seznam odborné literatury:

- [1] Davies, E., R.: Machine Vision - Theory, Algorithms, Practicalities. Morgan Kaufmann Press. UK, 2005, ISBN 0-12-206093-8
- [2] Hlaváč V., Sedláček M.: Zpracování signálu a obrazu, Skripta FEL ČVUT, Praha 2000, ISBN 80-01-02114-9
- [3] Šonka, M., Hlaváč, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. PWS Publishing, USA, 1999, ISBN 0-534-95393-X

Vedoucí diplomové práce: doc. Ing. Josef Chaloupka, Ph.D.  
Ústav informačních technologií a elektroniky  
Konzultant diplomové práce: Ing. Karel Paleček  
Ústav informačních technologií a elektroniky  
Datum zadání diplomové práce: 14. září 2015  
Termín odevzdání diplomové práce: 16. května 2016

  
prof. Ing. Václav Kopecký, CSc.  
děkan



  
prof. Ing. Zdeněk Plíva, Ph.D.  
vedoucí ústavu

V Liberci dne 14. září 2015

## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.


Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 16. 5. 2016

Podpis:



## **Poděkování**

Rád bych poděkoval vedoucímu doc. Ing. Josefu Chaloupkovi, Ph.D. za užitečné konzultace a věcné připomínky k mé diplomové práci. Dále bych chtěl poděkovat konzultantovi Ing. Karlu Palečkovi za jeho cenné rady ohledně metod rozpoznání obrazu. Oběma také děkuji za jejich vstřícnost a trpělivost. V neposlední řadě bych chtěl vřele poděkovat své rodině, přítelkyni a přátelům, kteří mě při vypracování této práce podporovali.

## Abstrakt

Diplomová práce se zabývá problematikou rozpoznání obličejů v obraze, a to zejména detekcí, klasifikací (pohlaví, věku a osob) a verifikací obličejů. Pro detekci byly otestovány dvě metody: detekce na základě Haar příznaků s využitím boostovací metody AdaBoost a detekce využívající Histogramů orientovaných gradientů (HOG). V úloze klasifikace byla nejdříve využita metoda pro redukci dimenze dat – PCA (Principal Component Analysis) a následně byly otestovány metody klasifikace pomocí Euklidovské vzdálenosti, Mahalanobisovy vzdálenosti, GMM (Gaussian Mixture Models) a lineárního SVM (Support Vector Machine). Pro úlohu verifikace bylo otestováno řešení využívající deskriptorů LARK (Locally Adaptive Regression Kernel) v kombinaci s PCA a lineárním SVM. Další částí této diplomové práce bylo vytvoření moderního webového rozhraní, které implementuje nejúspěšnější z otestovaných metod detekce, klasifikace a verifikace a umožňuje tak uživateli rozpoznávat vlastní obrazy s obličejí osob.

**Klíčová slova:** rozpoznávání obličejů, detekce, klasifikace, verifikace, webový systém

## **Abstract**

The diploma thesis deals with the issues of face recognition in image, especially detection, classification (gender, age and people) and verification of faces. For detection were tested two methods: detection based on Haar features using the boost algorithm AdaBoost and detection using Histograms of Oriented Gradients (HOG). In classification task was used method for reduction dimension of data - PCA (Principal Component Analysis) and subsequently tested these methods of classification: Euclidean distance, Mahalanobis distance, GMM (Gaussian Mixture Models) and linear SVM (Support Vector Machine). For verification task has been tested solution which uses LARK (Locally Adaptive Regression Kernel) descriptors in combination with PCA and linear SVM. Next part of the thesis was to create a modern Web interface that implements the most successful tested methods for detection, classification and verification. This system enables users to recognize their own images with faces of people.

**Keywords:** face recognition, detection, classification, verification, web system

## Obsah

Seznam použitých zkratk.....	11
Úvod.....	12
1 Rozpoznávání obličejů v obraze.....	13
1.1 Práce s obrazem.....	13
1.2 Detekce obličejů v obraze.....	14
1.2.1 Detekce na základě Haar příznaků s využitím metody AdaBoost.....	14
1.2.2 Detekce využívající Histogramů orientovaných gradientů (HOG).....	15
1.3 Klasifikace obličejů v obraze.....	16
1.3.1 Princip redukce dimenze příznakového vektoru metodou PCA.....	17
1.3.2 Klasifikace pomocí Euklidovské vzdálenosti.....	19
1.3.3 Klasifikace pomocí Mahalanobisovy vzdálenosti.....	19
1.3.4 Klasifikace pomocí Gaussian Mixture Models (GMM).....	19
1.3.5 Klasifikace pomocí lineárního Support Vector Machine (SVM).....	21
1.4 Verifikace obličejů osob.....	22
1.4.1 Využití deskriptorů LARK.....	23
1.4.2 Verifikace pomocí lineárního Support Vector Machine (SVM).....	24
2 Návrh systému pro rozpoznání obličejů v obraze.....	25
2.1 Úložiště dat a informací.....	25
2.2 Webový server.....	26
2.3 Metody rozpoznání obličejů v obraze.....	27
3 Testování metod rozpoznání obličejů.....	28
3.1 Databáze obrazů obličejů.....	28
3.1.1 Face Detection Data Set and Benchmark (FDDB).....	28
3.1.2 Unfiltered faces for gender and age classification (UFGA).....	28
3.1.3 Labeled Faces in the Wild (LFW).....	28
3.2 Testovací metriky.....	29
3.2.1 Úspěšnost.....	29
3.2.2 Přesnost (precision), úplnost (recall) a F1-skóre.....	29
3.3 Testování metod detekce obličejů v obraze.....	30
3.4 Předzpracování obrazu.....	31
3.4.1 Hromadné předzpracování obrazů.....	33
3.5 Klasifikace pohlaví osoby z obrazu obličeje.....	34
3.5.1 Normalizovaná norma obrazů (NN).....	34



3.5.2	Frontalizovaná norma obrazů (FN).....	36
3.5.3	Srovnání testovaných metod.....	38
3.6	Klasifikace věku osoby z obrazu obličeje.....	39
3.6.1	Euklidovská vzdálenost (ED).....	39
3.6.2	Mahalanobisova vzdálenost (MD).....	39
3.6.3	Gaussian Mixture Models (GMM).....	39
3.6.4	Lineární Support Vector Machine (SVM).....	40
3.7	Klasifikace (identifikace) konkrétní osoby z obrazu obličeje.....	41
3.7.1	Lineární Support Vector Machine (SVM).....	42
3.8	Verifikace osob na základě dvou obrazů obličeje.....	43
3.8.1	Lineární Support Vector Machine (SVM).....	43
4	Implementace webového systému.....	45
4.1	Návrh databáze.....	45
4.1.1	Kolekce people.....	46
4.1.2	Kolekce peopleImages.....	46
4.1.3	Kolekce users.....	46
4.2	Aplikace webového serveru – backend.....	47
4.2.1	Popis funkcionality Node.js aplikace.....	48
4.2.2	Spuštění aplikace webového serveru.....	51
4.3	Metody rozpoznání obrazu – RPC server.....	52
4.3.1	Třída CORE.....	52
4.3.2	Třída API.....	52
4.3.3	Vytvoření RPC serveru.....	53
4.3.4	Spuštění skriptu RPC serveru.....	54
4.3.5	Vytváření systémového datasetu.....	54
4.4	HTML 5 webová aplikace – frontend.....	54
4.4.1	Hlavní spouštěcí soubor index.html.....	55
4.4.2	Soubor aplikace app.js.....	55
4.4.3	Využití elementu canvas.....	57
4.4.4	Možnosti webové aplikace.....	57
5	Závěr.....	61
	Obsah příloženého DVD.....	64
	Přílohy.....	65

## Seznam obrázků

Obr. 1: Jednotlivé druhy Haarových příznaků.....	14
Obr. 2: Ukázka obrazu obličeje vyjádřeného deskriptory HOG.....	16
Obr. 3: Proces detekce využívající příznaků HOG.....	16
Obr. 4: Ukázka projekce dat 3 tříd, původně obsahující 4 příznaky za pomoci dvou PCA příznaků (převzato z [3]).....	18
Obr. 5: Jednorozměrná Gaussova funkce rozdělení pravděpodobností (převzato z [9]).	20
Obr. 6: Ukázka rozdělujících nadrovin lineárním SVM pro 3 rozdílné třídy.....	22
Obr. 7: Ukázka obrazu obličeje vyjádřeného příznakem LARK.....	23
Obr. 8: Architektura systému.....	25
Obr. 9: Rozdíl mezi tradičním životním cyklem webové stránky a architekturou SPA..	27
Obr. 10: Pozice 68 landmarků popisujících tvar obličeje (převzato z [16]).....	31
Obr. 11: Postup předzpracování obrazu do dvou obrazových norem.....	32
Obr. 12: Ukázka využití datových modelů v systémové databázi.....	47
Obr. 13: Základní rozhraní HTML5 aplikace s rozpoznáním obrazem obličeje.....	57
Obr. 14: Ukázka rozpoznání obličejů HTML5 aplikací.....	58
Obr. 15: Ukázka obrazovky detailu osoby.....	59
Obr. 16: Ukázka obrazovky HTML5 aplikace pro vyhledávání obrazů osoby na internetu.....	60

## Seznam grafů

Graf 1: Úspěšnost klasifikace pohlaví pomocí Euklidovské vzdálenosti, Mahalanobisovy vzdálenosti a lineárního SVM v závislosti na počtu prvních $n$ PCA komponent (normalizovaná norma).....	35
Graf 2: Úspěšnost klasifikace pohlaví pomocí GMM v závislosti na počtu Gaussových rozdělení (mixtur) (normalizovaná norma).....	36
Graf 3: Úspěšnost klasifikace pohlaví pomocí Euklidovské vzdálenosti, Mahalanobisovy vzdálenosti a lineárního SVM v závislosti na počtu prvních $n$ PCA komponent (frontalizovaná norma).....	37
Graf 4: Úspěšnost klasifikace pohlaví pomocí GMM v závislosti na počtu Gaussových rozdělení (mixtur) (frontalizovaná norma).....	38
Graf 5: Úspěšnost klasifikace věku pomocí Euklidovské vzdálenosti, Mahalanobisovy vzdálenosti a lineárního SVM v závislosti na počtu prvních $n$ PCA komponent (frontalizovaná norma).....	40

Graf 6: Úspěšnost klasifikace věku pomocí GMM v závislosti na počtu Gaussovských rozdělení (mixtur) (frontalizovaná norma).....	41
Graf 7: Úspěšnost klasifikace (identifikace) konkrétních osob pomocí lineárního SVM v závislosti na počtu prvních n PCA komponent (frontalizovaná norma).....	42
Graf 8: Úspěšnost verifikace osob pomocí lineárního SVM v závislosti na počtu prvních n PCA komponent (frontalizovaná norma).....	44

## **Seznam tabulek**

Tabulka 1: Definice pojmů binární klasifikace.....	29
Tabulka 2: Porovnání testovaných metrik algoritmů detekce.....	31
Tabulka 3: Srovnání testovaných metod klasifikace pohlaví.....	38
Tabulka 4: Srovnání testovaných metod klasifikace věku.....	41
Tabulka 5: Popis metod vytvořeného webového API.....	49

## Seznam použitých zkratk

PCA – Principal Component Analysis

HOG – Histogram orientovaných gradientů

GMM – Gaussian Mixture Models

SVM – Support Vector Machine

PIL – Python Imaging Library

LARK – Locally Adaptive Regression Kernel

NoSQL – z anglického „non SQL“

JSON – JavaScript Object Notation

HTML5 – HyperText Markup Language 5

CSS – Cascading Style Sheets

SPA – Single-Page Application

API – Application Programming Interface

HTTP – Hypertext Transfer Protocol

MIT – Massachusetts Institute of Technology (druh licence)

GNU AGPL – GNU Affero General Public License

BSD – Berkeley Software Distribution

REST – Representational State Transfer

RPC – Remote Procedure Call

## Úvod

Rozpoznání osob v obraze je v současné době často řešený problém v oblasti počítačového vidění, jelikož je stále více kladena poptávka právě na identifikaci osob, pohlaví, věku či emocí. Rozpoznání osob na základě obličeje je čím dál více využíváno jako bezpečnostní prvek například na letištích, nádražích či v kasinech. V supermarketech lze na základě správně rozpoznávaného obličeje zjistit, co nakupují muži nebo ženy, jaké věkové skupiny do supermarketu chodí nejčastěji nebo jak byli lidé s nákupem či s prodávací spokojeni (na základě klasifikace emocí). V počítačích je možné se pomocí obličeje přihlásit do operačního systému, a není tedy potřeba zadávat přihlašovací jméno a heslo.

Výhodou úlohy rozpoznání obličejů také je, že nám k němu stačí pouze snímací zařízení (nejčastěji obyčejná kamera), nepotřebujeme tak snímače biometrických údajů jako jsou čtečky otisků prstů nebo speciálně upravené kamery pro snímání oční sítnice. Kamera může být navíc chytře ukryta tak, že se účastník ani nedozví, že je právě rozpoznáván či detekován. Rozpoznání obličejů využívá také spousta moderních aplikací např. na sociální síti Facebook jsou v každé vložené fotografii detekovány obličeje, které mohou být později označeny, aby bylo zřejmé, kdo se na fotografii nachází. Chytré digitální fotoaparáty zase mívají možnost vyfotografovat snímek automaticky, když u fotografovaných osob rozpoznají úsměv.

Cílem diplomové práce je seznámit se s problematikou rozpoznávání obrazu a navrhnout případně modifikovat algoritmy vhodné pro detekci a identifikaci osob z obrazu obličeje. Tyto algoritmy budou v práci otestovány na databázích obrazů obličejů a následně bude vytvořen moderní webový systém, který algoritmy s nejvyšší dosaženou úspěšností využije pro rozpoznání obličejů osob na vlastních obrazech uživatele.

# 1 Rozpoznávání obličejů v obraze

Pojem rozpoznávání obličejů v obraze pod sebou skrývá několik různých samostatných úloh [1]. První úlohou je **detekce**, která má za úkol ve vstupním obraze správně detekovat oblast, případně oblasti, obsahující obličej. Další úlohou může být **klasifikace**, tedy správné zařazení detekovaného obličeje do předem určených skupin (tříd). Těmito třídami mohou v případě obličejů být: muž / žena, věkové intervaly, barva pleti, jednotlivé osoby apod. Poslední úlohou řešenou v této diplomové práci je **verifikace**. Jedná se o ověření, zdali dva detekované obličejové náleží stejné osobě či nikoliv.

## 1.1 Práce s obrazem

Pro práci s obrazem byl v této diplomové práci zvolen programovací jazyk Python ve verzi 2.7. Python je vysokoúrovňový skriptovací jazyk, který je vyvíjen jako open-source (s otevřeným kódem). Tento jazyk byl zvolen zejména kvůli jeho možné multiplatformnosti. Je totiž zdarma vydáván pro většinu běžných platforem (Unix, Windows, Mac OS). Další výhodou jazyka Python je možnost využití mnoha kvalitních knihoven pro práci s obrazem. První knihovnou, která byla pro práci s obrazem využita, je OpenCV. Je to velmi rozsáhlá multiplatformní knihovna zaměřená na počítačové vidění a zpracování obrazu v reálném čase. Tato knihovna je vydávána pod licencí BSD<sup>1</sup> (Berkeley Software Distribution). Další použitou knihovnou pro zpracování obrazu je knihovna Pillow, která je odnoží knihovny PIL (Python Imaging Library). Tato knihovna obsahuje mnoho metod a funkcí, zejména pro předzpracování obrazu (oříznutí, škálování apod.). Knihovna je vydávána zdarma pod licencí PIL Software License. Dále byla použita knihovna Dlib. Dlib je knihovna strojového učení, která je vydávána zdarma pod licencí Boost Software License [2]. Její výhodou je, že obsahuje mnoho naimplementovaných algoritmů – např. detektor objektů nebo prediktor landmarků (významných bodů). Poslední knihovnou pro práci s obrazem, která byla v rámci této práce využita, je knihovna scikit-learn. Tato je nejrozšířenější knihovnou

---

<sup>1</sup> BSD (Berkeley Software Distribution) je jednou z nejsvobodnějších softwarových licencí.

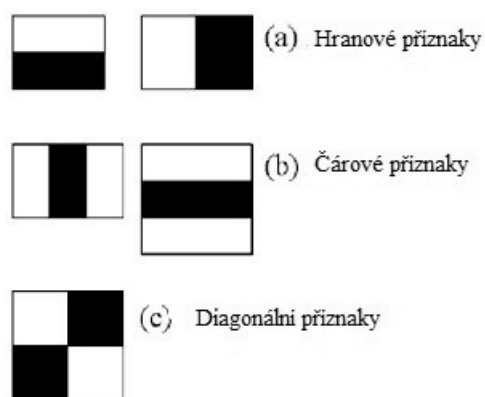
strojového učení jazyka Python [3]. Obsahuje metody pro redukci dimenze, klasifikaci, regresi, klastrování atp. Knihovna scikit-learn je vydávána pod licencí BSD.

## 1.2 Detekce obličejů v obraze

Základem úspěšného rozpoznání, klasifikace či verifikace osob je nutnost správně a přesně detekovat oblast v obraze, ve které se nachází obličej. K detekci se v současné době používá velká řada metod: detekce na základě Haar příznaků s využitím boostovacích metod, metoda barevné segmentace oblasti lidské kůže nebo algoritmy využívající histogramů orientovaných gradientů (HOG) [1]. Díky neustálému zvyšování výkonu počítačů jsou některé metody implementovány přímo ve standardních distribucích knihoven pro práci s obrazem. Výstupem detektoru jsou tedy obdélníkové oblasti (souřadnice) nalezených obličejů ve vstupním obraze.

### 1.2.1 Detekce na základě Haar příznaků s využitím metody AdaBoost

Tato metoda je součástí knihovny OpenCV a dále v této práci bude nazývána jako **HAAR detekce**. Základem je detekce objektů využívající kaskádu klasifikátorů založenou na *Haar* příznacích. Metoda byla poprvé představena již v roce 2001 P. Violou a M. Jonesem. Jedná se o metodu strojového učení, kde kaskádní klasifikační funkce je natrénována z mnoha pozitivních (obrazy obsahující hledaný objekt) a negativních obrazů (obrazy neobsahující hledaný objekt). Metoda pracuje s šedotónovým obrazem. Při trénování jsou z obrazů načteny *Haarovy* příznaky, které vycházejí z definice tzv. *Haarovy* vlnky. Hodnota takového příznaku se spočítá jako rozdíl sumy pixelů tmavé a světlé oblasti (viz obr. 1) [4].



Obr. 1: Jednotlivé druhy Haarových příznaků

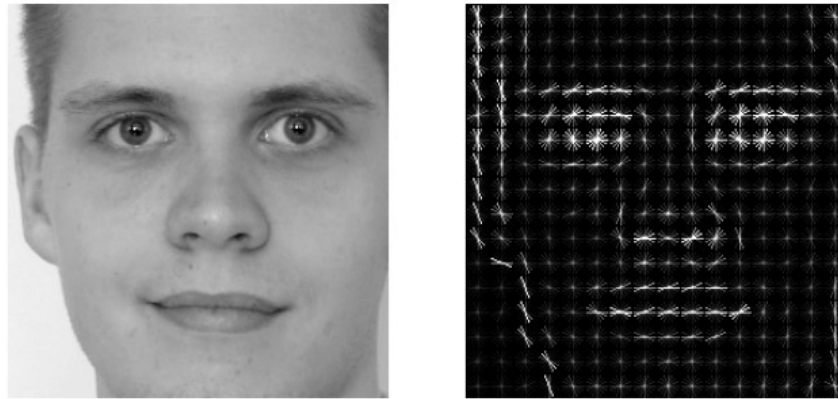
Jednotlivé příznaky jsou hledány v celém obraze, přičemž velikost příznaků je proměnná od velikosti  $1 \times 1$  do velikosti  $n \times n$ , kde  $n$  je výška (šířka) vstupního obrazu. Pro představu pro obraz o velikosti  $24 \times 24$  pixelů to znamená vypočítat více než 160 000 příznaků. Proto se jednotlivé sumy pixelů nevypočítávají z obrazů přímo, ale nejprve dojde k převedení obrazu na tzv. integrální obraz [4], kde pro vypočítání sumy pixelů libovolné oblasti stačí pracovat pouze se čtyřmi hodnotami. Proto je výpočet těchto příznaků jednoduchý a velmi rychlý. Takto vzniklé příznaky jsou poté vstupem klasifikačního algoritmu *AdaBoost*, který z nich vybere ty nejdůležitější a vytvoří z nich kaskádu klasifikátorů, která dokáže nejlépe nalézt trénovaný objekt (obličej). Takto natrénovaná kaskáda je poté vstupem detekčního algoritmu.

Při detekci je vstupní obraz zpracováván po částech. Tyto části jsou vybírány pomocí posuvného podokna, které mění svoji pozici a velikost v závislosti na předpokládané velikosti detekovaného objektu. Vstupem klasifikátoru jsou tedy příznaky daného podokna a klasifikátor rozhodne, zda toto podokno obsahuje nějaký obličej z pozitivních natrénovaných dat či nikoliv [5].

### 1.2.2 Detekce využívající Histogramů orientovaných gradientů (HOG)

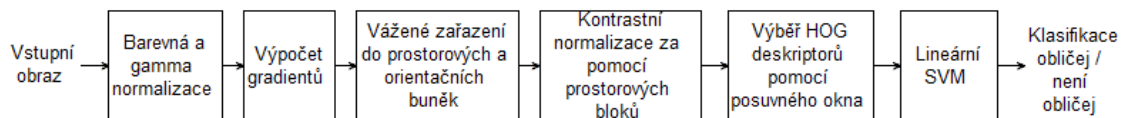
Tato metoda je přímo dostupná v knihovně Dlib [2] a dále bude nazývána jako **detekce HOG**. Metoda spočívá v rozdělení obrazu do malých buněk, kde pro každou buňku je z pixelů vypočítán jednorozměrný (1-D) histogram gradientních směrů [6]. Informace z obrazu je tedy v této metodě popsána pomocí intenzity gradientů. Pro snížení závislosti na osvětlení jsou jednotlivé buňky dále normalizovány za pomoci lokálně vypočtených histogramů pro oblasti větší, než-li je velikost jednotlivých buněk. Tato oblast se tedy skládá z několika buněk a nazývá se blok. Takto normalizované buňky nazýváme deskriptory HOG (Histogram orientovaných gradientů) [6]. Na obr. 2 je ukázka obličeje vyjádřeného pomocí deskriptorů HOG.





Obr. 2: Ukázka obrazu obličeje vyjádřeného deskriptory HOG

Po vypočtení HOG deskriptorů jsou metodou posuvného okénka postupně vybírány oblasti těchto deskriptorů, které jsou klasifikovány za pomoci lineárního Support Vector Machine (SVM) algoritmu. Lineární SVM je klasifikační metoda strojového učení určená k rozdělení zkoumaného souboru do dvou klasifikovaných tříd. V případě detektoru do tříd: obsahuje oblast hledaný objekt / oblast neobsahuje hledaný objekt. Konkrétně v obličejovém detektoru tedy: oblast obsahuje obličej člověka nebo oblast neobsahuje obličej člověka. Metodě lineární SVM je věnována kapitola 1.3.5. Celý proces detekce využívající příznaků HOG je znázorněn na následujícím obrázku.



Obr. 3: Proces detekce využívající příznaků HOG

### 1.3 Klasifikace obličejů v obraze

Úlohou klasifikace je správné zařazení detekovaného obličeje do předem určených klasifikačních tříd. V případě klasifikace obličeje mohou rozpoznávanými třídami například být: pohlaví (muž, žena), věkový interval (0-5, 5-15, 15-25, ..), barva pleti (světlá, tmavá), jednotlivé osoby apod. Klasifikace lidského obličeje na základě jeho detekované oblasti není pro počítač snadnou úlohou, protože není jednoduché z detekované oblasti získat ty správné příznaky, na základě kterých by klasifikátor správně rozlišil klasifikovanou třídu za všech podmínek. Navíc v reálném světě má člověk informací mnohem více a k zařazení osoby do správné třídy mu může pomoci např. mohutnost postavy, styl a barva oblečení, délka vlasů atp. V této práci byly

vstupními daty pro klasifikaci šedotónové obrazy obličejů ve velikosti  $64 \times 64$  pixelů. Za předpokladu, že barva každého pixelu obrazu je jeden příznak, bylo by takto získáno pro každý obraz vektor obsahující 4 096 příznaků. Při použití takového příznakového vektoru by mohla celá úloha klasifikace být velmi pomalá. Navíc samotné pixely obsahují velmi mnoho zbytečných informací např. pixely pozadí nám k rozpoznání správné třídy nijak nepomohou. Z těchto důvodů je dobré redukovat počet příznaků, aby rozpoznávání mohlo být výpočetně rychlejší. K redukci příznaků byla použita metoda analýzy hlavních komponent (dále jen PCA) [7]. Cílem je transformace původních příznaků na nové, nekorelované proměnné, zvané hlavní komponenty.  $N$ -tá hlavní komponenta je lineární kombinací takového rozměru původních dat, který vykazuje  $n$ -tý nejvyšší rozptyl hodnot a je vždy ortogonální na  $n-1$  předchozích komponent.

### 1.3.1 Princip redukce dimenze příznakového vektoru metodou PCA

Pro demonstraci výpočtu PCA komponent je zapotřebí dvou množin obrazů: trénovací množiny a testovací množiny.

Pro  $M$  vstupních trénovacích obrazů o velikosti  $x \times x$  je metodu možno popsat následovně. Z každého obrazu, který je načten jako šedotónový, je vytvořen řádkový vektor o velikosti  $x^2$  tak, že jednotlivé řádky vstupního obrazu jsou poskládány za sebe. Takto vytvořené vektory se poskládají pod sebe a vytvoří se tak matice  $W_p$  o velikosti  $M \times x^2$ . Ze sloupců matice je vytvořen průměrný vektor  $w_p$  s délkou ( $x^2$ ). Tento vektor je odečten od jednotlivých řádků matice  $W_p$  a vznikne tak matice  $W$ . Dále je vztahem 1.1 vytvořena kovarianční matice o velikosti  $M \times M$ .

$$C = W \times W^T \quad (1.1)$$

V kovarianční matici  $C$  jsou nalezena vlastní čísla a jim odpovídající vlastní vektory. Vlastní vektory se seřadí podle velikosti vlastních čísel a poskládají za sebe tak, že vznikne matice  $E_p$  o velikosti  $M \times M$ . Nyní je vztahem 1.2. vytvořen vlastní prostor  $E$ .

$$E = W^T \times E_p \quad (1.2)$$

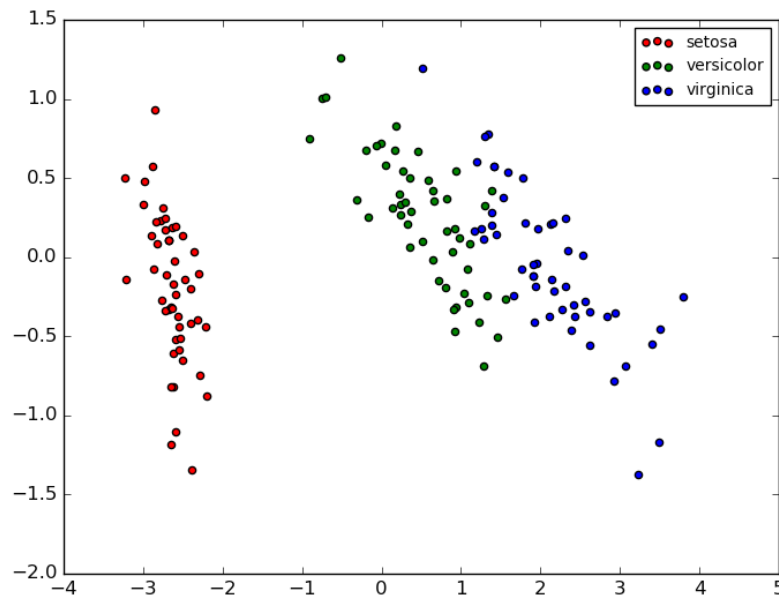
Řádky matice  $E$  jsou poté vyděleny jejich normou. Posledním krokem je projekce známých vektorů do vlastního prostoru  $PI$  vztahem 1.3, čímž jsou získány pro každý obraz trénovací sady PCA komponenty [7].

$$PI = W \times E \quad (1.3)$$

PCA komponenty pro testovací sadu obrazů jsou získány podobně jako pro množinu trénovací. Pro  $K$  vstupních testovacích obrazů o velikosti  $x \times x$  je metodu možno popsat následovně. Z každého obrazu, který je načten jako šedotónový, je vytvořen řádkový vektor o velikosti  $x^2$  tak, že jednotlivé řádky vstupního obrazu jsou poskládány za sebe. Takto vytvořené vektory se poskládají pod sebe a vytvoří tak matici  $T_p$  o velikosti  $K \times x^2$ . Od jednotlivých řádků matice  $T_p$  odečteme průměrný vektor  $w_p$  a vznikne tak matice  $T$ . Matici  $T$  promítneme do vlastního prostoru  $PT$  následujícím vztahem.

$$PT = T \times E \quad (1.4)$$

Tímto byly vypočteny PCA komponenty i pro testovací sadu [7]. PCA komponent pro každý vstupní obraz je právě tolik, kolik je obrazů v trénovací sadě. V případě, že máme v trénovací sadě 4 000 obrazů získali bychom pro každý obraz právě 4 000 PCA komponent. Využití všech PCA komponent je ale ve většině případů zbytečné a zpomalovalo by celý proces klasifikace. Je tedy vhodné využít pouze prvních  $N$  komponent. Vhodné číslo  $N$  lze získat za pomoci testování výsledné úspěšnosti s reálnými daty, obvykle to pro větší databáze může být 20 až 500. Číslo  $N$  tedy udává počet příznaků po redukci příznakového vektoru metodou PCA. Ukázka redukce dimenze metodou PCA je znázorněna následujícím obrázkem.



Obr. 4: Ukázka projekce dat 3 tříd, původně obsahující 4 příznaků za pomoci dvou PCA příznaků (převzato z [3])

### 1.3.2 Klasifikace pomocí Euklidovské vzdálenosti

Tato nejjednodušší klasifikační metoda spočívá v prostém porovnávání příznakových vektorů testovacích obrazů k příznakovým vektorům trénovacích obrazů a zjišťováním rozdílů mezi nimi. Třída testovacího vektoru je určena podle třídy trénovacího vektoru, ke kterému má testovací vektor nejmenší vzdálenost podle vzorce 1.5, kde  $s_r$  označuje vektor z trénovací množiny a  $s_i$  obsahuje vektor z testovací množiny. Odmocninu lze ze vztahu vynechat, jelikož neovlivní výsledek a operace bude rychlejší.

$$D_E(s_r, s_i) = \sqrt{\sum_{m=1}^N (s_r(m) - s_i(m))^2} \quad (1.5)$$

### 1.3.3 Klasifikace pomocí Mahalanobisovy vzdálenosti

Mahalanobisova vzdálenost byla představena P. C. Mahalanobisem v roce 1936. Oproti Euklidovské vzdálenosti zohledňuje vzájemné vztahy mezi příznaky. Metoda tedy uvažuje míru korelovanosti jednotlivých příznaků. Na druhé straně potlačuje vliv rozdílů ve variabilitě příznaků na výsledek. V případě, že příznaky jsou nekorelovány, pak Mahalanobisova vzdálenost odpovídá Euklidovské vzdálenosti. Vzorec pro

Mahalanobisovu vzdálenost lze při využití redukce dimenze příznakového vektoru metodou PCA a zjednodušením Moonovy definice [8] vyjádřit následovně:

$$D_{Ma}(s_r, s_i) = - \sum_{m=1}^N \left( \frac{1}{\sqrt{\lambda_m}} s_r(m) \cdot s_i(m) \right) \quad (1.6)$$

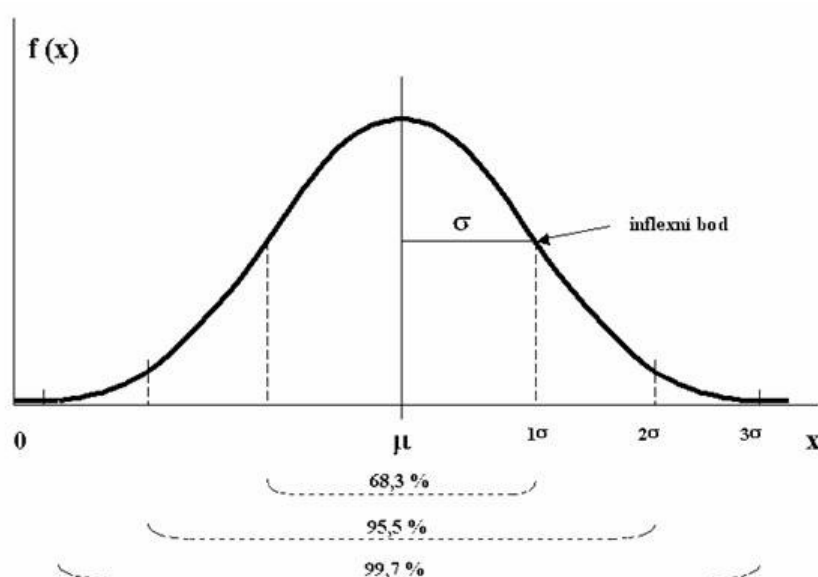
kde  $s_r$  označuje vektor z trénovací množiny,  $s_i$  obsahuje vektor z testovací množiny  $\lambda_m$  je  $m$ -té vypočítané vlastní číslo při realizaci algoritmu PCA. Třidu testovacího vektoru následně určíme obdobě jako při využití Euklidovské vzdálenosti.

### 1.3.4 Klasifikace pomocí Gaussian Mixture Models (GMM)

Metoda GMM je založena na statistickém rozpoznání tříd využívající pravděpodobnosti [1]. Využívá předpoklad, že vektory stejné klasifikační třídy mohou mít nějaké společné statistické vlastnosti. GMM je pravděpodobnostní model, který uvažuje, že všechny příznaky vznikly z konečného počtu Gaussových rozdělení (mixtur) s neznámými parametry. Jednorozměrná funkce Gaussova rozdělení pravděpodobností má vzorec:

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \exp\left(\frac{-(x-\mu)^2}{2 \cdot \sigma^2}\right) \quad (1.7)$$

kde  $\mu$  je střední hodnota a  $\sigma^2$  označuje rozptyl [1]. Graf průběhu této funkce je znázorněn na obr. 5.



Obr. 5: Jednorozměrná Gaussova funkce rozdělení pravděpodobností (převzato z [9])

Klasifikace GMM potom vzniká lineární kombinací více funkcí Gaussových rozdělení pravděpodobností (mixtur). Klasifikace GMM tak může být vypočtena podle následujícího vzorce:

$$f(x) = \sum_{i=1}^M \alpha_i \cdot \left( \frac{1}{(2 \cdot \pi)^d \cdot \det(\Sigma_i)} \right) \exp\left( \frac{-(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)}{2} \right) \quad (1.8)$$

kde  $\alpha$  je váhovací parametr,  $\Sigma$  je kovarianční matice a  $\mu$  je vektor středních hodnot. Tyto parametry se určují například za pomoci EM (Expectation Maximization) iteračního algoritmu. Ten pracuje ve dvou krocích:

- Expectation (odhad): zde se nejdříve provede odhad parametrů a výpočet tzv. likelihood (věrohodnostní) funkce [1].
- Maximization (maximalizace): v tomto kroku jsou parametry upravovány tak, aby došlo k maximalizaci likelihood (věrohodnostní) funkce [1].

Pro každou klasifikační třídu bude za pomoci dat z trénovací množiny vytvořen jeden GMM model. Při testování je následně zjištěna pravděpodobnost ke každému GMM modelu. Testovací vzorek je následně klasifikován podle třídy GMM modelu, u kterého byla výsledná pravděpodobnost nejvyšší.

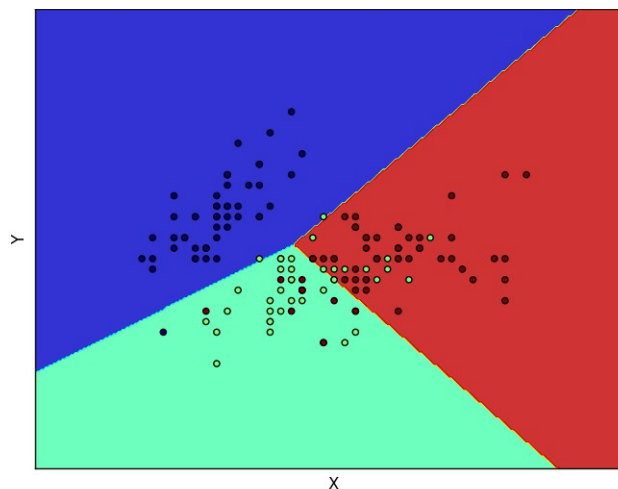
### 1.3.5 Klasifikace pomocí lineárního Support Vector Machine (SVM)

Lineární SVM je metodou strojového učení, která v úloze klasifikace hledá v prostoru příznaků takovou nadrovinu, která optimálně rozděluje trénovací data [1]. Metoda je binární, tedy rozděluje data dvou rozdílných tříd. Optimální nadrovinou je taková nadrovina, kde všechny body dvou rozdílných tříd leží na opačných stranách této nadroviny a minima vzdáleností bodů od nadroviny jsou co nejvyšší. Tedy v okolí nadroviny je z obou stran co nejširší pásmo bez bodů. Pro popis této nadroviny stačí pouze několik nejbližších bodů, které se nazývají podpůrné vektory (anglicky support vectors). Rozdělovací nadrovinou je lineární funkce. Hledání této nadroviny na trénovacích datech se nazývá tzv. trénování. Pro nalezení co nejoptimálnější nadroviny pro konkrétní úlohu je důležité správně zvolit regularizační parametr  $C$ . Pokud je pro parametr  $C$  zvoleno vysoké číslo, hovoříme o tzv. hard-margin, který se využívá, pokud je úloha úplně lineárně separabilní. Hard-margin SVM nedovolí, aby se z trénovací

množiny dat jakákoliv data nacházela na nesprávné straně nadroviny. Pokud je  $C$  zvoleno jako nízké číslo, hovoříme o tzv. soft-margin, který najde uplatnění v úlohách, které nejsou úplně lineárně separabilní, a tedy SVM při hledání nadroviny neklade takový důraz na správné rozdělení všech trénovacích dat. Ve chvíli, kdy je SVM natrénován, je možné použít tzv. predikci, kdy SVM pro vstupní testovací vektor vrací odhadovanou (predikovanou) třídu.

### Klasifikace do více tříd

Vzhledem k faktu, že SVM je binární klasifikátor, tak rozhodovací neboli diskriminační funkce rozhoduje, zdali příslušný vzorek patří do první nebo druhé třídy. Proto je nutné úlohu klasifikace do více tříd rozdělit na několik různých binárních SVM klasifikátorů. V této diplomové práci bylo pracováno s algoritmy, které v sobě implementují metodu one-vs-rest (jeden proti zbytku). Pro každou klasifikovanou třídu je tedy vytvořen jeden samotný SVM klasifikátor, kde první třídou klasifikátoru je ta klasifikovaná a druhou třídu tvoří vzorky všech ostatních tříd dohromady. Pro  $N$  klasifikovaných tříd je tedy vytvořeno  $N$  binárních SVM klasifikátorů. Klasifikovaný (testovací) vzorek je následně dosazen do diskriminačních funkcí všech těchto  $N$  klasifikátorů a je vybrána ta třída, kde hodnota diskriminační funkce byla nejvyšší.



Obr. 6: Ukázka rozdělovacích nadrovin lineárním SVM pro 3 rozdílné třídy

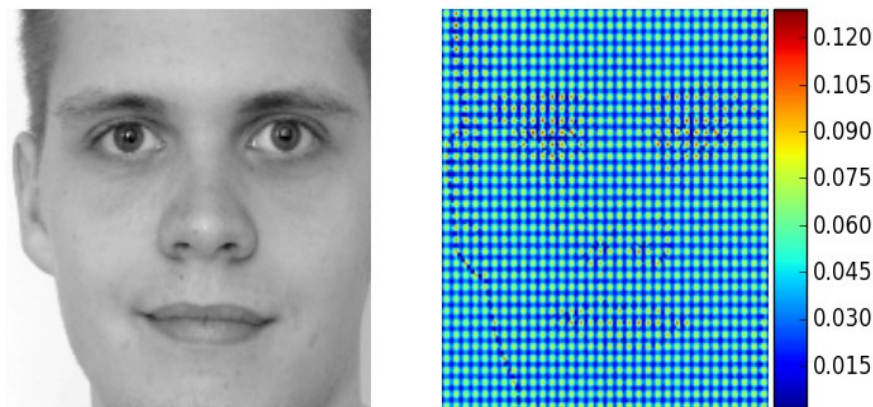
## 1.4 Verifikace obličejů osob

Úlohou verifikace se rozumí správně rozhodnout, zdali dva různé obrazy obličeje náleží jedné osobě či nikoliv. K tomu, aby verifikace mohla být úspěšná, je nutné

nejdříve správně zvolit takové příznaky, které dobře popisují konkrétní obličej. Stejně jako v úloze klasifikace byly pro úlohu verifikace vstupními daty šedotónové obrazy obličejů ve velikosti  $64 \times 64$  pixelů. Úlohu verifikace by bylo tedy možné formovat jednoduše: použít trénovací databázi obrazů, vytvořit prostor PCA a redukovat metodou PCA příznakové vektory verifikovaných obrazů. Spočítat k vektorům např. Euklidovskou vzdálenost (popsána v kapitole 1.3.2) a vypočtenou vzdálenost porovnat s předem určeným prahem, pomocí kterého by bylo rozhodnuto zdali obličej na vstupních obrazech patří stejné osobě či nikoliv. Bohužel kvůli vysoké variabilitě dat nelze pro tuto úlohu pouze za pomoci redukce dimenze metodou PCA vhodný práh nalézt. Proto bylo nutné získat z obrazů příznaky jiným způsobem.

#### 1.4.1 Využití deskriptorů LARK

Pro získání příznaků reprezentujících obličej byla využita metoda LARK (Locally Adaptive Regression Kernel). Tato metoda účinně a efektivně zachycuje lokální geometrickou strukturu obrazu, přičemž využívá výhod soběpodobnosti obrazu založené na gradientech [10]. Tato metoda byla poprvé představena v roce 2011 právě pro úspěšnou verifikaci obrazů obličeje [10]. Metoda pracuje s šedotónovým obrazem, který převede na matici obsahující deskriptory LARK. Ukázka obrazu vyjádřeného pomocí deskriptorů LARK je zobrazena na obr. 7.



Obr. 7: Ukázka obrazu obličeje vyjádřeného příznaky LARK

Po převedení obrazu na matici deskriptorů LARK je následně zredukována dimenze této matice metodou PCA (popsána v 1.3.1 ) na prvních  $N$  komponent.



Výsledný vektor obsahující  $N$  komponent nazveme  $F$ . Poté jsou funkcí 1.9 nelineárně převedeny jednotlivé hodnoty vektoru do hodnot mezi -0.5 do 0.5. Tato funkce dále zajišťuje, že vzniklý vektor  $G$  bude více řídový. Hodnota parametru  $c$  byla zvolena 80, stejně jako ve vědeckém článku [10], kde pro tuto hodnotu vycházelo nejvyšší skóre verifikace.

$$G = \frac{1}{1 + \exp(-c \cdot F)} - 0.5 \quad (1.9)$$

Následně je pro dosažení vyšší úspěšnosti verifikace vhodné pro verifikaci použít lineární SVM, který, jak bylo ověřeno při testování úlohy klasifikace, má při rozpoznávání obličejů velmi dobrou úspěšnost.

#### 1.4.2 Verifikace pomocí lineárního Support Vector Machine (SVM)

Jelikož lineární SVM je klasifikátor, pro úlohu verifikace byly zvoleny pouze dvě klasifikované třídy, a to: obličej na obou obrazech patří stejné osobě / obličej nepatří stejné osobě. Pro natrénování takového SVM je použita množina trénovacích dat, kterou vyjádříme pomocí deskriptorů LARK přesně tak, jak je vysvětleno v kapitole 1.4.1. Následujícím způsobem je z původní množiny trénovacích dat vytvořena nová množina:

1. Pro každý vektor množiny trénovacích dat je podle vzorce 1.10 vypočítán absolutní rozdíl ke všem ostatním vektorům, které reprezentují stejnou osobu. Takto vypočtené rozdílové vektory reprezentují v klasifikátoru SVM první třídu.
2. Dále je pro každý vektor množiny trénovacích dat podle vzorce 1.10 vypočítán absolutní rozdíl k pseudonáhodně vybraným vektorům, které reprezentují rozdílnou osobu. Takto vypočtené rozdílové vektory v SVM klasifikátoru reprezentují druhou třídu. Počet těchto rozdílových vektorů je omezen tak, aby jejich počet byl podobný jako počet rozdílových vektorů z bodu 1.

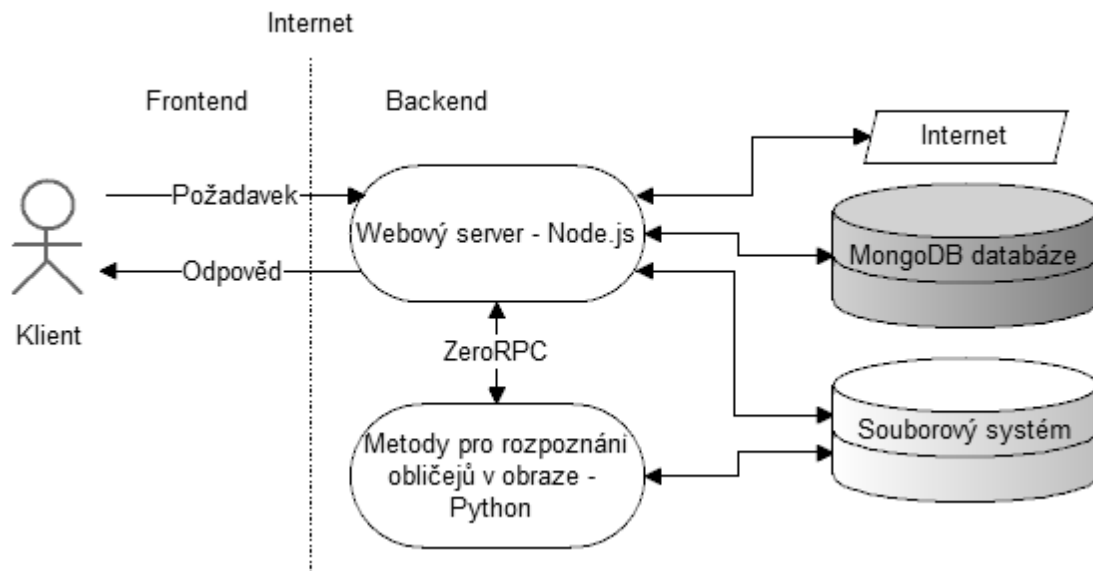
Tímto způsobem je tedy vytvořena nová množina trénovacích dat, která je použita pro natrénování lineárního SVM. Množina testovacích dat pro otestování úspěšnosti verifikace je vytvořena stejným způsobem.

$$D = |X - Y| \quad (1.10)$$

kde  $X$  a  $Y$  jsou vektory reprezentující dva obličeje a  $D$  je výsledný rozdílový vektor.

## 2 Návrh systému pro rozpoznání obličejů v obraze

Z technologického hlediska je možné celý systém rozpoznání obličejů v obraze rozdělit na tři samostatné části. První část reprezentuje úložiště dat a informací. Druhou část tvoří webový server, který slouží pro zpracování požadavků od klienta a také klientovi poskytuje webovou aplikaci. Třetí část se stará o pokročilé metody pro rozpoznání obličejů v obraze. Celá architektura systému je znázorněna na obr. 8.



Obr. 8: Architektura systému

### 2.1 Úložiště dat a informací

Pro uložení informací o rozpoznávaných osobách, obrazech a uživatelích systému byla vybrána moderní databáze MongoDB. Tato databáze byla představena teprve v roce 2007 a v roce 2009 se z ní stal open-source. V současné době je databáze vydávána zdarma pod kombinací licencí GNU AGPL (GNU Affero General Public License) a Apache Licence. Databáze patří do skupiny tzv. NoSQL databází, a tedy nevyužívá tabulky jako v klasických relačních databázích, ale dokumenty podobné formátu JSON (JavaScript Object Notation) [11]. Motivací pro zvolení databáze MongoDB byla zejména jednoduchost jejího návrhu, její škálovatelnost a kvalitní rozhraní pro prostředí Node.js. Pro uložení samotných obrazů osob byl zvolen obyčejný souborový systém, a to hlavně kvůli možnosti mít obrazy lehce přístupné pro programovací jazyk Python a pro jednoduchou možnost zálohy.

## 2.2 Webový server

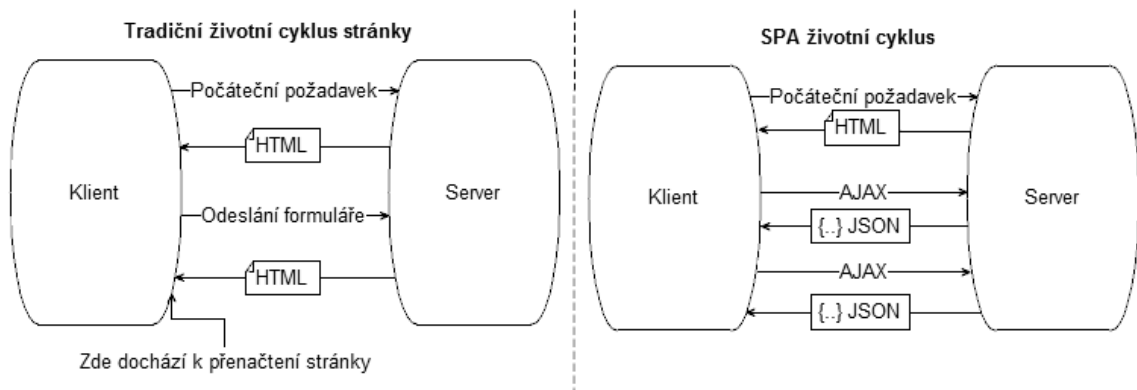
Druhou část tvoří webový server, který byl pro potřeby této práce založen na prostředí Node.js. Prostředí Node.js bylo vytvořeno v roce 2009 pro psaní škálovatelných aplikací a webových serverů. Je vydáváno zdarma pod licenci MIT. Node.js využívá V8 JavaScript engine vyvinutý společností Google pro internetový prohlížeč Google Chrome. Aplikace Node.js jsou tedy psané v programovacím jazyce JavaScript (JS) a využívají událostmi řízenou architekturu a asynchronní I/O operace. Dalo by se tedy říct, že webový server založený na Node.js je aplikace, která asynchronně reaguje na různé události. Navržená aplikace webového serveru využívá pro svou funkci mnoho závislostí (modulů) např. Express, ZeroRPC, OpenCV atd. Tyto moduly jsou instalovány pomocí balíčkovacího manažeru NPM, který je součástí standardní distribuce Node.js. Všechny použité moduly jsou vydávány pod licencemi, které umožňují použití zdarma minimálně pro nekomerční použití. Aplikace webového serveru v této práci přináší dvě důležité funkcionality.

V první řadě aplikace poskytuje webové API (Application Programming Interface) založené na požadavcích protokolu HTTP (Hypertext Transfer Protocol). Toto API využívá architektury REST (Representational State Transfer) a pro komunikaci jsou tedy použity klasické HTTP metody (GET, POST a DELETE). Webové API na jednotlivé požadavky vrací odpovědi ve formátu JSON. Pro úspěšné zpracování některých požadavků vyžadujících práci s obrazem je použita přímo v Node.js knihovna OpenCV a dále webové API komunikuje pomocí RPC (Remote procedure call, konkrétně ZeroRPC) s třetí systémovou součástí, která zajišťuje pokročilé metody pro rozpoznání obličejů v obraze. Komunikace částí systému je znázorněna v architektuře na obr. 8. Webové API představuje v systému tzv. *backend*.

Druhou úlohu, kterou zajišťuje Node.js aplikace webového serveru, je poskytování HTML5 (HyperText Markup Language 5) webové aplikace pro uživatele. Tato aplikace v systému zastává roli tzv. *frontendu*. Je založená na architektuře SPA (Single-Page Application), což znamená, že zdrojový kód HTML, CSS a JavaScript je načten při jediném načtení stránky. Další dynamický obsah je dále asynchronně načítán

na pozadí podle potřeby, obvykle jako reakce na různé akce uživatele. Rozdíl architektury SPA a klasického životního cyklu webové stránky je znázorněn na obr. 9 .

Výsledný vzhled HTML5 aplikace je nadefinován pomocí kaskádových stylů CSS a je založen na populárním frameworku Bootstrap. Tento framework byl využit zejména proto, aby aplikace byla responzivní a mohla tak být použita na velkých displejích počítačů, tabletech či chytrých telefonech. Funkcionalitu HTML5 aplikace zajišťuje programovací jazyk JavaScript, který pomocí AJAX volání komunikuje s webovým API, tedy se systémovým *backendem*. Pro jednodušší programování *frontend* aplikace je využito několika knihoven včetně knihovny jQuery, což je rychlá, malá a velmi bohatá knihovna jazyka JavaScript vydávána pod licencí MIT.



Obr. 9: Rozdíl mezi tradičním životním cyklem webové stránky a architekturou SPA

### 2.3 Metody rozpoznání obličejů v obraze

Poslední část systému zajišťuje pokročilé metody zpracování obrazu a rozpoznání obličejů v obraze (detekce, klasifikace a verifikace). V této části systému je v jazyku Python vytvořen RPC server za pomoci knihovny ZeroRPC. Serveru byly nadefinovány potřebné metody pro rozpoznání obličejů, které pro práci s obrazem využívají knihoven jazyka Python: Dlib, PIL a scikit-learn. Metody RPC serveru jsou volány z aplikace webového serveru viz kapitola 2.2.

### **3 Testování metod rozpoznání obličejů**

Aby výsledný systém dosáhl v rozpoznání obličejů nejlepšího výsledku, bylo nutné nejdříve zjistit, jaké ze zkoumaných metod dosahují při jednotlivých úlohách nejvyšší úspěšnosti. Proto byly v jazyku Python naprogramovány skripty pro otestování jednotlivých metod na dostupných databázích obličejů.

#### **3.1 Databáze obrazů obličejů**

V diplomové práci byly postupně použity tři různé obrazové databáze obličejů. Databáze se liší počtem obrazů a hlavně informacemi, které jsou ke každému obrazu zaznamenány (tzv. anotace). Obrazy všech použitých databází jsou zachyceny v reálném životě za různých podmínek – osvětlení, pozice, druh fotoaparátu atp.

##### **3.1.1 Face Detection Data Set and Benchmark (FDDB)**

Tato databáze obsahuje celkem 2 845 různých obrazů, na kterých je zachyceno 5 171 obličejů v různých pózách a polohách [12]. Databáze má správně anotovány všechny obličeje a jejími tvůrci jsou Vidit Jain a Erik Learned-Miller. Databáze byla použita pro otestování detekčních metod.

##### **3.1.2 Unfiltered faces for gender and age classification (UFGA)**

Databázi UFGA vytvořili Eran Eiding, Roe Enbar a Tal Hassner a obsahuje 26 580 obrazů, na kterých je zachyceno 2 284 různých osob [13]. Všechny obrazy v databázi jsou staženy z alb uživatelů webové aplikace Flickr. Ke všem obrazům jsou anotace obsahující informace o pohlaví, věkovém intervalu a označení konkrétní osoby. Tato databáze byla použita při testování úloh klasifikace pohlaví, věku a při úloze verifikace.

##### **3.1.3 Labeled Faces in the Wild (LFW)**

Tato databáze obsahuje 13 233 obrazů 5 749 různých všeobecně známých osob např. Andre Agassi, Angelina Jolie, Bill Gates atd. Každý zachycený obličej je anotován jménem a příjmením dané osoby. 1 680 různých osob má v databázi více než dva různé obrazy. Tuto databázi vytvořili Gary B. Huang, Manu Ramesh, Tamara Berg,

and Erik Learned-Miller [14]. Databáze byla v této práci využita pro otestování klasifikace osob a verifikace. Data některých osob byly také později naimportovány přímo do finálního webového systému.

### 3.2 Testovací metriky

Pro správné vyhodnocení jednotlivých testů (experimentů) jsou níže specifikovány metriky, které celkovou úspěšnost metod vyhodnocují.

#### 3.2.1 Úspěšnost

Tato nejjednodušší metrika popisuje úspěšnost testu, tedy kolik předpovězených hodnot bylo správně. Úspěšnost se dá vyjádřit následovně:

$$úspěšnost(y, \hat{y}) = \frac{1}{1+n} \cdot \sum_{i=1}^{n-1} 1(\hat{y}_i = y_i) \quad (3.1)$$

kde  $y$  je vektor správných hodnot,  $\hat{y}$  je vektor předpovězených hodnot a  $n$  je celkový počet vzorků.

#### 3.2.2 Přesnost (precision), úplnost (recall) a $F_1$ -skóre

Tyto tři metriky velmi dobře vyhodnocují testovanou metodu. Přesnost (precision) určuje schopnost metody neoznačit pozitivně vzorek, který je negativní. Úplnost (recall) definuje schopnost metody najít všechny pozitivní vzorky.  $F_1$ -skóre může být interpretováno jako vážený harmonický průměr přesnosti a úplnosti.

V úloze binární klasifikace se pojem pozitivní (positive) a negativní (negative) vztahuje k predikci klasifikátoru a pojem pravdivý (true) a falešný (false) říká, zdali tato predikce odpovídá vnějšímu pozorování. Tyto pojmy jsou znázorněny v tabulce 1.

Tabulka 1: Definice pojmů binární klasifikace

	Aktuální třída (pozorování)	
Předpovězená třída (očekávání)	TP (pravdivě pozitivní) správný výsledek	FP (falešně pozitivní) nečekaný výsledek
	FN (falešně negativní) chybějící výsledek	TN (pravdivě negativní) správná absence výsledku

V této souvislosti je možné definovat přesnost, úplnost a  $F_1$ -skóre následovně:

$$\text{přesnost} (\textit{precision}) = \frac{TP}{TP + FP} \quad (3.2)$$

$$\text{úplnost} (\textit{recall}) = \frac{TP}{TP + FN} \quad (3.3)$$

$$F_1 \text{ skóre} = 2 \frac{\text{přesnost} \cdot \text{úplnost}}{\text{přesnost} + \text{úplnost}} \quad (3.4)$$

Při klasifikaci do více tříd jsou metriky vypočítány nejdříve pro každou klasifikovanou třídu zvlášť a následně je vypočten průměr metrik pro všechny třídy. Tyto výsledné průměrné hodnoty značíme velkými písmeny P (precision – přesnost), R (recall – úplnost) a  $F_1$  ( $F_1$ -skóre).

### 3.3 Testování metod detekce obličejů v obraze

V rámci této diplomové práce byly otestovány dvě úspěšné metody detekce obličejů v obraze: detekce HAAR a detekce HOG. Metody jsou popsány v kapitole 1.2 Pro detekci HAAR byla použita natrénovaná kaskáda Rainera Lienharta pro rozpoznání obličejů (*haarcascade\_frontalface\_alt.xml*) a byly využity výchozí hodnoty parametrů. Pro detekci HOG byl využit natrénovaný detektor obličejů z knihovny Dlib (*get\_frontal\_face\_detector()*) a parametr *upsample* byl nastaven na 1. Metody byly testovány za stejných podmínek na celé databázi obrazů FDDB. Celý experiment je naprogramován ve skriptu *test\_detection.py*. Ve skriptu je možné zvolit metodu detekce a cestu k datům FDDB databáze. Po úspěšném proběhnutí skriptu je výsledkem testu posledních 8 řádků standardního výstupu. Výsledek běhu skriptu může vypadat následovně, za znakem # je uveden význam jednotlivého řádku (ten není součástí standardního výstupu).

```
Time consumption 382.170s #časová náročnost testu
Precision: 0.937 #výsledná přesnost
Recall: 0.732 #výsledná úplnost
F1 score: 0.822 #výsledné F1-skóre
True positive: 3786 #počet TP vzorků
False positive: 256 #počet FP vzorků
False negative: 1385 #počet FN vzorků
Right num of detected faces: 5171 #celkový počet test.
obličejů
```



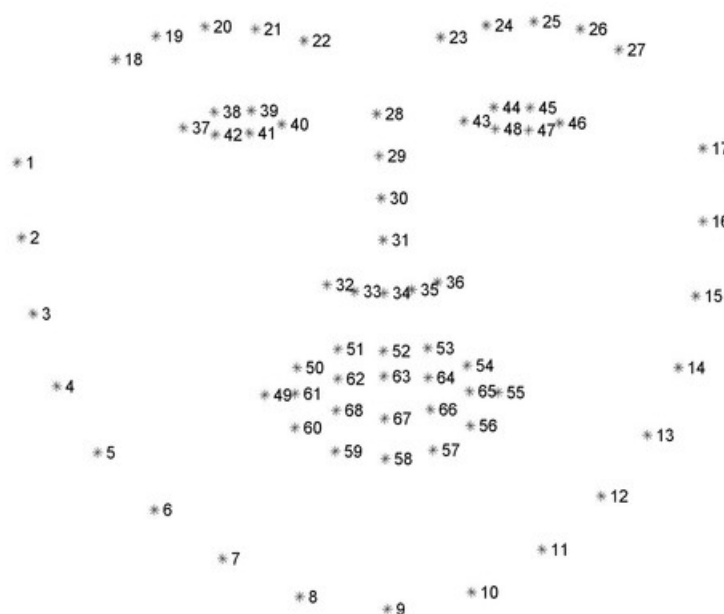
Následující tabulka obsahuje výsledky obou testovaných metod. Z tabulky je zřejmé, že úspěšnější je metoda detekce HOG.

Tabulka 2: Porovnání testovaných metrik algoritmů detekce

Metoda	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
detekce HAAR	93,70 %	73,20 %	82,20 %
detekce HOG	<b>96,30 %</b>	<b>74,50 %</b>	<b>84,00 %</b>

### 3.4 Předzpracování obrazu

Pro úlohu klasifikace nebo verifikace bylo ze vstupních obrazů nutné správně vyříznout a normovat oblast obsahující obličej. Proto byla nejdříve na vstupní obrazy použita metoda detekce HOG (knihovna Dlib *get\_frontal\_face\_detector()*), která v obraze detekovala oblast obsahující obličej. Následně byl opět z knihovny Dlib využit prediktor tzv. landmarků, tedy významných bodů popisujících tvar obličeje. Tyto landmarky jsou vstupem níže uvedeného frontalizačního algoritmu. Tento prediktor byl v Dlib vytvořen na základě vědeckého článku [15] a natrénován na iBUG 300-W datasetu [16]. Vstupem prediktoru je pole obsahující původní obraz a objekt popisující detekovanou obdélníkovou oblast obličeje. Výstupem prediktoru je objekt obsahující 68 landmarků – bodů popisujících tvar obličeje dle následujícího obrázku.



Obr. 10: Pozice 68 landmarků popisujících tvar obličeje (převzato z [16])

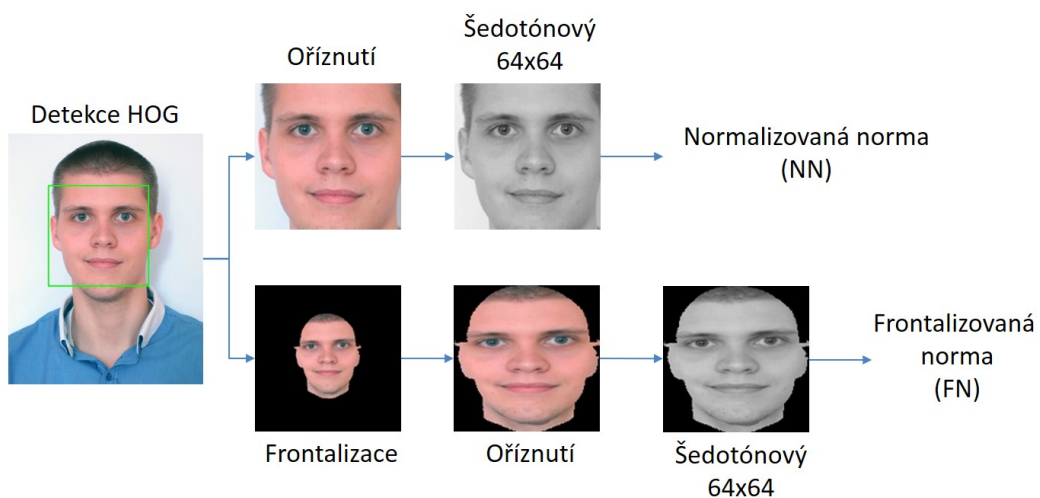
Nyní bylo možné vytvořit dvě normy obrazů, které jsou v práci dále nazývány: normalizovaná norma (NN) a frontalizovaná norma (FN).

Pro normu NN byl obraz oříznut podle nalezené obdélníkové oblasti, následně byl obraz transformován na velikost  $64 \times 64$  pixelů a převeden do stupňů šedi (256 hodnot).

Pro normu FN byl obličej z obrazu nejprve frontalizován. To znamená, že obličej byl chytré transformován tak, jako by se nacházel čelem k zařízení, které ho snímalo. Druhou funkcí frontalizačního algoritmu je, že v obraze zachová pouze obličej, ostatní části obrazu jsou obarveny černě. Pro frontalizaci byl použit frontalizační algoritmus popsáný ve vědeckém článku [17]. Implementaci tohoto algoritmu pro jazyk Python vytvořil Heng Yang a publikoval ho na serveru GitHub. Vstupem frontalizačního algoritmu je:

- pole obsahující původní obraz
- objekt popisující obdélníkovou oblast detekce
- pole obsahující landmarky popisující obličej

Následně byl frontalizovaný obraz oříznut, transformován na velikost  $64 \times 64$  pixelů a převeden do stupňů šedi (256 hodnot). Postup předzpracování obrazu do těchto dvou norem je znázorněn na obr. 11.



Obr. 11: Postup předzpracování obrazu do dvou obrazových norem

### 3.4.1 Hromadné předzpracování obrazů

Pro hromadné předzpracování obrazů (vytvoření datasetu) např. pro celé databáze byla vytvořena třída `DATASET_IMPROVED`, která se nachází ve skriptu `dataset_improved.py`. Vytvořením instance třídy `DATASET_IMPROVED` dochází k výše vysvětlenému předzpracování veškerých použitelných obrazů z daného zdrojového adresáře. V tomto adresáři musí mít každá klasifikační třída právě jeden podadresář, který obsahuje obrazy obličejů dané třídy. V každém obrazu musí být detekován právě jeden obličej, v jiném případě je obraz ignorován.

Parametry konstruktoru této třídy jsou:

- název souboru (pro možnost uložení vytvořeného datasetu)
- cesta k zdrojovému adresáři
- příznak použití frontalizace (*True* – použít, *False* - nepoužít)

Po úspěšném vytvoření instance třídy `DATASET_IMPROVED` je možné přistoupit k jednotlivým atributům třídy:

- *filename* – název souboru (pro možnost uložení vytvořeného datasetu)
- *path* – cesta k zdrojovému adresáři
- *frontalization* – příznak, zdali byla použita frontalizace
- *data* – obsahuje pole obrazů obličejů (vzorků), kde každý obraz je vyjádřen jako pole obsahující 4 096 (64×64) hodnot
- *target\_names* – obsahuje pole s názvy klasifikovaných tříd (získáno z názvů podadresářů)
- *target* – obsahuje pole indexů tříd pro každý obraz
- *landmarks* – obsahuje pole, kde každý řádek obsahuje pole landmarků pro daný obraz

Třída *DATASET\_IMPROVED* obsahuje také metody, které umožňují: uložit vytvořený dataset do souboru (*save*), zobrazit informace o datasetu (*print\_info*) nebo normalizovat počet obrazů pro každou třídu (*normalize* a *best\_normalize*). Statická metoda *load*, dokáže načíst dříve vytvořený dataset ze specifikovaného souboru.

### 3.5 Klasifikace pohlaví osoby z obrazu obličeje

Pro klasifikaci pohlaví osoby byla použita databáze UFGA (kapitola 3.1.2 ). Z této databáze byly vytvořeny 4 datasety. Dva datasety pro normalizovanou normu (*dataset\_gender\_nn\_train.ds* a *dataset\_gender\_nn\_test.ds*) a dva datasety pro frontalizovanou normu (*dataset\_gender\_fn\_train.ds* a *dataset\_gender\_fn\_test.ds*). Datasety končící *train.ds* byly použity pro trénovací část algoritmů (PCA, GMM a lineární SVM). Pro trénování bylo vždy použito 2000 obrazů obličejů žen a 2000 obrazů obličejů mužů. Datasety končící *test.ds* byly využity pro testování úspěšnosti algoritmů. Při testování klasifikace pohlaví bylo vždy použito 400 obrazů obličejů žen a 400 obrazů obličejů mužů. Testování probíhalo na datech redukováných pomocí metody PCA, kde maximální počet použitých prvních PCA komponent byl 350. Test je naprogramován ve skriptu *test\_classification.py*, kde na začátku skriptu je nutné specifikovat dva vstupní datasety (trénovací a testovací). Při běhu skriptu jsou postupně vypisovány hodnoty jednotlivých algoritmů na standardní výstup a po doběhnutí skriptu jsou vykresleny grafy úspěšností jednotlivých testovaných metod.

#### 3.5.1 Normalizovaná norma obrazů (NN)

##### Euklidovská vzdálenost (ED)

Nejvyšší naměřené úspěšnosti klasifikace **56,75 %** pomocí metody Euklidovské vzdálenosti bylo dosaženo při využití prvních **320**, 330, 340 nebo 350 PCA komponent (zobrazeno v grafu 1). Z grafu je také zřejmé, že úspěšnost metody se v závislosti na počtu komponent téměř nemění. Podrobné výsledky viz příloha 1.

##### Mahalanobisova vzdálenost (MD)

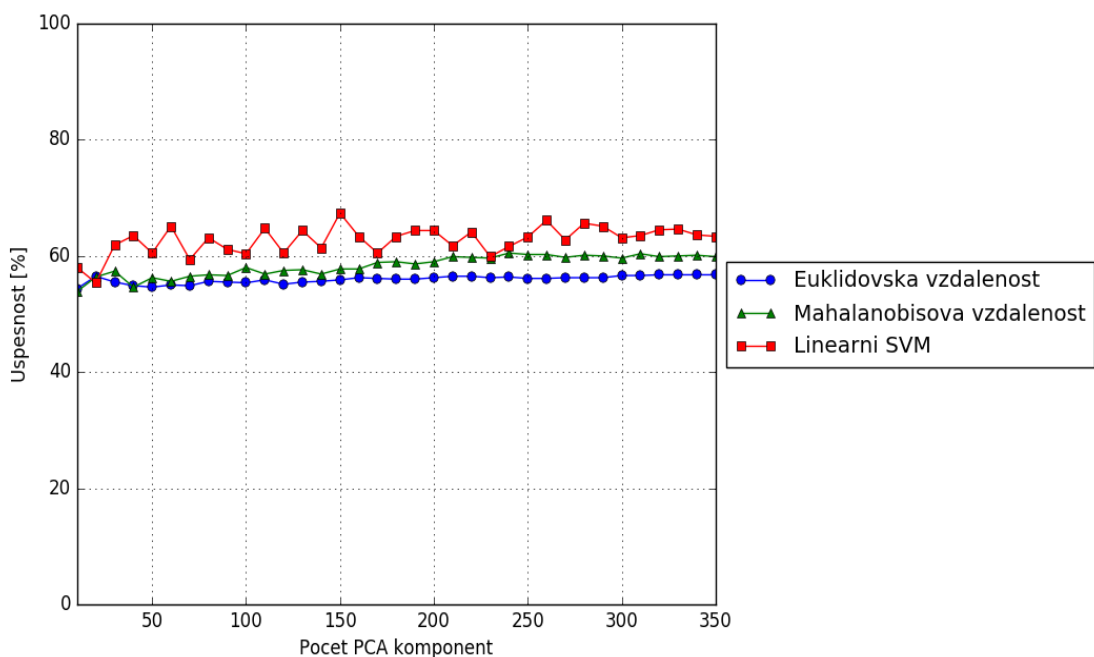
Metodou Mahalanobisovy vzdálenosti bylo dosaženo zhruba o 4 % vyšší úspěšnosti než u předešlé metody, a to **60,5 %** při využití prvních **240** PCA komponent. Výsledky jsou zobrazeny na grafu 1, případně v podrobných výsledcích viz příloha 2.

### Gaussian Mixture Models (GMM)

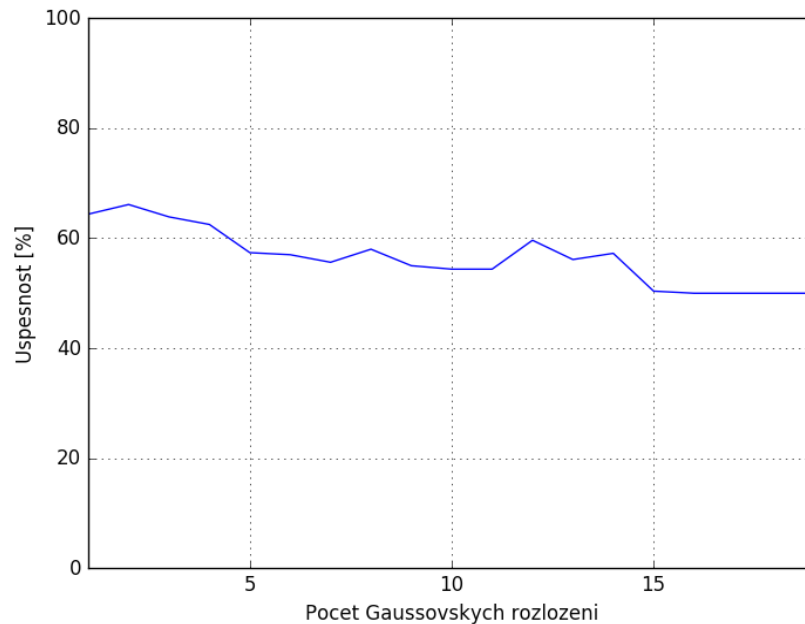
Pro klasifikaci metodou GMM bylo použito 240 prvních PCA komponent, protože právě tento počet komponent měl v předchozí metodě nejvyšší úspěšnost. Následně byl otestován vliv počtu použitých Gaussových rozdělení (mixtur) na úspěšnost metody. Nejvyšší úspěšnosti **66,13 %** bylo dosaženo pro **2** mixtury. Znázorněno v grafu 2 a v podrobných výsledcích v příloze 3.

### Lineární Support Vector Machine (SVM)

Pro metodu lineární Support Vector Machine bylo nutné zvolit regularizační parametr  $C$ . Hodnota tohoto parametru byla zvolena jako 0,0001, kde toto číslo bylo experimentálně zjištěno jako nejvhodnější pro tuto úlohu. Následně byl klasifikátor SVM testován pro 10 až 350 prvních PCA komponent s krokem 10. Jak vyplývá z grafu 1, nejvyšší dosažené úspěšnosti: **67,38 %** bylo dosaženo pro prvních **150** PCA komponent. Podrobné výsledky viz příloha 4.



Graf 1: Úspěšnost klasifikace pohlaví pomocí Euklidovské vzdálenosti, Mahalanobisovy vzdálenosti a lineárního SVM v závislosti na počtu prvních  $n$  PCA komponent (normalizovaná norma)



Graf 2: Úspěšnost klasifikace pohlaví pomocí GMM v závislosti na počtu Gaussových rozdělení (mixtur) (normalizovaná norma)

### 3.5.2 Frontalizovaná norma obrazů (FN)

#### Euklidovská vzdálenost (ED)

Nejvyšší úspěšnosti klasifikace pomocí metody Euklidovské vzdálenosti bylo dosaženo **62,38 %** při využití prvních 20 PCA komponent viz graf 3. Z grafu je také zřejmé, že úspěšnost se v závislosti na počtu komponent, stejně jako v případě normalizované normy, mění pouze minimálně. Podrobné výsledky viz příloha 5.

#### Mahalanobisova vzdálenost (MD)

Nejvyšší úspěšnosti klasifikace pomocí metody Mahalanobisovy vzdálenosti bylo dosaženo **61 %** při využití prvních 170 PCA komponent, zobrazeno na grafu 3. Podrobné výsledky viz příloha 6.

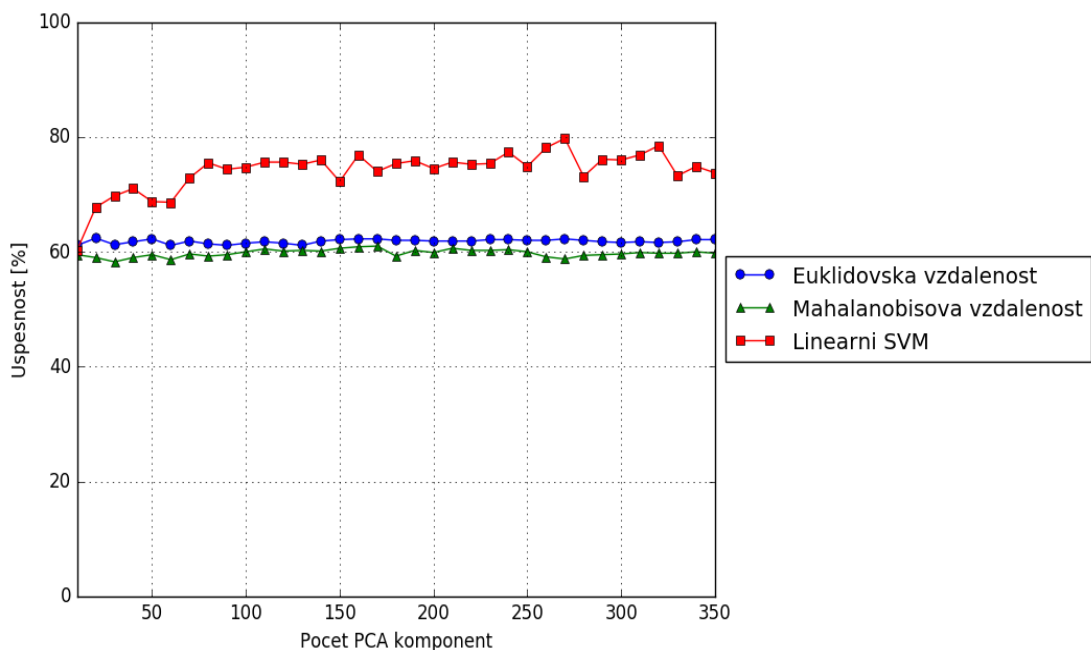
#### Gaussian Mixture Models (GMM)

Pro klasifikaci metodou GMM bylo použito 170 prvních PCA komponent, jelikož předchozí metoda pro tento počet komponent získala nejvyšší úspěšnost, a zároveň nedocházelo k tak vysoké redukci dimenze jako u metody Euklidovské vzdálenosti. Následně byl otestován vliv počtu použitých Gaussovských rozdělení (mixtur) na

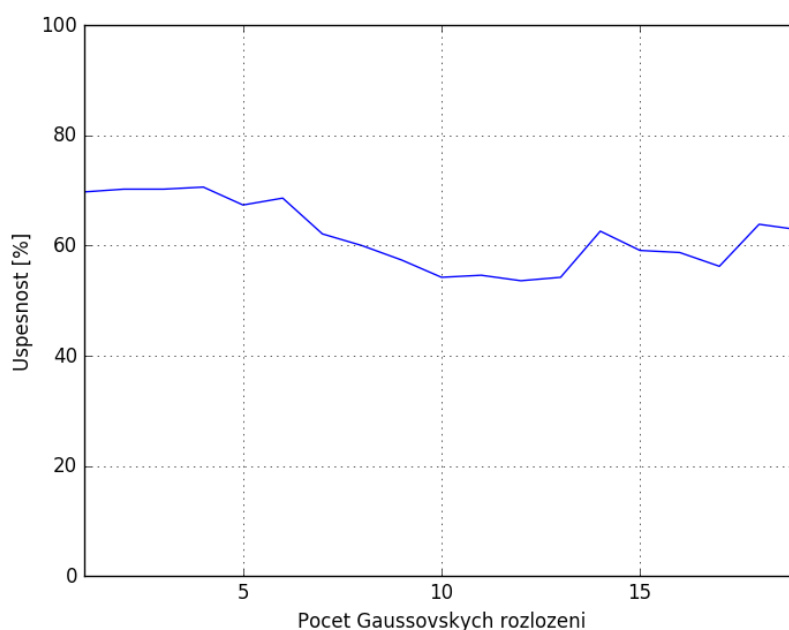
úspěšnost metody. Nejvyšší dosažená úspěšnost byla **70,75 %**, která odpovídala použití 4 Gaussových rozdělání (mixtur) viz graf 4 nebo v podrobných výsledcích v příloze 7.

### Lineární Support Vector Machine (SVM)

Pro metodu lineární Support Vector Machine byl stejně jako pro normalizovanou normu zvolen regularizační parametr  $C=0,0001$ . Následně byl klasifikátor SVM testován pro 10 až 350 prvních PCA komponent s krokem 10. Nejvyšší dosažené úspěšnosti: **79,75 %** bylo dosaženo při použití pro prvních **270** PCA komponent viz graf 3. Podrobné výsledky viz příloha 8. Tento klasifikátor tedy v celém experimentu klasifikace pohlaví dosáhl nejvyšší úspěšnosti, a byl proto využit pro klasifikaci pohlaví ve webovém systému.



Graf 3: Úspěšnost klasifikace pohlaví pomocí Euklidovské vzdálenosti, Mahalanobisovy vzdálenosti a lineárního SVM v závislosti na počtu prvních  $n$  PCA komponent (frontalizovaná norma)



Graf 4: Úspěšnost klasifikace pohlaví pomocí GMM v závislosti na počtu Gaussových rozdělení (mixtur) (frontalizovaná norma)

### 3.5.3 Srovnání testovaných metod

Testováním normalizované normy (NN) a frontalizované normy (FN) bylo zjištěno, že frontalizovaná norma dosahuje v průměru o **5,8 %** vyšší úspěšnosti a při využití klasifikační metody lineární SVM bylo dosaženo dokonce o 12,4 % vyšší úspěšnosti než při využití normalizované normy. V následující tabulce nalezneme srovnání všech testovaných metod pro klasifikaci pohlaví.

Tabulka 3: Srovnání testovaných metod klasifikace pohlaví

<b>Použitá metoda</b>	<b>Úspěšnost</b>
NN+PCA+ED	56,8 %
NN+PCA+MD	60,5 %
NN+PCA+GMM	66,1 %
NN+PCA+SVM	67,4 %
FN+PCA+ED	62,4 %
FN+PCA+MD	61 %
FN+PCA+GMM	70,8 %
<b>FN+PCA+SVM</b>	<b>79,8 %</b>



### 3.6 Klasifikace věku osoby z obrazu obličeje

Pro klasifikaci věku byla opět použita databáze obrazů UFGA (kapitola 3.1.2 ). Z této databáze byly vytvořeny dva datasety pouze pro frontalizovanou normu obrazů (*dataset\_ages\_fn\_train.ds* a *dataset\_ages\_fn\_test.ds*). Datasety pro normalizovanou normu již vytvářeny nebyly, protože z předchozího experimentu klasifikace pohlaví bylo zjištěno, že použití frontalizované normy obrazů obličejů je pro všechny klasifikační metody výhodnější. Dataset *dataset\_ages\_fn\_train.ds* byl v experimentech použit pro trénovací část algoritmů (PCA, GMM a lineární SVM). Při trénování bylo použito 279 obrazů pro každou klasifikovanou třídu. Klasifikovanými třídami bylo 8 věkových intervalů: 0-3, 3-7, 7-14, 14-22, 22-34, 34-45, 45-56, 56-100. Pro testování bylo z datasetu *dataset\_ages\_fn\_test.ds* použito 91 obrazů pro každou třídu. Testování opět probíhalo na obrazových datech redukovaných metodou PCA, kde maximální počet použitých PCA komponent byl 350. Test byl naprogramován ve skriptu *test\_classification.py*, kde na začátku skriptu je nutné specifikovat dva vstupní datasety (trénovací a testovací). Testovací skript ani výstup skriptu se neliší od skriptu použitého pro testování klasifikace pohlaví.

#### 3.6.1 Euklidovská vzdálenost (ED)

Nejvyšší naměřené úspěšnosti klasifikace věku pomocí metody Euklidovské vzdálenosti bylo dosaženo **23,63 %** při využití prvních **200, 230, 310** PCA komponent, jak je zobrazeno v grafu 5, nebo v podrobných výsledcích viz příloha 9.

#### 3.6.2 Mahalanobisova vzdálenost (MD)

Metodou Mahalanobisovy vzdálenosti bylo dosaženo úspěšnosti **26,1 %** při využití prvních **80** PCA komponent. Výsledky jsou zobrazeny na grafu 5, případně v podrobných výsledcích v příloze 10.

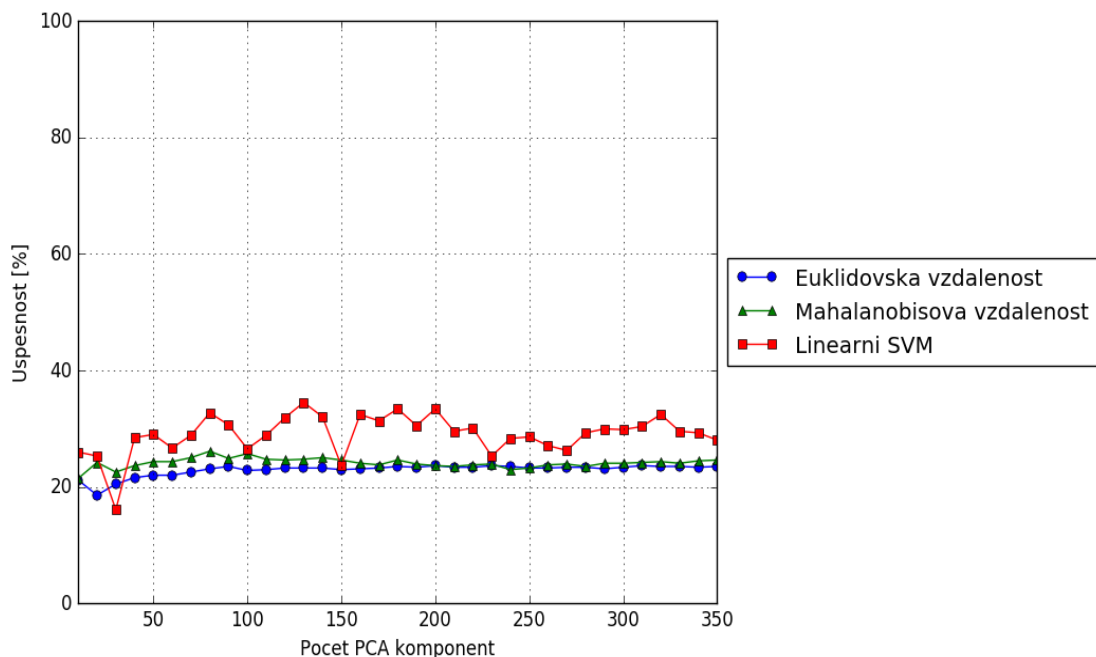
#### 3.6.3 Gaussian Mixture Models (GMM)

Pro klasifikaci metodou GMM bylo použito 80 prvních PCA komponent, protože právě tento počet komponent měl v předchozí metodě nejvyšší úspěšnost. Následně byl otestován vliv počtu použitých Gaussových rozdělení (mixtur) na úspěšnost metody.

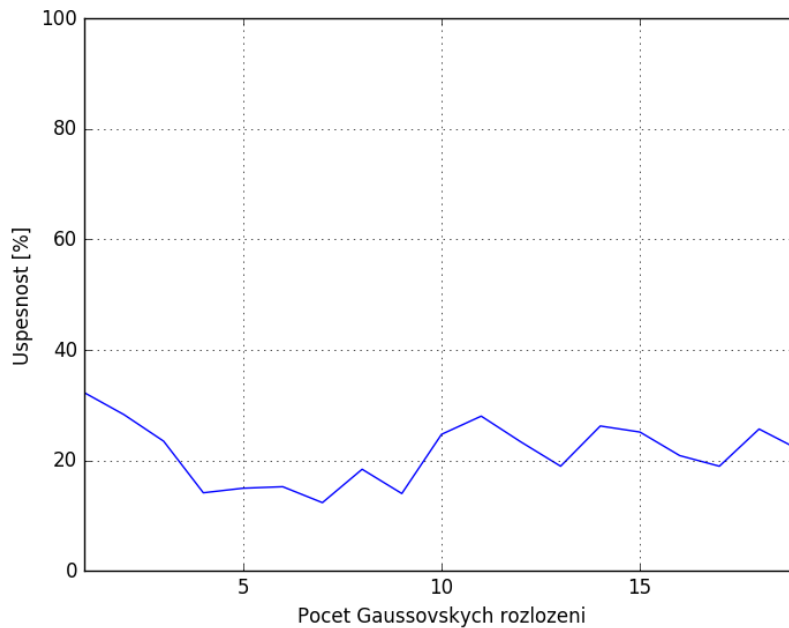
Nejvyšší úspěšnosti bylo dosaženo pro **jednu** mixturu: **32,28 %**. Znáznorněno na grafu 6 a v podrobných výsledcích viz příloha 11.

### 3.6.4 Lineární Support Vector Machine (SVM)

Pro metodu lineární Support Vector Machine byl opět zvolen regularizační parametr  $C=0,0001$ . Následně byl klasifikátor SVM stejně jako v případě klasifikace pohlaví testován pro 10 až 350 prvních PCA komponent s krokem 10. Nejvyšší dosažené úspěšnosti **34,48 %** bylo dosaženo při použití prvních **130** PCA komponent viz graf 5. Podrobné výsledky viz příloha 12. Při porovnání úspěšnosti testovaných metod a grafů jejich průběhů bylo zjištěno, že klasifikátor lineární SVM dosahuje při klasifikaci obličejů vždy nejvyšší úspěšnosti (viz tabulka 3 a 4). Tento klasifikátor byl následně použit pro klasifikaci věku ve webovém systému.



Graf 5: Úspěšnost klasifikace věku pomocí Euklidovské vzdálenosti, Mahalanobisovy vzdálenosti a lineárního SVM v závislosti na počtu prvních  $n$  PCA komponent (frontalizovaná norma)



Graf 6: Úspěšnost klasifikace věku pomocí GMM v závislosti na počtu Gaussovských rozdělení (mixtur) (frontalizovaná norma)

Tabulka 4: Srovnání testovaných metod klasifikace věku

Použitá metoda	Úspěšnost
FN+PCA+ED	23,6 %
FN+PCA+MD	26,1 %
FN+PCA+GMM	32,3 %
<b>FN+PCA+SVM</b>	<b>34,5 %</b>

### 3.7 Klasifikace (identifikace) konkrétní osoby z obrazu obličeje

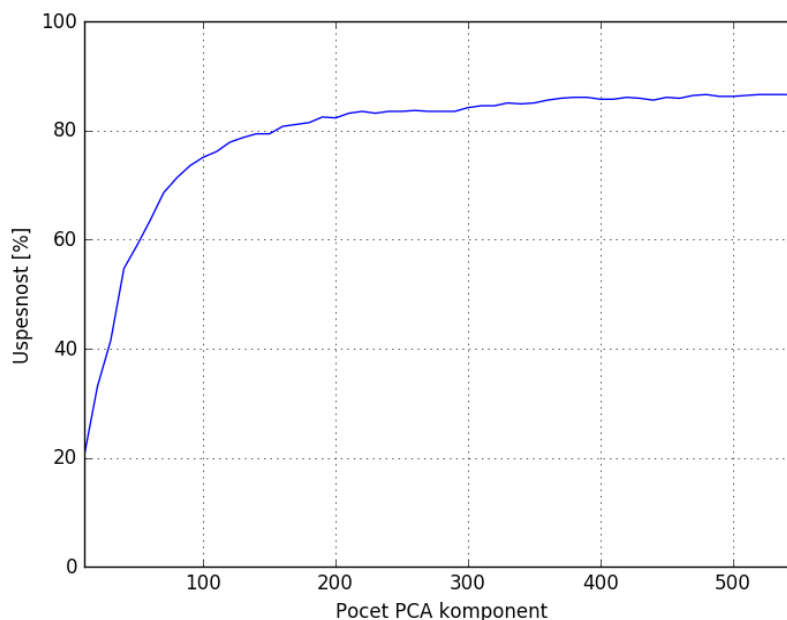
Pro otestování klasifikace osob byla použita databáze obrazů LFW (viz kapitola 3.1.3). Z této databáze byl vytvořen jeden dataset pro frontalizovanou normu obrazů (*dataset\_people\_lfw\_fn.ds*). Dataset pro normalizovanou normu vzhledem k výsledkům úspěšností u experimentů klasifikace pohlaví již vytvářen nebyl. Z datasetu byly použity pouze osoby (třídy), které obsahovaly mezi 10-300 obrazy (vzorky). Celkem bylo tedy pro klasifikaci použito 84 osob (tříd), kde každá třída byla v datasetu reprezentována průměrně 32 vzorky.

Dataset byl následně rozdělen na trénovací a testovací část. Pro trénovací část bylo zvoleno 80 % vzorků a pro testování zbylých 20 % vzorků. Testovaným

algoritmem byl v případě klasifikace osob pouze lineární Support Vector Machine, který, jak vyplynulo z předešlých experimentů, dosahuje při klasifikaci obličejů nejvyšší úspěšnosti. Testování probíhalo na obrazových datech redukováných metodou PCA, kde maximální počet použitých PCA komponent byl 550. Původně byl horní limit počtu PCA komponent zvolen 350 jako u předešlých úloh klasifikace, ale vzhledem k vývoji úspěšností křivky lineárního SVM, byl tento limit vhodně zvýšen. Celý test byl naprogramován ve skriptu *test\_classification\_people.py*, kde na začátku skriptu je nutné specifikovat vstupní dataset. Při běhu skriptu jsou vypisovány výsledné metriky testovaných lineárních SVM a po doběhnutí skriptu je vykreslen graf úspěšnosti jednotlivých lineárních SVM v závislosti na počtu prvních PCA komponent.

### 3.7.1 Lineární Support Vector Machine (SVM)

Pro klasifikaci osob pomocí lineárního SVM byl zvolen regularizační parametr  $C=1 \times 10^{-7}$ , který byl experimentálně zjištěn jako nejvhodnější pro tuto úlohu. Následně byl klasifikátor SVM testován pro 10 až 550 prvních PCA komponent s krokem 10. Výsledná křivka úspěšnosti (viz graf 7) je velmi podobná průběhu logaritmické funkce. Nejvyšší úspěšnosti **86,62 %** bylo dosaženo při využití **480, 520, 530, 540** prvních PCA komponent. Podrobné výsledky viz příloha 13.



Graf 7: Úspěšnost klasifikace (identifikace) konkrétních osob pomocí lineárního SVM v závislosti na počtu prvních n PCA komponent (frontalizovaná norma)

### 3.8 Verifikace osob na základě dvou obrazů obličeje

Pro otestování úspěšnosti verifikace byly použity dvě databáze obrazů: UFGA (viz kapitola 3.1.2) a databáze LFW (viz kapitola 3.1.3). Obrazy obou databází byly použity ve frontalizované normě. Z databáze UFGA byl vytvořen jeden dataset anonymních osob (*dataset\_anonymous\_people\_fn.ds*). Z datasetu bylo použito 606 obrazů 202 různých osob. Z těchto obrazů byly získány příznaky LARK a následně byl pomocí těchto příznaků vytvořen vlastní prostor PCA (viz kapitola 1.3.1).

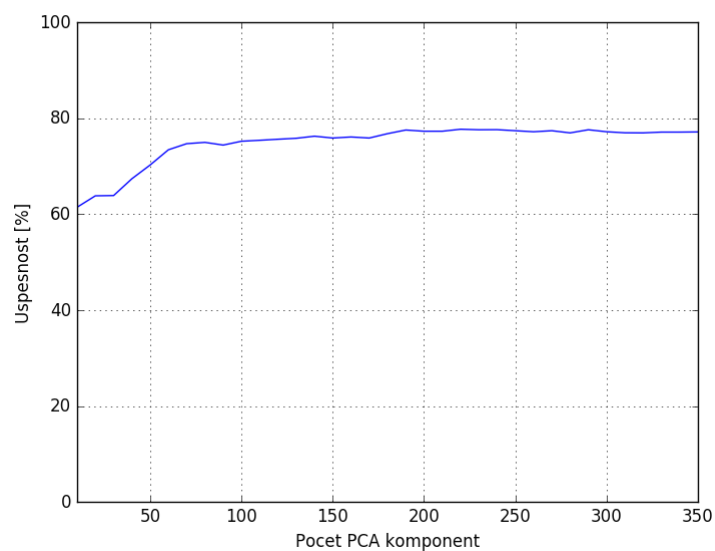
Z databáze LFW byl použit stejný dataset (*dataset\_people\_lfw\_fn.ds*) jako pro případ klasifikace osob. Z tohoto datasetu bylo načteno 38 tříd (osob), kde každá třída je průměrně reprezentována 48 obrazy. Dataset je následně rozdělen na trénovací a testovací část. Pro trénovací část bylo použito 80 % vzorků a pro testování zbylých 20 % vzorků. Následně jsou vzorky pro trénovací a testovací část převedeny na vektory deskriptorů LARK, kde tyto vektory jsou projektovány do výše zmíněného vlastního prostoru PCA. Celý postup převodu na příznaky LARK je popsán v kapitole 1.4.1. Následně byla úloha verifikace převedena na úlohu klasifikace a jako klasifikátor byl využit lineární SVM, který dosahuje v úspěšnosti klasifikace obličejů nejlepších výsledků. Postup převodu úlohy je popsán v kapitole 1.4.2.

Celý test byl naprogramován ve skriptu *test\_verification.py*, kde na začátku skriptu je nutné specifikovat dva vstupní datasety. Při běhu skriptu jsou vypisovány výsledné metriky testovaných lineárních SVM a po doběhnutí skriptu je vykreslen graf úspěšnosti verifikace za pomoci lineárních SVM v závislosti na počtu prvních PCA komponent.

#### 3.8.1 Lineární Support Vector Machine (SVM)

Pro verifikaci osob pomocí lineárního SVM byl zvolen regularizační parametr  $C=0,005$ . Tento parametr byl experimentálně zjištěn jako nejvhodnější pro tuto úlohu. Následně byl klasifikátor SVM testován pro 10 až 350 prvních PCA komponent s krokem 10. Výsledná závislost klasifikace na počtu PCA komponent je zobrazena na grafu 8. Nejvyšší úspěšnosti verifikace **77,74 %** bylo dosaženo při využití **220** prvních

PCA komponent. Podrobné výsledky viz příloha 14. Tato verifikační metoda byla následně využita pro řešení úlohy verifikace ve webovém systému.



Graf 8: Úspěšnost verifikace osob pomocí lineárního SVM v závislosti na počtu prvních  $n$  PCA komponent (frontalizovaná norma)

## 4 Implementace webového systému

Součástí této diplomové práce je systém umožňující uživateli rozpoznávat vlastní obrazy s obličejí osob. Prvním krokem k úspěšné implementaci tohoto systému bylo navržení databáze.

### 4.1 Návrh databáze

Pro účely této práce byla vytvořena databáze s názvem *fapi*, která, jak je uvedeno v kapitole 2.1, je typu MongoDB. V této databázi jsou uloženy všechny informace, které systém pro svou funkci využívá:

- Informace o osobách, které je možné systémem rozpoznávat.
- Informace o obrazech reprezentujících jednotlivé osoby.
- Informace o uživateli resp. administrátorech systému.

V MongoDB je pro uložení dat využito kolekcí a dokumentů. Kolekce v MongoDB lze přirovnat k tabulce klasické relační databáze. Kolekce v sobě uchovává jednotlivé dokumenty, které reprezentují konkrétní záznam. V MongoDB existují dva datové modely ukládání dokumentů. **Normalizovaný datový model** využívá k propojení více typů dokumentů reference. Tento model má výhodu vyšší přehlednosti kolekcí a jejich flexibility. Naopak nevýhodou tohoto modelu je vyšší složitost při vytváření nových dokumentů a vyhledávání dokumentů při využití referencí. Druhým datovým modelem je tzv. **vkládající model**, kde přímo v hlavním dokumentu jsou uloženy i další pod-dokumenty jiného typu. Při čtení je tedy postačující načíst pouze hlavní dokument a ihned jsou získány i všechny pod-dokumenty, které tento dokument obsahuje. Databáze systému využívá kombinace zmíněných modelů a pro uložení informací jsou použity tři kolekce: *people*, *peopleImages* a *users*. V kolekcích MongoDB není nutné předem specifikovat datové typy atributů jednotlivých ukládaných dokumentů. Datový typ je implicitně nastaven až podle právě ukládaných dat.

#### 4.1.1 Kolekce *people*

V této kolekci jsou uloženy informace o jednotlivých osobách, které je možné systémem rozpoznávat. Každý vytvořený dokument v této kolekci obsahuje následující atributy:

- *\_id* – identifikátor dokumentu (*ObjectId*)
- *name* – jméno a příjmení osoby
- *dateofBirth* – datum narození osoby (pokud není známo, je zde uloženo *null*)
- *gender* – pohlaví osoby (pokud není známo, je zde uloženo *null*)

#### 4.1.2 Kolekce *peopleImages*

Tato kolekce uchovává dokumenty s informacemi o obrazech obličejů, které reprezentují jednotlivé osoby v kolekci *people*. Systém tyto obrazy následně využívá pro natrénování klasifikátoru osob. Atributy dokumentů v této kolekci jsou následující:

- *\_id* – identifikátor dokumentu (*ObjectId*)
- *people\_id* – identifikátor (*ObjectId*) dokumentu *people*, kterému obraz náleží
- *url* – URL adresa, ze které byl obraz stažen nebo hodnota *null*
- *path* – cesta k obrazu v souborovém systému
- *roi* – objekt specifikující detekovanou obdélníkovou oblast v obraze obsahující obličej pomocí atributů objektu (*x*, *y*, *width*, *height*)
- *landMarks* – pole 68 bodů specifikující landmarky obličeje (viz obr. 10)

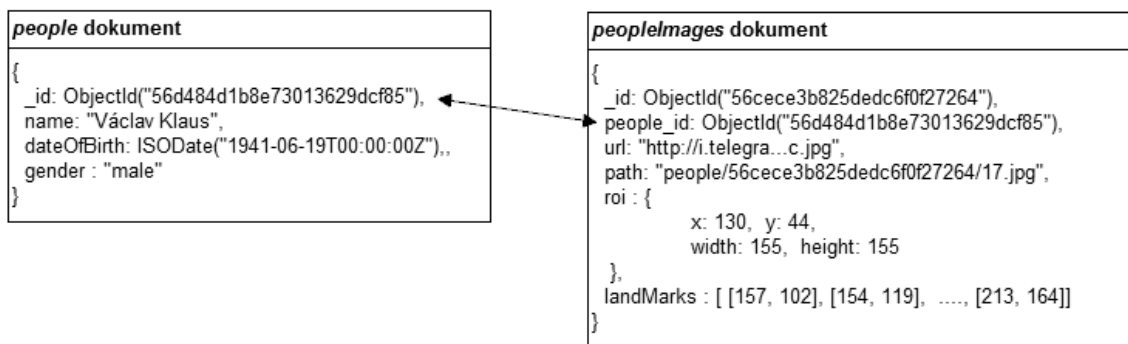
#### 4.1.3 Kolekce *users*

Kolekce *users* uchovává přihlašovací informace o administrátorech systému, tedy o uživateli, kteří mají možnost spravovat systémem rozpoznávané osoby. Jednotlivé dokumenty kolekce mají následující atributy:



- `_id` – identifikátor dokumentu (*ObjectId*)
- `username` – uživatelské jméno
- `password` – zašifrované heslo uživatele šifrou SHA-256

Na následujícím příkladu (obr. 12) je znázorněno využití obou datových modelů MongoDB v databázi *fapi*. Normalizovaný datový model je použit pro vazbu mezi dokumenty kolekcí *people* a *peopleImages*. Pro vytvoření této vazby je využit referující atribut `people_id` kolekce *peopleImages*, který odkazuje na příslušnou osobu v kolekci *people*. Vkládající datový model je využit pro uložení atributů `roi` a `landMarks` v dokumentech kolekce *peopleImages*.



Obr. 12: Ukázka využití datových modelů v systémové databázi

## 4.2 Aplikace webového serveru – *backend*

Aplikace webového serveru je založena na platformě Node.js a naprogramována v jazyku JavaScript. Poskytuje celému systému, jak bylo zmíněno v kapitole 2.2, dvě důležité funkcionality. Poskytuje webové API, které zastává funkci systémového *backendu*. V druhé řadě klientovi dodává statickou HTML5 webovou aplikaci (*frontend*), která klientovi poskytuje grafické rozhraní a funkce pro komunikaci s *backendem*.

Node.js aplikace webového serveru byla nazvána *fapi*. Z tohoto důvodu byla aplikace umístěna do stejnojmenného hlavního adresáře *fapi*, kde byly pro aplikaci vytvořeny následující podadresáře:

- *include* – adresář obsahující pomocné vytvořené moduly (*config.js* a *functions.js*)
- *public* – adresář obsahující veškerá veřejně dostupná statická data *frontendu* (HTML, JS, CSS, ...) a také adresář s obrazy rozpoznávaných osob
- *node\_modules* – adresář obsahující nainstalované moduly aplikace

Dále byl v hlavním adresáři vytvořen soubor *package.json*, který obsahuje důležité informace o aplikaci a o jejích potřebných závislostech. Díky tomuto souboru je např. možné provést instalaci všech závislostí pomocí jediného příkazu balíčkovacího manažeru NPM (*npm install .*).

Následně byl vytvořen hlavní spouštěcí skript – *app.js*. V tomto skriptu je nadefinována veškerá funkcionální aplikace.

#### 4.2.1 Popis funkcionality Node.js aplikace

Po spuštění aplikace *app.js* je provedeno následující. Nejdříve je do příslušných proměnných načteno 19 modulů pomocí příkazu *require*. První dva moduly jsou z podadresáře *include*, a to: *config* a *functions*. *Config* obsahuje konfigurační parametry aplikace jako přihlašovací údaje k databázi, port, na kterém běží HTTP server nebo cestu k adresáři *public*. Modul *functions* obsahuje pomocné funkce např. pro zajištění, že adresář pro ukládání fotek existuje, nebo pro ověření, zdali je uživatel přihlášen. Moduly *image-type* a *validator* jsou pro možnost kontroly a validace vstupů od klienta. Další moduly *fs* a *path* obsahují metody pro práci se souborovým systémem. Modul *opencv* zase umožňuje práci s obrazem a modul *sync* dává aplikaci možnost synchronního zpracování asynchronní metody. Modul *zerorpc* obsahuje metody pro připojení k serveru RPC serveru (ZeroRPC). Modul *mongodb* umožňuje práci s databází. Moduly *express*, *http* a *express-session* a *body-parser* jsou určeny pro vytvoření HTTP serveru a zpracování příchozích požadavků. Modul *request* umožňuje vytvořit webový požadavek na jiný server (např. stáhnout soubor z URL). Modul *multer* umožňuje zpracovávat příchozí nahrání souboru a modul *crypto* obsahuje metody pro šifrování nebo dešifrování (použito pro zašifrování hesla). Posledními načítanými

moduly jsou *passport* a *passport-local*, ty v aplikaci usnadňují vytvořit přihlášení uživatele.

Po načtení těchto modulů jsou inicializovány další pomocné prostředky a je vytvořen HTTP server, který naslouchá na definovaném portu (80). V dalším kroku je vytvořeno statické směrování do adresáře *public*, díky kterému klient může přistupovat ke statickým souborům poskytované HTML5 aplikace a případně také k obrazům osob. Následně je vytvořeno připojení do databáze a také je vytvořena strategie pro přihlašování uživatelů. Dále je vytvořeno připojení na RPC server, který v této práci poskytuje metody pro rozpoznání obličejů v obraze (viz kapitola 4.3.3). Následně je nadefinována funkcionality jednotlivých HTTP metod webového API viz následující tabulka.

Tabulka 5: Popis metod vytvořeného webového API

Cesta (název)	HTTP metoda	Popis funkcionality
<b>/login</b>	POST	Metoda umožňuje přihlášení do systému pomocí uživatelského jména a hesla.
<b>/logout</b>	GET	Metoda umožňuje odhlášení právě přihlášeného uživatele
<b>/user</b>	GET	Vrací informaci, zdali je uživatel přihlášen či nikoliv. V případě že ano, vrací jeho uživatelské jméno.
<b>/user/password</b>	POST	Umožňuje změnu hesla právě přihlášeného uživatele.
<b>/detection</b>	POST	Metoda ve vstupním obraze zakódovaném Base64 rozpoznává obličeje. Vrací pole objektů kde každý objekt má následující vlastnosti o nalezeném obličej: <ul style="list-style-type: none"> <li><i>x</i> – X souřadnice levého horního rohu detekované oblasti obličeje</li> <li><i>y</i> – Y souřadnice levého horního rohu detekované oblasti obličeje</li> <li><i>width</i> – šířka detekované oblasti obličeje</li> <li><i>height</i> – výška detekované oblasti obličeje</li> <li><i>landmarks</i> – pole 68 bodů predikujících landmarky obličeje (viz obr. 10)</li> <li><i>gender</i> – klasifikované pohlaví</li> </ul>

Cesta (název)	HTTP metoda	Popis funkcionality
		<ul style="list-style-type: none"> <li>• <i>age</i> – klasifikovaný věkový interval</li> <li>• <i>person</i> – objekt s atributy o klasifikované osobě</li> </ul> <p>Obličeje jsou v obraze rozpoznávány pomocí volání metody <i>detection</i> RPC serveru.</p>
<b>/detection/url</b>	POST	Metoda v obraze specifikovaným URL adresou detekuje a klasifikuje obličeje. Vrací pole objektů, kde každý objekt má stejné vlastnosti o obličeji jako v předchozí metodě. Obličeje jsou v obraze opět rozpoznávány pomocí volání metody <i>detection</i> RPC serveru.
<b>/verification/url</b>	POST	Verifikuje, zdali se na dvou obrazech obličejů, které jsou specifikovány URL adresami, nachází stejná osoba. Verifikace v metodě probíhá pomocí volání metody <i>verification</i> RPC serveru.
<b>/reload</b>	GET	Načte obraz z URL adresy a poskytne ho pod doménou serveru, kvůli odstranění problému CORS (Cross-origin resource sharing).
<b>/people</b>	GET	Metoda vrací pole objektů reprezentující osoby, které je možné systémem rozpoznávat. Každý objekt má následující atributy: <ul style="list-style-type: none"> <li>• <i>_id</i> – identifikátor</li> <li>• <i>name</i> – jméno a příjmení osoby</li> <li>• <i>dateofBirth</i> – datum narození osoby (pokud není známo tak <i>null</i>)</li> <li>• <i>gender</i> – pohlaví osoby (pokud není známo tak <i>null</i>)</li> </ul>
<b>/people</b>	POST	Vytvoří v systému novou osobu pro rozpoznání (dokument v kolekci <i>people</i> ). Vstupem jsou data z formuláře obsahující pole pro jméno a příjmení ( <i>name</i> ), datum narození ( <i>date</i> ) a pohlaví osoby ( <i>gender</i> ). V závorkách jsou názvy jednotlivých polí.
<b>/people/:id</b>	GET	Získá informace a obrazy osoby specifikované parametrem <i>:id</i> . Tento parametr je roven atributu <i>_id</i> osoby v kolekci <i>people</i> .

Cesta (název)	HTTP metoda	Popis funkcionality
/people/image	GET	Metoda vrací informaci o přiřazení obrazu specifikovaného URL nějakým systémem rozpoznávaným osobě. V případě, že obraz je přiřazen nějaké z osob, vrací v atributu <i>people_id</i> identifikátor osoby z kolekce <i>people</i> . Jinak vrací <i>null</i> .
/people/:id/image/url	POST	Uloží obraz specifikovaný URL adresou a vytvoří nový dokument v kolekci <i>peopleImages</i> . Referující atribut <i>people_id</i> nastaví na <i>:id</i> .
/people/:id/image/upload	POST	Uloží formulářem nahrávaný obraz a vytvoří nový dokument v kolekci <i>peopleImages</i> . Referující atribut <i>people_id</i> nastaví na <i>:id</i> .
/people/:id/image/:iid	DELETE	Smaže konkrétní obraz a dokument z kolekce <i>peopleImages</i> . Parametr <i>:id</i> je <i>_id</i> z kolekce <i>people</i> a parametr <i>:iid</i> je <i>_id</i> dokumentu v kolekci <i>peopleImages</i> .

**Tučně** jsou v tabulce zvýrazněny ty metody, které jsou přístupné bez přihlášení uživatele. Ostatní metody jsou přístupné až po úspěšném přihlášení uživatele pomocí metody */login*. S výjimkou metody */reload* všechny vracejí odpověď ve formátu JSON tak, aby byly jednoduše zpracovatelné HTML5 aplikací. Vstupní data každé metody jsou nejdříve validována a v případě špatně zadaných dat je vrácena chyba ve formátu JSON. Metody */people/:id/image/url* a */people/:id/image/upload* ukládají obrazy obličejů do adresáře *public/people/{\_id osoby}/*.

#### 4.2.2 Spuštění aplikace webového serveru

Spuštění aplikace webového serveru probíhá následovně. Nejdříve je spuštěn terminál nebo příkazový řádek v hlavním adresáři aplikace *fapi*. Dále je aplikace spuštěna příkazem: „*node .*“.

### 4.3 Metody rozpoznání obrazu – RPC server

Pro účely rozpoznání obrazu – detekce, klasifikace a verifikace byl v jazyce Python naprogramován skript vytvářející RPC server, se kterým dokáže komunikovat aplikace webového serveru. Skript RPC serveru byl naprogramován do souboru *my\_detector.py*. Pro vytvoření RPC serveru je využito knihovny ZeroRPC.

Na začátku skriptu jsou provedeny importy použitých knihoven jako jsou: *zerorpc*, *dlib*, *sklearn* apod.

#### 4.3.1 Třída *CORE*

Následně je nadefinována třída *CORE* obsahující 5 metod. Metoda *init\_gender\_recognition* slouží pro vytvoření prostoru PCA a natrénování klasifikátoru pro rozpoznání pohlaví. Jako klasifikátor byl použit lineární SVM, který dosáhl úspěšnosti 79,75 % (viz kapitola 3.5.2). Metoda *init\_ages\_recognition* slouží pro vytvoření prostoru PCA a natrénování klasifikátoru pro rozpoznání věkových intervalů. Byl použit lineární SVM, který dosáhl úspěšnosti 34,48 % (viz 3.6.4). Metoda *init\_people\_recognition* slouží pro vytvoření PCA prostoru a natrénování klasifikátoru osob. Tato metoda využívá testovaného lineárního SVM z kapitoly 3.7.1. Pro natrénování ale využívá systémový dataset (viz kapitola 4.3.5). Metoda *init\_verification* inicializuje klasifikátor pro verifikaci osob. Metoda implementuje přesnou podobu klasifikátoru z kapitoly 3.8. Metoda *frontalize* umožňuje vytvořit frontalizovanou normu obrazů obličeje (viz kapitola 3.4). V konstruktoru této třídy jsou volány všechny 4 inicializační metody a při vytvoření instance třídy *CORE* jsou tedy inicializovány všechny potřebné klasifikátory.

#### 4.3.2 Třída *API*

*API* je další definovanou třídou ve skriptu. Tato třída obsahuje metody, které bude RPC server poskytovat. Obsahuje tedy pouze metody *detection*, *verification* a konstruktor. V konstruktoru třídy je vytvořena instance třídy *CORE*, která je uložena do proměnné *self.CORE* tak, aby bylo možné využívat její klasifikátory a metodu *frontalize*.

### **Metoda *detection***

V metodě *detection* jsou ve vstupním obraze pomocí detekce HOG detekovány všechny obličeje (za pomoci knihovny Dlib, viz kapitola 1.2.2). Pro každý detekovaný obličej jsou získány landmarky, následně je vytvořena jeho frontalizovaná norma, která je po transformaci do příslušného prostoru PCA vstupem klasifikátoru pohlaví, věku a osob. Metoda *detection* nakonec vrací ve formátu JSON pole objektů popisující jednotlivé detekované obličeje s následujícími atributy:

- *rect* – pole obsahující souřadnice obdélníku opisující detekovaný obličej
- *landmarks* – pole 68 bodů predikujících landmarky obličeje
- *gender* – predikované pohlaví osoby
- *age* – predikovaný věkový interval
- *people\_id* – *\_id* predikované osoby

### **Metoda *verification***

Metodě *verification* jsou vstupem dva obrazy. V obou obrazech jsou nejprve detekovány obličeje pomocí detekce HOG knihovny Dlib. Pokud na některém z obrazů detektor detekuje více či méně než jeden obličej, metoda *verification* okamžitě vrací hodnotu *None*. Pokud se ale v každém z nich nachází právě jeden obličej, algoritmus pokračuje predikcí landmarků a převedením obrazů na frontalizovanou normu. Následně jsou oba obrazy transformovány do prostoru příznaků LARK a dále projektovány do prostoru PCA a normovány. Celý postup je popsán v kapitole 1.4.1. Následuje vytvoření absolutního rozdílu mezi oběma získanými vektory (podle vzorce 1.4). Takto vzniklý rozdíl je vstupem klasifikátoru verifikace. V případě, že klasifikátor predikuje, že se jedná o stejné osoby, tak metoda *verification* vrací hodnotu 1, v opačném případě vrací hodnotu 0.

### **4.3.3 Vytvoření RPC serveru**

Po definici tříd je následujícím kódem vytvořen a spuštěn ZeroRPC server na portu 4242, který přebírá metody třídy *API*.

```
s = zerorpc.Server(API(), pool_size=2)
s.bind("tcp://0.0.0.0:4242")
s.run()
```

#### 4.3.4 Spuštění skriptu RPC serveru

Spuštění Python skriptu poté probíhá následovně. Nejdříve je spuštěn terminál nebo příkazový řádek v adresáři obsahujícím skript *my\_detector.py* a potřebné knihovny. Následujícím příkazem je spuštěn skript a tedy ZeroRPC server:

```
python my_detector.py
```

#### 4.3.5 Vytváření systémového datasetu

Systémový dataset (*dataset\_people\_system.ds*) je dataset vytvořený ze všech obrazů osob, které se v systému nachází. Obrazy osob jsou uloženy v adresáři *public/people/*. Systémový dataset je vytvářen pomocí hromadného předzpracování obrazů z kapitoly 3.4.1 pomocí skriptu *dataset\_improved.py*. Skript je nastaven tak, aby vytvářený dataset měl název *dataset\_people\_system.ds* a byl vytvořen pro frontalizovanou normu obrazů. Tento skript byl automaticky spuštěn jednou denně pomocí nástroje pro plánování úloh Cron.

### 4.4 HTML 5 webová aplikace – *frontend*

Tato aplikace poskytuje koncovému uživateli grafické rozhraní, díky kterému může uživatel pohodlně využívat metod webového API poskytovaných *backendem*. Tato aplikace je uživateli dostupná přes libovolný webový prohlížeč na hlavní URL webového serveru (/) a má následující adresářovou strukturu:

- *index.html* – hlavní HTML spouštěcí soubor
- *css* – obsahuje soubory kaskádových stylů CSS definujících vzhled rozhraní
- *js* – obsahuje všechny JavaScript knihovny a soubor *app.js* s kódem aplikace
- *fonts* – obsahuje pomocné fonty
- *images* – obsahuje pomocné obrazy aplikace (ikony)



#### 4.4.1 Hlavní spouštěcí soubor *index.html*

Tento soubor je vstupním bodem celé aplikace. Sekce *head* tohoto souboru obsahuje nejdříve definice o kódování a škálovatelnosti aplikace (kvůli responzivitě). Dále obsahuje veškeré definice CSS stylů, které zajišťují výsledný vzhled rozhraní.

V sekci *body* je obsažen statický HTML kód všech obrazovek aplikace a formulářů. Elementy *div* obsahující obrazovky jednotlivých možností aplikace jsou ve výchozím stavu skryty pomocí CSS a zobrazovány JavaScriptem až v případě potřeby. Na konci sekce *body* jsou umístěny importy knihoven jazyka JavaScript:

- jQuery – rychlá a obsáhlá knihovna, která usnadňuje práci s jazykem JavaScript
- jQuery Canvas – využita pro zjednodušení práce s HTML5 elementem *canvas*
- jQuery BBQ – užitečná knihovna pro možnost obsluhy historie prohlížení a práci s *hash* částí URL adresy
- jQuery Double Tap – knihovna přidává možnost využití události dvojitého doteku na dotykových displejích
- Bootstrap – knihovna frameworku Bootstrap, která přináší rozšiřující možnosti jako dialogové okno apod.
- Datepicker for Bootstrap – u formulářového prvku pro datum přidává možnost výběru data za pomoci kalendáře
- Bootbox.js – pro jednoduché vytváření Bootstrap dialogových oken

Naposledy je naimportován hlavní soubor aplikace – *app.js*.

#### 4.4.2 Soubor aplikace *app.js*

Aplikační soubor *app.js* je naprogramován v jazyce JavaScript a obsahuje logiku řízení celé aplikace. Základem této aplikace je hlavní anonymní funkce, ve které je nadefinováno 13 proměnných a 27 anonymních funkcí. Většina z těchto funkcí jsou takzvané *callbacky*, což znamená, že jsou použity jako funkce, které se provedou při vzniku nějaké události, např. klepnutí na tlačítko nebo odeslání formuláře. Jednou

z těchto *callbackových* funkcí je *app.init*, která je volána ve chvíli, kdy je celý dokument zcela načten. Tato funkce má za úkol inicializovat celou aplikaci:

- načíst důležité prvky HTML dokumentu do proměnných,
- zjistit, zda je uživatel přihlášen, a podle toho mu umožnit správné možnosti rozhraní,
- zobrazit správnou obrazovku aplikace a přiřadit *callback* funkce událostem.

Dalším typem funkcí definovaných v hlavní anonymní funkci jsou funkce pomocné. Tyto jsou definovány kvůli vyšší přehlednosti kódu a patří mezi ně např. *app.alert*, která uživateli na určitou dobu zobrazí hlášku. Některé funkce např. přihlášení, rozpoznání obličejů v obraze apod. využívají komunikace s webovým API, které poskytuje webový server systému (*backend*). Tato komunikace probíhá pomocí volání AJAX, která probíhají asynchronně na pozadí aplikace a dovolují tak uživateli pracovat s aplikací i při jejich průběhu. Průběh AJAX volání je v aplikaci signalizován pomocí otáčivého kolečka, které je zobrazeno uprostřed obrazovky. Po úspěšném či neúspěšném dokončení požadavku volání jsou *backendem* vrácena data a je proveden určitý *callback* specifikovaný anonymní funkcí. Na následujícím příkladu je AJAX volání pro přihlášení uživatele. Za znaky „//“ je uveden komentář jednotlivých řádků.

```
$.ajax({
  type: "POST", //definice HTTP metody
  url: "/login", //definice cesty (názevu) API metody
  data: $(e.target).serialize(), //odesílaná data
  success: function (data) { //callback funkce
v případě úspěšného volání
    app.alert('success', "You have been successfully
logged in.", 1500); //zobrazení hlášky o úspěšném
přihlášení
    app.login(data.username); //zavolání funkce pro
obsluhu přihlášení
  },
  error: function (xhr) { //callback funkce v případě
neúspěchu volání
    app.alert('danger', xhr.responseJSON.error, 1500);
//zobrazení hlášky o neúspěšném přihlášení
  }
});
```

### 4.4.3 Využití elementu *canvas*

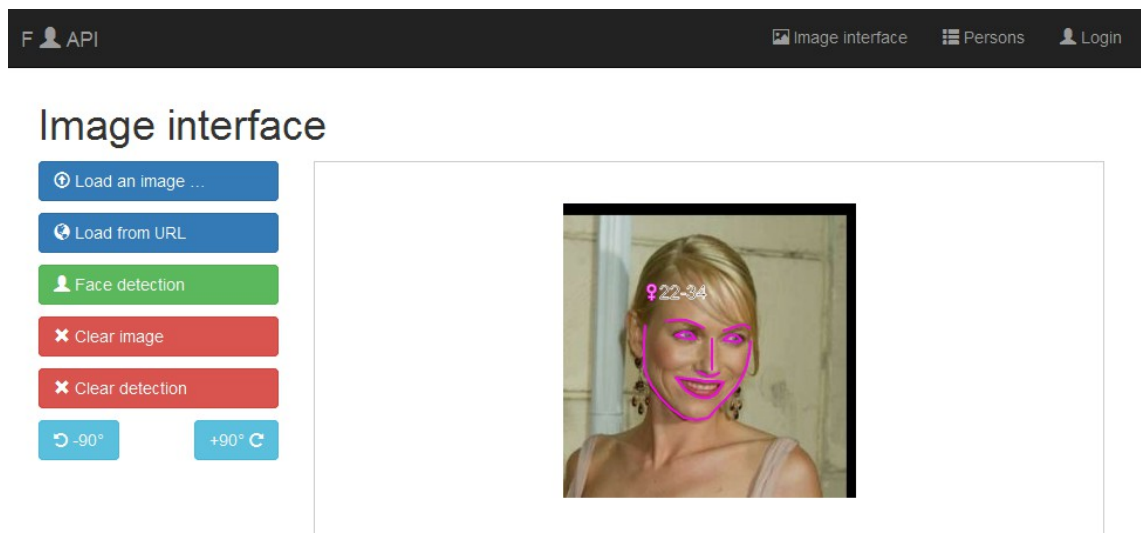
Ve webovém rozhraní je pro vykreslování rozpoznávaného obrazu využíváno HTML5 elementu *canvas* (plátno), který v HTML5 umožňuje dynamické kreslení a práci obrazem. Tento element umožňuje do obrazu přidávat vrstvy, vykreslovat čáry, obrazy, text apod. Kreslení ve vrstvách je v aplikaci využito po úspěšném rozpoznání obličejů viz kapitola 4.4.4.

### 4.4.4 Možnosti webové aplikace

Webová aplikace nabízí dvě kategorie možností, a to – možnosti přístupné bez přihlášení uživatele (veřejně) a možnosti přístupné pouze po přihlášení.

#### Možnosti nepřihlášeného uživatele

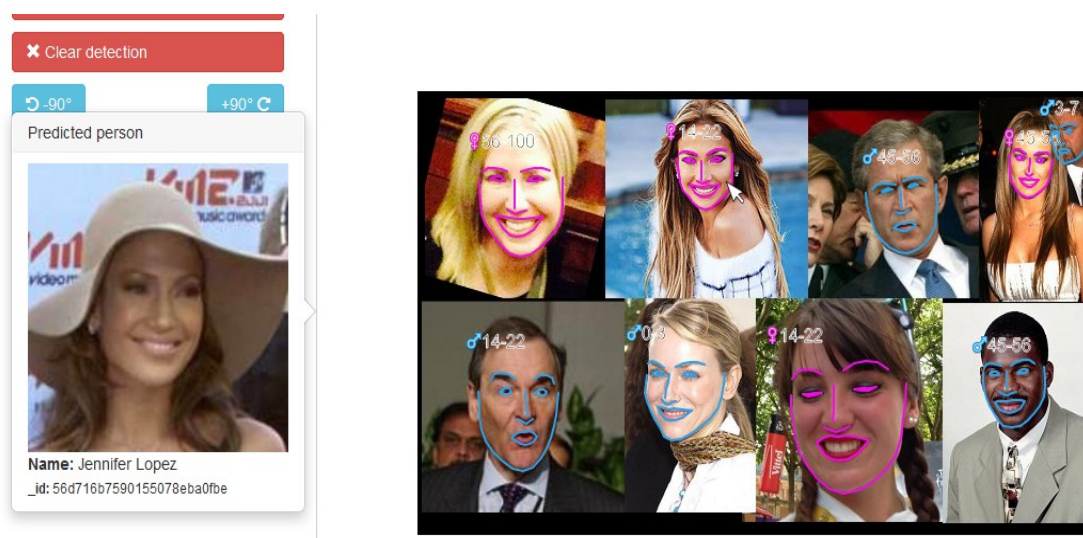
Bez přihlášení má uživatel v hlavním menu zobrazeny 3 položky: *Image interface*, *Persons* a *Login*. Po klepnutí na příslušnou položku je zobrazena příslušná obrazovka či formulář. Výchozí obrazovkou aplikace je *Image interface* viz následující obrázek.



Obr. 13: Základní rozhraní HTML5 aplikace s rozpoznáním obrazu obličeje

V této obrazovce může uživatel pomocí tlačítek *Load an image* a *Load from URL* načíst obraz typu JPEG ze souboru v počítači resp. z URL adresy. Po úspěšném načtení je obraz zobrazen v *canvasu* umístěném v pravé části obrazovky (v případě verze pro chytré telefony je *canvas* umístěn níže pod tlačítky). Následně tlačítky *90°* a *-90°* může uživatel otáčet obrazem v *canvasu* o  $90^\circ$  doprava či doleva. Pomocí tlačítka *Clear image* je celý obsah *canvasu* nenávratně smazán. Zelené tlačítko *Face detection* je

určeno pro rozpoznání obličejů v obrazu z *canvasu*, tedy pro detekci a klasifikaci pohlaví, věkového intervalu a rozpoznávaných osob. Po stisku tlačítka je obraz z *canvasu* rozpoznán metodou webového API a po dokončení požadavku je v *canvasu* u každého rozpoznávaného obličeje vykreslena v nové vrstvě: ikona symbolizující pohlaví, textový popis věkového intervalu a predikované landmarky. Tyto landmarky jsou po skupinách spojeny křivkou a vykresleny růžovou (žena) nebo modrou (muž) barvou. Po najetí myši (příp. doteku na chytrém telefonu) na oblast rozpoznávaného obličeje je zobrazen rámeček zobrazující klasifikovanou osobu. Ukázka výsledku rozpoznání obličejů aplikací je na obr. 14.



Obr. 14: Ukázka rozpoznání obličejů HTML5 aplikací

Poslední tlačítko *Clear detection* je určeno k odstranění vrstvy zobrazující detekci, načtený obraz v *canvasu* zůstane.

Další obrazovkou aplikace je *Persons*, v té je zobrazen abecedně seřazený seznam rozpoznávaných osob v systému viz příloha 15.

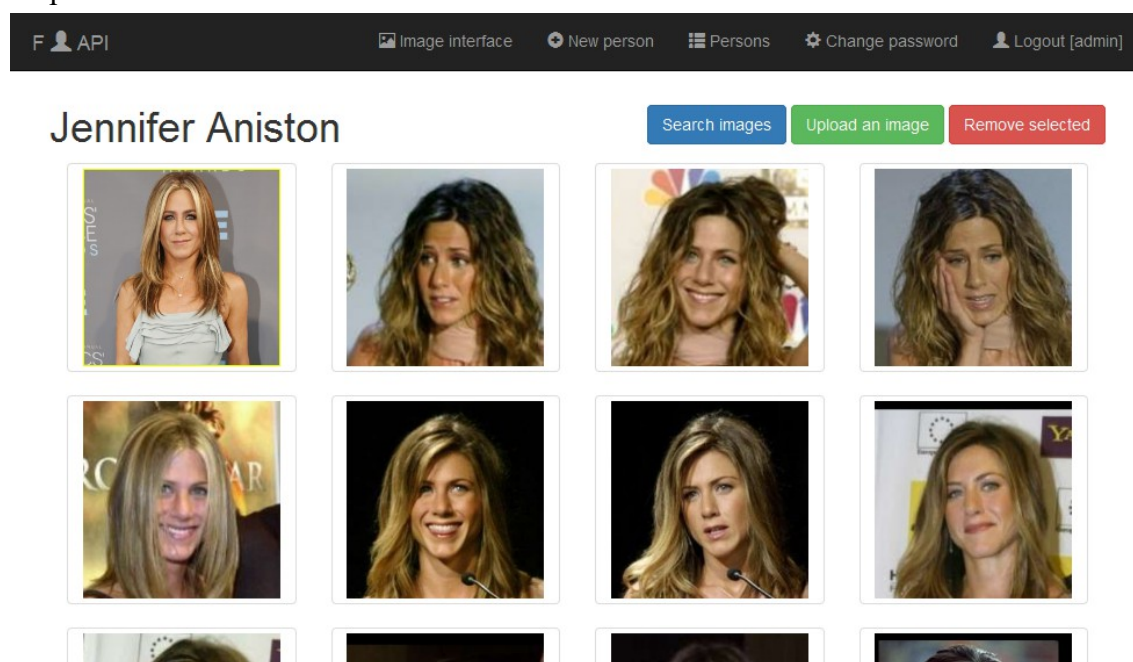
Stiskem položky *Login* v menu je zobrazen formulář pro přihlášení v dialogovém okně. Ukázka dialogového okna přihlášení je zobrazena v příloze 16.

### **Možnosti přihlášeného uživatele**

Přihlášený uživatel má v hlavním menu zobrazeny navíc dvě položky – *Change password* a *New person*. Položka *Login* je nahrazena položkou *Logout* se jménem přihlášeného uživatele v hranatých závorkách. Stiskem této položky dojde k odhlášení

uživatel. Klepnutím na položku *Change password* je v dialogovém okně zobrazen formulář pro změnu hesla. Po správném vyplnění prvků a odeslání formuláře je uživateli zobrazena hláška o úspěšné změně přihlašovacího hesla. Vzhled formuláře je k nahlédnutí v příloze 18. Na obrazovce *New person* se nachází formulář pro vytvoření nové rozpoznávané osoby v systému. Pro vytvoření osoby je nutné zadat její jméno, datum narození a pohlaví. Po odeslání vyplněného formuláře je zobrazena hláška o vytvoření nové osoby (obrazovka viz příloha 17).

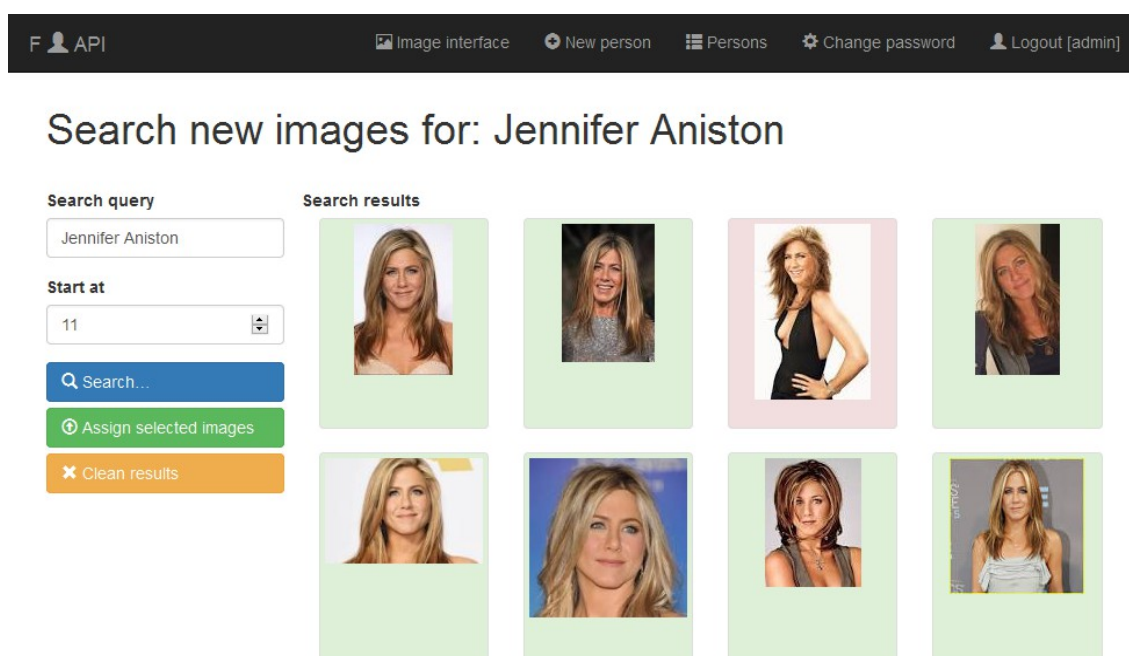
Na obrazovce *Persons* má přihlášený uživatel možnost klepnutím na jméno osoby zobrazit její detail. V tomto detailu (viz obr. 15) jsou zobrazeny obrazy osoby a další možnosti, které lze pro osobu provádět jsou specifikovány barevnými tlačítky. Zelené tlačítko *Upload an image* zobrazí dialog pro výběr souboru JPEG z počítače. Po výběru je soubor nahrán na *backend* metodou webového API a po úspěšném zpracování je zobrazena hláška o přiřazení obrazu dané osobě. Pomocí červeného tlačítka *Remove selected*, dojde ke smazání předem označených obrazů. Tyto obrazy lze označit pomocí klepnutí.



Obr. 15: Ukázka obrazovky detailu osoby

Tlačítkem *Search images* je zobrazena obrazovka (viz obr. 16), která umožňuje pomocí Google JSON Custom Search API chytrě vyhledávat obrazy pro osobu na internetu. Nejdříve je ve formuláři specifikován hledaný dotaz (nejčastěji jméno osoby)

a index prvního výsledku pro vrácení (ve výchozím stavu 1). Následně je pomocí modrého tlačítka *Search..* prohledán internet pro 10 následujících výsledků po specifikovaném indexu. Z těchto výsledků jsou použitelné obrazy zobrazeny v pravé části rozhraní. Následně je možné obrazy označit (zelená) či odznačit (červená) klepnutím na příslušný obraz. Dvojitým klepnutím na obraz je tento vybrán jako vzor a následně je verifikován ke všem ostatním zobrazeným obrazům. V případě úspěšné verifikace je obraz automaticky označen. Tlačítkem *Assign selected images* jsou vybrané obrazy přiřazeny osobě a nahrány do systému. Klepnutím na tlačítko *Clean results* jsou výsledky vyhledávání smazány a index vyhledávání je nastaven na číslo 1.



Obr. 16: Ukázka obrazovky HTML5 aplikace pro vyhledávání obrazů osoby na internetu

## 5 Závěr

V rámci této diplomové práce byly navrženy a otestovány metody detekce, klasifikace a verifikace obličejů v obraze. K otestování byly postupně použity tři různé obrazové databáze obličejů, kde obrazy z těchto databází byly zachyceny v reálném životě za různých podmínek. Pro detekci byla otestována metoda na základě Haar příznaků s využitím boostovací metody AdaBoost, která dosáhla přesnosti **93,70 %**. Druhou otestovanou metodou byla detekce využívající Histogramů orientovaných gradientů, která dosáhla o něco vyšší přesnosti **96,30 %**. V úloze klasifikace obličejů byla nejdříve vytvořena normalizovaná a frontalizovaná norma obrazových dat a následně byla využita metoda PCA pro redukci dimenze dat. Dále byly provedeny testy úspěšnosti klasifikace pohlaví, věkových intervalů a rozpoznávaných osob. Pro klasifikaci pohlaví byly na obou obrazových normách vyzkoušeny metody Euklidovské vzdálenosti, Mahalanobisovy vzdálenosti, GMM a lineárního SVM. Nejvyšší úspěšnosti **79,75 %** bylo dosaženo při využití frontalizovaných dat a metody lineární SVM. Dále bylo zjištěno, že frontalizovaná norma dat dosahuje pro všechny metody vyšší úspěšnosti, proto byla v dalších testech uvažována pouze tato norma. V klasifikaci věkových intervalů byly testovány stejné metody jako v případě klasifikace věku a nejvyšší dosažené úspěšnosti **34,5 %** opět dosáhl lineární SVM. U klasifikace rozpoznávaných osob již byla testována pouze metoda lineární SVM a dosáhla úspěšnosti až **86,62 %**. Pro úlohu verifikace bylo využito příznaků LARK, redukce dimenze PCA a následně převodu na klasifikační úlohu řešenou lineárním SVM. Toto řešení přineslo úspěšnost verifikace **77,74 %**.

Následně byl navržen moderní webový systém pracující s technologiemi HTML5, Node.js, MongoDB atp. Tento webový systém uživatelům umožňuje rozpoznávat vlastní obrazy s obličejí osob a využívá přitom výše zmíněných metod, které v experimentech dosáhly nejvyšší úspěšnosti. Po přihlášení do webového systému má uživatel možnost spravovat systémem rozpoznávané osoby a obrazy jejich obličejů.

## Seznam použité literatury

- [1] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. Image processing, analysis, and machine vision. Fourth edition. Stamford, CT: Cengage Learning, 2015. ISBN 978-1-133-59360-7.
- [2] Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research [online]. 2009, 2009(10), 1755-1758 [cit. 2016-03-14]. Dostupné z: <http://jmlr.csail.mit.edu/papers/volume10/king09a/king09a.pdf>
- [3] Pedregosa et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research [online]. 2011(12), 2825-2830 [cit. 2016-03-16]. Dostupné z: <http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>
- [4] VIOLA, Paul a Michael J. JONES. Robust Real-Time Face Detection. International Journal of Computer Vision [online]. 2004, 2004(57 (2)), 137–154 [cit. 2016-03-16]. Dostupné z: <http://link.springer.com/content/pdf/10.1023%2FB%3AVISI.0000013087.49260.fb.pdf>
- [5] SEO, Naotoshi. OpenCV haartraining (Rapid Object Detection With A Cascade of Boosted Classifiers Based on Haar-like Features). [online]. [cit. 2016-03-17]. Dostupné z: <http://note.sonots.com/SciSoftware/haartraining.html#n738280e>
- [6] DALAL, N. a B. TRIGGS. Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) [online]. IEEE, 2005, , 886-893 [cit. 2016-03-19]. DOI: 10.1109/CVPR.2005.177. ISBN 0-7695-2372-2. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1467360>
- [7] TURK, Matthew a Alex PENTLAND. Eigenfaces for Recognition. Journal of Cognitive Neuroscience [online]. 1991, 3(1), 71-86 [cit. 2016-03-19]. DOI: 10.1162/jocn.1991.3.1.71. ISSN 0898-929x. Dostupné z: <http://www.mitpressjournals.org/doi/abs/10.1162/jocn.1991.3.1.71>
- [8] CHRISTENSEN, H. I. a P. Jonathon PHILLIPS. Empirical evaluation methods in computer vision. River Edge, N.J.: World Scientific, c2002. Series in machine perception and artificial intelligence, v. 50.
- [9] BEDÁŇOVÁ, Iveta. Pravděpodobnostní rozdělení spojité náhodné veličiny pro základní soubory [online]. [cit. 2016-03-19]. Dostupné z: <http://cit.vfu.cz/statpotr/POTR/Teorie/Predn2/rozdelZS.htm>
- [10] SEO, Hae Jong a Peyman MILANFAR. Face Verification Using the LARK Representation. IEEE Transactions on Information Forensics and Security [online]. 2011, 6(4), 1275-1286 [cit. 2016-03-25]. DOI: 10.1109/TIFS.2011.2159205. ISSN



1556-6013. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5872024>

[11] WEI-PING, Zhu, Li MING-XIN a Chen HUAN. Using MongoDB to implement textbook management system instead of MySQL. 2011 IEEE 3rd International Conference on Communication Software and Networks [online]. IEEE, 2011, , 303-305 [cit. 2016-04-05]. DOI: 10.1109/ICCSN.2011.6013720. ISBN 978-1-61284-485-5. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6013720>

[12] Vidit Jain a Erik Learned-Miller. Fddb: A Benchmark for Face Detection in Unconstrained Settings. Technical Report UM-CS-2010-009, Dept. of Computer Science, University of Massachusetts, Amherst. 2010. [cit. 2016-04-09].

[13] EIDINGER, Eran, Roe ENBAR a Tal HASSNER. Age and Gender Estimation of Unfiltered Faces. IEEE Transactions on Information Forensics and Security. 2014, 9(12), 2170-2179 [cit. 2016-04-09]. DOI: 10.1109/TIFS.2014.2359646. ISSN 1556-6013. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6906255>

[14] Gary B. Huang, Manu Ramesh, Tamara Berg a Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49, 2007 [online]. [cit. 2016-04-09]. Dostupné z: <http://vis-www.cs.umass.edu/lfw/>

[15] KAZEMI, Vahid a Josephine SULLIVAN. One millisecond face alignment with an ensemble of regression trees. 2014 IEEE Conference on Computer Vision and Pattern Recognition [online]. IEEE, 2014, , 1867-1874 [cit. 2016-04-16]. DOI: 10.1109/CVPR.2014.241. ISBN 978-1-4799-5118-5. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909637>

[16] SAGONASAGONAS, Christos, Epameinondas ANTONAKOS, Georgios TZIMIROPOULOS, Stefanos ZAFEIRIOU a Maja PANTIC. 300 faces In-the-wild challenge: Database and results. Image and Vision Computing [online]. IEEE, 2016, , - [cit. 2016-04-16]. DOI: 10.1016/j.imavis.2016.01.002. ISBN 978-1-4799-5118-5. ISSN 02628856. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0262885616000147>

[17] HASSNER, Tal, Shai HAREL, Eran PAZ a Roe ENBAR. Effective face frontalization in unconstrained images. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [online]. IEEE, 2015, , 4295-4304 [cit. 2016-04-19]. DOI: 10.1109/CVPR.2015.7299058. ISBN 978-1-4673-6964-0. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7299058>

## Obsah příloženého DVD

- *diplomova\_prace\_2016\_Bumba\_Martin.pdf* – text diplomové práce ve formátu PDF
- *test\_skripty* – adresář obsahuje všechny Python skripty použité pro testování metod (neobsahuje testovací datasety)
- *fapi* – adresář obsahuje:
  - aplikaci webového serveru Node.js (bez nainstalovaných Node.js modulů)
  - soubor *ctime.txt* s instrukcemi pro instalaci a spuštění aplikace
- *rpc\_server* – adresář obsahuje Python skript RPC serveru (*my\_detector.py*) a jeho závislosti

## Přílohy

### Seznam příloh

Příloha 1: Tabulka výsledků klasifikace pohlaví pomocí Euklidovské vzdálenosti v závislosti na počtu PCA komponent (normalizovaná norma).....	66
Příloha 2: Tabulka výsledků klasifikace pohlaví pomocí Mahalanobisovy vzdálenosti v závislosti na počtu PCA komponent (normalizovaná norma).....	67
Příloha 3: Tabulka výsledků klasifikace pohlaví pomocí GMM v závislosti na počtu mixtur (normalizovaná norma).....	68
Příloha 4: Tabulka výsledků klasifikace pohlaví pomocí lineárního SVM v závislosti na počtu PCA komponent (normalizovaná norma).....	69
Příloha 5: Tabulka výsledků klasifikace pohlaví pomocí Euklidovské vzdálenosti v závislosti na počtu PCA komponent (frontalizovaná norma).....	70
Příloha 6: Tabulka výsledků klasifikace pohlaví pomocí Mahalanobisovy vzdálenosti v závislosti na počtu PCA komponent (frontalizovaná norma).....	71
Příloha 7: Tabulka výsledků klasifikace pohlaví pomocí GMM v závislosti na počtu mixtur (frontalizovaná norma).....	72
Příloha 8: Tabulka výsledků klasifikace pohlaví pomocí lineárního SVM v závislosti na počtu PCA komponent (frontalizovaná norma).....	73
Příloha 9: Tabulka výsledků klasifikace věku pomocí Euklidovské vzdálenosti v závislosti na počtu PCA komponent (frontalizovaná norma).....	74
Příloha 10: Tabulka výsledků klasifikace věku pomocí Mahalanobisovy vzdálenosti v závislosti na počtu PCA komponent (frontalizovaná norma).....	75
Příloha 11: Tabulka výsledků klasifikace věku pomocí GMM v závislosti na počtu mixtur (frontalizovaná norma).....	76
Příloha 12: Tabulka výsledků klasifikace věku pomocí lineárního SVM v závislosti na počtu PCA komponent (frontalizovaná norma).....	77
Příloha 13: Tabulka výsledků klasifikace konkrétních osob pomocí lineárního SVM v závislosti na počtu PCA komponent (frontalizovaná norma).....	78
Příloha 14: Tabulka výsledků verifikace osob pomocí lineárního SVM v závislosti na počtu PCA komponent (frontalizovaná norma).....	80
Příloha 15: Ukázka obrazovky Persons HTML5 aplikace.....	81
Příloha 16: Ukázka přihlašovacího dialogu HTML5 aplikace.....	81
Příloha 17: Ukázka obrazovky HTML5 aplikace pro vytvoření nového uživatele.....	81
Příloha 18: Ukázka dialogu pro změnu hesla HTML5 aplikace.....	82

Příloha 1: Tabulka výsledků klasifikace pohlaví pomocí Euklidovské vzdálenosti v závislosti na počtu PCA komponent (normalizovaná norma)

Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	54.375	54.617	51.750	53.145
20	56.500	56.599	55.750	56.171
30	55.500	55.584	54.750	55.164
40	54.875	54.887	54.750	54.819
50	54.625	54.707	53.750	54.224
60	55.000	55.181	53.250	54.198
70	54.875	55.118	52.500	53.777
80	55.625	55.937	53.000	54.429
90	55.500	55.851	52.500	54.124
100	55.375	55.733	52.250	53.935
110	55.875	56.233	53.000	54.569
120	55.125	55.438	52.250	53.797
130	55.500	55.820	52.750	54.242
140	55.625	56.000	52.500	54.194
150	55.875	56.267	52.750	54.452
160	56.250	56.684	53.000	54.780
170	56.125	56.568	52.750	54.592
180	56.000	56.486	52.250	54.286
190	56.000	56.522	52.000	54.167
200	56.250	56.757	52.500	54.545
210	56.500	57.065	52.500	54.688
220	56.500	57.065	52.500	54.688
230	56.250	56.793	52.250	54.427
240	56.375	56.948	52.250	54.498
250	56.125	56.712	51.750	54.118
260	56.125	56.712	51.750	54.118
270	56.250	56.831	52.000	54.308
280	56.250	56.868	51.750	54.188
290	56.250	56.868	51.750	54.188
300	56.625	57.260	52.250	54.641
310	56.625	57.260	52.250	54.641
<b>320</b>	<b>56.750</b>	<b>57.418</b>	<b>52.250</b>	<b>54.712</b>
<b>330</b>	<b>56.750</b>	<b>57.418</b>	<b>52.250</b>	<b>54.712</b>
<b>340</b>	<b>56.750</b>	<b>57.418</b>	<b>52.250</b>	<b>54.712</b>
<b>350</b>	<b>56.750</b>	<b>57.418</b>	<b>52.250</b>	<b>54.712</b>

Příloha 2: Tabulka výsledků klasifikace pohlaví pomocí Mahalanobisovy vzdálenosti v závislosti na počtu PCA komponent (normalizovaná norma)

Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	53.875	52.886	71.000	60.619
20	56.375	55.090	69.000	61.265
30	57.375	55.960	69.250	61.899
40	54.625	53.737	66.500	59.441
50	56.250	54.883	70.250	61.623
60	55.625	54.420	69.250	60.946
70	56.500	54.906	72.750	62.581
80	56.750	55.233	71.250	62.227
90	56.625	55.206	70.250	61.826
100	58.000	56.400	70.500	62.667
110	56.875	55.446	70.000	61.878
120	57.500	55.952	70.500	62.389
130	57.625	56.016	71.000	62.624
140	56.875	55.403	70.500	62.046
150	57.750	56.126	71.000	62.693
160	57.750	56.078	71.500	62.857
170	58.875	57.002	72.250	63.727
180	59.000	56.977	73.500	64.192
190	58.625	56.751	72.500	63.666
200	59.000	57.031	73.000	64.035
210	59.875	57.791	73.250	64.609
220	59.750	57.800	72.250	64.222
230	59.625	57.715	72.000	64.071
<b>240</b>	<b>60.500</b>	<b>58.434</b>	<b>72.750</b>	<b>64.811</b>
250	60.250	58.266	72.250	64.509
260	60.250	58.266	72.250	64.509
270	59.750	57.863	71.750	64.062
280	60.125	58.020	73.250	64.751
290	60.000	57.905	73.250	64.680
300	59.625	57.564	73.250	64.466
310	60.375	58.317	72.750	64.739
320	59.875	57.853	72.750	64.452
330	60.000	57.905	73.250	64.680
340	60.125	58.020	73.250	64.751
350	59.875	57.948	72.000	64.214

Příloha 3: Tabulka výsledků klasifikace pohlaví pomocí GMM v závislosti na počtu mixtur (normalizovaná norma)

Počet Gaussovských rozdělení (mixtur)	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
1	64.375	64.198	65.000	64.596
<b>2</b>	<b>66.125</b>	<b>64.365</b>	<b>72.250</b>	<b>68.080</b>
3	63.875	60.221	81.750	69.353
4	62.500	58.929	82.500	68.750
5	57.375	54.319	92.750	68.513
6	57.000	53.966	95.250	68.897
7	55.625	53.165	94.500	68.047
8	58.000	54.598	95.000	69.343
9	55.000	52.703	97.500	68.421
10	54.375	52.318	98.750	68.398
11	54.375	52.318	98.750	68.398
12	59.625	56.006	89.750	68.972
13	56.125	66.897	24.250	35.596
14	57.250	67.683	27.750	39.362
15	50.375	71.429	1.250	2.457
16	50.000	0.000	0.000	0.000
17	50.000	0.000	0.000	0.000
18	50.000	0.000	0.000	0.000
19	50.000	0.000	0.000	0.000

Příloha 4: Tabulka výsledků klasifikace pohlaví pomocí lineárního SVM v závislosti na počtu PCA komponent (normalizovaná norma)

Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	58.000	57.477	61.500	59.420
20	55.375	55.107	58.000	56.516
30	61.875	60.626	67.750	63.991
40	63.500	62.162	69.000	65.403
50	60.500	60.096	62.500	61.275
60	65.125	63.719	70.250	66.825
70	59.375	58.824	62.500	60.606
80	63.125	62.353	66.250	64.242
90	61.125	61.153	61.000	61.076
100	60.375	59.368	65.750	62.396
110	64.750	64.320	66.250	65.271
120	60.500	59.722	64.500	62.019
130	64.375	63.341	68.250	65.704
140	61.375	60.508	65.500	62.905
<b>150</b>	<b>67.375</b>	<b>66.276</b>	<b>70.750</b>	<b>68.440</b>
160	63.375	63.081	64.500	63.782
170	60.500	59.292	67.000	62.911
180	63.375	62.471	67.000	64.656
190	64.375	62.473	72.000	66.899
200	64.375	63.158	69.000	65.950
210	61.750	61.033	65.000	62.954
220	64.125	63.357	67.000	65.128
230	59.875	59.122	64.000	61.465
240	61.625	61.259	63.250	62.239
250	63.250	62.679	65.500	64.059
260	66.125	65.617	67.750	66.667
270	62.750	61.916	66.250	64.010
280	65.625	64.775	68.500	66.586
290	65.125	64.578	67.000	65.767
300	63.125	62.014	67.750	64.755
310	63.500	62.736	66.500	64.563
320	64.500	63.303	69.000	66.029
330	64.625	63.962	67.000	65.446
340	63.625	63.133	65.500	64.294
350	63.375	63.210	64.000	63.602

Příloha 5: Tabulka výsledků klasifikace pohlaví pomocí Euklidovské vzdálenosti v závislosti na počtu PCA komponent (frontalizovaná norma)

Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	61.125	61.440	59.750	60.583
<b>20</b>	<b>62.375</b>	<b>62.725</b>	<b>61.000</b>	<b>61.850</b>
30	61.250	60.922	62.750	61.823
40	61.750	61.634	62.250	61.940
50	62.250	62.312	62.000	62.155
60	61.125	61.266	60.500	60.881
70	61.875	62.211	60.500	61.343
80	61.375	61.461	61.000	61.230
90	61.125	61.266	60.500	60.881
100	61.500	61.616	61.000	61.307
110	61.750	61.929	61.000	61.461
120	61.500	61.917	59.750	60.814
130	61.125	61.440	59.750	60.583
140	61.875	62.148	60.750	61.441
150	62.125	62.532	60.500	61.499
160	62.250	62.694	60.500	61.578
170	62.250	62.694	60.500	61.578
180	62.000	62.371	60.500	61.421
190	62.000	62.371	60.500	61.421
200	61.875	62.274	60.250	61.245
210	61.875	62.211	60.500	61.343
220	61.875	62.211	60.500	61.343
230	62.125	62.404	61.000	61.694
240	62.125	62.468	60.750	61.597
250	62.000	62.308	60.750	61.519
260	62.000	62.308	60.750	61.519
270	62.250	62.564	61.000	61.772
280	62.000	62.371	60.500	61.421
290	61.750	62.051	60.500	61.266
300	61.625	61.954	60.250	61.090
310	61.750	62.051	60.500	61.266
320	61.625	61.893	60.500	61.188
330	61.750	61.929	61.000	61.461
340	62.125	62.341	61.250	61.791
350	62.125	62.404	61.000	61.694



Příloha 6: Tabulka výsledků klasifikace pohlaví pomocí Mahalanobisovy vzdálenosti v závislosti na počtu PCA komponent (frontalizovaná norma)

Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	59.500	56.507	82.500	67.073
20	59.000	56.691	76.250	65.032
30	58.250	56.371	73.000	63.617
40	59.000	57.087	72.500	63.877
50	59.500	57.540	72.500	64.159
60	58.625	56.886	71.250	63.263
70	59.625	57.594	73.000	64.388
80	59.250	57.283	72.750	64.097
90	59.500	57.631	71.750	63.920
100	60.000	58.130	71.500	64.126
110	60.500	58.468	72.500	64.732
120	60.125	58.182	72.000	64.358
130	60.250	58.300	72.000	64.430
140	60.125	58.182	72.000	64.358
150	60.625	58.621	72.250	64.726
160	60.875	58.824	72.500	64.950
<b>170</b>	<b>61.000</b>	<b>58.871</b>	<b>73.000</b>	<b>65.179</b>
180	59.250	57.460	71.250	63.616
190	60.250	58.333	71.750	64.350
200	59.875	58.144	70.500	63.729
210	60.625	58.656	72.000	64.646
220	60.250	58.367	71.500	64.270
230	60.250	58.402	71.250	64.189
240	60.375	58.487	71.500	64.342
250	60.000	58.130	71.500	64.126
260	59.125	57.374	71.000	63.464
270	58.750	57.056	70.750	63.170
280	59.375	57.606	71.000	63.606
290	59.500	57.661	71.500	63.839
300	59.625	57.778	71.500	63.911
310	59.875	58.012	71.500	64.054
320	59.750	58.025	70.500	63.657
330	59.750	57.959	71.000	63.820
340	60.000	58.230	70.750	63.883
350	59.750	58.058	70.250	63.575

Příloha 7: Tabulka výsledků klasifikace pohlaví pomocí GMM v závislosti na počtu mixtur (frontalizovaná norma)

Počet Gaussovských rozdělení (mixtur)	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
1	69.750	71.944	64.750	68.158
2	70.250	70.769	69.000	69.873
3	70.250	68.080	76.250	71.934
<b>4</b>	<b>70.625</b>	<b>68.456</b>	<b>76.500</b>	<b>72.255</b>
5	67.375	63.708	80.750	71.224
6	68.625	66.091	76.500	70.915
7	62.125	57.990	88.000	69.911
8	60.000	56.098	92.000	69.697
9	57.375	54.282	93.500	68.687
10	54.250	52.267	98.000	68.174
11	54.625	52.497	97.250	68.186
12	53.625	51.936	97.250	67.711
13	54.250	52.310	96.250	67.782
14	62.625	57.953	92.000	71.111
15	59.125	55.391	93.750	69.638
16	58.750	55.072	95.000	69.725
17	56.250	79.070	17.000	27.984
18	63.875	65.460	58.750	61.924
19	62.875	62.410	64.750	63.558

Příloha 8: Tabulka výsledků klasifikace pohlaví pomocí lineárního SVM v závislosti na počtu PCA komponent (frontalizovaná norma)

Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	60.250	58.952	67.500	62.937
20	67.750	65.991	73.250	69.431
30	69.750	67.955	74.750	71.190
40	71.000	70.192	73.000	71.569
50	68.750	67.943	71.000	69.438
60	68.625	68.486	69.000	68.742
70	72.875	72.263	74.250	73.243
80	75.500	75.888	74.750	75.315
90	74.375	75.064	73.000	74.018
100	74.750	75.781	72.750	74.235
110	75.625	75.434	76.000	75.716
120	75.625	76.762	73.500	75.096
130	75.250	75.635	74.500	75.063
140	76.000	77.083	74.000	75.510
150	72.250	73.545	69.500	71.465
160	76.875	77.215	76.250	76.730
170	74.000	74.870	72.250	73.537
180	75.375	76.501	73.250	74.840
190	75.875	76.071	75.500	75.784
200	74.500	75.521	72.500	73.980
210	75.625	76.081	74.750	75.410
220	75.250	75.505	74.750	75.126
230	75.375	76.501	73.250	74.840
240	77.375	78.590	75.250	76.884
250	74.875	75.578	73.500	74.525
260	78.125	79.070	76.500	77.764
<b>270</b>	<b>79.750</b>	<b>80.051</b>	<b>79.250</b>	<b>79.648</b>
280	73.125	73.779	71.750	72.750
290	76.125	77.285	74.000	75.607
300	76.000	76.804	74.500	75.635
310	76.875	76.942	76.750	76.846
320	78.500	79.231	77.250	78.228
330	73.250	74.603	70.500	72.494
340	74.875	76.253	72.250	74.198
350	73.750	75.538	70.250	72.798

Příloha 9: Tabulka výsledků klasifikace věku pomocí Euklidovské vzdálenosti v závislosti na počtu PCA komponent (frontalizovaná norma)

Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	21.154	21.214	21.154	21.124
20	18.544	19.483	18.544	18.843
30	20.467	21.770	20.467	20.858
40	21.566	22.768	21.566	21.922
50	21.978	23.604	21.978	22.448
60	21.978	23.530	21.978	22.434
70	22.527	24.447	22.527	22.997
80	23.077	25.151	23.077	23.590
90	23.489	25.881	23.489	24.071
100	22.802	25.094	22.802	23.345
110	22.940	25.349	22.940	23.567
120	23.214	25.289	23.214	23.732
130	23.214	25.396	23.214	23.754
140	23.214	25.408	23.214	23.734
150	22.940	25.240	22.940	23.487
160	23.077	25.278	23.077	23.565
170	23.214	25.709	23.214	23.793
180	23.489	25.994	23.489	24.057
190	23.352	25.776	23.352	23.891
<b>200</b>	<b>23.626</b>	<b>25.929</b>	<b>23.626</b>	<b>24.134</b>
210	23.352	25.887	23.352	23.865
220	23.352	25.993	23.352	23.918
<b>230</b>	<b>23.626</b>	<b>26.177</b>	<b>23.626</b>	<b>24.150</b>
240	23.489	26.217	23.489	24.008
250	23.214	25.762	23.214	23.687
260	23.214	25.762	23.214	23.687
270	23.352	26.026	23.352	23.839
280	23.352	26.017	23.352	23.839
290	23.077	25.685	23.077	23.576
300	23.352	26.026	23.352	23.835
<b>310</b>	<b>23.626</b>	<b>26.217</b>	<b>23.626</b>	<b>24.066</b>
320	23.489	26.095	23.489	23.939
330	23.489	26.115	23.489	23.934
340	23.352	26.165	23.352	23.848
350	23.489	26.282	23.489	23.965

Příloha 10: Tabulka výsledků klasifikace věku pomocí Mahalanobisovy vzdálenosti v závislosti na počtu PCA komponent (frontalizovaná norma)

Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	21.429	17.557	21.429	17.542
20	24.176	24.036	24.176	20.916
30	22.527	21.337	22.527	19.358
40	23.626	23.019	23.626	20.382
50	24.313	24.446	24.313	21.525
60	24.313	24.498	24.313	21.415
70	25.000	24.200	25.000	21.837
<b>80</b>	<b>26.099</b>	<b>25.282</b>	<b>26.099</b>	<b>23.063</b>
90	24.863	24.135	24.863	22.062
100	25.687	26.454	25.687	23.001
110	24.725	25.851	24.725	22.103
120	24.588	23.568	24.588	21.669
130	24.725	25.412	24.725	22.156
140	25.000	24.559	25.000	22.313
150	24.588	24.936	24.588	22.215
160	24.038	23.637	24.038	21.437
170	23.764	22.524	23.764	21.092
180	24.588	24.367	24.588	21.867
190	23.901	23.484	23.901	21.090
200	23.626	22.791	23.626	21.015
210	23.489	22.922	23.489	20.745
220	23.764	23.414	23.764	20.990
230	23.901	23.547	23.901	21.223
240	22.940	22.305	22.940	20.310
250	23.214	22.542	23.214	20.596
260	23.764	22.623	23.764	21.114
270	23.901	23.161	23.901	21.213
280	23.489	22.158	23.489	20.841
290	24.038	23.337	24.038	21.411
300	24.038	22.889	24.038	21.473
310	24.176	23.433	24.176	21.661
320	24.313	22.989	24.313	21.600
330	24.038	22.806	24.038	21.383
340	24.451	23.711	24.451	21.935
350	24.588	23.903	24.588	22.062

Příloha 11: Tabulka výsledků klasifikace věku pomocí GMM v závislosti na počtu mixtur (frontalizovaná norma)

Počet Gaussovských rozdělení (mixtur)	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
1	32.280	31.977	32.280	30.261
2	29.121	26.335	29.121	25.315
3	24.313	22.834	24.313	17.063
4	12.500	1.565	12.500	2.781
5	13.049	3.678	13.049	4.362
6	12.088	2.915	12.088	3.984
7	14.835	26.888	14.835	8.988
8	21.291	22.724	21.291	18.525
9	19.231	27.494	19.231	16.197
10	20.192	36.236	20.192	15.406
11	19.505	17.834	19.505	13.030
12	21.841	23.759	21.841	15.702
13	22.527	22.484	22.527	20.781
14	21.566	21.268	21.566	19.792
15	22.253	21.727	22.253	19.890
16	20.467	24.873	20.467	15.661
17	24.725	24.127	24.725	22.162
18	20.742	22.186	20.742	16.636
19	23.352	25.752	23.352	23.503

Příloha 12: Tabulka výsledků klasifikace věku pomocí lineárního SVM v závislosti na počtu PCA komponent (frontalizovaná norma)

Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	25.962	21.207	25.962	21.852
20	25.275	23.416	25.275	22.825
30	16.071	15.074	16.071	15.263
40	28.434	27.147	28.434	26.716
50	28.984	27.517	28.984	27.826
60	26.648	25.066	26.648	25.132
70	28.846	28.990	28.846	28.796
80	32.692	32.898	32.692	31.695
90	30.632	29.409	30.632	29.322
100	26.511	27.050	26.511	25.864
110	28.846	26.549	28.846	26.675
120	31.868	31.552	31.868	31.518
<b>130</b>	<b>34.478</b>	<b>33.823</b>	<b>34.478</b>	<b>32.987</b>
140	32.005	31.761	32.005	31.301
150	23.626	23.007	23.626	22.685
160	32.418	30.311	32.418	30.731
170	31.319	30.668	31.319	30.273
180	33.379	31.379	33.379	32.026
190	30.495	30.038	30.495	29.515
200	33.379	33.260	33.379	32.071
210	29.533	29.471	29.533	28.686
220	30.082	30.051	30.082	29.534
230	25.275	25.917	25.275	25.214
240	28.297	26.508	28.297	26.950
250	28.571	27.652	28.571	27.433
260	27.060	27.061	27.060	26.644
270	26.236	25.538	26.236	25.603
280	29.258	28.338	29.258	28.425
290	29.945	30.874	29.945	29.959
300	29.808	28.646	29.808	28.873
310	30.357	29.177	30.357	29.240
320	32.418	30.957	32.418	31.336
330	29.533	28.378	29.533	28.790
340	29.258	28.515	29.258	28.160
350	28.022	26.864	28.022	27.122

Příloha 13: Tabulka výsledků klasifikace konkrétních osob pomocí lineárního SVM v závislosti na počtu PCA komponent (frontalizovaná norma)

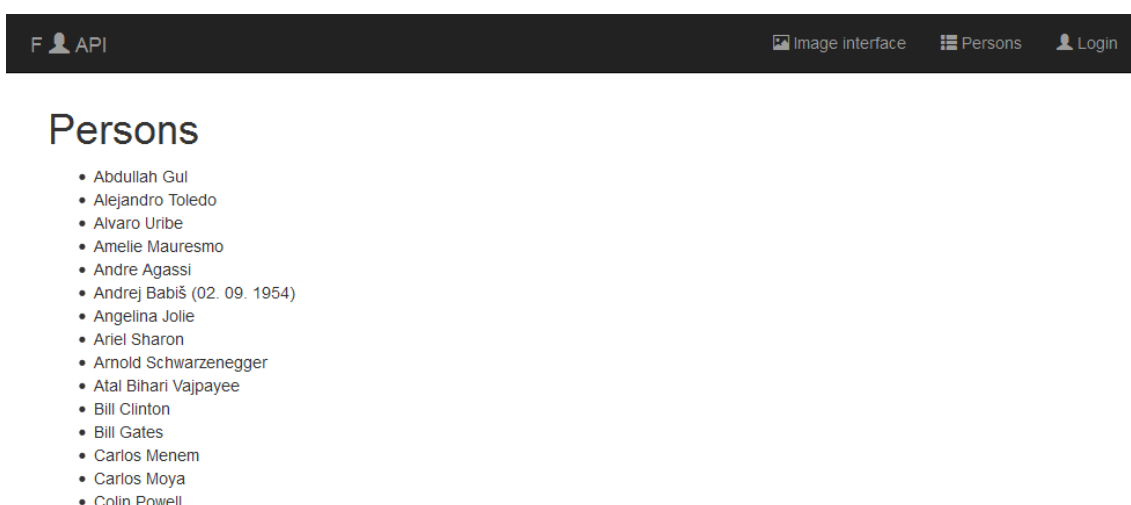
Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	20.412	9.908	11.654	7.972
20	33.105	21.965	21.858	19.160
30	41.509	37.472	29.761	28.402
40	54.717	55.693	43.403	43.495
50	59.005	56.092	47.332	47.084
60	63.636	66.641	53.002	54.570
70	68.611	75.331	59.132	61.484
80	71.355	74.974	62.883	64.296
90	73.585	78.093	65.322	67.086
100	75.129	81.080	67.401	69.769
110	76.158	82.034	68.082	70.320
120	77.873	82.891	70.144	72.606
130	78.731	81.987	71.122	73.482
140	79.417	83.288	72.218	74.715
150	79.417	85.484	72.305	75.211
160	80.789	88.476	74.232	77.513
170	81.132	87.091	74.867	77.894
180	81.475	86.348	75.312	77.786
190	82.504	88.951	77.140	79.791
200	82.333	87.448	77.067	79.623
210	83.190	89.477	77.769	80.667
220	83.533	89.674	78.275	81.107
230	83.190	89.847	77.887	81.283
240	83.533	90.028	78.550	81.819
250	83.533	89.593	78.621	81.347
260	83.705	89.587	78.870	81.581
270	83.533	90.567	78.405	81.439
280	83.533	89.382	78.278	81.092
290	83.533	89.804	78.604	81.449
300	84.220	91.784	79.488	82.819
310	84.563	91.641	79.776	83.013
320	84.563	91.826	79.835	82.992
330	85.077	91.917	80.609	83.531
340	84.906	90.685	80.252	83.138
350	85.077	90.696	80.550	83.281
360	85.592	91.267	81.442	84.343
370	85.935	92.269	82.038	84.907



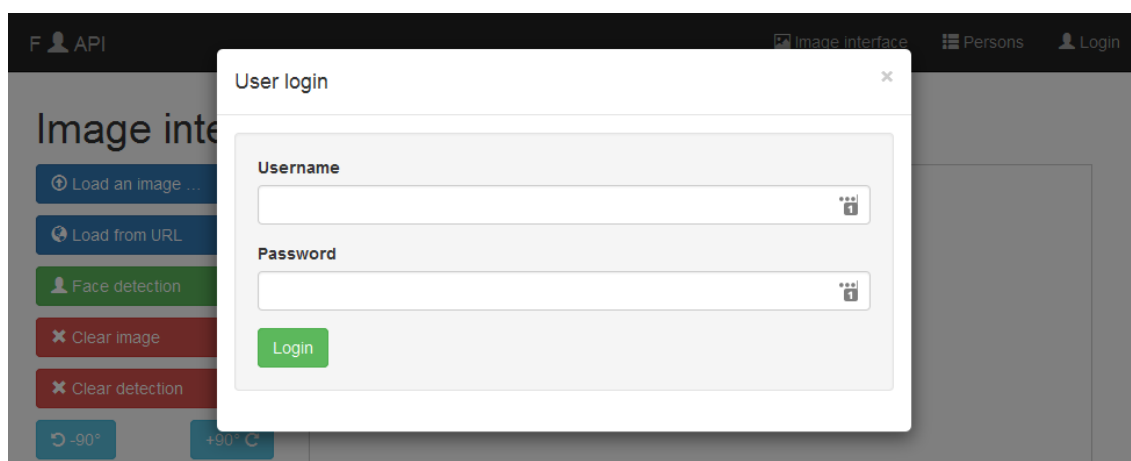
Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
380	86.106	92.311	82.064	84.989
390	86.106	92.174	82.335	85.058
400	85.763	92.247	81.800	84.763
410	85.763	92.227	81.588	84.640
420	86.106	92.406	82.124	85.023
430	85.935	92.142	81.885	84.755
440	85.592	92.107	81.561	84.465
450	86.106	92.392	82.335	85.174
460	85.935	92.199	81.885	84.762
470	86.449	92.289	82.481	85.179
<b>480</b>	<b>86.621</b>	<b>92.352</b>	<b>82.600</b>	<b>85.270</b>
490	86.278	92.230	82.243	85.026
500	86.278	92.185	82.243	84.997
510	86.449	92.170	82.302	85.007
<b>520</b>	<b>86.621</b>	<b>92.455</b>	<b>82.600</b>	<b>85.399</b>
<b>530</b>	<b>86.621</b>	<b>92.242</b>	<b>82.600</b>	<b>85.284</b>
<b>540</b>	<b>86.621</b>	<b>92.242</b>	<b>82.600</b>	<b>85.284</b>
550	86.449	92.170	82.302	85.007

Příloha 14: Tabulka výsledků verifikace osob pomocí lineárního SVM v závislosti na počtu PCA komponent (frontalizovaná norma)

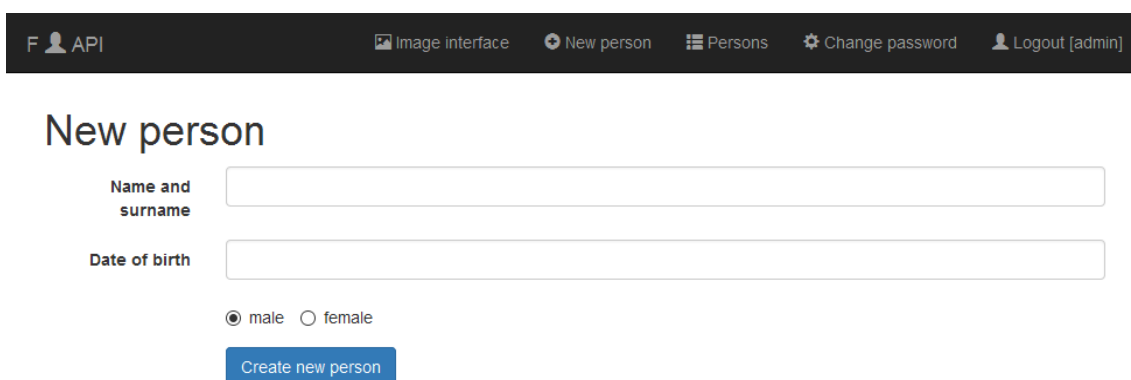
Počet PCA komponent	Úspěšnost	P (přesnost)	R (úplnost)	F <sub>1</sub> -skóre
10	61.477	62.388	65.669	63.986
20	63.854	65.363	65.181	65.272
30	63.909	66.517	61.908	64.130
40	67.411	70.613	64.171	67.238
50	70.278	73.283	67.618	70.337
60	73.453	76.716	70.439	73.443
70	74.723	77.853	71.971	74.796
80	74.995	78.020	72.423	75.117
90	74.433	77.203	72.284	74.663
100	75.231	78.274	72.632	75.348
110	75.431	78.577	72.667	75.507
120	75.649	78.043	74.130	76.036
130	75.848	78.379	74.095	76.177
140	76.284	78.653	74.791	76.673
150	75.903	78.134	74.652	76.353
160	76.120	78.107	75.279	76.667
170	75.903	77.650	75.487	76.554
180	76.810	78.898	75.766	77.300
190	77.572	79.573	76.636	78.077
200	77.318	79.257	76.497	77.853
210	77.318	79.320	76.393	77.829
<b>220</b>	<b>77.735</b>	<b>79.682</b>	<b>76.880</b>	<b>78.256</b>
230	77.627	79.299	77.228	78.250
240	77.645	79.603	76.776	78.164
250	77.427	79.092	77.054	78.060
260	77.209	78.754	77.054	77.895
270	77.427	78.886	77.403	78.137
280	76.973	78.332	77.159	77.741
290	77.627	79.091	77.577	78.327
300	77.209	78.511	77.472	77.988
310	76.991	78.220	77.403	77.809
320	76.973	78.133	77.507	77.819
330	77.137	78.045	78.099	78.072
340	77.137	78.261	77.716	77.987
350	77.173	78.780	76.915	77.837



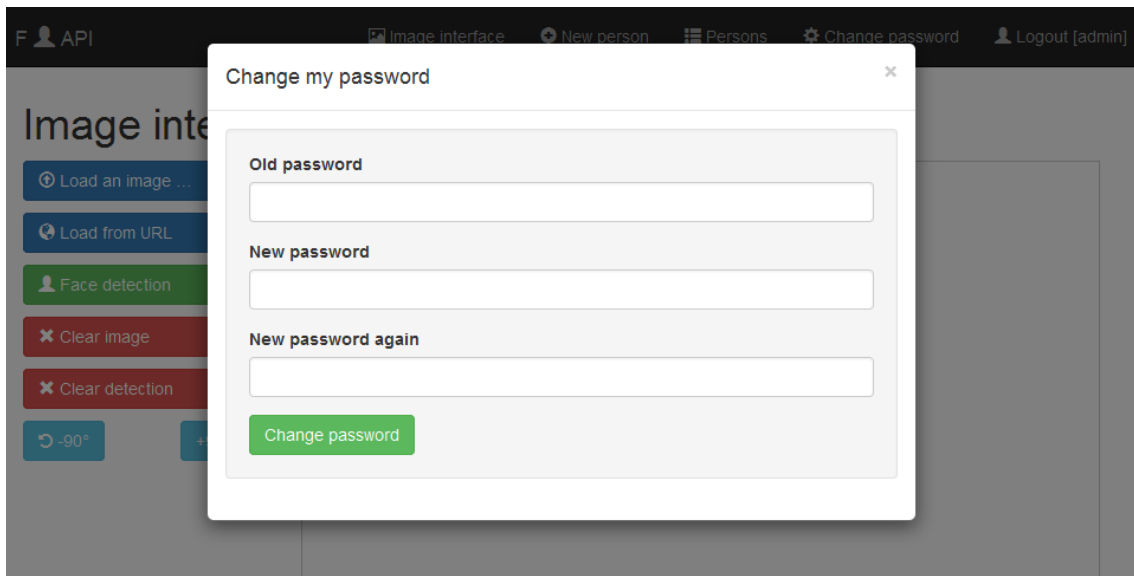
Příloha 15: Ukázka obrazovky *Persons* HTML5 aplikace



Příloha 16: Ukázka přihlašovacího dialogu HTML5 aplikace



Příloha 17: Ukázka obrazovky HTML5 aplikace pro vytvoření nového uživatele



Příloha 18: Ukázka dialogu pro změnu hesla HTML5 aplikace