UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

# DIPLOMOVÁ PRÁCE

2019 Bc. Tomáš Tichý

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

# Biometrické porovnávací algoritmy v procesech ověření identity

Bc. Tomáš Tichý

Diplomová práce

2019

**Univerzita Pardubice**
**Fakulta elektrotechniky a informatiky**
Akademický rok: **2018/2019**

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

| | |
|---|---|
| Jméno a příjmení: | **Bc. Tomáš Tichý** |
| Osobní číslo: | **I17224** |
| Studijní program: | **N2646 Informační technologie** |
| Studijní obor: | **Informační technologie** |
| Název tématu: | **Biometrické porovnávací algoritmy v procesech ověření identity** |
| Zadávající katedra: | **Katedra softwarových technologií** |

Zásady pro vypracování:

Cílem práce je analýza, návrh a implementace programu, který bude využívat biometrické algoritmy pro porovnání snímků obličeje.
V teoretické části budou popsány a vzájemně porovnány vybrané biometrické údaje, přičemž detailněji bude popsána biometrie obličeje. Důraz bude věnován procesu detekce obličeje a následnému porovnání množiny obličejů. Proces detekce a porovnání bude realizován pomocí vybraných biometrických algoritmů. Uvedené příklady biometrických algoritmů budou vzájemně porovnány podle zavedených srovnávacích kritérií. Nastíněna bude i problematika osobních dokladů, informací v nich uložených a jejich zabezpečení.
Praktická část bude obsahovat analýzu, návrh a implementaci porovnávacího programu. V programu budou využity biometrické algoritmy z open-source knihovny OpenCV. Výsledná aplikace bude napsaná dle vzoru MVVM v jazyce C#.

Rozsah grafických prací: **10**

Rozsah pracovní zprávy: **cca 40–50 stran**

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:


1. **LAGANIÉRE, Robert. OpenCV 2 computer vision application programming cookbook: over 50 recepies to master this library of programming functions for real-time computer vision. Brimmingham: Packt Publishing, 2011. Quick Answers to Common Problems. ISBN 978-1-849513-24-1.**

2. **LI, Jun-Bao, Shu-Chuan CHU a Jeng-Shyang PAN. Kernel learning algorithms for face recognition. New York: Springer, [2014]. ISBN 978-1-4614-0160-5.**


Vedoucí diplomové práce: **Ing. Petr Veselý**
Katedra softwarových technologií

Datum zadání diplomové práce: **22. října 2018**

Termín odevzdání diplomové práce: **18. května 2019**

Ing. Zdeněk Němec, Ph.D.
děkan

L.S.

prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 17. listopadu 2018

Prohlašuji:
Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 13. 05. 2019

Bc. Tomáš Tichý

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu své práce, Ing. Petru Veselému, za cenné rady a velmi vstřícný přístup. Dále bych rád poděkoval pracovnímu kolektivu firmy, ve které pracuji. Velké díky patří i mé rodině a blízkým přátelům, kteří mne podporovali a motivovali k dosažení cílů. Především bych rád poděkoval své sestře, Ing. Šárce Tiché a kamarádu Petru Zápotockému.

## ANOTACE

Teoretická část je zaměřená na popis biometrických údajů, především na biometrii obličeje. Nachází se zde popis a porovnání biometrických algoritmů pro detekci a porovnání obličejů a základní uvedení do problematiky osobních dokladů. Praktická část obsahuje analýzu, návrh a implementaci aplikace pro porovnání snímků obličeje.



## KLÍČOVÁ SLOVA

Biometrie, detekce, rozpoznání, identifikace, biometrické algoritmy, osobní doklad


## TITLE

Biometrical comparison algorithms in identity verification processes


## ANNOTATION

Theoretical part is focused on biometric information description, with most of the focus dedicated to the face biometrics. There is also description and comparison of biometric algorithms for both face detection and recognition, and basic information about personal documents. Practical part contains analysis, design and implementation of application for face images comparison.



## KEYWORDS

Biometrics, detection, recognition, identification, biometric algorithms, personal document

**Obsah**

# SEZNAM ILUSTRACÍ

# SEZNAM ZKRATEK A ZNAČEK

2DLDA – dvou-rozměrná lineární diskriminační analýza

AdaBoost – adaptive boosting – adaptivní zrychlení

AOI – area of interest – oblast zájmu

BAC – basic access control – základní řízení přístupu

BBF – best bin first – nejlepší zásobník první

CCD – charge-coupled device – zařízení s vázanými náboji

CVCA – country veryfying certificate authorities – certifikační autority ověřující zemi

DET – detection error tradeoff – porovnání chyb detekce

DG – data group – datová skupina

DNA – deoxyribonucleic acid – deoxyribonukleová kyselina

DOG – difference of gaussians – rozdíl Gaussovy funkce

DS – digital signature – digitální podpis

DVCA – document verifier certificate authorities – certifikační autorita ověřovatele dokumentů

EAC – extended access control – rozšířené řízení přístupu

ECMA – european computer manufacturers association – evropská asociace výrobců počítačů

eMRTD – electronic machine readable travel document – elektronický strojově čitelný cestovní doklad

FAR – false accept rate – míra falešného přijetí

FRR – false reject rate – míra falešného zamítnutí

FTA – failure to acquire – selhání při získávání

FTC – failure to capture – selhání při snímání

FTE – failure to enroll – selhání při zápisu

GAR – genuine accept rate – míra autentického přijetí

GDPR – general data protection regulation – obecné nařízení o ochraně osobních údajů

GUI – graphic user interface – grafické uživatelské rozhraní

HD – Hausdorff distance – Hausdorffova vzdálenost

HOG – histogram orientovaných gradientů

HSV – hue, saturation, value – odstín, saturace, hodnota

HTML – hypertext markup language – hypertextový značkovací jazyk

ICAO – international civil aviation organisation – mezinárodní organizace pro civilní letectví

IRGBY – intensity, red, green, blue, yellow – intenzita, červená, zelená, modrá, žlutá

KLDA – kernelova lineární diskriminační analýza

LBPH – local binary patterns histograms – histogramy lokálních binárních vzorů

LDA – lineární diskriminační analýza

LDS – logical data structure – logická struktura dat

MHD – modified Hausdorff distance – upravená Hausdorffova vzdálenost

MRZ – machine readable zone – strojově čitelná zóna

MVVM – model/view/viewmodel

PCA – principal component analysis – analýza hlavních komponent

PIN – personal identification number – osobní identifikační číslo

RBG – red, green, blue – barevné schéma červená, zelená, modrá

ROC – receiver operating characteristic – provozní charakteristika přijímače

SIFT – scale invariant feature transform – v měřítku neměnná transformace vlastností

SOD – security object document

SURF – speeded-up robust features – zrychlené robustní vlastnosti

WYSIWYG – what you see is what you get – co vidíš, to dostaneš

WPF – windows presentation foundation

XAML – extensible application markup language – značkovací jazyk

YAML – yaml aint markup language

# TERMINOLOGIE

Terminologie obsahuje sumarizaci všech pojmů, které jsou v práci vysvětlené v poznámkách pod čarou. V těchto poznámkách je vždy vysvětlen pojem, který je použit v daném kontextu. Stejně jako celá práce je i tento slovník pojmů orientován na znalostní doménu zpracování obrazu a pojmy, které mohou být využity i v jiné doméně jsou zde vysvětleny z hlediska zpracování obrazu.

*AdaBoost*: zkratka pro adaptive boosting neboli adaptivní zrychlení je algoritmus strojového učení, který je podrobněji popsán zde [21].

*Afinní transformace*: základní operace v počítačové grafice (posunutí, otočení, změna měřítka, zkosení, skládání jednotlivých operací).

*Aliasing*: efekt, díky kterému může rekonstruovaný vzorek být nerozpoznatelný vůči originálnímu obrázku.

*Amplituda*: maximální hodnota měnící se veličiny.

*Analýza hlavních komponent*: transformace sloužící k dekorelaci dat (také snížení dimenze s co nejmenší ztrátou informace). Tuto metodu je možné nastudovat zde [35].

*Anizotropie*: závislost na volbě směru.

*Autentizace*: proces ověření proklamované identity v režimu 1 ku 1.

*Best-Bin-First*: vyhledávací algoritmus, navržený pro efektivní nalezení přibližného řešení problému hledání nejbližšího souseda v multidimenzionálním prostředí.

*Bilineární interpolace*: rozšíření lineární interpolace pro interpolaci funkce dvou proměnných na pravidelnou prostorovou mřížku (jeden směr po druhém).

*Biometrie*: soubor vědních poznatků ohledně charakteristik živých organismů.

*Biometrika*: měřitelné charakteristiky živých organismů.

*Boxová-zpětná vzdálenost*: modifikace zpětné vzdálenosti pomocí výpočtu pouze pomocí bodů, které se nacházejí v blízkosti cílových objektů [19].

*CCD maticový detektor*: elektronické zařízení pro snímání obrazové informace.

*Degenerativní rozhodovací strom*: první klasifikátor eliminuje velké množství negativních pomocí velmi snadného zpracování. Následující vrstvy pokračují v eliminaci vždy o trochu složitějším způsobem. Po několika iteracích je počet pod-oken značně redukován.

*Desaturace*: ztráta původní vlastnosti (odbarvení původně barevného předmětu).

*Deskriptor*: jednoznačný unikátní popisovací vektor pro charakteristické vlastnosti tváře.

*Diferenciace*: rozlišení, porovnání, hledání rozdílů.

*Dilatace*: rozšíření objektů o určitou velikost pomocí skládání bodů dvou množin pomocí vektorového součtu za účelem zaplnění děr.

*Dopředná vzdálenost*: vzdálenost z množiny B do množiny A (z modelu do obrázku).

*EigenFaces – vlastní tváře*: označení pro hlavní komponenty rozdělení tváří

*EigenValue – Vlastní hodnota*: koeficient použitý při transformaci.

*EigenVector – vlastní vektor*: nenulový vektor, jehož směr se po transformaci nezmění.

*Euklidovská vzdálenost*: vzdálenost mezi 2 body v Euklidovském prostoru.

*Fenotyp*: soubor pozorovatelných vlastností a znaků živého organismu.

*Fetální vývoj*: období začínající 9. týdnem těhotenství a končící narozením jedince.

*Gaussova funkce*: funkce zpravidla využívaná za účelem reprezentace hustoty pravděpodobnosti náhodné proměnné s normálním rozdělením. Problematiku lze nastudovat z [40].

*Genetika*: biologická věda, zabývající se dědičností a proměnlivostí organismů.

*Gradientová orientace:* znázornění toku (směru růstu) světla od světlých oblastí k tmavým.

*Haarova vlnka*: nejstarší a nejjednodušší vlnka, kterou lze využít k výpočtu diskrétní vlnkové transformace.

*Hessova matice*: čtvercová matice druhých parciálních derivací skalární funkce. Dostupné na [39].

*Heteroskedasticita*: výskyt podmnožin s odlišnými variancemi od ostatních.

*Histogram*: grafické znázornění distribuce dat pomocí sloupcového grafu.

*Homoskedasticita* – homogenita: všechny proměnné dané sekvence nebo daného vektoru mají stejnou konečnou varianci (rozptyl).

*Chamtivý výběr vlastností*: algoritmus, který v každé iteraci vybírá nejlepší nebo odstraňuje nejhorší vlastnosti.

*Identifikace*: zjištění totožnosti hledáním shod se souborem dat (1 ku N).

*Identita*: totožnost, jednoznačné určení jedinečného objektu (subjektu).

Integrální obrázek: digitální reprezentace obrazu tak, že každý bod představuje součet intenzit předchozích pixelů doleva a nahoru (pravý spodní bod tedy obsahuje součet všech pixelů obrázku).

*Interpolace*: nalezení přibližné hodnoty funkce v daném intervalu, je-li známa hodnota jen v některých jiných bodech tohoto intervalu.

*Invariance*: neměnnost, nezávislost.

*Kanonická tvář*: jednotná forma reprezentace tváří (oříznutý snímek dané velikosti).

*Klasifikátor*: popis jednotlivé charakteristiky obličeje (obočí, nos, ústa) ve formě hran/linií.

*Konvoluce*: matematický operátor zpracovávající 2 funkce (v grafice většinou obrazu a filtru).

*Kovarianční matice* náhodné veličiny X o n složkách: čtvercová reálná matice o rozměru n x n, jejíž prvek s indexy i, j obsahuje kovarianci i-té a j-té složky náhodné veličiny X.

*Laplaceova transformace*: integrální transformace využívaná k řešení obyčejných diferenciálních rovnic. Tato problematika může být nastudována z [38].

*Lineární diskriminační analýza*: metoda pro hledání lineárních kombinací vlastností, které charakterizují dvě nebo více tříd objektů nebo událostí. Více informací zde [37].

*Luminofor*: látka schopná pohlcovat energii a následně ji vyzařovat ve formě světla.

*Mahalanobisova vzdálenost*: vzdálenost mezi bodem P a rozdělením D, blíže popsána [34].

*Markanty*: charakteristické a významné příznaky.

*Mediánový filtr*: nejvyužívanější filtr pro odstranění šumu z obrázku. Problematiku je možné nastudovat z dokumentu [12].

*Metoda nejmenších čtverců*: matematicko-statistická metoda pro aproximaci řešení soustav rovnic. Možno nastudovat zde [36].

*Mezitřídní variace*: variabilita u setů dvou různých osob.

*Mono*: volně šiřitelný open-source projekt, jehož cílem je vytvoření sady nástrojů, kompatibilních s prostředím .NET, splňující standardy ECMA.

*Normalizace*: převedení detekované tváře na standardizovanou formu (rozlišení, jas, perspektiva, natočení).

*Otisk prstu*: vzor hřbetů a údolí na povrchu konečků prstů.

*Parabolická interpolace*: technika hledání extrémů spojité unimodální funkce postupným nasazením parabol k funkci jedné proměnné ve třech jedinečných bodech.

*Pozorná kaskáda*: princip konstrukce kaskády klasifikátorů za účelem zvýšení výkonu detekce a snížení výpočetního času.

*Prahování*: skupina metod pro automatické rozdělení obrazu na oblasti se společnými vlastnostmi (například identifikace popředí a hledání objektů).

*Prvek strukturování disků*: binárně označená oblast, ze které jsou do výpočtů zahrnuty pouze pixely s pravdivou hodnotou (1) a ostatní vynechány (hodnota 0).

*Převzorkování*: matematická technika pro změnu velikosti obrázků v pixelech.

*Saturace*: intenzita barev obrázku (velmi saturovaný obrázek má velmi jasné barvy).

*Security Object Document*: objekt, digitálně podepsaný vydávajícím státem, obsahující hashové hodnoty obsahu LDS.

*Segmentace*: dělení obrazu na části, které korespondují s konkrétními objekty v obraze.

*Singulární rozklad matice*: rozklad komplexní nebo reální matice na maticový součin.

*Skóre shody*: vzdálenost mezi 2 popisovými vektory (šablonami), čím nižší, tím vyšší shoda.

*Sobelův operátor*: operátor diskrétní diferenciace, počítající aproximaci gradientu funkce intenzity obrazu. Tato problematika může být více nastudována na [20].

*Šablona*: jednoznačný (alfa) numerický popisovač získaných biometrických údajů.

*Tokenový snímek*: oříznutá část vstupního obrázku, která obsahuje pouze detekovaný obličej (vyříznutá ohraničená oblast získána z procesu detekce obličeje).

*Validace*: ověření na základě reálných, skutečných informací.

*Verifikace*: ověření, kontrola na základě modelu či jiných předpokladů.

*Vnitro-třídní variace*: variabilita pozorovaná u setu rysů jednoho jedince.

*Vzorkovací hustota*: počet zaznamenaných vstupních vzorků vzhledem k časovému úseku.

*Zpětná vzdálenost*: vzdálenost z množiny A do množiny B (z obrázku do modelu).

# Úvod

Tato diplomová práce se zabývá analýzou, návrhem a implementací porovnávacího programu, který pomocí biometrických algoritmů porovnává snímky obličeje a určuje míru shody. Téma práce bylo vybráno na základě zvyšujícího se tlaku na bezpečnost a kvůli nedostatečnosti klasických způsobů zabezpečení, jakými jsou například heslo nebo přístupová karta. Z těchto důvodů je čím dál více využívána biometrie lidského těla, a to jak fyziologická, tak behaviorální.

Vzhledem k velmi specifické a složité problematice, kterou se tato práce zabývá, zabírá značnou část teoretické části její popis, porovnání biometrických charakteristik a nastínění základních procesů s nimi spojených. Hlavní důraz je věnován procesu detekce obličeje a následnému porovnání množiny obličejů, k čemuž jsou využívány biometrické algoritmy. U těchto biometrických algoritmů je detailně popsán princip, na kterém jsou založeny a následně je provedeno vzájemné porovnání. Kromě algoritmů, použitých v programu zde jsou pro účely porovnání uvedeny i konkurenční, modernější postupy.

Díky stále rozsáhlejšímu využití biometrické identifikace a verifikace jsou k těmto účelům využívány i osobní doklady. Teoretická část proto obsahuje i stručné nastínění této problematiky, tedy která data a jak jsou uložena na čipu cestovního dokladu, jakým způsobem probíhá komunikace a jak je tento čip zabezpečen.

Neodmyslitelnou součástí vývoje software je analýza a návrh. Z tohoto důvodu je součástí práce zpracování v programu Enterprise Architect. Výstup z tohoto programu je v příloze a základní části jsou popsány v praktické části. Vzhledem k širokým možnostem využitelných technologií je zahrnuta i kapitola, zaměřená právě na výběr technologií a jejich popis.

Hlavním cílem této práce je samotná realizace a implementace programu pro biometrické porovnávání snímků obličeje. Po představení použitých technologií, analýze a návrhu je tedy v praktické části popsán způsob implementace a hlavní funkce programu. Součástí je rovněž popis práce v aplikaci a základní uživatelská příručka.

# 1 Úvod do problematiky biometrie

Vzhledem k rostoucímu tlaku na zvyšování bezpečnosti a minimalizaci hrozeb mají lidé snahu hledat a zkoušet nejrůznější způsoby ochrany dat a majetku. Postupně se tak přechází z něčeho, co máme (klíče, přístupové karty, čipy) přes něco, co známe (hesla, piny, kontrolní otázky) k něčemu, čím jsme (lidské charakteristiky, jako například otisky prstů, snímek obličeje atd.). Snižuje se tak riziko ztráty a následného odepření přístupu, jelikož až na mimořádné situace není možné zcela přijít o část sebe sama.

Biometrie umožňuje identifikaci a autentizaci osoby na základě souboru rozpoznatelných a ověřitelných anatomických nebo fyziologických charakteristik, které jsou pro každého člověka unikátní a časově neměnné. Pojem **biometrie** označuje soubor vědních poznatků, založených převážně na statistickém a analytickém přístupu, které mají za účel zkoumání a následné praktické využití měřitelných charakteristik živých organismů z důvodu pozdější jednoznačné identifikace a verifikace [1].

Označení **biometriky** se používá pro samotné měřitelné biometrické charakteristiky živého organismu, které se snímají, zpracovávají, vyhodnocují a uchovávají. Biometriky se dále dělí na fyziologické a behaviorální, přičemž nelze říci, že obě disponují stejnou mírou spolehlivosti. Fyziologické prvky jsou považovány za výhodnější z hlediska určité stability v průběhu lidského života. Navíc tyto nebývají většinou ovlivněny stresem, na rozdíl od behaviorálních charakteristik [1].

Každá z těchto charakteristik musí splňovat určité požadavky, mezi které patří například univerzálnost (každý jedinec, přistupující k aplikaci musí disponovat tímto rysem), jedinečnost (daný rys musí být dostatečně odlišný mezi osobami v rámci celé populace), stálost (biometrika by měla být přijatelně invariantní po daný časový úsek vzhledem k porovnávacímu algoritmu), praktická měřitelnost (možnost pořízení a digitalizace biometrického prvku bez narušení pohodlí osoby), výkon (přesnost rozpoznání a prostředky k tomu potřebné by měly splňovat omezení aplikace), přijatelnost (cílová skupina uživatelů by měla být ochotná podstoupit snímání daných biometrických rysů) a obcházení (jednoduchost imitace za pomoci artefaktů – například falešné otisky prstů) [2].

## 1.1 Anatomicko-fyziologické biometriky

Tyto biometriky splňují podmínku unikátnosti a výhodou je zde i časová stálost. Vychází se zde z vědeckých poznatků o oční duhovce, oční sítnici, tváři, stavbě vnějšího ucha, otiscích

prstů, dlaní a chodidel, geometrii prstů a ruky, topografii žil zápěstí, lidském tělesném pachu, obsahu solí v lidském těle, skladně DNA, rozměrech, váhách a dalších charakteristikách lidského těla.

### 1.1.1 Otisky prstů

Systémy využívající biometriky otisků prstů jsou považovány za nejrozšířenější, ať už za účelem autentizace, nebo identifikace (tyto pojmy budou vysvětleny v další kapitole). Vývoj probíhal na základě znalostí, ze kterých vyplývá, že každá osoba má unikátní otisky prstů, které ji odlišují od ostatních. Na každém prstě má navíc každá osoba jiný otisk, stejně jako se liší otisky prstů u dvojčat.

Pojem otisk prstu označuje vzor hřbetů a údolí na povrchu konečků prstů. Tyto vzory jsou formovány během prvních 7 měsíců fetálního vývoje[1]. Hlavními účely této kožní struktury jsou usnadnění vylučování potu, zlepšení hmatového smyslu a poskytnutí uchopovacího povrchu. Otisky jsou součástí lidského fenotypu[2], tudíž jsou jen velmi málo ovlivněny genetikou[3] [3].

Snímání otisků více prstů od jedné osoby poskytuje doplňující informace, díky kterým můžeme vyhledávat v rozsáhlejších databázích, čítajících i milióny identit[4]. Při práci s rozsáhlejšími databázemi (zejména v procesu identifikace) jsou největším problémem nároky na výkon počítače. Dalším problémem může být fakt, že otisky prstů určité části populace mohou být nevhodné pro proces automatické identifikace kvůli genetice, stáří, environmentálním nebo pracovním důvodům (např. dělníci mohou mít zjizvené konečky prstů) [2]].

### 1.1.2 Otisk dlaně

Lidská dlaň obsahuje, obdobně jako špičky prstů, vzory hřbetů a údolí. Plocha dlaně je však mnohonásobně větší než plocha špičky prstu, proto se od této metody očekává spolehlivější jedinečnost. Vzhledem k potřebě snímání velké plochy jsou snímače tohoto rysu objemnější a dražší než snímače otisků prstů. Kromě celé dlaně dokáží tyto senzory většinou snímat i malíkové nebo palcové hrany. Při využívání nákladnějších zařízení je možné kombinovat jednotlivé části lidské ruky pro vytvoření velmi přesného a spolehlivého biometrického systému [2].

[1] *Fetální vývoj*: období začínající 9. týdnem těhotenství a končící narozením jedince.

[2] *Fenotyp*: soubor pozorovatelných vlastností a znaků živého organismu.

[3] *Genetika*: biologická věda, zabývající se dědičností a proměnlivostí organismů.

[4] *Identita*: totožnost, jednoznačné určení jedinečného objektu (subjektu).

### 1.1.3 Oční duhovka

Oční duhovka je prstencová část oka ohraničená zorničkou a bělmem. Vizuální textura je formovaná během fetálního vývoje a stabilizuje se během prvních 2 let života (pigmentace se však ustaluje delší dobu). V této části oka se nacházejí unikátní identifikační body, které se dají využít k velmi přesnému stanovení identity osoby. Duhovka sestává z náhodně rozmístěných, v čase neměnných barevných struktur, které připomínají sněžné vločky. Není možné najít dvě shodné duhovky [2].

Ke snímání se využívá standardní video technologie, které umožňují detekci kontaktních čoček s falešnou vytištěnou duhovkou. Drobné pohyby oka mohou být použity ke zjištění živosti tohoto biometrického rysu. Přestože systémy využívající tuto charakteristiku mají poměrně nízké FAR, může se zde vyskytovat vyšší FRR (tyto pojmy budou dále vysvětleny).

### 1.1.4 Oční sítnice

Ke snímání dochází pomocí světelného paprsku, jehož část je pohlcena bílou sítnicí lidského oka. Druhá část je odražena, díky čemuž dochází ke zmapování řečiště drobných žilek a cévek sítnice, které zůstávají po dobu života jedince téměř neměnné.

### 1.1.5 Geometrie ruky

Metoda spočívá ve dvou nebo třírozměrném měření délek nebo šířek jednotlivých prstů, kloubů nebo kostí. Environmentální faktory, jako například suché počasí nebo individuální anomálie (např. suchá kůže) neovlivňují přesnost autentizace systémů, které využívají tuto charakteristiku. Tento rys však není dostatečně individuální na to, aby se dal využívat pro identifikaci jedince z velkého vzorku lidí. Navíc tato biometrika ne úplně dokonale splňuje podmínku stálosti, především pak v období růstu dítěte.

Další komplikací může být snímání jedince, který má na rukou šperky nebo který má omezenou obratnost. Fyzické rozměry systému pro snímání geometrie ruky jsou velké, proto nemohou být obsažené například v noteboocích. Existují autentizační systémy, které využívají rozměrů pouze některých prstů (z pravidla ukazováku a prostředníku) namísto měření celé ruky [2].

### 1.1.6 Tvář

Rozpoznávání tváře je metoda, která snímaného jedince příliš neomezuje na pohodlí a využívá atributy obličeje, které jsou nejběžnějším biometrickým rysem. Tento způsob využívají lidé denně k rozpoznání svého okolí. Nejrozšířenější přístupy jsou založeny

buď na lokaci a tvaru atributů tváře (jako např. očí, obočí, nosu, rtů, brady) a jejich vztahů, nebo celkové analýze snímku obličeje, která reprezentuje tvář jako vážené kombinace počtu kanonických tváří[5].

Aplikace využívající tento charakteristický rys jsou různé, a to od aplikací, které vyžadují statický, kontrolovaný snímek obličeje po aplikace schopné rozpoznat obličej v členitém snímku. Mezi další možnosti patří detekce a identifikace z video sekvence (například z bezpečnostních kamer), z trojrozměrného obrázku (techniky založené na geometrii kůže nebo lebky) nebo infračerveného snímku (tato technika se využívá jako řešení problému s osvětlením) [3].

Přestože provedení autentizace v komerčních biometrických systémech je obstojné, jsou zde zpravidla velká omezení ohledně pořízení snímku obličeje. Často je vyžadován snímek s jednoduchým pozadím a kontrolovaným osvětlením. Dalším problémem bývá správné spárování obrázků, pořízených z různých úhlů, s různým osvětlením, nebo v různý čas. Z toho vychází otázka, zda je samotná tvář, bez jakéhokoliv kontextu, dostatečná znalostní báze pro rozpoznání osoby na dostatečné úrovni spolehlivosti [2].

Pro správné fungování biometrického systému pro identifikaci na základě snímku obličeje by měl být systém schopný automaticky detekovat přítomnost obličeje ve snímku, lokalizovat obličej, pokud se zde nějaký nachází a rozpoznat obličej v jakékoliv póze s libovolnými okolními podmínkami [2].

## 1.2 Behaviorální biometrie

Behaviorální charakteristiky jsou v praxi využívány méně často než výše uvedené anatomicko-fyziologické především z důvodu časové nestálosti. Uplatňují se poznatky o lidském hlase, pohybu těla (lokomoci) a znalostech a dovednostech psaní (rozlišujeme psaní souvislého textu, podpis osoby, ale i psaní na počítačové klávesnici).

### 1.2.1 Psaní na klávesnici

Existuje teorie, podle které každá osoba píše na klávesnici charakteristickým způsobem. Nejedná se o rys, u kterého by se očekávala absolutní jedinečnost, ale poskytuje dostatečnou rozlišovací informaci pro umožnění biometrické verifikace[6]. Jakožto u behaviorální biometriky

[5] *Kanonická tvář*: jednotná forma reprezentace tváří (oříznutý snímek dané velikosti).

[6] *Verifikace*: ověření, kontrola na základě modelu či jiných předpokladů.

se zde dá očekávat silná vnitro-třídní variace[7] vzhledem k emočnímu stavu, umístění uživatele vzhledem ke klávesnici, typu klávesnice atd. Psaní na klávesnici může být monitorováno nenápadně, zatímco uživatel zpracovává informace. Tento rys navíc umožňuje kontinuální verifikaci identity jedince během relace [2].

### 1.2.2 Hlas

Tato biometrika obsahuje jak fyziologické, tak i behaviorální charakteristiky, přičemž první druh je zde založen především na tvaru a velikosti tzv. přívěsků (vokální rozsah, ústní a nosní dutina a ústa), které se podílejí na syntéze zvuku. Tato stránka biometriky je pro jedince neměnná, na rozdíl od behaviorální stránky hlasu, která se v průběhu času mění vlivem věku, zdravotního stavu (např. obyčejné nachlazení), emočního stavu a dalších. Ve velkém měřítku tento rys navíc není dostatečně osobitý, pro efektivní nasazení [2].

Existují dva druhy systémů, využívajících tuto biometriku. Prvním je textově závislý systém, který je založen na vyslovení předem definované fráze. Druhý je schopný rozpoznání nezávisle na obsahu projevu, jedná se však o variantu mnohem náročnější na implementaci (která však nabízí podstatně vyšší ochranu proti podvodu). Nevýhodou této biometriky je vysoká závislost na vedlejších faktorech, ze kterých nejvýznamnějším je okolní zvuk v pozadí. Nejčastěji je tato technika využívána pro rozpoznání hlasu v telefonním hovoru, kde je však kvalita hlasového signálu často degradována komunikačním kanálem [2].

### 1.2.3 Podpis

Metoda, při které nedochází jen k zaznamenání statického obrazu, ale i vyhodnocení dynamických charakteristik (rychlost pera, přítlak, směr atp.). Přestože tato technika vyžaduje kontakt s psacím nástrojem a vyvinutí značného úsilí ze strany uživatele, jedná se o poměrně rozšířený a používaný způsob ověření totožnosti. Stejně jako u ostatních behaviorálních biometrik, i zde dochází ke změně v čase a značenému ovlivnění z hlediska fyzického a emočního stavu. U některých jedinců je odlišnost tak vysoká, že i dva po sobě sejmuté podpisy se výrazně liší. Na druhou stranu existuje mnoho případů úspěšného oklamání systému pro verifikaci podpisu [2].

### 1.2.4 Chůze

Velkou výhodou této metody je fakt, že jako jedna z mála umožňuje vzdálené zkoumání dané biometriky. Díky tomuto faktu se postupně rozšiřuje její využití v bezpečnostních aplikacích

[7] *Vnitro-třídní variace*: variabilita pozorovaná u setu rysů jednoho jedince.

pro nenápadnou identifikaci osoby. Princip je založen na extrakci lidské siluety za účelem sledování časově-prostorových atributů pohybujícího se jedince. Stěžejní je tedy volba dobrého modelu pro reprezentaci lidského těla. Některé algoritmy využívají optický tok, spojený se sadou dynamicky získávaných pohybujících se bodů na lidském těle. Další výhodou této biometriky je možnost sledování jedince po delší časovou periodu. Stinnou stránkou však je ovlivnitelnost tohoto charakteristického rysu například volbou obuvi, oblečení, poranění nohou nebo povrch, po kterém se osoba pohybuje [2].

## 1.3 Využití biometrie

Důvěryhodné prokázání totožnosti osoby se stává kritickým bodem dnešní nesmírně provázané společnosti. Potřeba metody spolehlivé autentizace uživatelů se neustále zvyšuje spolu s požadavky na bezpečnost, rychlými pokroky v počítačových sítích, komunikaci a mobilitě. Z tohoto důvodu je biometrie čím dál více zakomponována do několika různých typů aplikací. Tyto aplikace mohou být zařazeny do následujících kategorií:

- Komerční aplikace (např. přihlašování do počítačové sítě, ochrana elektronických dat, e-reklama, přístup k internetu, bankomaty a kreditní karty, mobilní telefony, správa zdravotnických záznamů),
- Vládní aplikace (např. občanské průkazy, vězeňská správa, řidičské průkazy, hraniční kontrola, pasová kontrola, vyplácení sociálních dávek),
- Forenzní aplikace (např. identifikace mrtvol, kriminální vyšetřování, určení rodičovství) a další [2].

## 1.4 Porovnání vybraných biometrik

Jak již bylo uvedeno výše, z hlediska časové stálosti a relativní neměnnosti je výhodnější užití fyziologicko-anatomických biometrik. Ty totiž daleko méně podléhají vnějším vlivům i samotnému procesu vývoje jedince, a kromě extrémních případů (např. amputace končetiny nebo ztráta oka) nejsou tak vysoce ovlivnitelné.

Z hlediska pohodlí sledované, či prověřované osoby se však jeví přívětivější behaviorální rysy, jelikož jejich zkoumání je v menší či větší míře možné bez nutné výrazné spolupráce jedince (podpis či psaní na klávesnici jsou typické procesy, které mohou být nenápadně monitorovány, zatímco snímání otisků prstů, obrazu krevního řečiště či snímku duhovky je něco tak zásadního, co není možné přehlédnou či splést s jiným procesem).

Výjimkou pak je biometrie obličeje, která jako jedna z mála fyziologicky-anatomických biometrik jde analyzovat nezávisle na součinnosti osoby. Je tedy možné tento rys zkoumat i bez vědomí jedince, což především v bezpečnostních aplikacích je velmi žádoucí prvek. Díky jedné z metod snímání biometrických dat obličeje, která bude popsána v následující kapitole, je navíc možné získat obraz i ze zahalené tváře a z různých úhlů pohledu prověřované osoby. Z tohoto důvodu bude tato diplomová práce zaměřena především na zkoumání procesu získání a následného ověření biometriky obličeje a procesy s tímto spojené.

# 2 Biometrické procesy

Na rozdíl od systémů, založených na principu hesel, ve kterých je vyžadována naprostá shoda dvou alfanumerických řetězců pro validaci[8] uživatelské identity, u biometrického systému jen zřídka nastane situace, kdy dva záznamy uživatelských rysů vyústí v naprosto stejný set. Příčinou jsou nedokonalé podmínky biometrických senzorů (např. nedokonalý otisk prstu z důvodu selhání senzoru), alternace uživatelských biometrických charakteristik (např. respirační indispozice ovlivňující rozpoznávání hlasu), změny okolních podmínek (např. nekonzistentní stupně osvětlení při rozpoznávání obličeje) nebo variace v uživatelské interakci se senzory (např. částečný otisk prstu).

Z toho tedy vyplývá, že málokdy vypadají dva sety znaků stejného biometrického rysu jednoho uživatele přesně stejně. Kompletní shoda mezi dvěma sety může tedy indikovat spíše pokus o útok na systém. Variabilita pozorovaná u setu rysů jedince je označovaná jako vnitro-třídní, zatímco o variabilitě setů dvou různých osob se mluví jako o mezitřídní. Použitelný set rysů disponuji nízkou vnitro-třídní a vysokou mezitřídní variací.

## 2.1 Sběr biometrických dat

První fází každého biometrického procesu je nepochybně sběr dat. Podle typu vybrané charakteristiky je nutné správně zvolit senzor, pomocí kterého k načtení dojde. Různé biometriky mají odlišné nároky na patřičné zařízení. Zatímco pro získání snímku obličeje stačí obyčejný fotoaparát nebo i bezpečnostní kamera, pořízení otisku prstu nebo mapy krevního řečiště vyžaduje specifické systémy. Pro bližší nastínění vyšších nároků na senzory otisků prstů bude obsažen oddíl popisující různé druhy snímání.

### 2.1.1 Snímání otisku prstu

Jak již bylo zmíněno v úvodu, pro získání otisku prstu je zapotřebí mít specifické zařízení. Nicméně i to není tak jednoznačné, jelikož druhů snímačů je více. Základní principy budou vysvětleny v této části. Výhodou této biometriky je fakt, že ze senzoru již dostáváme biometrická data, připravená ke zpracování. Není potřeba rozsáhlejší zásah ze strany software pro následné zpracování. Podklady pro tuto problematiku jsou čerpány z [4].

[8] *Validace*: Ověření na základě reálných, skutečných informací.

**Optoelektronické biometrické snímače**

Princip činnosti tohoto snímače je založen na rozdílném odrazu světla. Digitální zobrazení otisku je zachyceno pomocí viditelného světla na rozhraní plochy hranolu a přiloženého prstu. Pod vrstvou, na kterou se přikládá prst, se nachází vrstva fosforu, osvětlující celou plochu prstu. Světlo, které je od povrchu prstu odraženo (k odrazu světla dojde pouze od papilárních linií – vrcholů, od rýh – údolí nikoliv) a prochází luminoforní[9] vrstvou k CCD maticovému detektoru[10]. Zde dojde k tvorbě obrazu otisku prstu, jeho digitalizaci a předání k dalšímu zpracování.

Tento princip disponuje vysokou kvalitou, odolností proti statickým výbojům a minimální ovlivnitelností okolním prostředím. Pokud je však snímaný prst poškozen nebo znečištěn, může dojít ke špatnému vykreslení otisku. Existuje zde také riziko zachycení předchozí stopy otisku při nedostatečném očištění snímací plochy. Velkou nevýhodou pak mohou být větší rozměry tohoto typu snímače v porovnání s ostatními.

**Kapacitní biometrické snímače**

Jak již napovídá název, tato zařízení využívají rozdílu kapacity mezi deskou snímače a povrchem prstů. Prst je přiložen na čtecí plochu, která je osazena velkým počtem elektrod. Vzhledem k tomu, že papilární linie jsou k podložce více přilehlé než mezery mezi nimi, mají vyšší kapacitní odpor. Otisk prstu je získáván v digitální formě, která je připravena k dalšímu zpracování.

Mezi velké výhody tohoto typu patří malé rozměry, jednoduchý princip funkčnosti a vysoká kvalita. Na druhou stranu často dochází ke zničení snímače vlivem statické elektřiny, což značně snižuje dobu životnosti (snímače je potřeba měnit většinou v rozmezí 3 let).

**Teplotní biometrické snímače**

Hlavní komponentou teplotního snímače je malý citlivý čip nazývaný pyrodetektor. Ten snímá rozdíl teplot mezi jednotlivými papilárními liniemi a rýhami mezi nimi. Snímání probíhá přejížděním snímaným prstem po snímací ploše. Výstupem snímání jsou digitální pásy, které jsou následně složeny do výsledného obrazu tisku. Tato metoda se však potýká s velkým množstvím nevýhod. Mezi ty nejvýznamnější patří například nízká kvalita nebo problémy

[9] *Luminofor*: látka schopná pohlcovat energii a následně ji vyzařovat ve formě světla.

[10] *CCD maticový detektor*: elektronické zařízení pro snímání obrazové informace

s algoritmy pro zpracování markant[11]. Náročná je i tvorba databáze, jelikož při přejíždění prstu po snímači mohou vznikat otisky různých částí prstu.

**Elektroluminiscenční biometrické snímače**

V tomto snímači se nachází speciální vrstva, která reaguje na tlak, způsobený luminiscenčním efektem. Důležitou roli zde hraje světlo, které je filtrováno eliminující vrstvou (místy přiložení papilárních linií). Ke zpracování obrazu zde slouží fotodiody a výstup má digitální podobu. Terminály s tímto typem snímače mají velmi malé rozměry a dovedou číst při srovnatelné kvalitě i otisky extrémně suchých prstů. Na druhou stranu jsou tyto senzory náchylnější ke znečištění prachem či vodou a méně odolné vůči mechanickému poškození.

**Radiofrekvenční biometrické snímače**

Princip spočívá v připojení generátoru střídavého napětí na 2 rovnoběžné desky (deskami je zde myšlen snímač a plocha otisku prstu). Důležitou roli zde hraje fakt, že vlnová délka je mnohem větší než délka desek, díky čemuž se zde vyskytuje pouze složka elektrického pole, bez pole magnetického. Pokud bude jedna z desek prst, tvar tohoto pole se změní a bude kopírovat tvar papilárních linií. Přiložením prstu dochází tedy ke zvlnění pole a na senzory dopadá signál s rozdílnými velikostmi. Výběžky (vrcholy) mají větší signál a rýhy nižší. Výhodou této technologie je odolnost vůči nečistotám. Při snímání navíc dochází k vytvoření většího počtu snímků, které jsou postupně optimalizovány.

**Multispektrální biometrické snímače**

Tato technologie umožňuje snímat a zpracovat vlastnosti prstu i pod kůží. Senzor je složen ze 2 částí (zdroje světla a zobrazovacího systému), přičemž světlo prochází pod povrch kůže a díky vícero způsobům nasvícení (různé směry a intenzity světel) umožňuje shromáždit více identifikačních údajů z prstu. Tento princip může bez problémů fungovat i za extrémních podmínek okolního prostředí (tekoucí voda, okolní světlo atp.).

### 2.1.2 Snímání tváře

Proces snímání biometrie obličeje sestává ze dvou kroků. Prvním je extrakce prvků a druhým je klasifikace objektů. V nejjednodušším případě je tento proces z hardwarového hlediska mnohem méně náročný než u většiny ostatních biometrik, jelikož není potřeba disponovat specializovaným zařízením, jako třeba čtečka otisků prstu nebo krevního řečiště.

[11] *Markanty*: charakteristické a významné příznaky.

**Tradiční zpracování obrazu**

Tato metoda nevyžaduje žádný specifický senzor, ani specializované zařízení. Potřebujeme zde pouze snímek (může se jednat jak o fotografii, tak o snímek z videa). Pro správnou funkci je zde potřeba dobré softwarové vybavení, které se postará o zpracování vstupních dat. Způsobů detekce obličeje ze získaného snímku je více, podrobněji bude tato problematika pospána v následující kapitole.

**Trojrozměrné rozpoznávání**

Tento způsob využívá 3 D modelu lidské tváře, ve kterém jsou výrazně znázorněny rozlišovací rysy, především tvar očních důlků, nosu a brady. Tyto prvky jsou pro každého jedince jedinečné a v průběhu času nedochází k jejich modifikaci. Velkými výhodami této techniky je relativní nezávislost na osvětlení a možnost rozpoznání obličeje ze snímku, pořízeného z různých pozorovacích úhlů (např. i fotka z profilu). Tohoto je dosaženo za pomocí hloubkového a osového měření, ze kterého je následně vytvořena šablona[12] [5][6].

K měření dochází pomocí mřížky, která je na obličej promítaná z jiného místa, než kde se nachází kamera. Neprobíhá tedy pouze distanční měření, ale také úhlové, což je hlavní rozdíl mezi 2 D a 3 D technologií detekce obličeje [7].

**Analýza struktury kůže**

Dalším rostoucím trendem je využití unikátnosti struktury kůže pro dosažení přesnějších výsledků. Princip této funkcionality je podobný jako u tradičního zpracování obrazu. Po získání snímku vzorku kůže je tento algoritmicky převeden na matematický, měřitelný prostor, ve kterém jsou systematicky odlišeny linie, póry a samotná textura kůže. Pro zpracování struktury kůže jsou využity například následující postupy:

- Vektorová šablona (velmi malá, využívaná pro rychlé hledání v celé databázi),
- Analýza lokálních komponent (sekundární vyhledávání seřazených shod),
- Analýza struktury kůže (nejrozsáhlejší, finální analýza detailních informací).

Ani tento postup však není dokonalý. Mohou zde vznikat problémy při rozpoznání z důvodu nošení brýlí (odlesky od skel), zakrytí části obličeje dlouhými vlasy, špatného osvětlení nebo nedostatečného rozlišení snímku [5].

[12] *Šablona*: jednoznačný (alfa) numerický popisovač získaných biometrických údajů.

**Snímek z termální kamery**

Alternativou výše uvedených možností je pořízení snímku obličeje pomocí termální kamery. Tento snímek obsahuje pouze tvar hlavy, umožňuje tedy ignorovat doplňky jako brýle, pokrývky hlavy nebo make-up. Další výhodou je možnost pořizování snímků za špatných světelných podmínek, dokonce i v noci, a to bez použití blesku, který by mohl odhalit pozici senzoru. Navíc není potřeba v databázi ukládat přímo snímky z termální kamery. Stačí klasické snímky, pořízené tradičním zpracováním obrazu. Toto porovnání je možné pomocí mezi-spektrální syntézy, ve které dochází k porovnání několika oblastí obličeje a jejich detailů [8].

## 2.2 Biometrický zápis (enrollment)

Účelem biometrického zápisu je získání a archivace biometrických vzorků a generování numerických šablon pro pozdější porovnání. V praxi tedy tento proces zahrnuje jak samotné pořízení biometrických dat (popsáno v předcházející kapitole), tak vytvoření záznamu o uživateli a jeho archivaci. Přestože ukládání surových dat do databáze nebo na zabezpečené přenosné úložiště umožňuje generování nových šablon při případné změně nebo přidání biometrického algoritmu do systému, nařízení a požadavkům na ochranu osobních dat více odpovídá model ukládání vytvořených popisových vektorů, jelikož tyto vektory není možné zpětně převézt na fotografie [26].

Obecně platí, že na kvalitu referenčních snímků jsou kladeny daleko větší nároky než na snímky pro verifikaci nebo identifikaci. Z tohoto důvodu je často nutné, aby použité snímače nejprve prošly procesem zajištění kvality, ve kterém je ověřeno, že sejmutá data odpovídají patřičným normám a požadavkům z hlediska dostatečné přesnosti následujících procesů.

## 2.3 Biometrická verifikace

Biometrická verifikace označuje proces porovnání dat, charakterizujících člověka, s biometrickou šablonou proklamované osoby za účelem určení podobnosti. Prověřovaný jedinec se nejprve prokáže, většinou zadáním PINu, uživatelského jména nebo za pomocí chytré karty. Proklamovaná totožnost je následně vyhledána v databázi a je vrácen příslušný záznam biometrické šablony. Ten je porovnán v režimu 1 ku 1 s nově pořízenými biometrickými údaji prověřované osoby. Výsledek je buď pozitivní nebo negativní, tedy dostatečná či nedostatečná shoda. Tento princip se využívá především za účelem předcházení využívání jedné identity (v rámci daného systému) vícero osobami a označuje se jako pozitivní rozpoznávání. Příkladem

může být snaha zamezení předávání jedné přístupové karty mezi vícero zaměstnanci. V této fázi hledáme odpověď na otázku „*Opravdu se jedná o pana/paní X?“[1]*.

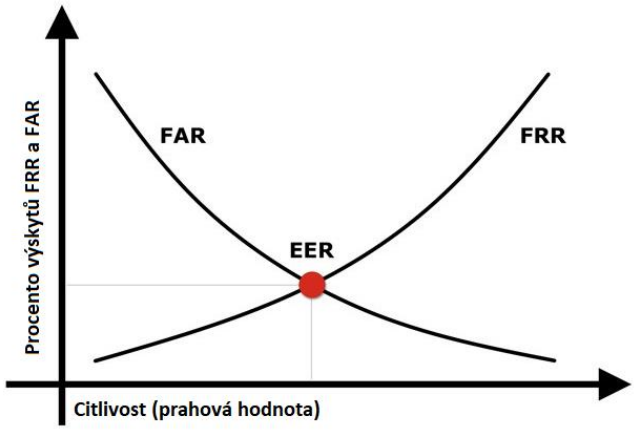
## 2.4 Biometrická identifikace

Oproti tomu biometrická identifikace slouží k jednoznačnému určení identity dané osoby. Dochází k tomu pomocí zachycení položky biometrických dat od snímané osoby. Může se jednat například o fotografii obličeje, záznam hlasu nebo snímek otisku prstu. Tato data jsou v pozdější části procesu porovnána s biometrickými daty všech osob, uložených v databázi. Mluvíme tedy o porovnávání 1 ku N. Nejlepší shoda je určována pomocí zkoumání skóre shody (označující většinou vzdálenost mezi šablonami, bude popsáno později) související se všemi porovnáními, při kterém je uchována identita šablony s největší hodnotou shody [1].

Účelem tohoto tzv. negativního rozpoznávání je prevence situace, kdy jedna osoba využívá vícero identit (například jeden zaměstnanec, kterému kolegové zapůjčili přístupové karty za účelem falšování docházky). Otázka, na kterou zde hledáme odpověď zní „*Kdo je tato osoba?“*. Zatímco k pozitivnímu rozpoznávání mohou být využity i jiné prostředky (hesla, klíče, tokeny), negativního rozpoznávání lze dosáhnout pouze za využití biometrik.

## 2.5 Chyby v biometrických procesech

Úroveň shody dvou setů je prezentována pomocí skóre podobnosti (skóre shody). Pokud hovoříme o výsledku porovnání vzorků od stejného uživatele, označujeme jej jako autentický, v opačném případě se jedná o podvodný výsledek. Podvodné skóre, které překročí práh přípustnosti je označováno jako falešné přijetí (nebo také falešná shoda), obdobně jako autentické skóre s hodnotou nižší, než práh je známo pod pojmem falešné zamítnutí. Z tohoto pocházejí ukazatele **FAR** (False Accept Rate – Míra falešného přijetí) a **FRR** (False Reject Rate – Míra falešného zamítnutí). Třetím ukazatelem je **GAR** (Genuine Accept Rate – Míra autentického přijetí) který udává část autentických výsledků přesahující práh přípustnosti. Tento oddíl čerpá z podkladů [2].




**Obrázek 1 – Graf chyb FRR a FAR [27]**

Regulací hodnoty přípustného prahu je možné ovlivnit hodnoty ukazatelů FRR a FAR, v rámci jednoho systému však není možné snížit obě chybové hodnoty najednou. Tuto dvojici ukazatelů je dále možné sumarizovat pro různé hodnoty prahu pomocí křivky **DET** (Detection Error Tradeoff – Porovnání chyb detekce). Pokud v tomto grafu použijeme lineární, logaritmické nebo semi-logaritmické měřítko, hovoříme o křivce **ROC** (Receiver Operating Characteristic – provozní charakteristika přijímače). Místo v grafu, kde je míra obou chyb stejná je označováno jako EER (Equal Error Rate – shodný výskyt obou druhů chyb).

Obecně platí, že výskyty falešného přijetí i zamítnutí nejsou mezi uživateli biometrického systému rovnoměrně distribuované. Příčinou je odlišná vrozená rozpoznatelnost různých uživatelů. Na základě těchto odlišností rozeznáváme 4 kategorie uživatelů (také známé jako Doddingtonova ZOO):

- Ovce – uživatel s velmi výraznými rysy a nízkou vnitro-třídní variací, očekávaná nízká hodnota výskytu falešného přijetí i zamítnutí,
- Koza – uživatel náchylný k falešnému zamítnutí (vysoká vnitro-třídní variace),
- Jehně – uživatel, jehož rysy se rozsáhle překrývají s rysy jiných jedinců, nízká vnitro-třídní variace, velká míra falešného přijetí jiného jedince jako jehněte,
- Vlk – jedinec se sklony k úspěšnému manipulování s biometrickými rysy (především behaviorálními) za účelem imitace legitimně zapsaného uživatele systému, zvýšení výskytu falešného přijetí.

Dalším chybovým ukazatelem je míra **FTA** (Failure to Acquire – selhání při získávání) někdy také označován jako **FTC** (Failure to Capture – selhání při snímání). Oba termíny označují

poměr případů, kdy se biometrickému zařízení nepodařilo sejmout vzorek prezentované biometrické charakteristiky. Tato situace může nastat například když senzor není schopný lokalizovat biometrický signál dostatečné kvality (např. nejasný otisk prstu). Vliv na výskyt tohoto typu chyby může mít i opotřebení zařízení, a proto je třeba dbát na pravidelnou údržbu biometrických senzorů.

Míra **FTE** (Failure to Enroll – selhání při zápisu) udává poměr uživatelů, kteří nemohou být úspěšně zapsáni do biometrického systému. Z hlediska správné interakce se systémem i usnadnění získávání kvalitních biometrických dat je nezbytné vhodné seznámení uživatelů se systémem.

# 3 Vybrané biometrické detekční algoritmy

Důležitým prvkem je schopnost rozlišit rozdíl mezi obličejem a zbytkem fotografie. Programy zaměřené na rozpoznání obličeje jsou založeny na detekci tváře a následném měření jejich parametrů. Každá tvář má jisté charakteristické orientační body (především vzdálenost mezi očima, šířka nosu, hloubka očních důlků, tvar lícních kostí a délka čelisti). Každá osoba má přibližně 80 těchto mezníků (markant). Tyto mezníky jsou změřeny a je vytvořen numerický kód, nazývaný otisk obličeje. Tento přístup se však často setkává s problémy z důvodu pořizování snímků v nekontrolovaném prostředí. Pro efektivní a přesnou funkčnost totiž systém vyžaduje fotografii s přímým pohledem tváře do kamery a s co nejmenšími odchylkami osvětlení [5].

Pokud máme k dispozici snímek s kontrolovaným (jednobarevným a statickým) pozadím, stačí jej odebrat a získáme ohraničení obličeje.

## 3.1 Hledání obličeje pomocí barvy kůže

Návrh systému pro automatické rozpoznání obsahu snímku je netriviální úkol, jež byl studován pro širokou škálu využití. V minulosti již bylo navrženo mnoho systémů za účelem nalezení lidí nebo tváří na obrázku. Jedná se například od systém Rowley, Baluja a Kanade[13], který využívá trénování neuronové sítě a počítání vzdáleností mezi snímky, které jsou obsaženy v sadě pro trénování. Další softwarové nástroje jsou určeny pro rozpoznání rysů tváře ve snímku, o kterém je známo, že je obsahuje. Metoda, která bude popisována v tomto oddílu se liší v zaměření, které spočívá v detekci obličeje na libovolném snímku. Není zde ani nutnost náročného trénování neuronové sítě nebo počítání vzdáleností mezi všemi oblastmi obrázku. Tento oddíl je založen na poznatcích sepsaných na stránkách [9].

Proces sestává ze dvou kroků. Prvním je filtrování obrázku pro označení pouze těch oblastí, ve kterých je pravděpodobnost výskytu tváře. Filtr využívá základní funkce pro matematické operace a zpracování obrazu z programu MATLAB a je založen na kožním filtru, vyvinutém pro Berkeley-Iowa vyhledávač nahých lidí[14]. Samozřejmostí bylo provedení úprav za účelem subjektivního vylepšení výstupu. Druhý krok zahrnuje odstranění nejtmavších a nejsvětlejších oblastí z filtrovaných kožních oblastí. Odstraněné prvky by na základě empirických testů měly odpovídat očím, obočí, nosním dírkám a puse. Oblasti

[13] Detekce tváří nezávisle na rotaci pomocí neuronových sítí, dostupné z [10].
[14] Hledání nahých lidí, práce vycházející ze spolupráce univerzit Berkeley a Iowa, dostupná z [11].

se vzniknutými děrami, mohou být považovány za kandidáty na obličeje. Volitelně se zde může vyskytovat třetí krok, jímž je rozlišení rozměrů otvorů a jejich prostorové uspořádání pro zvýšení robustnosti programu.

**Obrázek 2 – Vstupní RGB obrázek [9]**

Vstupním souborem je obrázek ve formátu RGB (viz Obrázek 2) s intenzitou barev v rozsahu 0 až 255. Matice RGB jsou vynulovány, aby se předešlo desaturaci[15] při převodu obrázku z barevného prostoru RGB do IRGBY. Nejnižší hodnota intenzity vyskytující se alespoň u 10 pixelů z jakékoli hrany libovolné ze třech barevných úrovní je nastavena jako nulová odezva obrázku, tedy hodnota, která je odebrána ze všech třech barevných úrovní. Snímek je převeden z RGB do IRGBY a dojde ke kalkulaci amplitudy[16], odstínu a saturace[17]. K převodu dochází pomocí variace vzorce Fleck a Forsyth[18]

$$\boldsymbol{I = \frac{[L(R)+L(B)+L(G)]}{3}} \quad \textbf{(1)}$$

$$\boldsymbol{Rg = L(R) - L(G)} \quad \textbf{(2)}$$

$$\boldsymbol{By = L(B) - \frac{[L(G)+L(R)]}{2}} \quad \textbf{(3)}$$

, kde operace L(x) je definována jako:

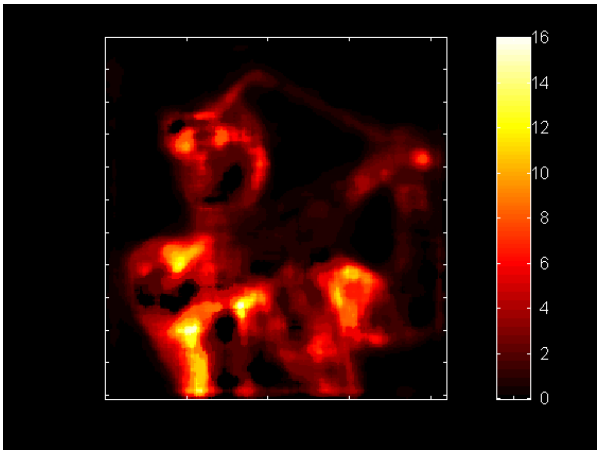
$$\boldsymbol{L(x) = 105 * \log_{10}(x+1).} \quad \textbf{(4)}$$

Matice RG a BY jsou následně filtrovány pomocí mediánového okenního filtru[19] s délkami stran 4*SCALE (hodnota počítaná jako nejbližší integer k výsledku (výška+šířka)/320). Toto

[15] *Desaturace*: ztráta původní vlastnosti (odbarvení původně barevného předmětu).
[16] *Amplituda*: maximální hodnota měnící se veličiny.
[17] *Saturace*: intenzita barev obrázku (velmi saturovaný obrázek má velmi jasné barvy).
[18] Vzorec vychází z aplikace pro hledání nahých lidí, dostupné z [11].
[19] *Mediánový filtr*: nejvyužívanější filtr pro odstranění šumu z obrázku. Problematiku je možné nastudovat z dokumentu [12], kde je popsána i volba velikosti okna.

filtrování je krok, který nejvíce omezuje rychlost celého procesu detekce a může být vylepšen aproximací mediánového okenního filtru navrhovanou Fleckovým několika okruhovým operátorem[20].

Pro nalezení oblastí s nízkým obsahem informací o textuře je využita mapa amplitud textury (Obrázek 3). Tohoto se využívá na základě předpokladu o přítomnosti velmi hladké (jednotné) struktury v oblastech obrázku, které odpovídají kůži. Texturová mapa je vytvořena z matice I pomocí třech kroků. Prvním krokem je mediánový filtr I pomocí okna o délce 8*SCALE. Další krok zahrnuje vytažení filtrovaného obrazu z původní matice I. Posledním krokem je získání absolutní hodnoty rozdílu a vytvoření mediánového filtru výsledku pomocí okna o velikosti 12*SCALE.




**Obrázek 3 – Mapa amplitud textury vstupního obrázku [9]**

Odstín a saturace jsou využity k vybrání těch oblastí, které barevně odpovídají barvě kůže. Převod z formátu IRGBY na hodnotu odstínu je pomocí vzorce

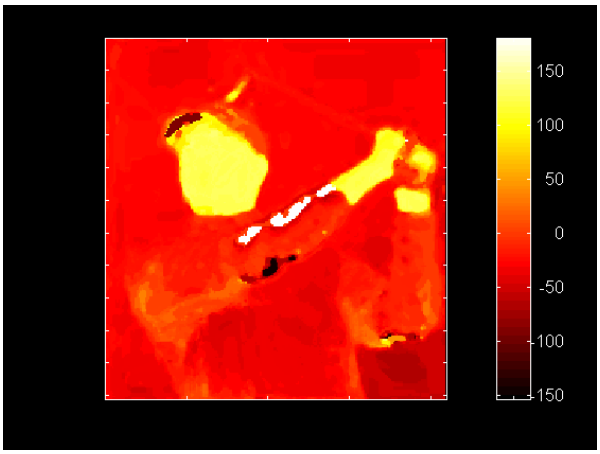
$$\boldsymbol{odstín} = (\boldsymbol{atan^2(Rg, By)}), \qquad (5)$$

kde výsledná hodnota vyjde ve stupních. Převod na saturaci je zajištěn pomocí rovnice

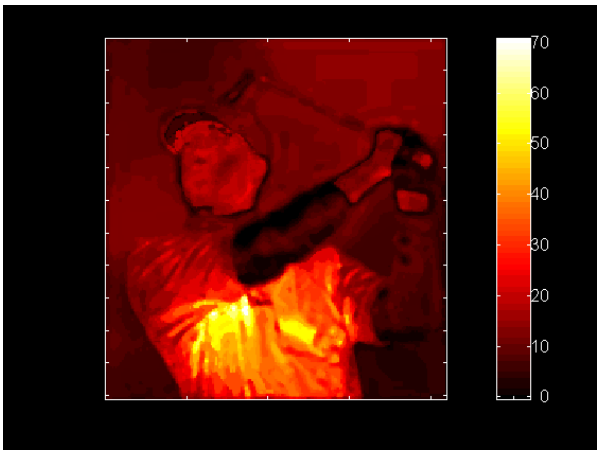
$$\boldsymbol{saturace} = \boldsymbol{sqrt(Rg^2 + By^2)}. \qquad (6)$$

Díky využití texturové amplitudy, odstínu (Obrázek 4) a saturace (Obrázek 5) je možné označit oblasti kůže.

[20] Postup, navrhnutý M. M. Fleckem, jehož princip lze prostudovat z [13].




**Obrázek 4 – Odstíny vstupního obrázku [9]**




**Obrázek 5 – Saturace vstupního obrázku [9]**

Pokud pixel patří do jednoho z daných rozsahů [textura $\langle 4{,}5; 120)$, textura $\langle 4{,}5; 150)$] je označen jako součást kůže v binárním poli mapování kůže, kde 1 vypovídá o přítomnosti kůže na daném pixelu původního obrázku a 0 odpovídá pixelu neobsahujícímu kůži. Tato mapa je reprezentována černobílým obrázkem (Obrázek 6), kde kůže je vyobrazena bíle. Mapa je navíc rozšířena pomocí operace dilatace[21] a prvku strukturování disků[22]. Díky tomuto se rozšíří mapa kožních oblastí o hraniční pixely, oblasti v blízkosti vlasů nebo oblasti s nízkou saturací. Dilatace přidává 8 připojených pixelů na každou stranu okraje objektu. Mapa je následně porovnána s hodnotami odstínu a saturace. Pokud jsou splněny podmínky (odstín v rozsahu 110-180 a saturace v rozsahu 0-130), hodnota 1 je v mapě ponechána. Výsledek

[21] *Dilatace*: Rozšíření objektů o určitou velikost pomocí skládání bodů za účelem zaplnění děr.
[22] *Prvek strukturování disků*: Binárně označená oblast, ze které jsou do výpočtů zahrnuty pouze pixely s pravdivou hodnotou (1) a ostatní vynechány (hodnota 0).

kožního filtru nemusí být vždy dokonalý, vzhledem k tendenci označovat vysoce saturované červené nebo žluté oblasti jako části kůže.

**Obrázek 6 – Kožní mapa vstupního obrázku [9]**

Binární kožní mapa je následně spolu s původním vstupním obrázkem použita pro detekci obličejů. Technika spočívá v prahování[23] oblastí kůže za účelem zobrazení otvorů na místech, kde se nachází obočí, oči, ústa nebo nos. Vychází se zde z předpokladu, že žádné jiné oblasti kůže nemají výrazné rysy, na jejichž místech by se objevily díry. Tento přístup se může zdát dosti zjednodušený, na základě dalších omezení z hlediska rozměrů děr a jejich umístění lze však dosáhnout kvalitních výsledků jako rychlejší a jednoduší alternativy neuronových sítí.

Podstatnou podmínkou funkčnosti je uzavření binární mapy do souvislých oblastí. Eliminace děr je důležitá, neboť program dále předpokládá, že jediné otvory jsou ty, které byly vygenerovány procesem prahování. Uzavírání je prováděno na kožní mapě pomocí prvku strukturování disků. Matice tohoto obrázku je následně skalárně vynásobena maticí originálního obrázku, převedeného do stupňů šedi. Výsledkem tohoto kroku je černobílý obrázek, obsahující pouze části obrázku, na kterých byl nalezen výskyt kůže (Obrázek 7).

[23] *Prahování*: skupina metod pro automatické rozdělení obrazu na oblasti se společnými vlastnostmi (například identifikace popředí a hledání objektů).

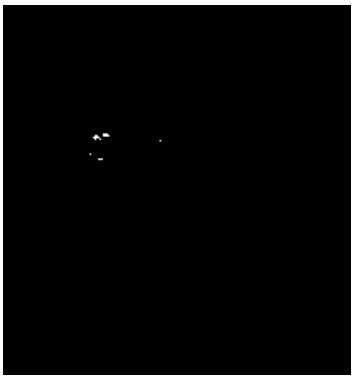
**Obrázek 7 – Kožní mapa s černobílým vstupním obrázkem [9]**

Pro zvýšení kontrastu je využito roztažení histogramu[24] výsledného černobílého obrázku (Obrázek 7). Díky tomuto mohou tmavé a světlé oblasti zapadnout do více předvídatelných rozsahů intenzit a dojde ke kompenzaci efektu různého osvětlení. Nyní proběhne další fáze prahování, kde jsou odstraněny nejtmavší a nejsvětlejší pixely. Ze vzniklého obrázku je následně vytvořen tzv. pozitivně označený obrázek (Obrázek 8), na kterém jsou vyobrazeny různé oblasti kůže.

**Obrázek 8 – Pozitivně označený obrázek [9]**

Analogicky je vytvořen tzv. negativně označený obrázek (Obrázek 9), na kterém jsou vyobrazeny pouze otvory jako samostatné objekty. Uzavírání děr binárního obrázku, vytvořeného prahováním je provedeno pomocí prvku strukturování disků. Výsledek je získán z původního binárního obrázku a zobrazuje pouze otvory v oblastech kůže.

[24] *Histogram*: grafické znázornění distribuce dat pomocí sloupcového grafu.



**Obrázek 9 – Negativně označený obrázek (otvory) [9]**

Negativně označený obrázek (Obrázek 9) je použit spolu s pozitivně označeným obrázkem (Obrázek 8) pro nalezení objektů na snímku, které by mohly být obličeje. Nejprve jsou odstraněny díry z negativního obrázku, které mají velikost pouze 1 pixel (jelikož se většinou jedná o anomálie). Alternativním přístupem může být odstranění všech děr, kromě 3 největších (2 oči a nos). Objekty otvorů jsou dále expandovány využitím dilatace a matice tohoto binárního obrázku je skalárně vynásobena maticí pozitivně označeného obrázku. Výsledkem je obrázek (Obrázek 10), kde jsou pouze objekty, které se nacházejí v bezprostředním okolí děr.

**Obrázek 10 – Objekty tváře [9]**

Vzhledem k tomu, že tento proces spoléhá především na hledání děr v prahovaných objektech, je zde vyšší šance nalezení tváří nezávisle na perspektivě. Nevýhodou může být vyšší riziko detekce jiných objektů než tváří. Tato metoda vykazuje skvělé výsledky u snímků, kde tvář zabírá převážnou část obrázku a dostatečné výsledky, když je tvář součástí větší kompozice. Pro zvýšení robustnosti programu je možné zabudovat minimální poměrnou část snímku, kterou musí obličej zabírat, aby byl přijat [9].

## 3.2 Hledání obličeje pomocí detekce pohybu

Pokud můžeme využít video záznam (případně živý přenos) jako vstup pro program, využijeme toho, že tvář se v realitě takřka neustále hýbe. Stačí pouze vymezit pohybující se oblast, je zde však riziko detekce jiného pohybujícího se objektu. Prokladem pro tento oddíl je elektronický článek [14].

Detekce pohybu probíhá ve 4 krocích. Nejprve je potřeba provést diferenciaci[25] aktuálního snímku z videa se snímkem bezprostředně předcházejícím. Pokud je rozdíl větší než hodnota daného prahu, jedná se o dostatečný pohyb pixelu a tento pixel je následně nastaven na černou barvu. Pokud je hodnota menší než práh, pixel je nastaven na bílou barvu. Výsledný obrázek (Obrázek 11) indikuje zda, případně kde, došlo k nějakému pohybu.

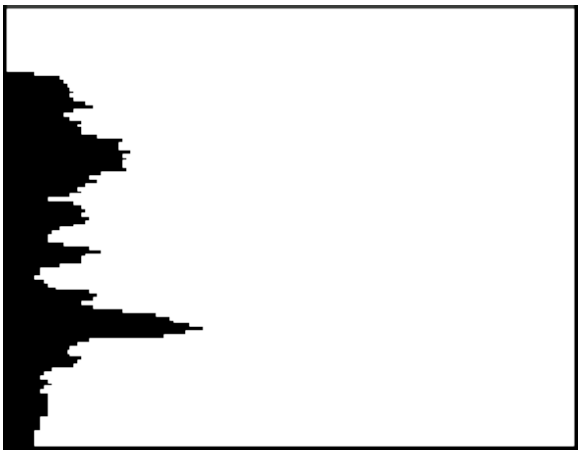
**Obrázek 11 – Pohybový obrázek [14]**

V prahovaném obrázku se může vyskytovat šum. Za účelem odstranění tohoto šumu se provádí skenování snímku pomocí mediánového filtru a odstranění všech černých pixelů, které jsou izolovány v bílém prostoru. Pokud je uprostřed rámce dané velikosti černý pixel a počet černých pixelu rámce je menší než šířka rámce, dojde k odebrání černého centrálního pixelu, jelikož se pravděpodobně jedná o šum. V opačném případě zůstane pixel černý. Tímto způsobem zajišťujeme detekci pohybu pouze velkých objektů.

Dalším krokem je specifikace počtu posunutých (černých) pixelů pro každý řádek. Takto vytvořený obrázek (Obrázek 12) se využívá pro nalezení horních pohybujících se objektů. Pokud se zde nachází dostatečný počet po sobě jdoucích řádků s výrazným posunem, předpokládáme, že se jedná o objekt, tedy ne pouze o jediný posunutý pixel. Z informace

[25] *Diferenciace*: Rozlišení, porovnání, hledání rozdílů.

o množství pohybu na každém řádku zjistíme bod uprostřed pohybujícího se objektu (vypočítáme střed objektu tvaru čtverce). Dojde k vypočtení průměrné šířky a centrální pixel se nachází na prostředním řádku a sloupci, jehož hodnota se rovná polovině průměrné šířky. Tento proces je zopakován pro rámce různých velikostí.



**Obrázek 12 – Počet pixelu na každém řádku pohybového obrázku [14]**

Alternativní přístup hledání obličeje pomocí detekce pohybu využívá faktu, že každý člověk musí pravidelně mrkat, za účelem dostatečného zvlhčování oka. Mrkání je nedobrovolné a velmi rychlé, většina lidí si jej ani neuvědomuje. Detekce mrkajícího vzoru v sekvenci snímků je ale snadným a spolehlivým způsobem detekce přítomnosti obličeje. Mrkání poskytuje časově prostorový signál, který je snadno detekovatelný a pro tváře unikátní. Navíc skutečnost, že obě oči mrkají současně nabízí redundantní ověření pro nezaměnitelnost tohoto pohybu s jiným možným pohybem v rámci scény. Symetrická poloha očí s pevnou separací může sloužit jako prostředek pro normalizaci velikosti a orientace hlavy [15].

Princip je do jisté míry podobný s předchozím popisovaným způsobem. Při získání nového snímku je uložen snímek předchozí. Výsledný diferencovaný obrázek obsahuje malou hraniční oblast kolem hlavy. Pokud nastala situace, kdy na jednom byly oči zavřené, vzniknou kolem nich kulaté oblasti. Diferencovaný obrázek je dále prahován a pro každou komponentu je spočítán hraniční rámeček. Oblast s předpokládaným výskytem oka musí mít hranice v určitém vertikálním i horizontálním rozsahu. Dvě takovéto oblasti musí být detekovány s určitou separací ve směru osy x a minimální diferencí ve směru osy y. Při detekci dvou hraničních boxů dojde k předpokladu mrkajících očí. Pozice obrázku je dále určena dle prostředku mezi těmito oblastmi. Vzdálenost k tváři je měřena na základě této separace. Díky tomu je možné určit velikost okna pro extrakci tváře ze snímku [15].

## 3.3 Hledání obličeje na základě Hausdorffovy vzdálenosti

Přestože oba výše zmíněné způsoby mají své výhody, jsou případy, ve kterých je jejich užití značně neefektivní, či dokonce nemožné. Detekci založenou na pohybu není možné použít u statických snímků a detekce pomocí barvy kůže nefunguje stejně dobře pro všechny barvy. V této kapitole bude popsán přístup založený na modelech, který je ideální pro statické obrázky v odstínech šedé. Informace pro popis této metody pocházejí z [17].

Princip spočívá ve využití Hausdorffovy vzdálenosti[26] (HD – Hausdorff Distance), což je označení pro metriku mezi dvěma množinami bodů. Vzhledem k účelu použití omezíme tuto vzdálenost na dvě dimenze. Pro účely definice považujme $A = \{a_1, \dots, a_m\}$ a $B = \{b_1, \dots, b_m\}$ za dvě konečné množiny bodů. Hausdorffova vzdálenost je tedy definovaná jako:

$$\boldsymbol{H(A,B) = \max(h(A,B), h(B,A))}, \qquad \textbf{(7)}$$

$$\text{kde } \boldsymbol{h(A,B) = \max_{a \in A} \min_{b \in B} \|a - b\|}. \qquad \textbf{(8)}$$

Z toho vyplývá, že h(A, B) se nazývá směrová Hausdorffova vzdálenost ze setu A do setu B se základní normou $\|.\|$ na bodech setů A i B. Pro účely zpracování obrazu se aplikuje lehce upravené opatření, (řízená) upravená Hausdorffova vzdálenost (MHD – Modified Hausdorff Distance), která byla představena Dubuissonem ve tvaru:

$$\boldsymbol{h_{mod}(A,B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} \|a - b\|}. \qquad \textbf{(9)}$$

Díky využití průměrů jednotlivých vzdáleností mezi body snižuje tato verze dopad odchylek, díky čemuž se jedná o daleko vhodnější model pro účely rozpoznávání.

Tohoto principu je v procesu detekce využito tak, že dvourozměrné množiny A i B považujeme za reprezentace snímku a objektu. Tedy každý pixel z množiny B je označován jako prvek obrázku (množiny A), například hraniční bod. Cílem je nalezení transformačních parametrů p, $p \in P$ takových, pro které je Hausdorffova vzdálenost mezi transformovaným modelem $T_p(B)$ a A minimální. Volba přijatelných transformací (např. měřítka a překladů) a jejich parametrů P záleží na případu užití. Efektivní kalkulace HD umožňuje vyčerpávající vyhledávání v diskretizovaném transformačním prostředí. Formulace detekčního problému může být následující:

[26] Podrobněji je tuto problematiku možné nastudovat na webu [18].

$$d_{\hat{p}} = \min_{p \in P} H(A, T_p(b)). \quad (10)$$

Dále je provedena operace $h(T_p(B), A)$, tedy výpočet dopředné vzdálenosti[27] a $h(A, T_p(B))$, tedy výpočet zpětné vzdálenosti[28]. Jelikož nám stačí uvažovat jen tu část obrázku, která je popisována modelem, zaměníme zpětnou vzdálenost za boxovou zpětnou vzdálenost[29] $h_{box}$.

Technika využívající HD se skládá ze 2 fází. Nejprve musí proběhnout hrubá detekce, ve které dojde k definici oblastí zájmu (AOI – Area of Interest), s pevně nastavenou šířkou i výškou, pro každý příchozí snímek. Tyto oblasti jsou dále převzorkovány[30] na danou pevnou velikost, aby byly nezávislé na rozměrech snímků. Následuje fáze segmentace[31], ve které dojde ke kalkulaci obrázku hraniční intenzity z upravené oblasti zájmu pomocí Sobelova operátoru[32]. Lokální prahování zajišťuje rovnoměrné rozdělení výsledných binárních hraničních bodů v rámci celé oblasti snímku.

Za využití modelu B a binární reprezentace A, získané ze segmentace, může nyní proběhnout lokalizace obličeje v obrázku. Na základě pozorování by měla stačit dopředná vzdálenost pro prvotní odhad pravděpodobné polohy. Set parametrů $\hat{p}$ je použit jako vstup do následující fáze.

Pomocí výše zmíněného setu parametrů $\hat{p}$ je vymezena nová oblast zájmu, která pokrývá očekávanou oblast pro výskyt obličeje. Tato část je převzorkována z originálního snímku do černobílého obrázku obličejové oblasti. Nad tímto je následně provedena segmentace a lokalizace, obdobně jako v první fázi procesu s tím rozdílem, že je kladen větší detail na oblast očí. Hodnota modifikované boxové zpětné vzdálenosti na dané pozici, zejména po vynásobení modifikovanou dopřednou vzdáleností, může být použita jako hodnotící kritérium kvality odhadu. Pozice očí je stanovena ze vstupního setu parametrů.

Verifikace této metody se provádí pomocí porovnání zjištěné polohy očí (po proběhnutí celého procesu) s pevně zadanou skutečnou polohou očí. Z testování nad velkou testovací sadou vyplývá, že tato technika je robustní vůči různým pozadím snímků i měnícímu se osvětlení. Tato technika je navíc vhodná i pro využití pro živé přenosy videí.

---

[27] *Dopředná vzdálenost: Vzdálenost z množiny B do množiny A (z modelu do obrázku).*
[28] *Zpětná vzdálenost: Vzdálenost z množiny A do množiny B (z obrázku do modelu).*
[29] *Boxová zpětná vzdálenost*: Modifikace zpětné vzdálenosti pomocí výpočtu pouze pomocí bodů, které se nacházejí v blízkosti cílových objektů [19].
[30] *Převzorkování*: matematická technika pro změnu velikosti obrázků v pixelech.
[31] *Segmentace*: dělení obrazu na části, které korespondují s konkrétními objekty v obraze.
[32] *Sobelův operátor*: operátor diskrétní diferenciace, počítající aproximaci gradientu funkce intenzity obrazu. Tato problematika může být více nastudována na [20].

## 3.4 Hledání obličeje pomocí kaskádových klasifikátorů

Velkým průlomem v detekci obličejů byl přístup využívající tzv. slabé klasifikátory[33]. Tato metoda může, především po rozsáhlém trénování, přinést působivé výsledky. V poslední době se jedná o jeden z nejvyužívanějších algoritmů pro detekci. Základní implementace je využita i v open-source knihovně OpenCV. Tento oddíl vychází z poznatků [22].

Tato technika využívá 3 zásadních komponent, ze čehož první je nová reprezentace snímku nazývaná integrální obrázek[34], díky kterému je možné dosahovat mnohem rychlejšího vyhodnocení vlastností. Tento nový typ obrázku je možné získat z původního pomocí několika operací pro každý pixel. Po dokončení převodu je možné spočítat kteroukoliv z vlastností v libovolném měřítku v konstantním čase. Druhou komponentou je zde jednoduchý a efektivní klasifikátor, který je sestaven pomocí výběru malého počtu důležitých vlastností z rozsáhlé knihovny. Každý obrázek obsahuje velké množství těchto prvků, daleko více než pixelů. Z důvodu zajištění rychlého průběhu klasifikace je proto nutné vynechat většinu častých prvků a soustředit se na menší množství kritických vlastností.

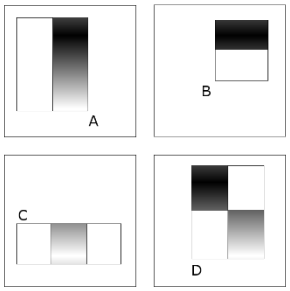
Výběr vlastností probíhá pomocí algoritmu AdaBoost[35], který omezuje každý klasifikátor pouze na jeden prvek. Poslední zásadní komponentou je metoda pro postupné kombinování více komplexních klasifikátorů do kaskádové struktury, což vede k dramatickému zvýšení rychlosti detekce pomocí zaměření pozornosti na slibné oblasti snímku. Slibná oblast je určena pomocí rychlého odhadu, kde na obrázku by se mohla tvář vyskytovat. Komplexnější zpracování je realizováno právě pro tyto slibné oblasti.

Tato podokna (slibné oblasti), pokud nejsou odmítnuty počátečním klasifikátorem, jsou zpracovávány sekvencí klasifikátorů, přičemž každý je o něco komplexnější než ten předchozí. Pokud kterýkoliv klasifikátor odmítne podokno, k dalšímu zpracování již nedochází. Struktura kaskádového detekčního procesu odpovídá v podstatě degenerativnímu rozhodovacímu stromu[36].

[33] *Klasifikátor*: popis jednotlivé charakteristiky obličeje (obočí, nos, ústa) ve formě hran/linií.
[34] *Integrální obrázek*: digitální reprezentace obrazu tak, že každý bod představuje součet intenzit předchozích pixelů doleva a nahoru (pravý spodní bod tedy obsahuje součet všech pixelů obrázku).
[35] *AdaBoost*: Zkratka pro Adaptive Boosting neboli adaptivní zrychlení je algoritmus strojového učení, který je podrobněji popsán zde [21].
[36] *Degenerativní rozhodovací strom*: První klasifikátor eliminuje velké množství negativních vyhodnocení pomocí velmi snadného zpracování. Následující vrstvy pokračují v eliminaci vždy o trochu složitějším způsobem. Po několika iteracích je počet pod-oken značně redukován.

Jak již bylo zmíněno, tento princip využívá klasifikaci snímků na základě hodnot jednoduchých prvků. Pro použití prvků namísto pixelů existuje vícero důvodů, mezi ty hlavní patří například schopnost na vyžádání se naučit doménové znalosti, které by se složitě učily konečným množstvím trénovacích dat. Hlavní výhodou v tomto případě ale je mnohem rychlejší zpracování prvků než samotných pixelů.

Využívají se zde 3 druhy vlastností. Nejzákladnější je vlastnost dvou-obdélníková (Obrázek 13 – A, B), která označuje rozdíl mezi součty intenzit pixelů ve dvou obdélníkových oblastech. Oblasti mají stejnou velikost i tvar a jsou horizontálně nebo vertikálně přilehlé. Troj-obdélníková vlastnost počítá součet ve dvou venkovních odečtený od součtu centrálního obdélníku (Obrázek 13 – C) a čtyř-obdélníková vlastnost počítá rozdíl mezi intenzitami diagonálními páry obdélníků (Obrázek 13 – D). Pokud uvažujeme základní rozlišení detektoru 24x24, set obdélníkových vlastností obsahuje 160 000 vlastností.




**Obrázek 13 – Obdélníkové vlastnosti v detekčních oknech [22]**

Tyto obdélníkové vlastnosti mohou být velmi rychle vypočítány pomocí pokročilé reprezentace obrázku, označované jako integrální obrázek. Tento na pozici x, y obsahuje součet pixelů nahoru a vlevo od souřadnic, včetně:

$$\boldsymbol{ii}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\boldsymbol{x}'\leq \boldsymbol{x},\boldsymbol{y}'\leq \boldsymbol{y}} \boldsymbol{i}(\boldsymbol{x}',\boldsymbol{y}'), \qquad \textbf{( 11 )}$$

Kde ii je integrální, i je originální obrázek a (x, y) jsou souřadnice v příslušném obrázku.

Pokud tedy máme výše uvedených 160 000 prvků v každém podokně, pracujeme s číslem daleko větším, než je počet pixelů. Přestože každá vlastnost může být spočítána velmi efektivně, počítání celé sady může být nepřiměřeně náročné. Hypotéza této metody spočívá v kombinaci malého počtu vlastností do podoby efektivního klasifikátoru. Důležitou výzvou tak zůstává hledání těchto kombinovatelných vlastností.

Tuto roli zde zastává algoritmus AdaBoost stejně jako roli trénování klasifikátoru. Tento algoritmus původně sloužil pouze jako podpora výkonu klasifikátorů jednoduchých

samoučících se algoritmů. Takovémuto algoritmu se také říká tzv. slabý samouk (slabý proto, že nikdy nečekáme dokonalou klasifikaci tréninkových dat). Samoučící algoritmus například hledá v setu a výsledkem je klasifikátor, který má nejnižší množství klasifikačních chyb. Slabší algoritmus je podpořen tím, že je zavolán, aby vyřešil sérii samoučících problémů.

Formální záruky samoučící procedury jsou poměrně vysoké. Na základě testování se počet trénovacích chyb u silných algoritmů blíží nule s rostoucím počtem trénovacích kol. Konvenční podpůrná procedura AdaBoost může být přirovnána k procesu chamtivého výběru vlastností[37]. Uvažujme problém podpory, ve kterém je kombinován velký počet klasifikačních funkcí pomocí váženého většinového hlasování. Důležité tedy je přiřadit velkou váhu k dobré klasifikační funkci a nízkou váhu k té špatné.

Pozorná kaskáda popisuje princip konstrukce kaskády klasifikátorů za účelem zvýšení výkonu detekce a snížení výpočetního času. Princip spočívá ve využití jednodušších klasifikátorů pro odmítnutí většiny podoken před tím, než jsou zavolány komplexnější, pro docílení co možná nejnižšího počtu falešně pozitivních výsledků. Jednotlivé fáze kaskády jsou sestaveny pomocí trénování klasifikátorů za pomoci AdaBoost. Z jednoduchého dvou-prvkového silného klasifikátoru můžeme dostat efektivní klasifikátor tváří pomocí úpravy prahu silného klasifikátoru, což vede ke snížení počtu falešně negativních výsledků. Nižší hodnota prahu má za následek vyšší procento detekcí, ale i vyšší výskyt falešně pozitivních výsledků [22].

Dvou-prvkový klasifikátor ani zdaleka nesplňuje požadavky na systém pro detekci tváří. Může však sloužit k redukci počtu pod-oken pro další zpracování pomocí vyhodnocení obdélníkových vlastností, vypočítání jednoduchých klasifikátorů pro každou vlastnost (vyžaduje 1 prahovou operaci pro každou vlastnost) a kombinování slabých klasifikátorů (vyžaduje násobení, sčítání a prahovou operaci).

## 3.5 Hledání obličeje pomocí histogramu orientovaných gradientů

Tato metoda (HOG) je založená na vyhodnocování důkladně normalizovaných lokálních histogramů gradientových orientací[38] obrázku. Základní myšlenka stanovuje, že lokální vzhled a tvar objektu může být velmi dobře charakterizován rozdělením gradientu lokální intenzity

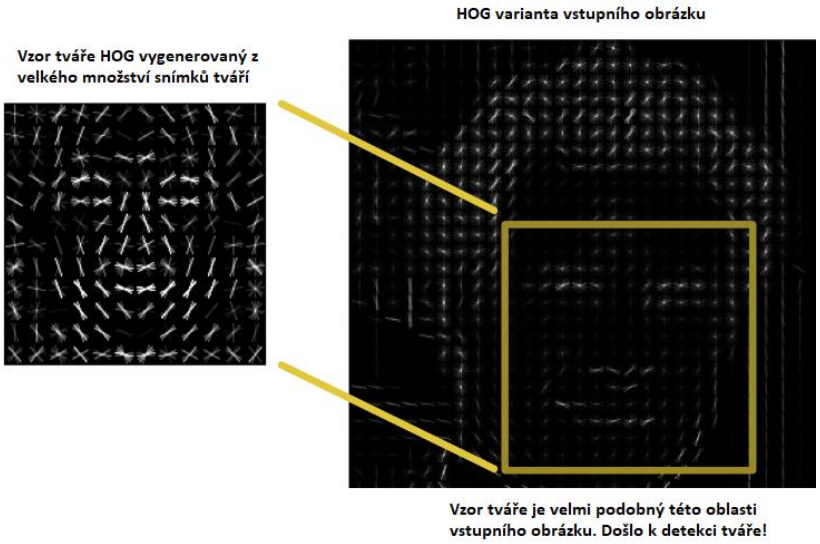
[37] *Chamtivý výběr vlastností*: Algoritmus, který v každé iteraci vybírá nejlepší nebo odstraňuje nejhorší vlastnosti.
[38] *Gradientová orientace* – znázornění toku (směru růstu) světla od světlých oblastí k tmavým.

nebo hranové orientace, a to dokonce bez přesné znalosti korespondujícího gradientu nebo polohy hran.

V praxi je tento postup aplikován pomocí rozdělení obrázku do malých oblastí (tzv. buněk), z čehož pro každou je vytvořen histogram gradientové nebo hranové orientace všech obsažených pixelů. U každého pixelu se vyberou pixely s tímto přímo sousedící a jsou porovnány jejich odstíny. Na základě tohoto porovnání je nakreslena šipka, směřující k tmavší části obrázku. Provedením tohoto kroku pro každý pixel dojde ke kompletnímu nahrazení všech pixelů šipkami, které se nazývají gradienty a zobrazují tok od světlých k tmavým oblastem v rámci celého snímku [23][25].

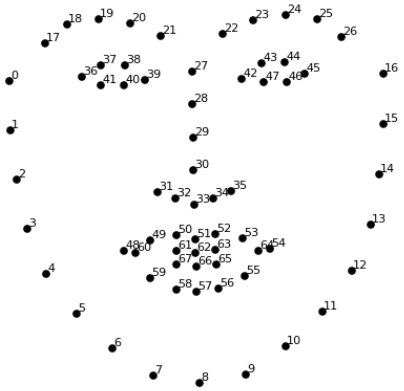
Pokud by probíhala analýza samotných pixelů, nacházely by se odlišné hodnoty u velmi tmavých a velmi světlých obrázků stejné osoby. Zaměřením se na směr změny jasu lze dosáhnout naprosto stejné reprezentace pro oba typy snímků. Tento krok však poskytuje přílišnou míru detailu, která je pro účely detekce obličeje zbytečná. Pro jakési zobecnění se využívá princip hromadění energie lokálních histogramů větší oblasti o velikosti 16x16 pixelů (velikost vychází z rozsáhlého testování pro dosažení nejlepších výsledků). V každém z těchto bloků proběhne sčítání gradientových bodů v každém z 8 hlavních směrů a nahrazení šipkou směru s nejvyšším zastoupením. Výsledkem je jednoduchá reprezentace, zachycující základní struktury (Obrázek 14 – vpravo) [25].

Pro nalezení tváří v tomto HOG snímku stačí najít část obrázku, která je nejvíce podobná naučenému HOG vzoru, získaného z trénovacího procesu (Obrázek 14).




**Obrázek 14 – Hledání HOG vzoru v HOG snímku [25]**

Zdánlivým problémem může být detekce na snímku, kde jsou obličeje nějakým směrem otočené. Pro další procesy je podstatné, aby oči a ústa byly vždy na stejném místě ve snímku. K tomuto se zde využívá odhad tvářových mezníků, přičemž zde bude popsána metoda vynalezená Vahidem Kazemi a Josephine Sullivan[39] v roce 2014. Základní myšlenka spočívá v předpokladu existence 68 specifických bodů (mezníků) v každém obličeji (vrchol brady, vnější hrana každého oka, vnitřní hrana každého obočí atd.). Stačí tedy pouze naučit algoritmus tyto body rozpoznat ve kterékoliv tváři[24][25].



**Obrázek 15 – 68 mezníků tváře [25]**

Po nalezení těchto mezníků stačí natočit nebo oříznout obrázek, případně změnit jeho velikost pro dosažení maximálního možného vycentrování.

## 3.6 Shrnutí a porovnání detekčních algoritmů

Jak již bylo nastíněno v úvodu kapitoly, detekce tváře je podstatný, ne však úplně jednoduchý proces. Výjimkou jsou případy, kdy máme kontrolované pozadí. Takové ideální podmínky však v praxi nejsou běžné, proto bylo potřeba přijít s pokročilejšími způsoby. Obecně platí, že postupem času se přístupy k této problematice stávaly robustnější a spolehlivější. Často i proto, že byly v podstatě založené na některém z předchozích postupů, ale jednalo se o zdokonalení, či jen vyřešení určitých nedostatků.

Hledání tváře podle barvy kůže je sice poměrně jednoduché na implementaci, nefunguje to však se všemi barvami kůží a často se vyskytují problémy při různých osvětleních stejného obličeje. Ze snahy o potlačení nedostatků vzniklo v průběhu času více implementací, mezi které patří například detekce v barevných obrázcích pomocí PCA, detekce barvy kůže při měnících se světelných podmínkách nebo detekce tváře v barevných snímcích

[39] Metoda podrobněji popsaná v publikaci [24]

s komplexním pozadím. Tento přístup byl také experimentálně kombinován s jinými, zejména s 3 D snímáním a odstraňováním pozadí.

O něco spolehlivější je přístup, založený na detekci pohybu ve snímku, jelikož v reálném světě dochází na každém obličeji neustále k určitému pohybu (mimo jiné například mrkání). Problém zde však může nastat, když se na obrázku (respektive na snímku z videozáznamu, či živého přenosu) vyskytují i jiné, pohybující se objekty v pozadí. Nejen pohybující se, ale celkově objekty v pozadí (tedy členitý snímek, kde se nenachází pouze jedna tvář) představuje největší výzvu pro počítačové vidění a automatickou detekci objektů.

Největším průlomem a doposud jedním z nejlepších přístupů je využití kaskádových klasifikátorů (postup využit i v této diplomové práci) na základě Haarových vlastností. Dokonalejším přístupem je aktuálně pouze HOG za využití hloubkového učení, který kromě detekce umožňuje i samotné porovnání (tedy kompletní rozpoznání).

# 4 Vybrané biometrické porovnávací algoritmy

Hledání odpovídajících bodů mezi dvěma snímky jedné scény nebo jednoho objektu je součástí mnoha aplikací, zaměřených na počítačové vidění. Hledání těchto bodů se dá rozdělit do 3 hlavních kroků. Nejprve jsou vybrány tzv. zájmové body ve výrazných částech obrázku, jako například hrany, skvrny nebo křížení. Nejhodnotnější vlastností detektoru je opětovná použitelnost, která označuje schopnost nalezení stejných fyzických zájmových bodů za různých zobrazovacích podmínek [32].

Dále je sousedství každého z těchto bodů reprezentováno pomocí vektoru vlastností. Tento tzv. deskriptor[40] musí být velmi specifický a zároveň odolný vůči šumu, detekčnímu přemístění a geometrickým a fotometrickým deformacím. Třetím krokem je porovnání deskriptorů mezi různými snímky, které je založeno na vzdálenosti mezi vektory (například Mahalanobisova[41] nebo Euklidovská vzdálenost[42]). Rozměry vektorů mají zásadní vliv na rychlost, ale i na věrohodnost porovnání [32].

## 4.1 Algoritmus EigenFaces

V tomto oddíle bude popsán jeden z prvních funkčních přístupů k rozpoznání tváře, který však od svého původního vývoje doznal velké spousty změn a rozšíření. EigenFaces v překladu znamená *vlastní tváře* a v kontextu tohoto algoritmu bude v této kapitole využíváno slovo *vlastní* jako překlad slova *eigen*. Podkladem pro tuto část bude zdroj [28].

Tato metoda označuje vzhledově orientovaný přístup rozpoznávání tváře, který hledá odchylky ve sbírce snímků obličeje a tyto využívá pro zašifrování a následné porovnání tváří jako celků (na rozdíl od přístupu založeném na částech nebo vlastnostech obličeje). Označení *vlastní tváře* se konkrétně používá pro hlavní komponenty rozdělení tváří, nebo také pro tzv. *vlastní* vektory[43] kovarianční matice[44] sady snímků tváří, kde obrázek obsahující N pixelů je považován za bod (nebo také vektor) v N-rozměrném prostoru. Metoda využívající hlavní komponenty pro reprezentaci lidské tváře byla vyvinuta Sirovichem

[40] *Deskriptor*: jednoznačný unikátní popisovací vektor pro charakteristické vlastnosti tváře.
[41] *Mahalanobisova vzdálenost*: vzdálenost mezi bodem P a rozdělením D. Blíže popsána zde [34].
[42] *Euklidovská vzdálenost*: vzdálenost mezi 2 body v Euklidovském prostoru.
[43] *EigenVector – vlastní vektor*: nenulový vektor, jehož směr se po transformaci nezmění.
[44] *Kovarianční matice* náhodné veličiny X o N složkách: čtvercová reálná matice o rozměru N x N, jejíž prvek s indexy i, j obsahuje kovarianci i-té a j-té složky náhodné veličiny X.

a Kirbym v roce 1987 a poprvé využita pro detekci a rozpoznání obličeje Turkem a Pentlandem v roce 1991.

Stále je možné najít případy využití tohoto algoritmu, i když často pouze jako referenční porovnávací metodu pro demonstraci minimálních výkonnostních požadavků systému. První ze dvou hlavních funkcí je extrakce relevantních informací o obličeji, které mohou, ale také nemusí být přímo spojené s lidským vnímáním tváře (např. oči, nos nebo rty). Přístup zde využitý se zaměřuje na variace mezi snímky obličeje. Druhá funkce pak zahrnuje efektivní reprezentaci tváře, tedy minimalizaci výpočtů a prostorové komplexity. Každá tvář může být popsána pomocí nízkého počtu parametrů.

Před samotným generováním *vlastních tváří* musí být všechny snímky normalizovány[45], z hlediska umístění očí a úst, a následně převzorkovány na stejné rozlišení. K získání *vlastní tváře* je využita analýza hlavních komponent[46], která se snaží nalézt hlavní osy, které stanovují ortonormální souřadnicový systém pro zachycení většiny odchylek v datech. Pro tyto účely uvažujme M snímků obličeje o rozměrech h x w, přičemž každý je transformován na vektor o velikosti D (h*w) a umístěn do sady $\{\Gamma_1, \Gamma_2, \dots, \Gamma_M\}$. Obrázky by v této fázi již měly být patřičně upraveny, především co se týče velikosti a zarovnání a pozadí by mělo být jednotné, či odebrané.

Každá tvář se od té průměrné (která je definována jako

$$\boldsymbol{\Psi} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\Gamma_i} \qquad (12)$$

) liší vektorem

$$\boldsymbol{\Phi_i} = \boldsymbol{\Gamma_i} - \boldsymbol{\Psi}. \qquad (13)$$

Z toho vyplývá, že kovarianční matice $C \epsilon R^{DxD}$ je definovaná rovnicí

$$\boldsymbol{C} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\Phi_i}\boldsymbol{\Phi_i^T} = \boldsymbol{AA^T}, \qquad (14)$$

$$\text{kde } \boldsymbol{A} = \{\boldsymbol{\Phi_1}, \boldsymbol{\Phi_2}, \dots, \boldsymbol{\Phi_M}\}\boldsymbol{\epsilon R^{DxD}}. \qquad (15)$$

Určení vlastních vektorů této kovarianční matice je téměř neřešitelný úkol pro obvyklé rozměry obrázků, kde $D \gg M$. Nicméně pro efektivní vypočítání těchto vektorů je možné nejprve spočítat vektory mnohem menší M x M matice $A^T A$. Tyto jsou definovány jako

[45] *Normalizace*: převedení detekované tváře na standardizovanou formu (rozlišení, jas, perspektiva, natočení).
[46] *Analýza hlavních komponent*: transformace sloužící k dekorelaci dat (také snížení dimenze s co nejmenší ztrátou informace). Tuto metodu je možné nastudovat zde [35].

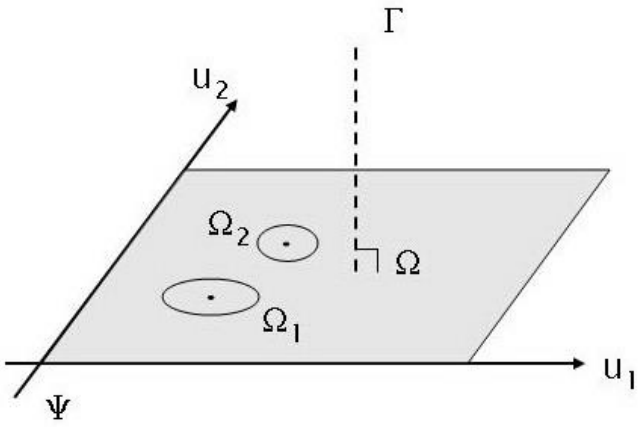
$V = (v_1, v_2, \dots, v_r)$, kde r je hodnost matice A a $v_i$ jsou vlastní vektory matice $A^T A$ spojené s vlastními hodnotami[47] $\lambda_i$.

Matice *vlastních* hodnot a *vlastních* vektorů jsou $\Lambda = diag\{\lambda_1, \lambda_2, \dots, \lambda_r\}$, $pro\ \lambda_1 \geq \lambda_2 \geq \cdots > 0$ a $U = A * V * \Lambda^{-1/2}$, kde $U = \{u_i\}$ označuje kolekci *vlastních tváří* a $\lambda$ patřičnou vlastní hodnotu. Pomocí Singulárního rozkladu matice[48] A dostaneme faktorizaci

$$\boldsymbol{A = U\Delta V^T = \sum \delta_i u_i v_i^T} \qquad \textbf{( 16 )}$$

, kde U a V jsou ortonormální matice (ortonormální vektory jsou ortogonální, to jest pravoúhlé nebo nezávislé, a zároveň mají jednotkovou délku). Diagonální matice Δ obsahuje singulární hodnotu $\delta_i$, která může být kladná, nulová i záporná. Při spojení s analýzou hlavních komponent jsou U a V matice *vlastních* vektorů $AA^T$ a $A^T A$ a platí $\delta_i = \lambda_i$ pro každou hodnotu i.

Aplikací výše uvedeného algoritmus v podstatě vyměří m-dimenzionální podprostor z originálního obrázku pomocí výběru podmnožiny *vlastních* vektorů $\widehat{U}$, spojených s m největšími vlastními hodnotami. Výsledkem je tzv. tvářový prostor, jehož počátkem je průměrný obličej a osami jsou *vlastní tváře*. Provedení detekce nebo rozpoznání je možné vypočítáním vzdálenosti v rámci tvářového prostoru nebo od něj.




**Obrázek 16 – Vizualizace dvourozměrného tvářového prostoru [28]**

Jelikož výše uvedený prostor definuje prostor obličejových snímků, proces detekce může být charakterizován jako detekce částí obrázku, které leží blízko tohoto prostoru. Jinými slovy projekční vzdálenost $\delta$ by neměla překračovat prahovou hodnotu $\theta_\delta$. Vzdálenost bodu k prostoru $\delta$ je odstup snímku obličeje od jeho projekce na tvářový prostor a je možné

[47] *EigenValue – Vlastní hodnota*: koeficient použitý při transformaci.
[48] *Singulární rozklad matice*: rozklad komplexní nebo reální matice na maticový součin.

ho spočítat jako $\delta = \left\|(I - \widehat{U}\widehat{U}^T)(\Gamma - \Psi)\right\|$, kde I označuje matici identity. Vzdálenost mezi obrázkem a jeho prostorovou projekcí je mnohem menší pro snímek obličeje než pro snímek neobsahující obličej.

Nová tvář je do prostoru promítnuta váženým vektorem

$$\boldsymbol{\Omega = \widehat{U}^T(\Gamma - \Psi)} \qquad \textbf{( 17 )}$$

, kde $\widehat{U}$ je sada výrazných *vlastních* vektorů. Tento vážený vektor je v podstatě reprezentace nové tváře v tvářovém prostoru. Jednoduchým způsobem pro určení, ke které tvářové třídě $\Gamma$ obličej patří je minimalizace Euklidovské vzdálenosti (vzdálenosti dvou bodů v Euklidovském prostoru)

$$\boldsymbol{\epsilon_k = \|\Omega - \Omega_k\|} \qquad \textbf{( 18 )}$$

, kde $\Omega_k$ je vážený vektor, reprezentující tvářovou třídu s pořadím k. Tvář $\Gamma$ je považována za náležitou k třídě s pořadím k, pokud minimální Euklidovská vzdálenost je menší než přednastavená prahová hodnota $\theta_\epsilon$, v opačném případě je tvář klasifikována jako neznámá.

Přestože tato metoda byla velkým průlomem ve vizuálním rozpoznávání v rámci počítačového vidění, je zde vysoká citlivost na změny osvětlení, velikosti, postoje a výrazu v obličeji. Pro správnou funkčnost musí obličej být snímám zepředu z přiměřené vzdálenosti, v podobných světelných podmínkách a s neutrálním výrazem. Pro zvládnutí nestálosti těchto podmínek vznikla velká spousta rozšíření tohoto algoritmu.

První myšlenkou byla modifikace na základě úhlu pohledu, ve které se nejprve sestaví sada separovaných *vlastních* prostorů, přičemž každý zachycuje variaci snímků obličeje z obvyklého pohledu. Prvním krokem při klasifikaci nové tváře je určení pohledu, čehož lze dosáhnout pomocí kalkulace minimální vzdálenosti $\delta_l$ k individuálnímu *vlastnímu* prostoru

$$\boldsymbol{\delta_l = \left\|(I - \widehat{U_l}\widehat{U_l}^T)(\Gamma - \Psi_l)\right\|} \qquad \textbf{( 19 )}$$

, kde $\Psi_l$ a $\widehat{U_l}$ jsou průměrná tvář a *vlastní tvář* v pohledu l. Jakmile dojde ke stanovení pohledu, pokračuje se v projekci a rozpoznání jako v klasickém průběhu algoritmu.

Dalším přístupem jsou tzv. *vlastní* vlastnosti nebo modulární *vlastní tváře*. Postup spočívá v počítání *vlastních* prostorů pro tvářové rysy, což přináší *vlastní* oči, *vlastní* nos a *vlastní* ústa. Myšlenka je založená na rozšíření *vlastních tváří* o *vlastní* vlastnosti, což může být vnímáno jako jakýsi vrstvený přístup, kde hrubá reprezentace (*vlastní tvář*) celé tváře je rozšířena

o reprezentaci obličejových prvků. Tento přístup je rovněž využíván pro reprezentaci kousku tváře, který je později kombinován ve více klasifikátorovém přístupu.

O velké škále rozšíření této techniky svědčí i fakt, že mnoho následujících algoritmů na tento navazuje (např. FisherFaces nebo Kernel EigenFaces). Využití analýzy hlavních komponent je skvělý způsob pro reprezentaci šablony, ne však už tolik pro klasifikaci. Proto pozdější algoritmy využívají jiné přístupy (například FisherFaces využívá rozdělení tváří na základě Fisher kritéria). Jelikož analýza hlavních komponent zachycuje pouze statistiku druhého řádu (např. kovarianční matici) snímků obličeje, byla vyvinuta metoda Kernel EigenFaces, využívající Kernelovu analýzu hlavních komponent. Technika vlastních tváří byla aplikována i do jiných oblastí, jako například EigenVoice (*vlastní* hlas) nebo EigenGain (*vlastní* chůze) [33].

## 4.2 Algoritmus LBPH

Tento oddíl bude zaměřen na algoritmus LBPH, což je zkratka pro Local Binary Patterns Histograms (Histogramy Lokálních Binárních Vzorů). Tato metoda byla poprvé popsána v roce 1994 a stejně jako metoda předešlá byla postupně vylepšována. Hlavním zdrojem tohoto oddílu je [29].

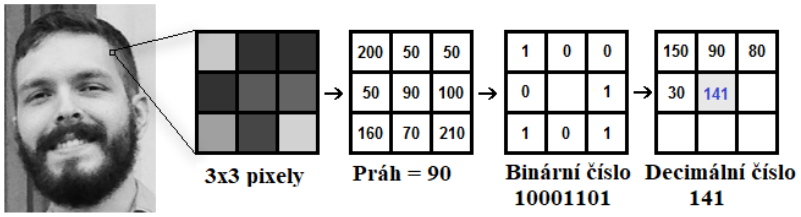
LBP tedy označuje Local Binary Pattern (lokální binární vzor), což je jednoduchý, přesto velmi efektivní strukturální operátor, který označuje jednotlivé pixely obrázku pomocí prahování sousedících pixelů a následného převodu výsledku na binární hodnotu. Lepších výsledků je možné dosáhnout kombinací s deskriptorem HOG (popisovaným v předchozí kapitole) a právě touto kombinací vznikl algoritmus LBPH. Výsledkem kombinace těchto procesů je jednoduchý datový vektor.

Tato technika pracuje se 4 parametry, z čehož prvním je poloměr (reprezentace okolí centrálního pixelu, většinou nastaven na hodnotu 1), díky kterému dochází ke konstrukci kruhového lokálního binárního vzoru (Obrázek 17 – druhá část zprava). Pro tento proces je dále potřeba znát počet sousedů (čím vyšší počet, tím vyšší výpočetní náročnost, obvykle roven 8). Tyto dva parametry mohou nabývat různých hodnot díky bilineární[49] interpolaci[50]. Pokud je vybrán bod v pomezí pixelů, je určena nová hodnota na základě 4 nejbližších pixelů

[49] *Bilineární interpolace*: Rozšíření lineární interpolace pro interpolaci funkce dvou proměnných na pravidelnou prostorovou mřížku (jeden směr po druhém).
[50] *Interpolace*: nalezení přibližné hodnoty funkce v daném intervalu, je-li známa hodnota jen v některých jiných bodech tohoto intervalu.

(2 x 2). Posledními dvěma parametry jsou tabulky X a Y, tedy počty buněk v horizontálním a vertikálním směru. Čím více buněk, tím lepší tabulka a vypovídající hodnota výsledného vektoru vlastností (nejčastěji se vyskytuje hodnota 8).

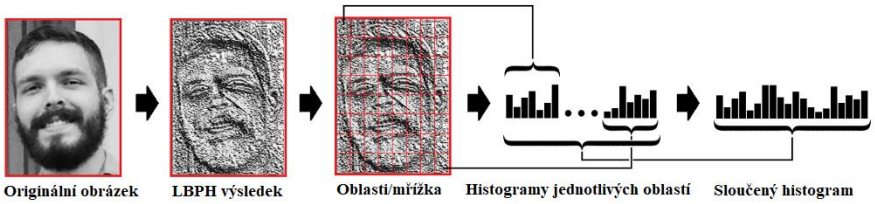
Prvním výpočetním krokem je tvorba pokročilého obrázku, který lépe charakterizuje originální snímek zvýrazněním charakteristických rysů tváře. K tomuto je využit koncept posuvného okna za pomocí parametrů poloměru a sousedů. Z obrázku v odstínech šedé barvy dojde k výběru části o určité velikosti (dle zmíněných parametrů, pro poloměr 1 a počet sousedů 8 to bude 3 x 3 pixely), reprezentované jako matice intenzit obsažených pixelů, každá v rozsahu od 0 do 255. Hodnota prostředního stanovuje hraniční práh, pomocí kterého budou definovány nové hodnoty pro všechny sousedy. Pro pixely s hodnotou vyšší nebo rovnou se nastaví nová hodnota 1, pro pixely s nižší hodnotou hodnota 0. Tento postup je znázorněn na následujícím obrázku (Obrázek 17 – kromě poslední části).




**Obrázek 17 – Tvorba pokročilého obrázku v algoritmu LBPH [29]**

Dalším krokem je spojení těchto binárních hodnot ze všech pozic (kromě centrální – prahové) do nové binární hodnoty. Existuje více přístupů, ale ať dojde ke spojení hodnoty podle / proti směru hodinových ručiček nebo řádek po řádku, konečný výsledek bude stejný. Tato binární hodnota je následně převedena na hodnotu decimální a nastavena jako nová hodnota centrálního pixelu. Tento postup se opakuje pro každý pixel obrázku, dokud nedojde k nahrazení každého pixelu pro dosažení obrázku lépe popisujícího charakteristiky původního snímku (tzv. pokročilý obrázek – Obrázek 18 LBPH výsledek).

Nyní dojde k využití zbylých dvou parametrů, tedy tabulky X a tabulky Y. Pomocí těchto je obrázek rozdělen do určité mřížky (Obrázek 18 oblasti/mřížka). Z každé oblasti mřížky je vytvořen histogram četností intenzit pixelů (pro každou intenzitu v rozsahu od 0 do 255). Jednotlivé histogramy jsou později sloučeny do finálního velkého histogramu, který reprezentuje charakteristiky originálního obrázku.




**Obrázek 18 – Práce s histogramy v algoritmu LBPH [29]**

Pro samotný proces rozpoznání obličeje je, jako u každého algoritmu, podstatné trénování pomocí referenční sady dat. K tomuto je potřeba sada snímků lidí, které by systém měl být schopný rozpoznat a k nim přiřazený příslušný unikátní identifikátor (pokud existuje více snímků jedné osoby, měly by obě mít stejný identifikátor). Pro každý obrázek z trénovací sady je vytvořen histogram, reprezentující zadaná data. Pro vstupní obrázek (u kterého má dojít k identifikaci) je provedena stejná operace vytvoření histogramu. Dále se hledá obrázek s histogramem nejbližším tomu, který odpovídá vstupnímu snímku. Pro hledání nejbližšího histogramu je možné využít vícero přístupů – Euklidovská vzdálenost, chí kvadrát, absolutní hodnota a další. Předpokládejme využití Euklidovské vzdálenosti, jelikož se jedná asi o nejrozšířenější variantu. Ta se dá spočítat pomocí vzorce:

$$\boldsymbol{D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}} \qquad \textbf{( 20 )}$$

Výsledkem algoritmu je identifikátor osoby s nejbližším histogramem a hodnota výsledné vzdálenosti. Tato hodnota se dá považovat za jakousi míru důvěryhodnosti výsledku (nutno podotknout, že čím nižší hodnota, tím lepší výsledek, jelikož to znamená menší vzdálenost mezi histogramy). Pro tyto účely lze rovněž využít prahové hodnoty, která automaticky vyhodnotí spolehlivost výsledku. Pokud bude výsledná vzdálenost nižší než práh, algoritmus úspěšně rozpoznal osobu, v opačném případě nebyla nalezena dostatečná shoda.

## 4.3 Algoritmus FisherFaces

Přestože z hlediska reprezentace dat (což je jeden z klíčových problémů počítačového vidění, rozpoznání vzorů i strojového učení) se může zdát metoda *vlastních tváří* (viz oddíl 4.1) optimální vzhledem k posuzování odchylky na snímku obličeje, z hlediska klasifikace nemusí vždy dojít k dosažení požadovaných výsledků. Z tohoto důvodu byla navržena alternativa R. A. Fishermanem roku 1936. Díky jménu autora myšlenky vznikl název FisherFaces pro algoritmus, kterému se bude věnovat tento oddíl s odkazem na zdroj [30], kde je rovněž dokázáno tvrzení o nedostatečnosti řešení pomocí *vlastních tváří*.

*Vlastní vektory* získané z *vlastních tváří* ve stejnojmenném algoritmu odpovídají řešení metodou nejmenších čtverců[51]. Právě tato metoda může být kritickým prvkem při procesu klasifikace. V těchto případech je totiž potřeba najít podprostor, který mapuje vzorové vektory stejné třídy v jediném bodě reprezentace charakteristického rysu, přičemž ostatní třídy se nacházejí tak daleko, jak jen je to možné. Za tímto účelem byla odvozená technika známá jako LDA (Linear Discriminant Analysis – Lineární Diskriminační Analýza[52]). Při použití LDA pro nalezení pod-prostorové reprezentace sady snímků obličeje jsou výsledkem bázové vektory definující tento prostor, označované jako FisherFaces.

V praxi je možné se setkat s více než 2 třídami. Je proto potřeba minimalizovat vnitro-třídní variaci, která může být odhadnuta za využití vnitro-třídní matice rozptylu. Ta je určena vzorcem

$$\boldsymbol{S_w = \sum_{j=1}^{C} \sum_{i=1}^{n_j} (x_{ij} - \mu_j)(x_{ij} - \mu_j)^T,} \qquad \textbf{( 21 )}$$

kde $x_{ij}$ je i-tý vzorek třídy s číslem j, $\mu_j$ označuje průměr třídy číslo j a $n_j$ počet vzorků v třídě j. Obdobně může být spočítána mezi-třídní variace (kterou je potřeba maximalizovat), tedy

$$\boldsymbol{S_b = \sum_{j=1}^{C} (\mu_j - \mu)(\mu_j - \mu)^T,} \qquad \textbf{( 22 )}$$

kde $\mu$ reprezentuje průměr všech tříd. Dále je potřeba najít bázové vektory V, pro které je $S_w$ minimální a $S_b$ maximální. V je zde označení pro matici, jejíž sloupce $v_i$ jsou bázové vektory, definující pod-prostor. Tyto jsou určeny pomocí

$$\frac{|V^T S_b V|}{|V^T S_w V|}. \qquad \textbf{( 23 )}$$

Řešení tohoto problému je možné dosáhnout dekompozicí zobecněných *vlastních hodnot*

$$\boldsymbol{S_b V = S_w V \Lambda,} \qquad \textbf{( 24 )}$$

kde V je matice *vlastních* vektorů a Λ je diagonální matice odpovídajících *vlastních hodnot*.

*Vlastní vektory* matice V přiřazené nenulovým *vlastním* hodnotám jsou tzv. FisherFaces. Těchto se zde může vyskytovat maximálně C-1, což vychází z definice $S_b$ (tato mezi-třídní diferenciace je definována jako kombinace C vektorů charakteristických rysů, z čehož každý vytváří podprostor o nejvýše C-1 dimenzích). Tato rovnost platí pouze pokud jsou vektory vzájemně lineárně nezávislé.

[51] *Metoda nejmenších čtverců*: matematicko-statistická metoda pro aproximaci řešení přeurčených soustav rovnic. Možno nastudovat zde [36].
[52] *Lineární diskriminační analýza*: metoda pro hledání lineárních kombinací vlastností, které charakterizují dvě nebo více tříd objektů nebo událostí. Více informací zde [37].

Pro získání výše zmíněných FisherFaces je potřeba získat inverzi mezi-třídní variace. Pokud nastane situace, kdy vzorové vektory vlastností jsou definovány v p-rozměrném prostoru a p je větší než celkový počet vzorů, je variace označována za singulární. Tyto případy lze řešit dvěma způsoby. První možností je promítnutí vzorových vektorů do prostoru analýzy hlavních komponent o r dimenzích (kde r je menší nebo rovno hodnosti diferenciace) a k výpočtu FisherFaces dojde v tomto prostoru. Alternativou je přidání omezujícího pravidla.

Vnitro a mezi třídní metriky mohou být nahrazeny jinými. Podstatné však je, aby nově zvolené měly za účel minimalizaci vnitro-třídní a maximalizaci mezi-třídní variace. Vnitro-třídní matice může být například nahrazena jednoduchou kovariační maticí.

Takto získané FisherFaces jsou založené na předpokladu linearity (jak již bylo uvedeno výše u LDA). Tento předpoklad je splněn pouze v případech, kdy třídy jsou homoskedastické[53] (mají stejnou kovarianci). Tento předpoklad je však v praxi často porušen, proto je nutné mapovat originální heteroskedastický [54]problém do problému homoskedastického. K tomuto mapování existuje více způsobů:

- Kernelova matice -> využití Kernelovy Lineární Diskriminační Analýzy [33],
- Zavedení heteroskedatického způsobu měření vnitro a mezi třídní diferenciace,
- Reprezentace vzorků ve třídách pomocí míchání normálních distribucí,
- Algoritmus analýzy pod-třídních diskriminantů a kernelovská alternativa [33].

Algoritmus FisherFaces byl rozšířen o dvou rozměrnou verzi LDA (2DLDA). Na rozdíl od klasické LDA zde není nutné převádět vstupní vzory na jedno-rozměrné vektory, jelikož dochází k přímému zpracování extrahovaných vlastností ze snímku.

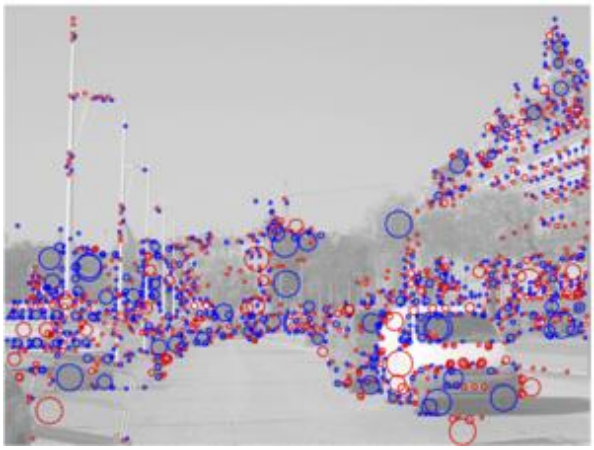
## 4.4 Algoritmus SIFT

Tento algoritmus zavádí nový deskriptor SIFT (Scale Invariant Feature Transform – v měřítku neměnná transformace vlastností), který se využívá pro rozpoznání a hledání shod v obrazu. Invariance[55] spočívá v nezávislosti na přeložení, rotace a změnu velikosti snímku a zároveň v robustním zpracování změny perspektivy nebo intenzity osvětlení. V tomto oddíle

[53] *Homoskedasticita – Homogenita*: všechny proměnné dané sekvence nebo daného vektoru mají stejnou konečnou varianci (rozptyl).
[54] *Heteroskedasticita*: výskyt podmnožin s odlišnými variancemi od ostatních.
[55] *Invariance*: neměnnost, nezávislost.

bude pozornost věnována algoritmu, který využívá tento deskriptor. Informace pochází od prof. Tonyho Lindeberga, ze Švédského Královského Institutu Technologie [31].

Původně obsahoval deskriptor SIFT metodu detekce zájmových bodů ze snímku v odstínech šedi. Princip spočíval ve shromažďování statistik o směrech lokálních gradientů pro jednotný popis struktury obrázku v blízkém okolí bodů zájmu. Tyto body je možné získat z lokálních extrémů DoG (Differences of Gaussians) v rámci stejnojmenné pyramidy. Ta je z původního obrázku získána pomocí postupného vyhlazování a podvzorkování. Výsledek je počítán z rozdílů přilehlých úrovní této pyramidy. Body zájmu jsou pak získány, jakožto hodnoty extrémů vzhledem k prostorovým souřadnicím v obrázku i k měřítku pyramidy.



**Obrázek 19 – Detekce zájmových bodů algoritmem SIFT [31]**

Poloměr kruhů (Obrázek 19) ilustruje vybrané detekční měřítko zájmových bodů. Červené kruhy označují světlé rysy, zatímco modré kruhy označují tmavé. Tento postup může být vnímán jako variace měřítku adaptabilní metody detekce skvrn, kde skvrny, spojené s úrovní měřítka, jsou detekovány z lokálních extrémů měřítkově normalizované Laplaceovy transformace[56].

Je dokázáno, že tento způsob detekce zájmových bodů vede k nezávislosti na měřítku, a to ze třech různých hledisek. Prvním základním faktem je, že tyto body jsou zachovány během měřítkových transformací. Dále jsou úrovně měřítka transformovány vzhledem k rozsahu změny velikosti. Z toho vyplývá, že hodnoty měřítka získané ze zájmových bodů mohou být použity pro normalizaci lokálních sousedů vzhledem ke změně měřítka,

[56] *Laplaceova transformace*: integrální transformace využívaná k řešení obyčejných diferenciálních rovnic. Tato problematika může být nastudována z [38].

což je nezbytné pro splnění měřítkové nezávislosti deskriptoru SIFT. Díky rotační nezávislosti Laplaceovy transformace jsou i zájmové body rotačně nezávislé.

Oba přístupy (DOG i Laplaceova transformace) zahrnují zakomponování kvadratického mnohočlenu do významných hodnot kolem každého lokálního extrému pro lokalizaci lokálního extrému s rozlišením vyšším, než je vzorkovací hustota[57] v prostoru a měřítku. Tato fáze následného zpracování je klíčová pro zvýšení přesnosti odhadu měřítka pro účely měřítkové normalizace.

Kromě detekce struktur připomínajících skvrny a rohy má Laplaceův operátor silné výsledky v okolí hran. K potlačení těchto bodů, které by značně komplikovaly porovnávání snímků, bylo formulováno kritérium z hlediska *vlastních* hodnot Hessovy matice[58], počítané na pozici a v určité vrstvě, příslušné danému bodu zájmu. Toto může být přeformulováno vzhledem ke stopě (trace) a determinantu této matice pro dosažení efektivnějších výpočtů

$$\frac{detHL}{trace^2 HL} = \frac{L_{xx}L_{yy} - {L_{xy}}^2}{\left(L_{xx}+L_{yy}\right)^2} \geq \frac{r}{(r+1)^2}, \qquad (25)$$

kde $r \geq 1$ značí horní hranici limitu povoleného poměru mezi větší a menší vlastní hodnotou. Pro potlačení rysů obrázku s nízkým kontrastem, jsou zájmové body většinou limitovány velikostí odezvy.

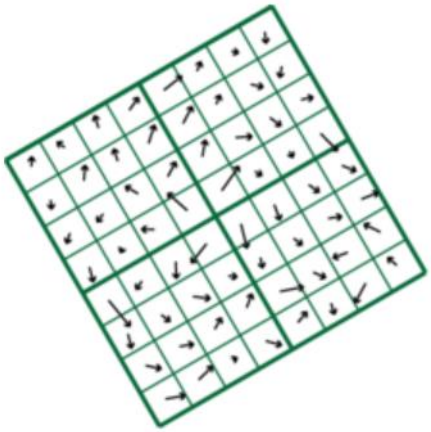
V každém zájmovém bodu je spočítán obrazový deskriptor, nejčastěji jako pozičně závislé histogramy směrů lokálních gradientů v okolí těchto bodů. Pro dosažení nezávislosti na měřítku musí být velikost okolí normalizována způsobem nezávislým na měřítku. Pro dosažení nezávislosti na rotaci musí nejprve být vybrána převládající orientace vektorů gradientů v daném okolí. Tato je použita ke změně orientace mřížky (Obrázek 20), nad kterou je počítán pozičně závislý histogram. Odhad velikosti deskriptoru je založen na detekčním měřítku, kde může být ovlivněn mechanismem pro výběr měřítka v detektoru zájmových bodů.

Výběr preferované orientace probíhá pomocí akumulace histogramu směrů gradientů v okolí zájmových bodů, přičemž tyto směry jsou získány z vektorů gradientů v daném detekčním měřítku. Při výpočtu jsou přírůstky zvažovány pomocí velikosti gradientu a Gaussovy funkce[59] se středem v bodě zájmu a velikostí proporční vůči detekčnímu měřítku. V souhrnném

[57] *Vzorkovací hustota*: počet zaznamenaných vstupních vzorků.
[58] Hessova matice: čtvercová matice druhých parciálních derivací skalární funkce. Dostupné na 104[39].
[59] *Gaussova funkce*: funkce zpravidla využívaná za účelem reprezentace hustoty pravděpodobnosti náhodné proměnné s normálním rozdělením. Problematiku lze nastudovat z [40].

orientačním histogramu je potřeba lokalizovat nejvyšší vrchol pomocí lokální parabolické interpolace[60] kolem maximálního bodu histogramu. Pokud se zde nachází více než jeden vrchol (akceptovány jsou všechny takové, jejichž výška dosahuje alespoň 80 % výšky nejvyššího), dojde k výpočtu několika deskriptorů, podle každého z vrcholů. Pro zvýšení přesnosti se využívá hrubého vzorkování orientací.

Po získání velikosti a orientace je umístěna čtvercová mřížka do oblasti obrázku, se středem v zájmovém bodě, orientací dle hlavního vrcholu histogramu a řádkováním dle detekčního měřítka bodu (nejčastěji se jedná o mřížku 4 x 4). Pro každý bod mřížky je následně vypočítán histogram směrů lokálních gradientů nad okolím bodu mřížky a směry gradientů jsou rozřazeny do 8 hlavních směrů. Přírůstky jsou váženy stejným způsobem jako v předchozím procesu. Pokud tento proces proběhne pro všech 16 bodů mřížky, výsledkem bude deskriptor se 128 dimenzemi pro každý zájmový bod. Tomuto výslednému obrázku se říká deskriptor SIFT (Obrázek 20).



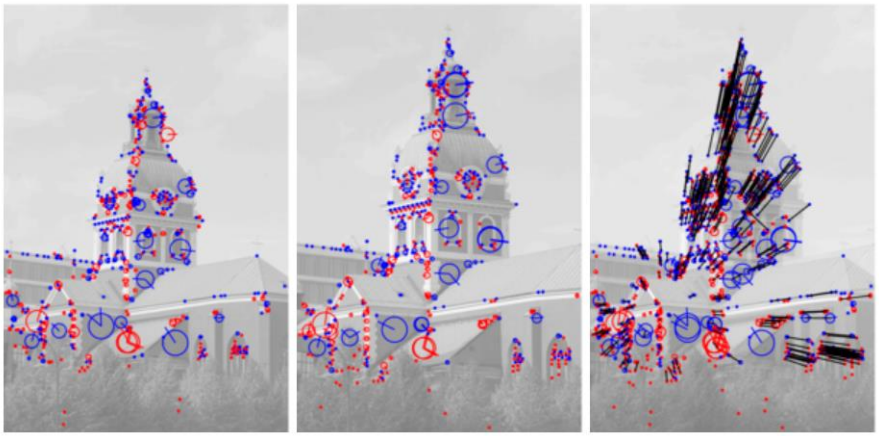
**Obrázek 20 – Deskriptor SIFT [31]**

Nezávislosti na kontrastu je možné dosáhnout pomocí normalizace deskriptoru SIFT do jednotkového součtu. Díky tomuto kroku se stanou vážené vstupy histogramu nezávislými na afinních transformacích[61] snímku kolem zájmových bodů, což zvyšuje robustnost deskriptoru vůči změnám osvětlení. Aby se předešlo situacím, kdy je v deskriptoru přidělen příliš velký důraz lokálnímu naměření vysokého kontrastu,

[60] *Parabolická interpolace*: technika hledání extrémů spojité unimodální funkce postupným nasazením parabol k funkci jedné proměnné ve třech jedinečných bodech.
[61] *Afinní transformace*: základní operace v počítačové grafice (posunutí, otočení, změna měřítka, zkosení, skládání jednotlivých operací).

provádí se normalizace ve dvou krocích, kdy v prvním kroku je hodnota normalizace omezena horní hranicí.

Samotné porovnání dvou obrázků probíhá relativně podobně jako u předchozích algoritmů. Jsou-li k dispozici sady deskriptorů pro dva různé obrázky, dochází k porovnání pomocí hledání příslušného bodu ve druhém obrázku pro každý zájmový bod obrázku prvního. To zde opět obstarává Euklidovská vzdálenost deskriptorů (zde 128 - dimenzionálních vektorů). Pro potlačení dvouznačných shod je zde opět něco, co by se dalo nazvat prahová hodnota, a sice poměr mezi nejbližším a druhým nejbližším bodem nesmí být větší než 0,8.



**Obrázek 21 – Zájmové body dvou různých obrázků a jejich porovnání [31]**

Tento postup by byl však velmi výpočetně náročný při porovnávání s rozsáhlou kolekcí objektů z databáze, jelikož by muselo dojít k porovnání všech deskriptorů. Pro zrychlení je možné využít algoritmus BBF[62] (Best-Bin-First – Nejlepší Zásobník První), hierarchické k-cestné stromy nebo náhodně uspořádané k-cestné stromy.

Prvotní účel spočíval pouze ve spojování příslušných bodů mezi různými obrázky. Později byla zvýšena schopnost deskriptoru v procesech kategorizace, klasifikace textur, zarovnání obrázku a biometrie díky aplikaci husté mřížky. Dalším vylepšením byl přechod z odstínů šedi na barevný obrázek, čehož je možné docílit různými způsoby. Jednou z těchto variant je počítání deskriptorů nad každým ze třech kanálu HSV (Hue, Saturation, Value – Odstín, Saturace, Hodnota), přičemž výsledkem je deskriptor HSV-SIFT o 3 x 128 dimenzích.

[62] *Best-Bin-First*: vyhledávací algoritmus, navržený pro efektivní nazelení přibližného řešení problému hledání nejbližšího souseda v multidimenzionálním prostředí.

Alternativou je zřetězení deskriptoru s váženým odstínem nebo histogramem soupeřících úhlů a následné porovnání složených deskriptorů při procesu porovnání dvou různých datových sad.

Třetí možností reprezentace barevného obrázku je sada deskriptorů, založených na barevné nezávislosti v osvětlovacím modelu. Tyto invarianty jsou opět vyjádřeny dle Gaussovy funkce. Set barevných deskriptorů SIFT byl zkonstruován pomocí nahrazení gradientů úrovní šedi v klasickém deskriptoru gradienty různých barev, které jsou invariantní vůči rozmanitým kombinacím lokální intenzity, stínů, odstínu a světlým místům. Tyto deskriptory byly následně vyhodnoceny pomocí testovacích datových sad, z čehož vyplynulo, že za využití jednoho z těchto deskriptorů, označovaného jako C-color-SIFT, je možné dosáhnout daleko lepších a přesnějších výsledků než pomocí kteréhokoliv z výše uvedených. Tento postup byl navržen Burghoutsem a Geusebroekem roku 2009.

Mezi další významné úpravy algoritmu patří zobecnění užití z 2D prostorových snímků na 2+1D časoprostorové video, čehož lze dosáhnout pomocí vypočítání pozičně závislých histogramů nad lokálním časoprostorovém sousedství kolem buď časoprostorových gradientových vektorů nebo optického toku na každé pozici 2+1D časoprostorové domény. Konkrétně jsou tyto deskriptory počítány v lokálních časoprostorových zájmových bodech za využití časoprostorového mechanismu pro výběr měřítka. Účelem je umožnění lokální adaptace, a tedy i nezávislosti na časovém i prostorovém měřítku.

Z experimentů vyplývá, že tento přístup umožňuje rozpoznání lidských činností na základě časoprostorových deskriptorů obdobně jako lokální prostorový deskriptor SIFT umožňuje rozpoznání a klasifikaci objektů. Pro dosažení invariance vzhledem k relativním pohybům mezi objektem v reálném světě a pozorovatelem je tento přístup často kombinován s mechanismem pro adaptaci rychlosti. Tento mechanismus využívá časoprostorové vyhlazovací operace pro lokální pohyby, což umožňuje rozpoznávání časoprostorových událostí i v zaplněných scénách.

## 4.5 Algoritmus SURF

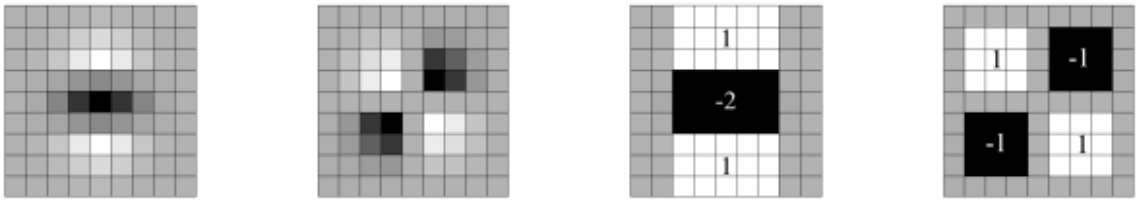
Na základě poznatků z předchozích algoritmů byl v roce 2006 představen nový postup, známý jako SURF (Speeded-Up Robust Features – Zrychlené Robustní Vlastnosti), jehož detektor a deskriptor disponují na základě provedených experimentů nejvyšší mírou znovu-použitelnosti. Pro zpracování tohoto oddílu byly čerpány informace ze zdroje [32].

Hlavní důraz je zde kladen na nezávislost na rotaci v rovině, díky čemuž algoritmus dosahuje odolnosti vůči obvyklým deformacím. Zkreslení, anizotropní[63] změny velikosti a změna perspektivy jsou chápány jako druhořadé defekty, které jsou do jisté míry vyřešeny samotným deskriptorem. Deskriptor může navíc být rozšířen pro nezávislost na dalších charakteristikách pomocí afinní normalizace elipsy, což se však promítne na času výpočtu. Informace o barvě není využita detektorem ani deskriptorem.

Pro detekci zájmových bodů je využita základní aproximace Hessovy matice, díky čemuž je možné využít integrální obrázky (přístup, navržený Violou a Jonesem, viz kapitola 3.4) pro výrazné zrychlení výpočtů. Skvrnovité struktury jsou detekovány v místech, kde dosahuje determinant Hessovy matice (ze kterého je rovněž získáno měřítko) maxima. Pro bod $x = (x, y)$ v obrázku I je Hessova matice $H(x, \sigma)$ v bodě x a měřítku $\sigma$ definovaná jako

$$\boldsymbol{H(x,\sigma)} = \begin{bmatrix} \boldsymbol{L_{xx}(x,\sigma)} & \boldsymbol{L_{yx}(x,\sigma)} \\ \boldsymbol{L_{xy}(x,\sigma)} & \boldsymbol{L_{yy}(x,\sigma)} \end{bmatrix}, \qquad \textbf{( 26 )}$$

kde $L_{xx}(x, \sigma)$ je konvoluce[64] Gaussovy derivace druhého řádu. Gaussova funkce je optimální pro analýzu prostorového měřítka. V praxi je však nutné provést diskretizaci a oříznutí, což ale vede ke ztrátě opakovatelnosti pro rotace obrázků sudými násobky $\frac{\pi}{2}$. I přes tuto nevýhodu, která je dána čtvercovým tvarem filtru, dosahuje detektor skvělých výsledků. Aproximace Hessovy matice je navíc ještě vylepšena boxovými filtry (viz Obrázek 22). Toto označení se využívá pro aproximaci Gaussovy derivace druhého řádu, kterou je možné velmi rychle vypočítat díky integrálním obrázkům.




**Obrázek 22 – Gaussova derivace 2. řádu a její aproximace [31]**

Boxové filtry velikosti 9 x 9 (Obrázek 22 - vpravo) jsou aproximací Gaussovy funkce s měřítkem 1,2, což reprezentuje nejnižší měřítko (například nejvyšší prostorové rozlišení)

[63] *Anizotropie*: závislost na volbě směru.
[64] *Konvoluce*: matematický operátor zpracovávající 2 funkce (v grafice většinou obrazu a filtru).

pro výpočet mapy odezvy skvrn. Tyto filtry pojmenujeme $D_{xx}, D_{yy}$ *a* $D_{xy}$. Váhy, aplikované na obdélníkové oblasti jsou pro zachování efektivity výpočtu zjednodušeny v podobě

$$\mathbf{det}\left(\boldsymbol{H_{approx}}\right) = \boldsymbol{D_{xx}D_{yy} - (\omega D_{xy})^2}. \qquad \textbf{( 27 )}$$

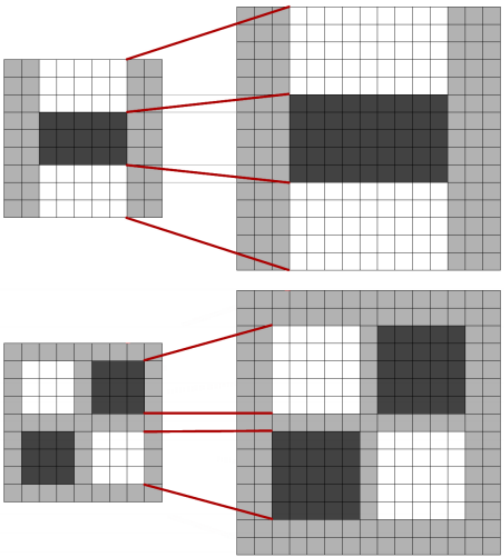
Relativní váha $\omega$ odezvy filtru vyvažuje výraz Hessova determinantu za účelem zachování energie mezi Gaussovými jádry a jejich aproximací. Tyto odezvy jsou dále normalizovány vzhledem k jejich velikosti, díky čemuž lze dosáhnout konstantní maticové normy pro každou velikost filtru. Aproximovaný Hessův determinant reprezentuje skvrnovou odezvu na obrázku v bodě x. Odezvy tohoto typu jsou uloženy v mapě skvrnových odezev v různých měřítkách, ze kterých jsou následně hledány lokální maxima.

Vzhledem k potřebě nalézt zájmové body na obrázcích v různém měřítku se reprezentuje měřítko prostoru pomocí obrázkových pyramid. K tomu dochází pomocí opakovaného vyhlazování snímků pomocí Gaussovy funkce a následného převzorkování pro dosažení vyšší úrovně pyramidy. Tato část postupu se rovněž vyskytuje u algoritmu SIFT (Předchozí kapitola 4.4), i zde je tedy možné odebrat vrstvu pyramidy pro získání DOG, kde je možné vyhledání hran a skvrn.

Díky využití boxových filtrů a integrálních obrázků není potřeba iterativně aplikovat stejný filtr na výstup filtrované vrstvy. Místo toho je možné použít boxový filtr libovolné velikosti při zachování stejné rychlosti přímo na originální obrázek. Měřítko prostoru je tedy analyzováno pomocí zvětšování velikosti filtru, namísto iterativního zmenšování obrázku. Výstup filtru o velikosti 9 x 9 je považován za počáteční vrstvu měřítka. Další vrstvy jsou získávány filtrováním obrázku postupně se zvětšujícími maskami. Hlavní motivací pro tento typ vzorkování je vysoká výpočetní efektivita a absence aliasingu[65].

Měřítko prostoru je rozděleno do oktáv, z čehož každá reprezentuje sérii map filtrových výstupů, které jsou získány pomocí konvoluce stejného vstupního obrázku a filtru o zvětšující se velikosti. Každá oktáva je dále rozdělena do úrovní. Vzhledem k diskrétní povaze integrálních obrázků záleží minimální rozdíl měřítka mezi 2 po sobě jdoucími měřítky na délce $l_0$ (pozitivní nebo negativní oblasti částečného derivátu druhého stupně ve směru derivace x nebo y), která je nastavena na třetinu velikosti filtru. Pro filtr o velikosti 9 x 9 je tato délka tedy rovna 3. Pro 2 po sobě jdoucí úrovně je nutné zvýšit tuto velikost o nejméně 2 pixely (1 na každé straně) pro zajištění přítomnosti centrálního pixelu.

[65] *Aliasing*: efekt, díky kterému může rekonstruovaný vzorek být nerozpoznatelný vůči originálnímu obrázku.



**Obrázek 23 – Filtry dvou po sobě jdoucích úrovní měřítka [32]**

Konstrukce prostorového měřítka začíná použitím filtrů o velikosti 9 x 9 (kde dojde k výpočtu skvrnové odezvy snímku pro nejmenší měřítko) a následují velikosti 15 x 15, 21 x 21 a 27 x 27. První a poslední Hessovy mapy odezev neobsahují lokální maxima, jsou tedy využívány především k referenčním účelům.

Pro každou novou oktávu se velikosti přírůstků filtrů zvětšují dvojnásobně (postupně 6, 12, 24 s 48). Spolu s velikostmi mohou být stejným způsobem zvětšovány vzorkovací intervaly pro extrakci zájmových bodů, díky čemuž lze docílit redukce výpočetního času. Ztráta přesnosti je zde srovnatelná s tradičními přístupy k převzorkování obrázku. Velikosti filtrů pro druhou oktávu jsou 15, 27, 39, 51, pro třetí oktávu 27, 51, 75, 99 a pokud je velikost originálního obrázku stále větší než velikosti filtrů, proběhne analýza pro čtvrtou oktávu, jejíž filtry mají velikosti 51, 99, 147 a 195.

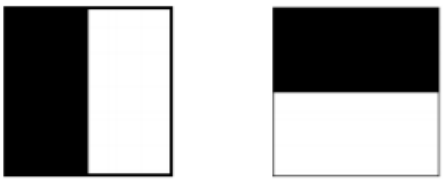
Největší změny měřítka (především mezi prvními filtry oktávy, například 15 je 1,7násobku 9) způsobují hrubé vykreslení vzorkování. Z tohoto důvodu je potřeba využít měřítka prostoru s jemnějším vzorkováním, kde dochází ke zdvojnásobení velikosti obrázku za využití lineární interpolace, který je následně filtrován první oktávou s filtrem o velikosti 15 (po které následují velikosti 21, 27, 33 a 39). Poté následují filtry druhé (kde dochází ke změně velikosti o 12 pixelů), třetí a čtvrté oktávy. Nyní je rozdíl mezi prvními dvěma filtry pouze 1,4.

Za účelem lokalizace zájmových bodů v obrázku a přes měřítka dochází k potlačení maximálních hodnot v sousedství 3 x 3 x 3. Maximální hodnoty determinantu Hessovy matice

jsou poté interpolovány v měřítku a prostoru obrázku. Tyto interpolace jsou v tomto postupu velice důležité, jelikož rozdíly měřítek mezi prvními vrstvami každé oktávy jsou poměrně vysoké.

Výsledný deskriptor popisuje rozdělení intenzity obsahu v okolí zájmových bodů. Podobně je tomu u algoritmu SIFT, kde se však využívá informací o gradientech. Místo gradientu jsou zde použity vlnové odezvy prvního řádu Haarova rozdělení pro směry os x a y, integrální obrázky pro rychlost a pouze 64 dimenzí. Díky tomu dochází zároveň k snížení potřebného výpočetního času a zvýšení robustnosti procesu. Prvním krokem je vymezení znovu-použitelné orientace na základě informací z kruhové oblasti kolem zájmového bodu. Z vybrané orientace se vytvoří čtvercová oblast, ze které je extrahován deskriptor SURF.

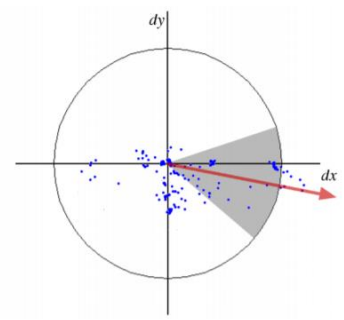
Pro dosažení nejvyšší možné invarianci na rotaci obrázku, je třeba identifikovat reprodukovatelnou orientaci zájmových bodů. Za tímto účelem jsou nejprve spočítány zmíněné Haarovy[66] vlnové odezvy ve směrech x a y v rámci kruhového okolí s poloměrem 6s, kde s je měřítko, ve kterém jsou zájmové body detekovány. Vzorkování i velikost vlnek jsou závislé na měřítku, které se volí opět s (4s pro délku vlnek). Pro urychlení filtrování jsou využity integrální obrázky.



**Obrázek 24 – Haarovy vlnkové filtry [32]**

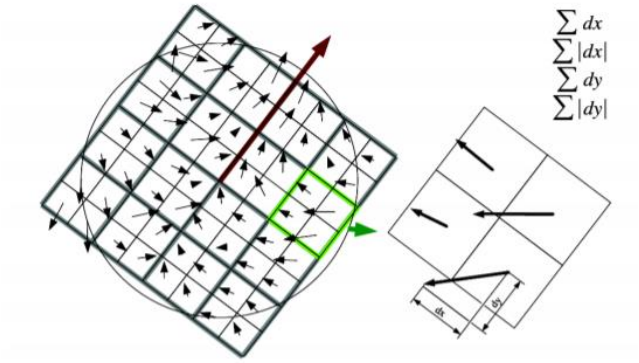
Jakmile jsou vlnkové odezvy vypočteny a ohodnoceny Gaussovou funkcí, která je má střed na zájmovém bodě, jsou odezvy reprezentovány jako body v prostoru s horizontální silou vyobrazenou na ose x a vertikální na ose y. Dominantní orientace je určena vypočítáním součtu všech odezev v rámci posuvného okna orientace o velikosti $\frac{\pi}{3}$ (tato velikost musí být důkladně vybrána - Obrázek 25, špatná volba může vést k chybnému určení orientace). Součet horizontálních a vertikálních odezev udává lokální orientační vektor, přičemž nejdelší ze všech těchto vektorů udává orientaci zájmového bodu.

[66] *Haarova vlnka*: nejstarší a nejjednodušší vlnka, kterou lze využít k výpočtu diskrétní vlnkové transformace.




**Obrázek 25 – Určení orientace zájmového bodu [32]**

Prvním krokem extrakce deskriptoru je konstrukce čtvercových oblastí o velikosti 20s, se středem v zájmovém bodě a orientací určenou dle výše popsaného postupu. Čtvercové oblasti jsou dále rozděleny do 4 x 4 menších čtvercových podoblastí, které uchovávají důležité prostorové informace. Pro každou z těchto menších oblastí dojde k vypočtení Haarových vlnkových odezev (horizontálních - $d_x$ a vertikálních - $d_y$). Pro zvýšení odolnosti vůči geometrickým deformacím a lokalizačním chybám jsou odezvy $d_x$ a $d_y$ nejprve ohodnoceny pomocí Gaussovy funkce se středem v zájmovém bodě.




**Obrázek 26 – Výpočet Haarových vlnkových odezev [32]**

Vlnkové odezvy $d_x$ a $d_y$ jsou sečteny pro každou pod-oblast a výsledek tvoří první sadu záznamů vektoru vlastností. Pro získání informací o polaritě změn intenzity dochází i k extrakci součtu absolutních hodnot odezev. Každá pod-oblast má tedy 4 rozměrný vektor deskriptoru pro základní strukturu intenzity:

$$\boldsymbol{v} = (\sum \boldsymbol{d_x}, \sum \boldsymbol{d_y}, \sum|\boldsymbol{d_x}|, \sum|\boldsymbol{d_y}|). \qquad (28)$$

Spojením těchto vektorů pro všech 4 x 4 pod-oblastí je získán vektor deskriptoru o délce 64. Tyto vlnkové odezvy jsou invariantní vůči sklonu i osvětlení. Nezávislosti na kontrastu je možné dosáhnout převodem deskriptoru na jednotkový vektor.

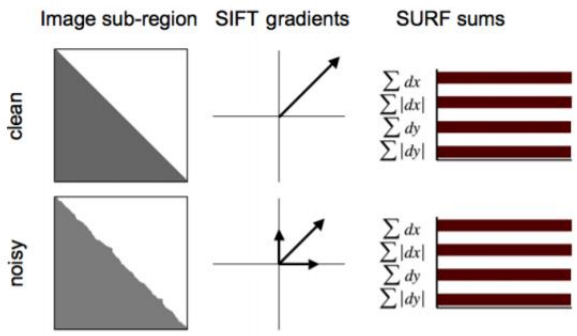
Pro rychlou indexaci během porovnávání je využit Laplaceův operátor (stopa Hessovy matice) pro základní zájmový bod. Obvykle jsou tyto body nalezeny na skvrnovitých strukturách, kde Laplaceův operátor umožňuje rozlišení světlých skvrn na tmavém pozadí od opačné situace. Při samotném porovnávání jsou porovnány vlastnosti, jen pokud mají stejný typ kontrastu (světlá na tmavém nebo tmavá na světlém). Tato minimální informace, která byla zjištěna již v procesu detekce, umožňuje výrazné zrychlení procesu. Tohoto lze využít i u složitějších indexových metod, jako je například k-cestný strom. Samotné porovnání je opět založeno na vzdálenosti vektorů (například Mahalanobisova nebo Euklidovská vzdálenost).

## 4.6 Shrnutí a porovnání rozpoznávacích algoritmů

U porovnávacích biometrických algoritmů platí obdobné tvrzení, jako u detekčních algoritmů. Tedy že s postupem času se přístupy neustále zdokonalují. Obecný princip průběhu je více či méně u všech algoritmů podobný. Podstatné je nějakým způsobem popsat hlavní vlastnosti (charakteristické rysy) obličeje do jednotného vektoru (různých velikostí dle zvoleného algoritmu) a tyto vektory následně porovnat. Tento vektor musí být pro každou osobu co možná nejvíce jedinečný. Čím kvalitněji a konkrétněji je obličej popsán, tím přesnější jsou výsledky následných porovnání. Bohužel to však souvisí i s výpočetní náročností, proto je účelem najít co možná nejlepší kompromis mezi počtem popisových bodů (dimenzí) a výpočetní náročností.

Téměř vždy k tomuto přímému porovnání lze využít Euklidovskou vzdálenost. Cílem je tuto vzdálenost minimalizovat, jelikož čím blíže si 2 vektory jsou, tím jsou si podobnější. Z tohoto vychází i tzv. skóre shody, u kterého platí nepřímá úměra, tedy čím nižší hodnota (menší vzdálenost), tím lepší shoda.

Jak již bylo zmíněno, není výjimkou, že nový algoritmus využívá do jisté míry postup některého ze starších (předchozích) algoritmů s důrazem na odladění chyb a nedostatků. Příkladem je například algoritmus SURF, jehož postup je v jistém ohledu podobný algoritmu SIFT, jelikož oba jsou zaměřeny na prostorové rozložení informací o gradientu. Výkonem má však SURF navrch, především proto, že jsou zde integrovány informace o gradientech v rámci pod-oblastí, zatímco algoritmus SIFT je zaměřen na individuální gradienty. Díky touto přístupu je SURF méně citlivý na šum.




**Obrázek 27 – Porovnání SIFT a SURF z hlediska šumu [32]**

Dalším příkladem návazného vývoje může být využití tzv. vlastních hodnot nebo vektorů, které se poprvé vyskytly v prvním funkčním přístupu nazývaném vlastní tváře (eigenFaces) v mnoha později navržených algoritmech (například FisherFaces nebo SIFT). Původní myšlenka byla taková, že součástí této práce bude i konkrétní srovnání všech algoritmů dle jednotných kritérií. Vzhledem k tomu, že se mezi vyjmenovanými algoritmy jednalo o více či méně plynulý, postupný vývoj, bylo by toto porovnání naprosto bezpředmětné.

Relevantní porovnání nabízí například Obrázek 27, kde jsou porovnány dva z nejmodernějších přístupů z hlediska náchylnosti na šum snímku. Zde je znázorněna situace, kdy jsou porovnávány dva snímky, přičemž jeden je čistý a na druhém se nachází šum. V případě využití algoritmu SIFT je vidět, že výsledkem jsou různé vektory gradientů, zatímco pomocí algoritmu SURF jsou výsledky totožné pro oba případy.

Poslední představený algoritmus, SURF, svým výkonem, znovu-použitelností i invariancí přesahuje všechny své předchůdce. Díky užití integrálních obrázků se jedná i o nejrychlejší z vyjmenovaných algoritmů a díky vysoké míře opakovatelnosti je možné užití i v procesu automatické kalibrace snímacího zařízení. Jak bylo uvedeno v oddílu 4.5, obrovskou úsporu času poskytuje i indexovací strategie, založená na Laplaceově operátoru.

# 5 Osobní doklad

V kapitole Biometrické procesy byl nastíněn rozdíl mezi biometrickou verifikací a identifikací, přičemž oddíl Biometrická verifikace popisoval usnadnění a urychlení ověření totožnosti pomocí různých způsobů proklamace totožnosti, která je následně ověřena způsobem 1 : 1. Popsaný přístup však stále předpokládá uložení biometrických šablon v databázi, ze které se podle způsobu proklamace vybere uživatel, jehož uložená data jsou porovnána s daty živými. Z hlediska ochrany osobních údajů (i nového nařízení GDPR) toto není problémová položka, jelikož biometrická data jsou ve valné většině případů uložená v podobě zpětně nepřeveditelných vektorů (na první pohled nesmyslných čísel, které nemohou vézt k jednoznačné identifikaci jedince).

Problém může nastat v případě, kdy je potřeba využívat biometrický způsob ověření totožnosti u velkého množství lidí (dokonce celé státní populace). V takovém případě je potřeba uchovávat ohromné množství dat a i přesto, že samotné biometrické údaje se nedají využít k určení totožnosti, ukládají se v záznamech společně s konkrétními osobními daty, jako jménem, příjmením, adresou, datem narození či osobní fotkou. To může vézt k problémům se získáním souhlasu ke snímání a uchování těchto osobních dat.

Z tohoto důvodu neexistuje žádná centrální databáze, ve které by byly uloženy biometrické údaje všech osob, ale využívá se řešení pomocí osobních dokladů. Od roku 2006 jsou totiž vydávány elektronické pasy a od roku 2012 elektronické občanské průkazy. Oba typy strojově čitelných dokladů disponují mikročipem, na kterém jsou uložena osobní a biometrická data jedince. Logická struktura dat (LDS – Logical Data Sctructure) je regulována Mezinárodní organizací pro civilní letectví (ICAO – International Civil Aviation Organisation), z jejíž předpisu bude vycházet tato kapitola [41].

## 5.1 Data uložená na mikročipu

ICAO specifikuje následující požadavky na LDS:

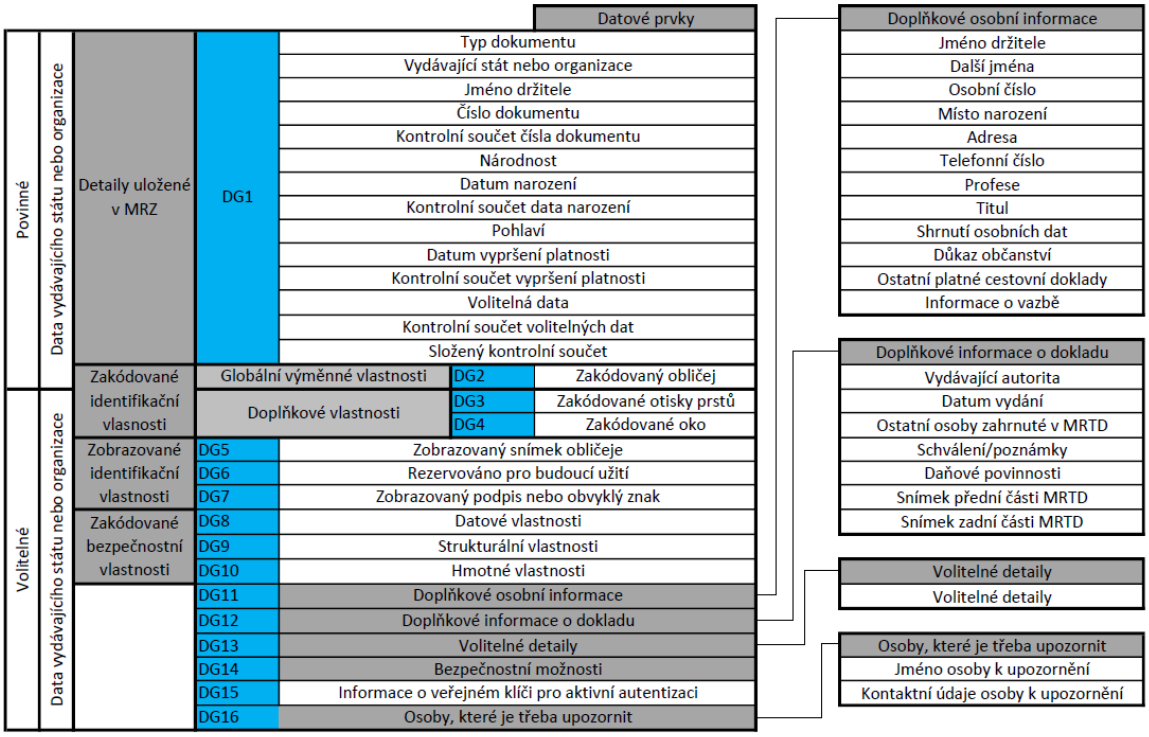
- účinné a optimální usnadnění užívání pro oprávněného držitele,
- ochrana detailů, zaznamenaných v doplňkové technologii rozšíření kapacity,
- globální interoperabilita dat rozšířené kapacity na základě jednotného LDS pro všechny eMRTD (electronic Machine Readable Travel Document – elektronický strojově čitelný cestovní doklad),

- řešení dílčích potřeb doplňkové technologie rozšíření kapacity příslušných států a organizací,
- poskytování rozšiřující kapacity dle potřeb uživatele a dostupných technologií,
- podpora různých možností ochrany dat,
- využití existujících mezinárodních specifikací, zejména pro globálně interoperabilní biometrické údaje.

Pro globální interoperabilitu je klíčové zachování datové integrity a autenticity, k čemuž slouží speciální objekt. Každý z těchto objektů je uložen v separátním souboru. Samotná data jsou organizována do datových skupin (Data Group – DG), které musí být chráněny proti zápisu. Přístup k zápisu do datových skupin může mít pouze vydávající stát nebo organizace. Tyto datové skupiny musí být uspořádány pomocí tzv. náhodného systému řazení, který je konzistentní s technologií doplňkového rozšíření kapacity pro umožnění přímého vyhledání specifických datových prvků i v případě jiného než postupného zápisu.

Jelikož LDS obsahuje velké množství datových prvků, předpokládá reprezentace souborů s náhodným přístupem podporu široké škály implementací. Tyto mohou být obsaženy pro usnadnění autentifikace samotného dokladu nebo oprávněného držitele. Datová struktura musí dále umožňovat zpracování omezené nebo rozsáhlé sady těchto prvků, vícenásobný výskyt specifického prvku i pokračující vývoj implementací.

LDS je považováno za jedinou soudržnou entitu, obsahující různý počet seskupení datových prvků, zaznamenaných v doplňkové technologii rozšíření kapacity v době strojového čtení. Tato struktura byla navržena s dostatečnou flexibilitou, aby mohla být aplikována na všechny typy eMRTD. V rámci LDS byly vytvořeny logické skupiny souvisejících datových prvků, které jsou nazývány datové skupiny (DG, Obrázek 28). Každé této skupině je přiřazeno referenční číslo (například DG2 identifikuje Data Group 2 – Datovou Skupinu číslo 2).




**Obrázek 28 – Popis datových skupin eMRTD [41]**

## 5.2 Komunikace s čipem a jeho zabezpečení

Čipy, na kterých jsou uloženy osobní údaje držitele jsou zabezpečeny několika úrovněmi. Při komunikaci dochází nejprve k **pasivní autentizaci** (nevyužívá se zde výpočetního výkonu čipu, vše probíhá na straně terminálu). Z dokumentu je nejprve vyčten $SO_D$[67], který obsahuje digitální podpis (DS) certifikační autority. Tento podpis je ověřen pomocí veřejného klíče kořenového certifikátu vydávající země. Po úspěšném ověření jsou přečteny všechny v pasu přítomné soubory a spočítány jejich hash hodnoty. Tyto hodnoty jsou porovnány s hodnotami, uloženými v $SO_D$. Tímto je zajištěno, že data v pasu jsou autentická a nezměněná [42].

Pro zabránění kopírování pasu je využívána i aktivní autentizaci, pomocí páru klíčů, přičemž soukromý klíč je uložen v pasu a není možné ho vyčíst, zatímco veřejný klíč je uložen v DG 15 (Obrázek 28). Komunikace opět probíhá mezi pasem a terminálem, kdy pas zašifruje příchozí zprávu pomocí soukromého klíče a terminál tuto zprávu následně dešifruje pomocí klíče veřejného. Tímto dochází ke kontrole shody veřejného a soukromého klíče (pravost veřejného byla již ověřena pasivní autentizací) [42].

[67] *Security Object Document*: objekt, digitálně podepsaný vydávajícím státem, obsahující hashové hodnoty obsahu LDS.

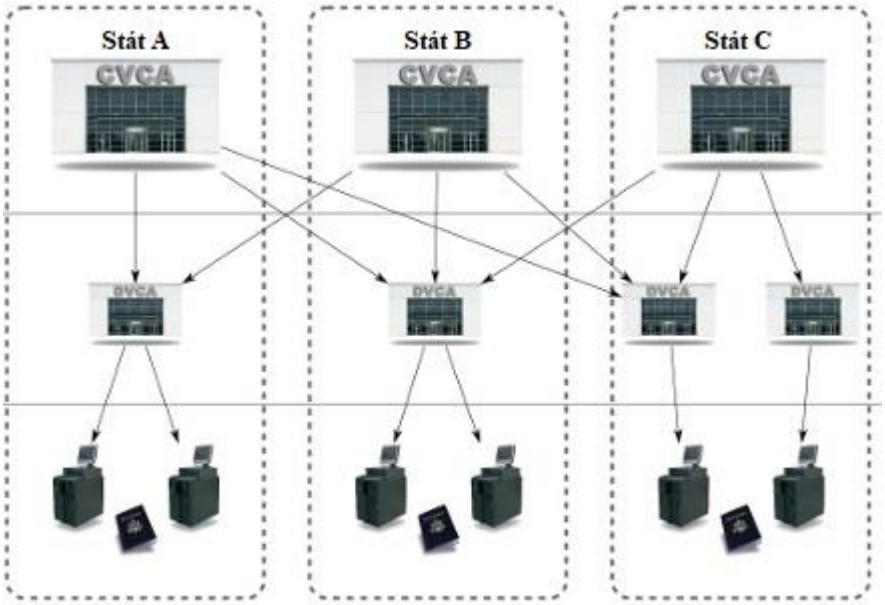
Kromě samotné autentizace dokladu je třeba i jistým způsobem řídit přístup k vyčítaným datům (jinými slovy je potřeba ověřit oprávněnost terminálu tyto data vyčítat). K tomuto slouží tzv. MRZ (strojově čitelná zóna), která sestává ze dvou řádků po 44 znacích (Obrázek 29). První řádek obsahuje typ dokumentu (P), informaci o vydávající zemi (CZE) a jméno držitele (SPECIMEN, VZOR). Na druhém řádku je číslo pasu (99009054), národnost (CZE), datum narození (690622), pohlaví (F), datum vypršení platnosti (160729) a na českých pasech i rodné číslo. Kromě vypsaného výčtu se zde u některých údajů nachází i kontrolní součet pro detekci chyb při strojovém čtení [43].


**Obrázek 29 – Český pas SPECIMEN [43]**

**BAC** (základní řízení přístupu) využívá číslo pasu, datum narození a datum vypršení platnosti. Tyto údaje se zřetězí za sebe včetně kontrolních součtů a dojde k výpočtu hashe. Ze 16 nejvýznamnějších bajtů se vytvoří dočasné klíče pro vzájemnou autentizaci a ustavení klíčů relace. Terminál si od pasu vyžádá 8 bajtů náhodných dat a sám rovněž tyto vygeneruje, spolu s dalšími 16 bajty náhodných dat. Tato náhodná data terminál zřetězí a vzniklý řetězec zašifruje a společně s kontrolním součtem pošle pasu [42].

Pas tyto data dešifruje a porovná s daty a porovná je s náhodnými daty, které terminálu dříve odeslal. Následně vygeneruje dalších 16 bajtů dat a odvodí klíče relace, pomocí kterých zašifruje vzniklá data. Tuto informaci zřetězí spolu s kontrolními součty a pošle jako odpověď terminálu. Ten provede dešifrování a odvození klíčů pro šifrovanou komunikaci pasu a terminálu [42].

Pro přístup k citlivým biometrickým údajům (otisk prstu, duhovka) je třeba využít rozšířeného řízení přístupu (**EAC**). V první fázi dojde k nahrazení klíčů, získaných z BAC, za silnější, přičemž dochází zároveň k ověření pravosti čipu (obdobně jako při pasivní autentizaci). Využívá se zde totiž opět páru klíčů (kde soukromý je opět z čipu nezískatelný a veřejný uložený v DG14). Druhou fází je autentizace terminálu, která je založená na certifikátech, které vydává CVCA (certifikační autority ověřující zemi) nebo DVCA (Certifikační autorita ověřovatele dokumentů) [43][44].




**Obrázek 30 – Struktura spolupráce CVCA a DVCA [44]**

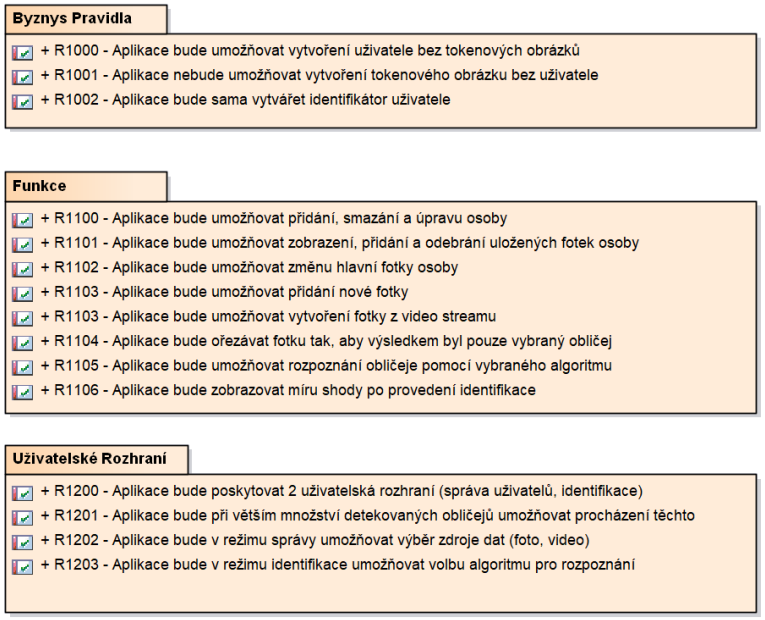
# 6 Analýza a návrh porovnávací aplikace

Při vytváření libovolné aplikace či programu je třeba před samotnou implementací projít jistými fázemi vývoje. Proces procházení těmito fázemi je označován jako analýza a návrh a pro tuto práci je realizace uložena v příloze. K realizaci byl využit program Enterprise Architect díky své rozsáhlé nabídce možností pro realizaci procesu.

První z těchto fází je zpravidla sběr a specifikace požadavků, které se dělí na funkční a nefunkční. Nefunkční požadavky byly rozděleny do 4 balíčků:

- Perzistence – operativní a výkonnostní kritéria vzhledem k úložišti informací (zálohy, databázový systém, soubory a další mechanismy ukládání),
- Škálovatelnost – operativní parametry vzhledem k velikosti (rozšiřitelnost, návrhový vzor, kapacita, počet uživatelů),
- Výkon – délka načítání formulářů, programovací jazyk, komunikace s databází,
- Zabezpečení – požadavky vzhledem k přístupu k datům.

Tyto požadavky tedy popisují jakési „pasivní vlastnosti“ systému, tedy základní charakteristiky, které se u systému očekávají. Na druhou stranu funkční požadavky specifikují funkce a možnosti systému. Rozdělení proběhlo do 3 balíčků (byznys pravidla, funkce a uživatelské rozhraní), jak je vidět na Obrázek 31.

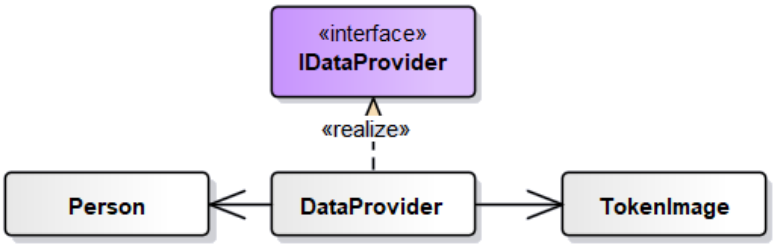



**Obrázek 31 – Funkční požadavky**

Dalším důležitým prvkem pro tvorbu metamodelu je diagram případů užití. Vzhledem k faktu, že se jedná spíše o testovací aplikaci (což vyplývá již ze sběru požadavků) a ne o rozsáhlejší podnikový systém, nebude ve finále výstup analýzy nijak rozsáhlý. Aplikace bude provádět spíše menší počet náročnějších operací než velké množství jednoduchých funkcí. Důraz zde bude tedy kladen hlavně na správu uživatelů a zpracování vstupních snímků.

Aktéři zde dle vstupních požadavků vznikli 2 – správce (který bude mít přístup k databázi uživatelů a tuto bude moci spravovat) a klasický uživatel (který bude moci využívat pouze funkce pro zpracování vstupních snímků). Kromě základních operací nad databází uživatelů (jako je změna osobních údajů, přidání nebo smazání uživatele), zde bude navíc možnost správy tzv. tokenových obrázků, tedy snímků obličeje, které slouží k naučení rozpoznávacího algoritmu. Funkce uživatele pak představují hlavně detekci obličeje a identifikaci osoby, vzhledem k datům z databáze. Každý případ užití je navíc konkrétněji specifikován pomocí scénáře, kde je popsána interakce uživatele (nebo správce) se systémem a jednotlivé kroky každého procesu.

Podstatným výstupem analýzy a návrhu je datový model a analytický model tříd. Jak již však bylo zmíněno, výsledky této analýzy, jakožto analýzy testovací aplikace, nepředstavují rozsáhlé modely. Podstatné bylo získání datového modelu pro vytvoření databáze, do které budou ukládány informace o uživatelích a jejich snímky obličeje. Analytický model tříd představuje pouze základ, který bylo potřeba rozšířit o View a ViewModel vzhledem k záměru využít návrhový vzor MVVM. Rovněž pro samotná data bylo potřeba zakomponovat EntityFramework, za účelem správy a reprezentace dat z databáze. Tyto technologie budou popsány v následující kapitole.




**Obrázek 32 – Diagram analytických tříd**

# 7 Použité technologie

Pro implementaci aplikace pro porovnávání snímků obličeje jsou využity biometrické algoritmy z open-source knihovny OpenCV. Jak vychází z požadavků z předchozí kapitoly, bude potřeba ukládat větší množství dat, k čemuž byla vybrána databáze Microsoft SQL, se kterou bude program komunikovat pomocí Entity Frameworku. Samotná aplikace bude napsána v programovacím jazyce C# dle návrhového vzoru MVVM.

## 7.1 OpenCV

OpenCV je označení pro open source computer vision library (volně šiřitelnou knihovnu zaměřenou na počítačové vidění a strojové učení). Vzhledem k licenci, která umožňuje volné šíření licencovaného obsahu (za podmínky uvedení autora a informace o licenci), poskytuje tato knihovna snadné využití a možnosti modifikace kódu. To je jedním z hlavních důvodů výběru této knihovny pro realizaci diplomové práce [45].

Knihovna obsahuje více než 2500 algoritmů, mezi kterými nechybí aktuální poznatky z oblasti počítačového vidění nebo strojového učení. Tyto algoritmy mohou být využity kromě rozpoznání tváří například i k identifikaci objektů, klasifikaci lidského chování ve videu, sledování pohybujících se objektů, získávání 3D modelů objektů, hledání podobných obrázků z databáze, odstranění rudých očí ze snímku vyfoceného s bleskem, sledování pohybu očí a dalším činnostem. Komunita čítá přes 47 tisíc lidí a počet stažení přesahuje 14 miliónů [45].

Využití této knihovny je opravdu široké, od světově známých značek, jako je Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda nebo Toyota až k start-upům. Další využití je například sledování důlních zařízení v Číně, asistence robotům v navigaci a zvedání objektů ve Willow Garage, detekce utonutí v bazénech v Evropě nebo kontroly přistávacích drah v Turecku [45].

OpenCV využívá vlastních základních struktur pro uchovávání informací. Pro informace o bodě ve dvourozměrném prostoru jsou to struktury Point a Point2f. Pro informace o objektech jsou to Size pro údaje o dvourozměrné velikosti, Rect pro dvourozměrné obdélníky a RotatedRect pro obdélníky natočené o určitý úhel. Nejdůležitější však je struktura pro uchování informací o obrázku – Mat (nachází se zde informace o řádcích a sloupcích, barevných kanálech i barevné hloubce) [48].

OpenCV podporuje tři typy reprezentace obrázků, 2 tříkanálové (BGR a HSV, kde H označuje barvu – hue, S saturaci a V světlost – value) a 1 jednokanálový (Grayscale). Pro zobrazení či uložení obrázku je však potřeba převést obrázek buď na BGR nebo Grayscale. Pro zpracování snímků nabízí knihovna i funkce pro normalizaci a prahování (tresholding). Mezi další funkce patří detekce hran (pomocí třech přístupů – Sobelova, Scharrova a Laplaceova) a vyhlazení obrázku (za účelem snížení ostrosti hran) [48].

Nejjednodušším způsobem instalace na operačním systému MS Windows je využití správce balíčků NuGet, pomocí kterého je možné stahovat a instalovat velké množství knihoven do vlastních projektů. Stačí vyhledat příslušnou knihovnu (EmguCV pro C#, nebo OpenCV), vybrat požadovanou verzi (v této práci je využívána verze 3.4.3.3016) a kliknout na tlačítko *instalovat*. Alternativním přístupem je stažení balíčků z webových stránek a jejich manuální instalace.

Vzhledem k tomu, že praktická část práce (samotná implementace) je realizována v programovacím jazyce C#, byl využit .Net wrapper pro zpracování obrazu, zvaný **EmguCV**. Díky tomuto wrapperu je možné volat funkce z knihovny OpenCV z jazyků kompatibilních s .Net (c#, VB, VC++, IronPython a další) a je možné jej provozovat na Windows, Linux, Mac OS X, iOS, Android i Windows Phone [46].

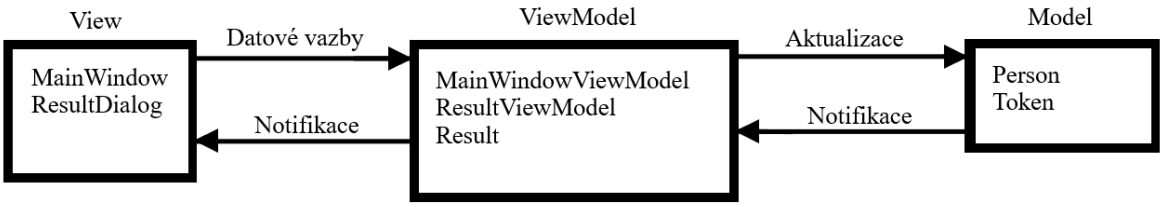
Tato relativní nezávislost na platformě je zajištěna skutečností, že EmguCV je zkompilovatelný v Mono[68], tedy je možné ho spustit na všech platformách, které Mono podporuje (všechny výše zmíněné). Přestože existuje mnoho pokusů o implementaci čistě v C#, EmguCV je stále nejvyužívanější pro implementace v jazyce C# právě z důvodu přenositelnosti [46].

## 7.2 MVVM (Model/View/ViewModel)

Tento návrhový vzor, využívaný mimo jiné v aplikacích WPF (Windows Presentation Foundation) je přizpůsoben moderním vývojovým platformám uživatelského rozhraní, kde View (GUI) má na starosti spíše designer, než programátor. Design je tvořen deklarativní formou, jako například HTML nebo XAML, většinou pomocí WYSIWYG nástroje. Uživatelské rozhraní navíc může být vytvořeno pomocí různých nástrojů nebo jazyků (na rozdíl od MVC, kde celá aplikace musí být napsána v jednom jazyce a jednom prostředí) [47].

[68] Mono: volně šiřitelný open-source projekt, jehož cílem je vytvoření sady nástrojů, kompatibilních s prostředím .NET, splňující standardy ECMA.

Definice modelu je shodná s MVC, jedná se tedy o data nebo byznys logiku, jejíž stav je ukládán (například do databáze nebo XML). View sestává z vizuálních elementů (tlačítek, oken a dalších ovládacích prvků), které je však málokdy možné napojit přímo na model (data, která je potřeba zobrazit). Tato data je před zobrazením často potřeba upravit nebo zmapovat a k tomu zde slouží ViewModel, který označuje model uživatelského rozhraní. Jedná se o jakéhosi prostředníka pro interakci View s Modelem (pomocí příkazů a data-transformátorů) [47].




**Obrázek 33 – Schéma komunikace v rámci MVVM**

## 7.3 Microsoft SQL Server a Entity Framework

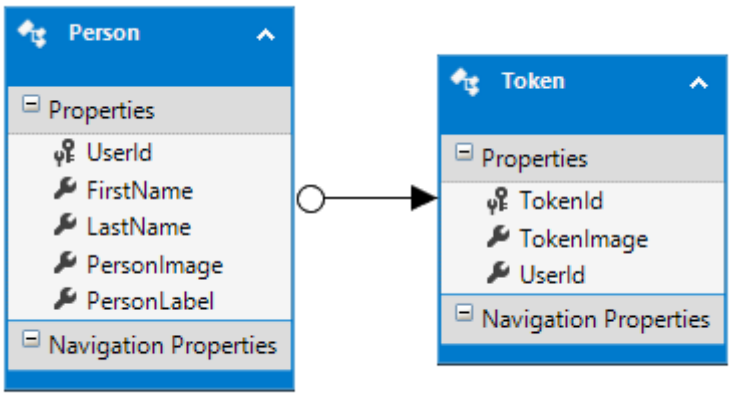
Microsoft SQL Server je relační databázový systém, vyvinutý společností Microsoft. Pro tuto práci byl vybrán především díky vysokému výkonu, škálovatelnosti a bezpečnosti. K napojení aplikace na tuto databázi byl využit open-source mapovač relací objektů pro platformu .NET – Entity Framework. Jedná se o sadu technologií, která podporuje vývoj datově orientovaných aplikací. Pro přidání tohoto frameworku do projektu stačí využít správu balíčků NuGet v prostředí Microsoft Visual Studio. Zde je vybrána poslední stabilní verze a tato je nainstalována do celého projektu.

# 8 Implementace a popis funkcí aplikace

Tato kapitola je zaměřena na popis implementace a hlavních funkcí porovnávacího programu. Bude zde nastíněn způsob implementace, aplikace a využití technologií, uvedených v předchozí kapitole.

## 8.1 Implementace

Z hlediska praktické části této kvalifikační práce byla prvním krokem implementace tvorba databázové struktury. Z analýzy již vyplývá, že se nebude jednat o velmi rozsáhlou databázi, nýbrž o pouhé uložení osob a jejich snímků obličeje. Výsledný model tedy vypadá následovně:




**Obrázek 34 – Databázový model**

S tímto databázovým modelem komunikuje modul, založený na výše zmíněném Entity Frameworku (v práci je použita poslední stabilní verze, tedy 6.2.0). Celé řešení je rozděleno do 3 balíčků – první (Core) obsahuje základní a společné prvky (především pro komunikaci s databází), ImageDemo je samotná porovnávací aplikace, které bude věnován prostor v oddíle 8.3 a PersonManager je správa osob, uložených v databázi (detailněji popsáno v oddíle 8.2).

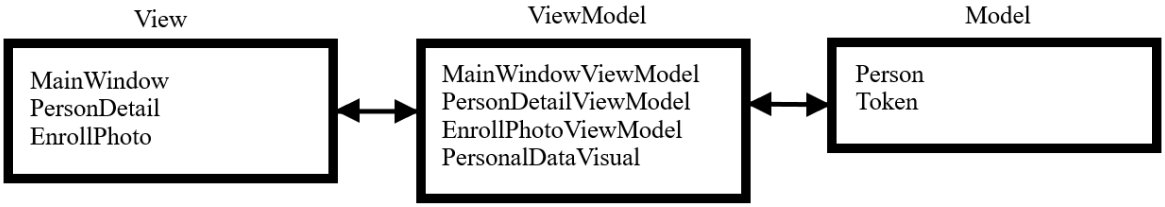
Celá implementace probíhala dle návrhového vzoru MVVM (viz oddíl 7.2), přičemž všechny dílčí aplikace využívají jednoho modelu – třídy Person a Token, které jsou třídními reprezentacemi stejnojmenných databázových tabulek. Kolekce těchto tříd jsou obsaženy v třídě DataProvider, která slouží jako prostředník pro přístup do databáze. Při kterékoliv práci s daty (tedy s modelem) je využívána tato třída. Ostatní jsou tedy pouze View nebo ViewModel, které budou popsány u jednotlivých aplikací.

Jak již naznačuje předchozí odstavec, nejedná se o jednu aplikaci, která by obsahovala více modulů, ale o jednotlivé samostatné aplikace. Prvotní myšlenka u tohoto rozhodování zahrnovala odlišnou distribuci na zařízení správce a na zařízení uživatele (správce by měl pouze aplikaci pro správu osob, zatímco uživatel pouze aplikaci pro zpracování obrázku). Struktury aplikací budou popsány v následujících oddílech, stejně jako jejich funkce a způsob použití. Pro ilustrační účely byly využity volně dostupné fotky známých osobností.

## 8.2 Správa osob

Jak již název napovídá, tato aplikace poskytuje kompletní možnosti pro správu osob, uložených v databázi. Aplikace sestává ze 3 dialogových oken (view) spolu s příslušícími viewmodely. Pro lepší komunikaci mezi okny a připojení k viewmodelům zde bylo využito vlastní rozhraní **IDialog**, které si uchovává základní znalosti o kontextu (DataContext), výsledku (DialogResult), vlastníkovi (Owner – okno, ze kterého nový dialog byl zavolán) a poskytuje metody pro zavření (Close) a zobrazení dialogu (ShowDialog).

Spolu s vlastním rozhraním je zde použita vlastní implementace třídy **DialogService** (jakožto implementace rozhraní IDialogService), jejíž hlavní odpovědností je správa View a přiřazení specifických ViewModelů. Je zde tedy obsažen slovník, do kterého se ukládají dvojice View – ViewModel. Dále je zde obsažena metoda pro registraci nové dvojice. Proces přiřazení ViewModelů k View a registrace dvojic probíhá v metodě OnStartup v souboru App.xaml.

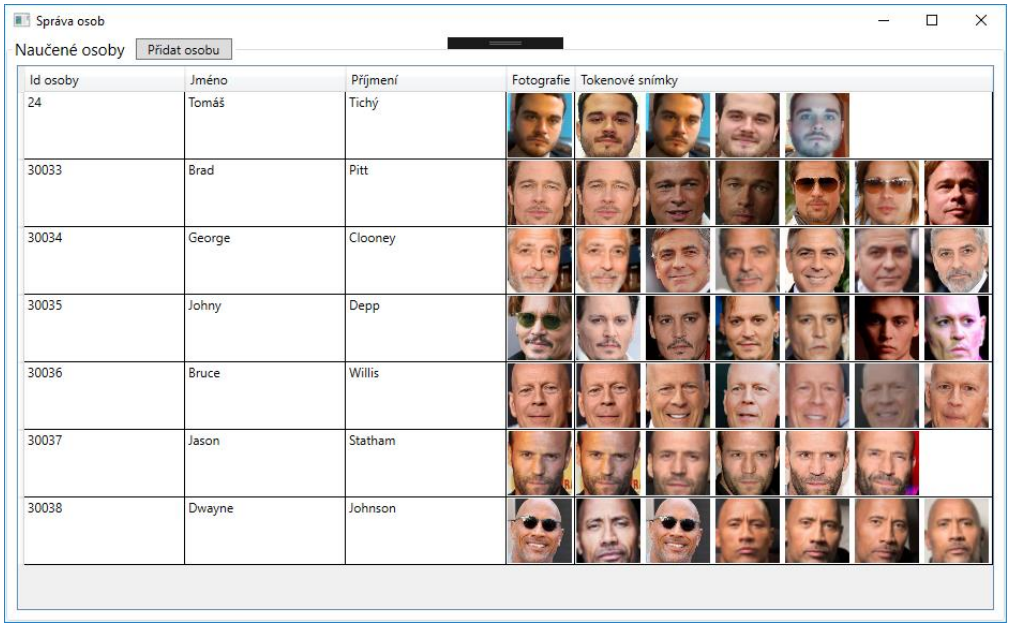



**Obrázek 35 – Rozdělení aplikace pro správu osob dle MVVM**

Z důvodu správného propojení s View implementují všechny ViewModely rozhraní INotifyPropertyChanged, které pomocí PropertyChangedEventHandler obstarává aktualizaci View při změně kterékoliv z vlastností, uložené ve ViewModelu a propojené s grafickým uživatelským rozhraním.

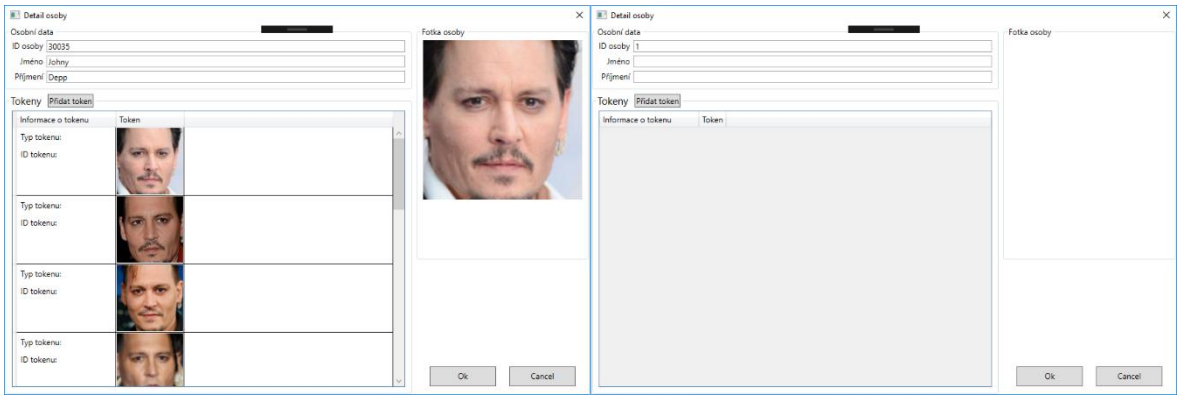
Hlavnímu oknu aplikace dominuje tabulka, zobrazující uložená data o všech osobách z databáze. Pro každou osobu je zde údaj o identifikačním čísle, jménu, příjmení a hlavní

profilové fotografii (data z databázové tabulky Person). Poslední sloupec obsahuje všechny tokenové snímky[69] osoby, které jsou pro danou osobu v databázi uloženy (tabulka Token).




**Obrázek 36 – Hlavní okno správy osob**

V levé horní části aplikace se nachází tlačítko pro přidání nové osoby. Úprava a mazání existující osoby jsou možné pomocí kontextové nabídky po pravém kliknutí na řádek tabulky (záznam reprezentující konkrétní osobu). Pro přidání nové osoby i editaci stávající slouží jednotné dialogové okno, přičemž rozdíl je, zda jsou tomuto dialogu poslána data o konkrétní osobě, nebo zda je dialog prázdný – připravený pro vytvoření nového záznamu.

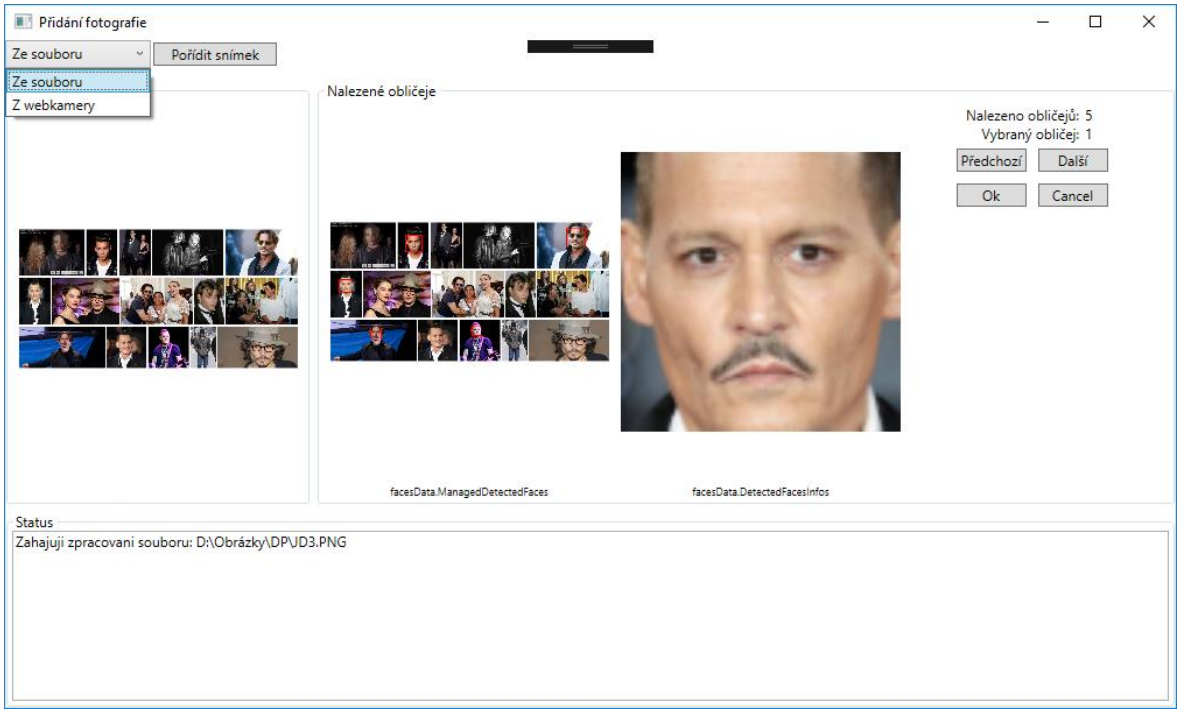



**Obrázek 37 – Dialogové okno pro editaci/přidání osoby**

Nejdůležitější částí tohoto dialogového okna je správa tokenových obrázků (v levé spodní části). Zde jsou zobrazeny existující, uložené snímky, přičemž po kliknutí pravým tlačítkem

[69] Tokenový snímek: oříznutá část vstupního obrázku, která obsahuje pouze detekovaný obličej (vyříznutá ohraničená oblast získána z procesu detekce obličeje). Tyto snímky budou použity jako trénovací sada rozpoznávacího algoritmu.

je možné daný snímek odstranit nebo jej nastavit jako profilovou fotku uživatele (větší fotografie v pravé horní části dialogu). Kromě těchto možností a změny jména a příjmení je zde tlačítko pro přidání tokenových snímků, které vyvolá nové dialogové okno.




**Obrázek 38 – Dialogové okno pro přidání tokenového snímku**

V tomto dialogovém okně již probíhá několik operací. Nejprve je třeba zvolit zdroj dat (jak je znázorněno na Obrázek 38 – rozbalený combo box v levém horním rohu). Možnosti jsou ze souboru, tedy nahrání uloženého obrázku nebo z webkamery. Při druhé volbě dojde k zahájení přenosu snímků z kamery a jejich následnému zpracování. Při obou volbách se nejprve načte vstupní obrázek do levé části dialogu (při volbě vstupu z webkamery se zde začne přehrávat vstupní proud dat).

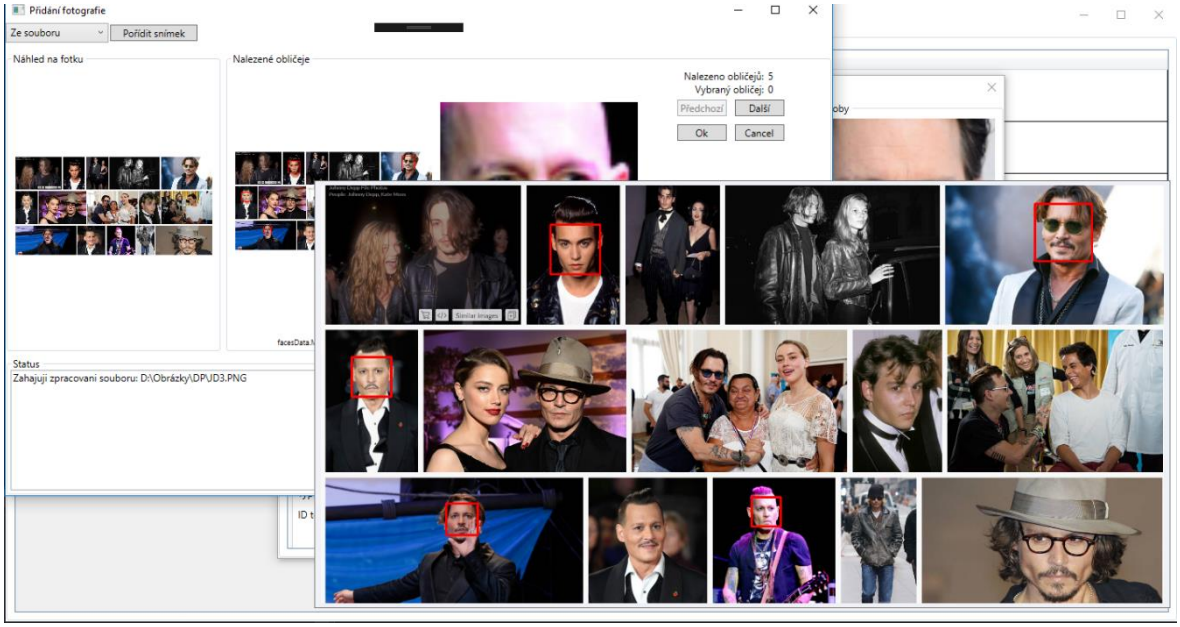
Prvním krokem zpracování obrázku je převod na formát `Image<Bgr, Byte>` a pro detekci následně na `Image<Gray, Byte>`, tedy do obrázku v odstínech šedé barvy. Převedený obrázek je následně podroben procesu detekce pomocí kaskádových klasifikátorů (metoda popsaná v oddíle 3.4).

```
private List<Rectangle> DetectFaces(Image<Bgr, Byte> inputImage){
  var grayframe = inputImage.Convert<Gray, Byte>();
  List<Rectangle> detected =
    _cascadeClassifier.DetectMultiScale(grayframe, 1.1, 10,
    System.Drawing.Size.Empty).ToList();
  return detected;
}
```

Všechny nalezené oblasti, ve kterých se nacházejí obličeje jsou vloženy do kolekce obdélníků. Pro každou z obdélníkových oblastí jsou do původního obrázku zakresleny okraje této oblasti a výsledek je zobrazen vpravo od původního stavu vstupního obrázku.

```
private void ProcessImage(Image<Bgr, Byte> loadedImage){
  NewFrame = Convert(loadedImage.ToBitmap(), PixelFormats.Bgr24);
  Image<Bgr, Byte> processedImage = loadedImage.Copy();
  foreach (var face in DetectFaces(loadedImage)){
    processedImage.Draw(face, new Bgr(System.Drawing.Color.Red), 3);
    loadedImage.ROI = face;
    _detectedFacesList.Add(loadedImage.Copy());
  }
  DetectedFacesCount = _detectedFacesList.Count;
  ManagedDetectedFaces = Convert(processedImage.ToBitmap(),
    PixelFormats.Bgr24);
  SwitchFace();
}
```

Přičemž metoda **`SwitchFace`** ve výše uvedeném zdrojovém kódu slouží k přepínání mezi detekovanými tvářemi, které jsou vyobrazeny v pravé části okna aplikace. Po najetí myší na libovolný element obrázku v dialogu se obrázek zvětší.




**Obrázek 39 – Zvětšený obrázek s detekovanými tvářemi**

Na Obrázek 39 je patrné, že tento princip detekce obličejů není zcela dokonalý a vyžaduje určité podmínky pro úspěšné detekování obličeje (jak již bylo naznačeno v oddíle 3.4). Všechny tváře jsou dále z původního obrázku vyříznuty do podoby tokenových snímků a přidány do kolekce pro zobrazení (Obrázek 38 – obrázek obličeje vpravo). Mezi jednotlivými obličeji v kolekci je možné přecházet pomocí tlačítek „Předchozí“ a „Další“ a po nalezení požadovaného lze potvrdit volbu kliknutím na tlačítko „Ok“. Proces přidávání snímku je zároveň možné zrušit pomocí tlačítka „Cancel“.

Obdobný proces probíhá při získávání snímku z živého přenosu z webové kamery. Pro každý příchozí snímek je provedena stejná sada operací. V druhé sekci zprava je tedy vstupní snímek s označenými oblastmi, ve kterých se nachází obličej a v pravé sekci jsou vyříznuté obličeje, mezi kterými je opět možné listovat pomocí tlačítek.

V této aplikaci tedy dochází pouze k detekování obličejů, nikoliv k samotnému porovnání. Účelem užití této aplikace je totiž vytvoření databáze osob a k nim přiřazených tokenových obrázků, které bude moci další aplikace využít k naučení porovnávacích algoritmů pro další užití.

## 8.3 Zpracování obrázku

Tato aplikace reprezentuje hlavní realizaci praktické části této práce. Předchozí oddíl byl zaměřen na přípravnou aplikaci pro vytvoření datových základů pro rozpoznávání obličejů. Databáze, vytvořená pomocí aplikace pro správu osob je využita v této aplikaci, kde dochází k porovnávání nových vstupních obrázků s naučenými daty z databáze.

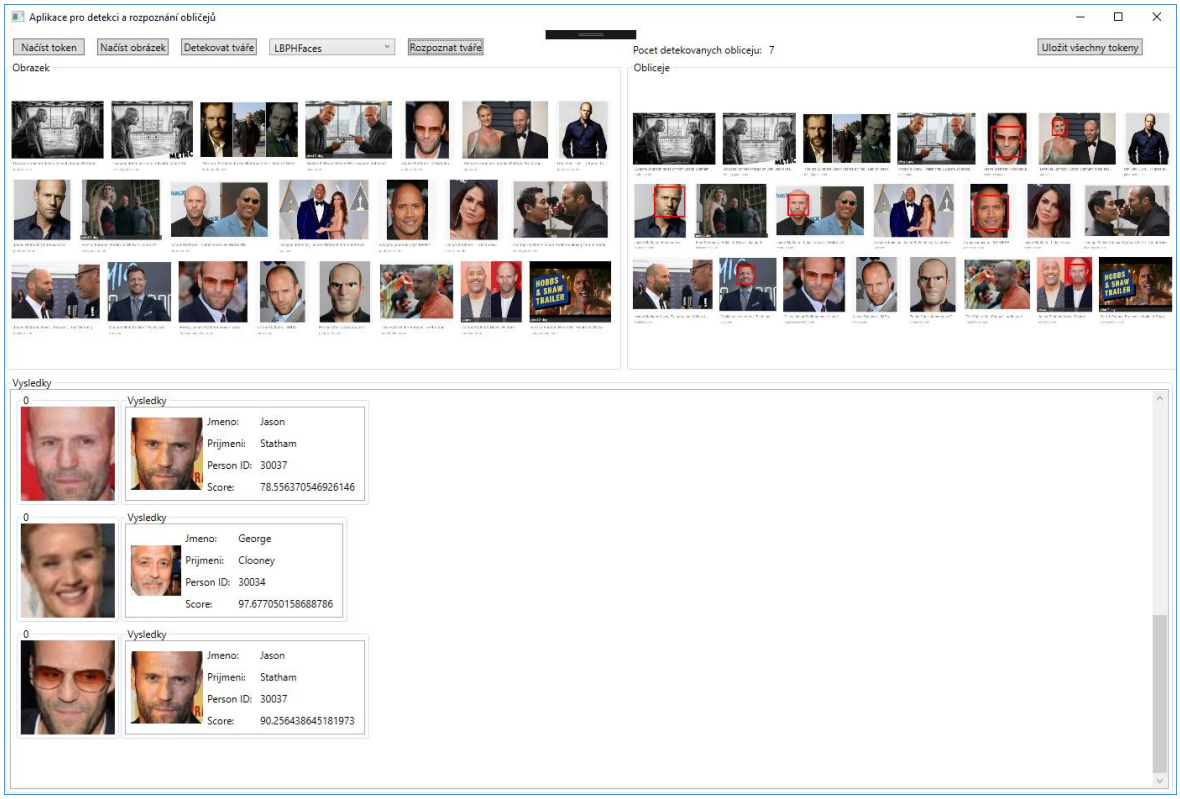
Při spuštění aplikace dojde k načtení všech dat z databáze, za účelem trénování všech 3 porovnávacích algoritmů. Dojde tedy k inicializaci `EigenFaceRecognizer`, `FisherFaceRecognizer` i `LBPHFaceRecognizer` a u každého je zavolána metoda **`Train`**, jejímiž parametry jsou pole tokenových obrázků a pole uživatelských identifikátorů, přičemž na určitém indexu těchto polí jsou související data (tokenu na indexu i v poli tokenů odpovídá identifikátor na indexu i v poli identifikátorů).

```
var allFaces = _dataProviderModule.GetAllTokens();
if (allFaces != null){
  var faceImages = new Image<Gray, byte>[allFaces.Count];
  var faceLabels = new int[allFaces.Count];
  for (int i = 0; i < allFaces.Count; i++){
    Stream = new MemoryStream();
    stream.Write(allFaces[i].TokenImage, 0,
      allFaces[i].TokenImage.Length);
    var faceImage = new Image<Gray, byte>(new Bitmap(stream));
    faceImages[i] = faceImage.Resize(100, 100, Inter.Cubic);
    faceLabels[i] = (int) allFaces[i].UserId;
  }
  _faceRecognizer.Train<Gray, byte>(faceImages, faceLabels);
  _faceRecognizer.Write(_recognizerFilePath);
}
```

Stav takto naučeného *`Recognizeru`* je uložen do souboru, ze kterého může být následně načten. Důvodem je možnost volby porovnávacího algoritmu před zahájením procesu rozpoznání (pomocí combo boxu v horní části aplikace, bude popsáno dále). K naučení všech najednou dochází z důvodu ušetření času v rámci dalších procesů a z důvodu znovu-

použitelnosti (jednou naučený a uložený *rekognizér* může být opakovaně využit k rozpoznání různých tváří).

Stavy jsou ukládány do souborů formátu YAML (YAML Aint Markup Language), což je formát pro serializaci dat textových souborů. Tyto soubory jsou čitelné strojem i člověkem, strukturované pomocí odsazení (pomocí mezer, nikoli tabulátorů) a není zde omezení počtu úrovní vnořování. K ukládání jsou využity knihovní funkce *Write* pro každý *rekognizér* a pro opětovné načtení funkce *Read* [49].




**Obrázek 40 – Aplikace pro detekci a rozpoznání obličejů**

Převážnou část aplikace pro detekci a rozpoznání obličejů zabírají zobrazovací prvky pro vykreslení obrázků v různých částech procesu. V horní části okna se nachází ovládací prvky, pomocí kterých je možné se v procesu detekce a rozpoznání pohybovat. Opět se zde vyskytuje pojem token (dříve použit jako tokenový snímek). I zde se jedná o označení předem připravené a částečně zpracované fotografie, kde již došlo k oříznutí na základě detekce (v této aplikaci pomocí zmíněných kaskádových klasifikátorů).

Tlačítko „Načíst token“ tedy slouží k načtení předpřipraveného obrázku, na kterém se nachází pouze 1 obličej, který zabírá dostatečnou poměrnou část obrázku. V tomto procesu je snímek načten rovnou jako obličej, je tedy přeskočena samostatná část detekce

a přidání do seznamu detekovaných (jak bude popsáno níže, pro klasické obrázky). Dále tedy stačí vybrat porovnávací algoritmus a kliknout na tlačítko „Rozpoznat tváře“.

Na rozdíl od prvního tlačítka, kde dochází k přeskočení jedné fáze procesu, pomocí tlačítka „Načíst obrázek“ dojde pouze k načtení a zobrazení obrázku v levé horní části. K rozfázování zde došlo za účelem dodržení analýzy, konkrétně jako realizace všech případů užití a jejich scénářů. Po úspěšném načtení obrázku je teprve možné kliknout na tlačítko „Detekovat obličeje“. Výsledkem je naplnění seznamu detekovaných tváří, které se zobrazí ve spodní části okna (Výsledky). Zároveň dojde ke zpracování vstupního obrázku podobným způsobem jako u první aplikace, tedy k orámování detekovaných obličejů. Takto zpracovaný obrázek je zobrazen v pravé horní části. Stejně jako v předešlé aplikaci, i zde jsou pro detekci využity kaskádové klasifikátory (viz oddíl 3.4).

V libovolné části procesu, kdy seznam detekovaných tváří není prázdný, je možné tyto tokenové snímky uložit. Jednou možností je postupné ukládání pomocí kontextového menu jednotlivých snímků, pokud je účelem uložení celého seznamu, je možné použít tlačítko v pravé horní části aplikace „Uložit všechny tokeny“.

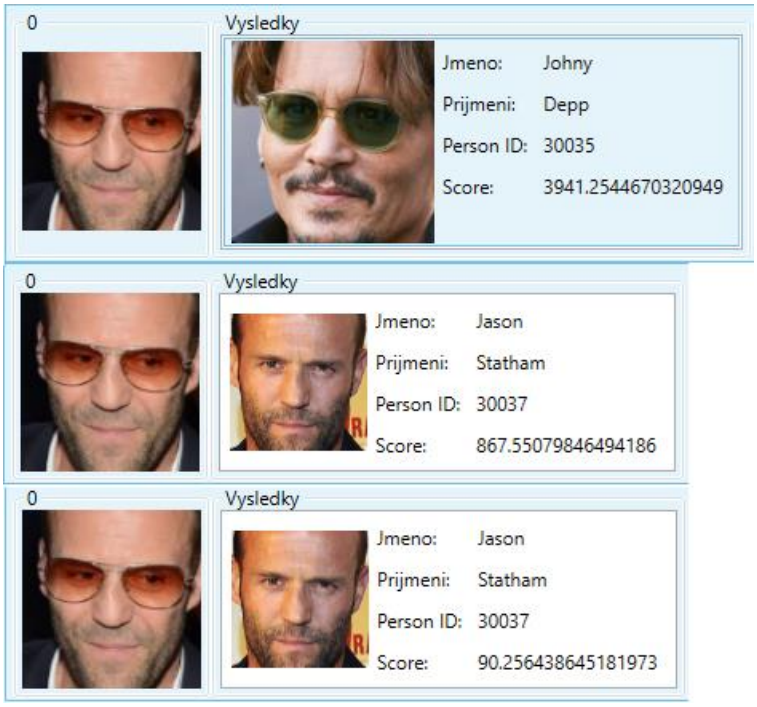

**Obrázek 41 – Kontextové menu snímku detekovaného obličeje**

Po provedení detekce je dle některých scénářů možné ukončit proces využití aplikace. Pokud je účelem užití například jen detekovat na snímku všechny tváře a uložit jejich snímky pro pozdější zpracování, je zbytečné provádět porovnání již zde. Pokud je však potřeba provést rozpoznání tváří na vstupním obrázku, je třeba zvolit porovnávací algoritmus k dalšímu zpracování.

To je možné pomocí combo boxu vedle tlačítka „Detekovat tváře“. Zde jsou k dispozici všechny 3 biometrické algoritmy, které nabízí open-source knihovna OpenCV (popisována v oddíle 7.1), tedy EigenFaces, FisherFaces a LBPH. Princip fungování těchto algoritmů

je vysvětlen v příslušných oddílech kapitoly 4, kde jsou tyto algoritmy vzájemně i porovnány a uvedeny jejich nástupci.

Volba porovnávacího algoritmu je velmi důležitá, neboť má velký vliv na výsledek porovnání. Jak je popisováno ve zmíněné kapitole 4, ani jeden ze 3 algoritmů není zcela dokonalý, a ne vždy tyto algoritmy poskytují spolehlivé a věrohodné výsledky. Jeden obličej může pomocí různých algoritmů být vyhodnocen různými způsoby (viz Obrázek 42, kde byly použity algoritmy postupně shora EigenFaces, FisherFaces a LBPH).




**Obrázek 42 – Výsledek porovnání pomocí různých algoritmů**

Na Obrázek 42 je tedy možné vidět, jakým způsobem dochází k výpisu výsledků porovnání. K tomuto výpisu je využita třída `DetectedFace`, která obsahuje pro každou detekovanou tvář základní údaje spolu se seznamem výsledků porovnání. Do tohoto seznamu jsou ukládány instance třídy `PersonMatchResult`, která obsahuje především profilovou fotografii, unikátní identifikátor, jméno a příjmení uživatele. Tato data jsou získána z databáze pomocí zmiňovaného `DataProvideru`.

```
private PersonMatchResult GetPersonMatchByImage(Image<Gray, Byte> image){
  _faceRecognizer.Read(_recognizerFilePath);
  var result = _faceRecognizer.Predict(image.Resize(100, 100,
    Inter.Cubic));
  var personData = _dataProviderModule.GetPersonalDataByID(result.Label);
  PersonMatchResult = new PersonMatchResult{
    FirstName = personData.FirstName,
    LastName = personData.LastName,
    PersonId = personData.UserId,
```

```
    Score = result.Distance,
    Photo = _dataProviderModule.GetPersonImageByID(result.Label)
  };
  return personMatchResult;
}
```

Kromě těchto dat je zde uložena i informace o skóre. Tato hodnota udává výsledek porovnání, tedy jakousi věrohodnost nalezené shody. V kapitole 4 bylo vysvětleno, že obrázky, respektive jejich reprezentující vektory, jsou porovnány pomocí měření (Euklidovské) vzdálenosti mezi nimi. Ideální hodnota skóre, tedy vzdálenost mezi obrázky, které se naprosto shodují, je 0. Není zde tedy přímá úměra věrohodnosti (čím vyšší skóre, tím přesnější a věrohodnější shoda), nýbrž přesně naopak.

V kapitole 4 byla také často skloňována prahová hodnota. U biometrických algoritmů je toto označení používáno pro jakousi hraniční hodnotu přípustnosti. Pokud je tedy tato hodnota překročena u nejbližší shody, výsledkem je nenalezení dostatečné shody, nikoliv zobrazení nejbližší možné shody (nesprávné určení nejbližší shody – Obrázek 43).


**Obrázek 43 – Nesprávné určení shody**

Na Obrázek 42 je patrné, že každý z uvedených porovnávacích algoritmů má skóre shody (tedy vzdálenost mezi popisovými vektory) v jiných řádech. Je tedy potřeba nalézt optimální hraniční hodnotu pro každý algoritmus a tuto nastavit jako práh. Názory na hodnotu tohoto prahu se rozcházejí a vzhledem k důležitosti volby této hodnoty je součástí této práce rovněž aplikace pro hledání prahové hodnoty za pomoci lidského faktoru.

# 9 Hledání prahových hodnot a testování algoritmů

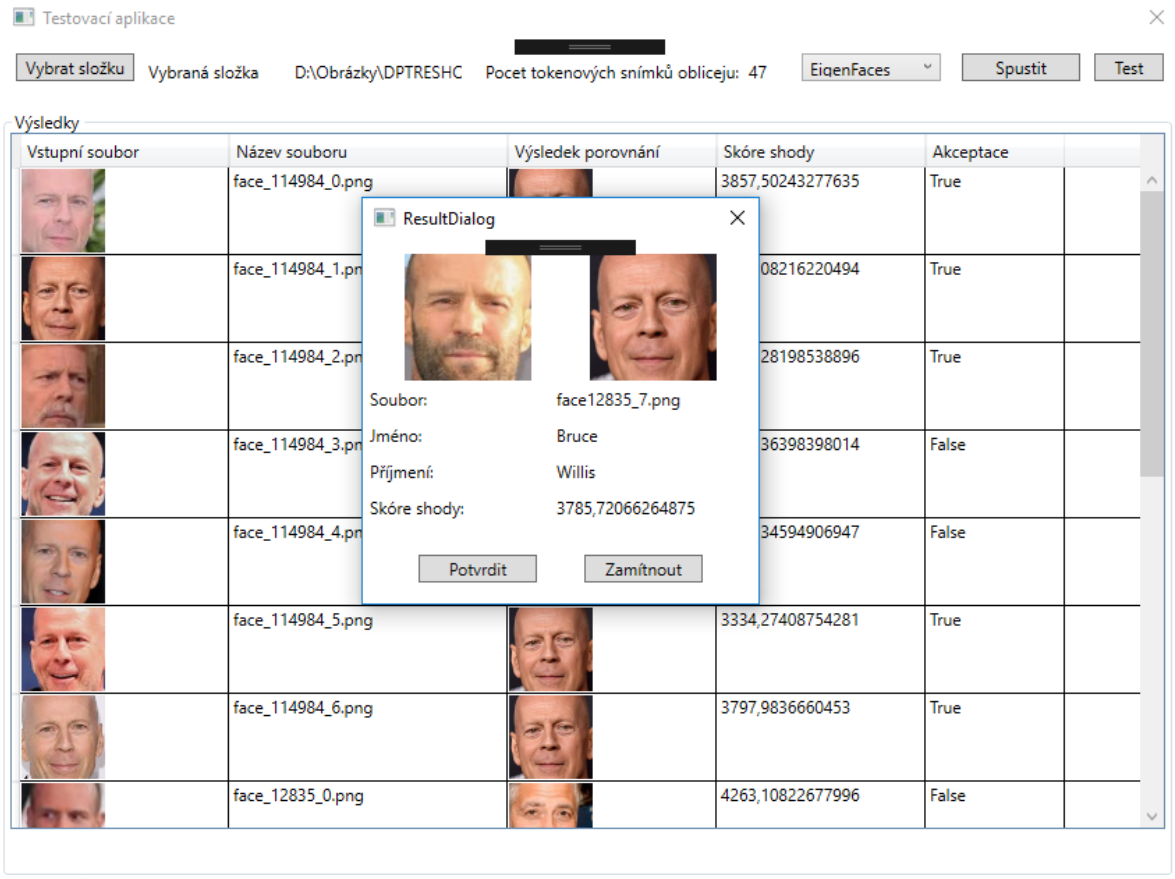
Tato kapitola bude zaměřena na popis aplikace, která bude sloužit jednak k nalezení prahových hodnot a jednak k otestování úspěšnosti určení shody pomocí jednotlivých algoritmů. Jedná se tedy o aplikaci, která má jedno uživatelské rozhraní a dvě funkcionality. První režim, hledání prahových hodnot, vyžaduje zásah operátora – lidské obsluhy, pro stanovení správnosti určení shody, zatímco druhá část je plně automatizovaná.

## 9.1 Hledání prahových hodnot

Tento režim aplikace plně navazuje na poslední oddíl předchozí kapitoly, kde bylo konstatováno, že správné určení prahové hodnoty je zcela zásadní pro správné vyhodnocení porovnávacího procesu. Než začne práce s touto aplikací, je třeba mít předpřipravená data v podobě tokenových obrázků. Tyto je možné (buď po jednom nebo hromadně) uložit z aplikace pro zpracování obrazu (oddíl 8.3).

Myšlenka této aplikace je zpracování co možná největšího množství těchto testovacích dat dle vybraného biometrického algoritmu. Samozřejmostí je také předem naplněná databáze osob, vůči kterým bude docházet k porovnávání. Pro každé porovnání bude následně vyžadován zásah operátora (uživatele), který bude muset potvrdit, zda algoritmus shodu vyhodnotil správně či nikoliv.

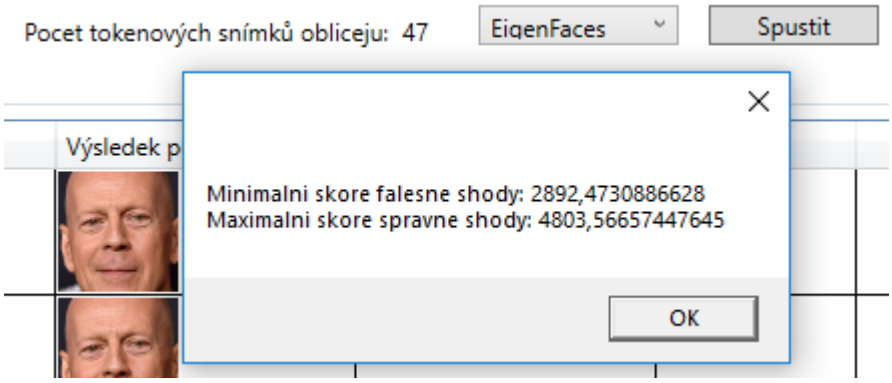
Prvním krokem je tedy výběr složky, ve které se připravené tokenové obrázky nacházejí. Následně dojde k naplnění seznamu názvu souborů z vybrané složky, přičemž vybírány jsou pouze soubory ve formátu png. Po výběru biometrického algoritmu již stačí jen stisknout tlačítko *Spustit* a vyhodnocovat všechny výsledky porovnání. Změna prahové hodnoty má vliv na ukazatele FRR a FAR, proto je potřeba určit, jaký je ideální stav. Toto se odvíjí od případu užití a způsobu nasazení, jelikož například u zabezpečené budovy, kde systém umožňuje přístup do skladu utajovaných informací jsou jiné preference než u obyčejného docházkového systému firmy.




**Obrázek 44 – Průběh hledání prahových hodnot**

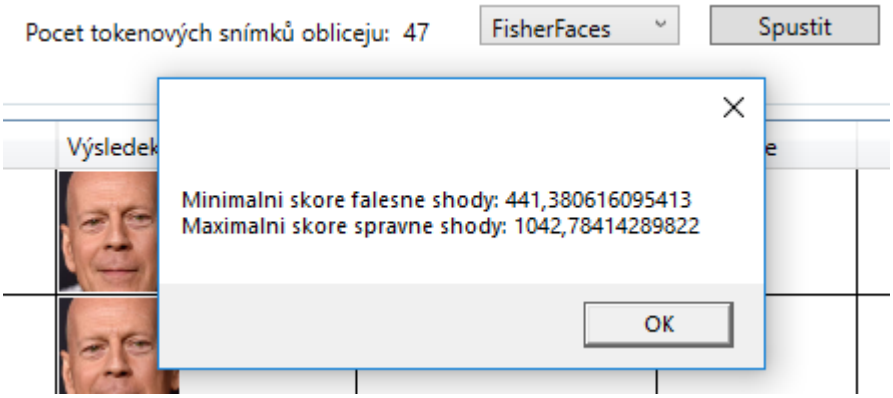
Na Obrázek 44 je možné vidět, jakým způsobem je řešeno grafické uživatelské rozhraní aplikace. Stejně jako u ostatních aplikací se veškeré ovládací prvky nachází v horní části. V levé části se nachází tlačítko pro výběr složky, ve které se nacházejí testovací data. Po vybrání se zobrazí hned vedle tohoto tlačítka vybrané umístění a počet tokenových snímků obličejů. Následuje combo box pro výběr biometrického porovnávacího algoritmu, pro který chceme prahovou hodnotu nalézt a tlačítko *Spustit*. Poslední tlačítko (Test) je ovládacím prvkem druhé části této aplikace, které bude věnován následující oddíl.

Již v oddíle 8.3, konkrétně na Obrázek 42 bylo řečeno, že hodnoty shody pro jednotlivé algoritmy se poměrně dost liší, proto je nutné provést hledání optimálních hodnot pro každý algoritmus zvlášť. Aplikace má podobný průběh jako aplikace pro zpracování obrázku (je přeskočena část detekce, jelikož se předpokládá práce s tokenovými obrázky). Vstupní token je tedy převeden do odstínů šedé barvy a je zavolána funkce `Predict` vybraného biometrického algoritmu. Výsledek je poslán do nového dialogového okna, kde je vyžadováno vyhodnocení ze strany operátora.



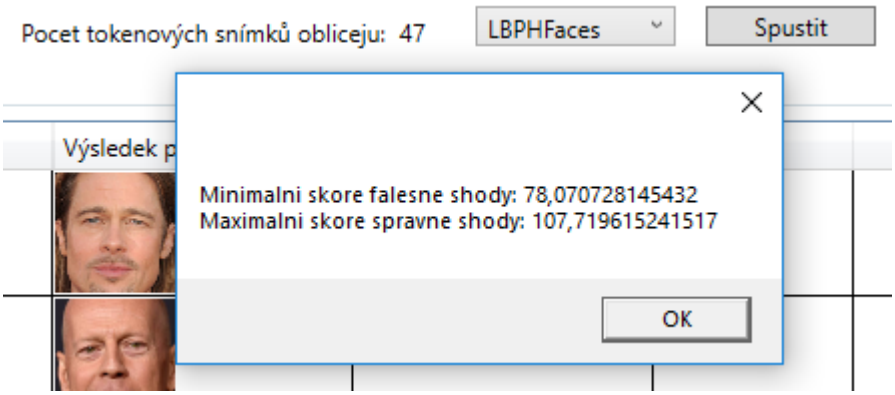

**Obrázek 45 – Výsledek hledání prahové hodnoty pro algoritmus EigenFaces**

Podstatným výsledkem hledání prahů je nalezení dvou hodnot – minimální hodnoty, kterou algoritmus nesprávně vyhodnotil jako shodu a maximální hodnotu, pro kterou algoritmus správně vyhodnotil shodu. Z těchto dvou hodnot je následně možné vytvořit novou prahovou hodnotu. Pokud by se například jednalo o vysoký požadavek na zabezpečení, byla by prahová hodnota nastavena dle minimálního skóre falešné shody. V takovém případě by ze všech vstupních 47 tokenových obrázků nedošlo ani k jednomu falešnému přijetí.




**Obrázek 46 – Výsledek hledání prahové hodnoty pro algoritmus FisherFaces**

Přestože by bylo možné docílit stavu, kdy ze všech vstupních dat by nedošlo ani k jednomu falešnému přijetí, negativním důsledkem by byl zvýšený výskyt falešného odmítnutí. Jak je vidět na Obrázek 45 i Obrázek 46, vyskytují se případy, kde skóre shody je vyšší, než při kterém došlo k nesprávnému párování. U takovýchto případů by při nastavení prahové hodnoty dle minimálního skóre falešné shody docházelo k falešnému zamítnutí.




**Obrázek 47 – Výsledek hledání prahové hodnoty pro algoritmus LBPH**

Vzhledem k velkým závislostem ukazatelů FRR a FAR na nastavení prahové hodnoty často dochází k opakovanému testování. Častou praktikou u systémů, které využívají zmíněné biometrické algoritmy, proto je na základě procesu hledání využití obou získaných hodnot (minimální skóre falešné shody i maximální skóre správné shody) a nastavení prahové hodnoty na hodnotu přesně mezi těmito hodnotami. Pro algoritmus **EigenFaces** je tedy na základě procesu hledání optimální prahová hodnota **3 847**, pro algoritmus **FisherFaces 741** a pro algoritmus **LBPH 92,5**.

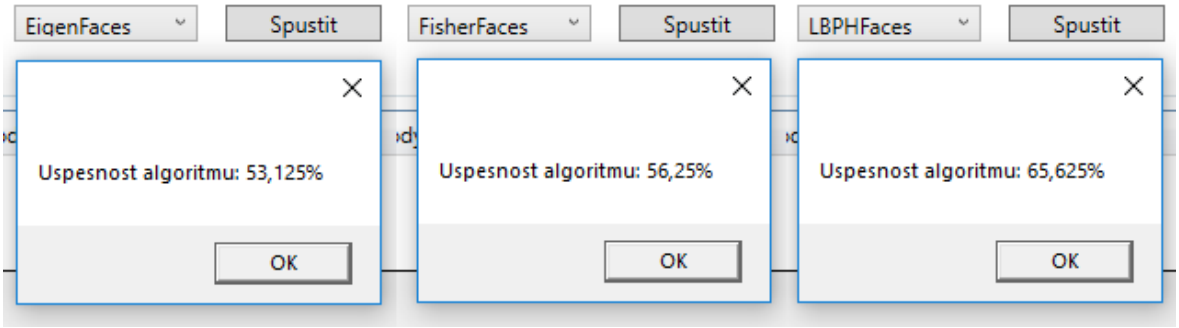
Takto nastavenou hodnotu je samozřejmě potřeba otestovat a v případě nepříznivých výsledků upravit. Je samozřejmě ideální minimalizovat oba chybové ukazatele (FRR i FAR), ale jak již bylo opakovaně zmíněno, snížením jednoho dochází ke zvýšení druhého.

## 9.2 Testování biometrických algoritmů

Před samotným zahájením testování je třeba mít nastavenou prahovou hodnotu, abychom pracovali s „ostrou“ verzí pro každý algoritmus. Dalším předpokladem je trochu upravená sada testovacích tokenových fotografií, a to tím způsobem, že první část názvu každého souboru (před dělícím znakem _) obsahuje identifikátor osoby, ke které fotografie patří. Tohoto lze rovněž dosáhnout pomocí upravené funkce aplikace pro zpracování obrázku, kde je v kontextové nabídce možnost „*Uložit pro test*“. Účelem je dosažení plně automatizovaného průběhu testování, kde je výsledek testování porovnán právě s první částí názvu souboru.

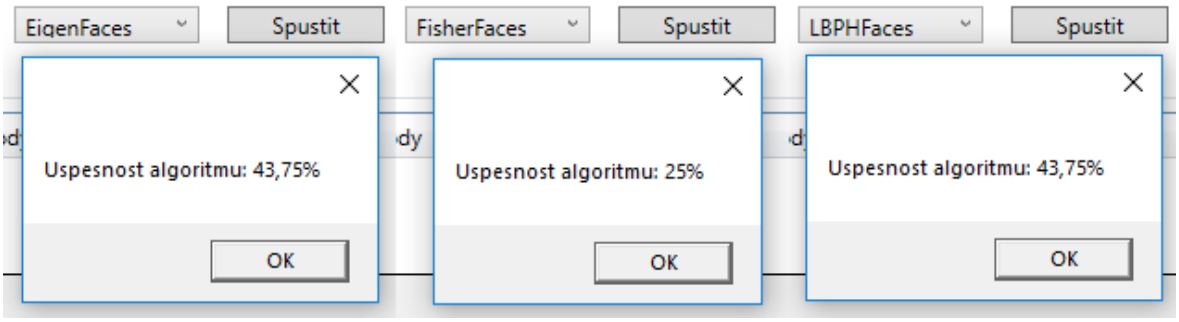
Začátek procesu je obdobný procesu hledání prahových hodnot. V horní části aplikace je stisknuto tlačítko pro výběr adresáře, po čemž následuje načtení názvů souborů do seznamu. Po zvolení požadovaného biometrického algoritmu k otestování je stisknuto tlačítko Test (pravá horní část aplikace, viz Obrázek 44). Toto testování probíhá automaticky

a délka běhu tohoto procesu tedy závisí na objemu testovaných dat (i na objemu referenčních dat, uložených v databázi). Po dokončení testování se zobrazí informační okno s výslednou procentuální úspěšností (procento správně vyhodnocených snímků daným algoritmem. Pro získání podrobných výsledků stačí spustit testovací aplikaci a vybrat složku se vstupními daty.




**Obrázek 48 – Úspěšnosti algoritmů před nastavením prahové hodnoty**

Na Obrázek 48 je vidět stoupající tendence vzhledem k novějšímu algoritmu. Již v teoretické části bylo popisováno, že s postupem času se algoritmy stávaly dokonalejšími a spolehlivějšími. Přestože 65 % se může zdát jako poměrně nízká hladina úspěšnosti, je nutné brát v potaz různé podmínky, za kterých byly pořízeny vstupní snímky a relativně malý objem trénovacích setů, uložených v databázi. První testování bylo provedeno před nastavením prahových hodnot na hodnoty získané v předchozím oddílu.




**Obrázek 49 – Úspěšnosti algoritmů po nastavení prahových hodnot**

Z Obrázek 49 je patrný skoro až alarmující pokles procentuální úspěšnosti. Na první pohled by se mohlo zdát, že nastavení prahové hodnoty má opačný než požadovaný účinek a že celý proces, popisovaný v předchozí kapitole je zbytečný. Opak je však pravdou, neboť přestože došlo ke zvýšení chyb typu False Reject (falešné odmítnutí, tedy nerozpoznání osoby, která je uložena v databázi), došlo ke značnému snížení chyb opačného typu, tedy False Accept

(falešné přijetí – chybné přiřazení snímku osoby, které není v databázi k osobě, která se tam nachází).

Ve výchozím stavu jsou totiž algoritmy inicializovány s velmi vysokou hodnotou prahu, takže výsledkem rozpoznání nikdy není *Neznámá osoba*. Účelem není najít věrohodnou shodu, nýbrž nejbližší možnou. Tento postup je založen na popisovaném principu minimalizace (Euklidovské) vzdálenosti bez dalších podmínek. Přidáním prahové hodnoty dojde k omezení maximální přípustné vzdálenosti, pro kterou je shoda považována za věrohodnou.

Bohužel však i u jedné osoby je možné pomocí různého úhlu a nasvícení pořídit 2 fotografie, které budou mít velmi vysokou hodnotu vzdálenosti (tzv. skóre shody), proto může dojít k falešnému zamítnutí (FR). Naopak nejsou výjimkou případy, kdy se jedna osoba natolik podobá jiné, že výsledná vzdálenost mezi snímky, pořízenými za podobných podmínek (světlo, úhel), je natolik nízká, že může projít pod hodnotou prahovou.

Možností řešení tohoto problému je více, z důvodu spolehlivosti a odolnosti vůči napadení je však většinou vyžadována součinnost operátora (uživatele – obsluhy). Existují systémy, ve kterých na konci určitého období dochází k manuální kontrole zamítnutých přístupů, přičemž pokud došlo k chybnému zamítnutí, je daný snímek osoby, která se již v databázi nachází, přidán do trénovacího setu za účelem možného vytvoření dokonalejšího vzoru – šablony dané osoby.

Jistou alternativou mohou být systémy, které mají nastavenou periodu, po které dochází k přidání do trénovacího setu. Tento přístup není založen na chybném zamítnutí, nýbrž naopak na správném přiřazení. Přičemž například při každém desátém správném přiřazení dojde k přidání správně rozpoznaného snímku do trénovacího setu pro daného uživatele. Tento proces je automatizovaný, což může představovat bezpečnostní riziko. Pokud by se totiž ve správný okamžik podařilo oklamat rozpoznávací systém, došlo by k zápisu podvržené fotografie do systému a postupně by pak bylo možné přetvořit uživatelskou šablonu na úplně jiný vzhled.

# Závěr

Autor se rozhodl začít práci uvedením do problematiky biometrie, kde se nachází základní popis, rozdělení a zařazení vybraných biometrik do kategorií. Tyto kategorie, ale i jednotlivé biometrické charakteristiky jsou vzájemně porovnány dle různých kritérií, přičemž největší důraz je, vzhledem k zaměření této kvalifikační práce, věnován biometrii lidské tváře. Větší množství charakteristik je zde uvedeno pro lepší nastínění možností využití charakteristických rysů lidského těla pro identifikaci. Z porovnání navíc vyplývá značné množství výhod využití právě biometriky obličeje.

Následující část je zaměřena na úvod do biometrických procesů, kde je mimo jiné nastíněn rozdíl mezi biometrickou verifikací a identifikací. Zároveň je zde popsána velká výhoda využití tváře pro identifikaci, a to z hlediska nenáročnosti na snímací senzor pro pořízení snímku obličeje. Pro zdůraznění této výhody je uveden odlišný postup při snímání biometrie otisků prstů, který vyžaduje specifické senzory (více, či méně složité a prostorově a finančně náročné).

Pravděpodobně nejpodstatnější pasáží teoretické části je popis vybraných detekčních a porovnávacích biometrických algoritmů. Kromě těch, které jsou použity v praktické části jsou zahrnuty další hojně využívané algoritmy. U Všech algoritmů z obou kategorií (detekce i porovnání) je detailně popsán princip fungování a ani zde nechybí vzájemné porovnání. Algoritmy jsou popisovány chronologicky, tedy v pořadí, ve kterém vznikaly a není výjimkou postupná návaznost nového algoritmu na ten předchozí. Teoretickou část práce uzavírá popis osobních dokladů a dat v nich uložených. V této části se nachází hlavně popis zabezpečení čipu dokladu a komunikace s ním.

Neodmyslitelnou částí vývoje libovolného software je analýza a návrh. Samozřejmě existují i jiné, více či méně správné, přístupy k procesu vývoje, tyto dvě části by však neměly být vynechány. Z tohoto důvodu je součástí této diplomové práce popis těchto procesů, využitých při vývoji aplikace pro biometrické porovnání snímků obličeje. Výsledek tohoto procesu je připojen v příloze, jako projekt programu Enterprise Architect. Dílčí části jsou popsány v první kapitole praktické části, jakožto nastínění průběhu.

V souladu se zadáním se autor rozhodl využít existující knihovnu OpenCV, jejíž stručný popis je rovněž obsažen v praktické části. Jelikož autor prováděl implementaci hlavních i testovacích aplikací v programovacím jazyce C#, musel být využit .NET wrapper nazývaný EmguCV. Společně se jeho popisem je v praktické části možné nalézt i popis Entity

Frameworku, který je využitý pro připojení k databázi a popis návrhového modelu MVVM, dle kterého je program implementovaný.

Hlavní cíl práce autor splnil pomocí svého programu pro biometrické porovnávání snímků obličeje, implementovaného v jazyce C#, dle návrhového vzoru MVVM s napojením na Microsoft SQL databázi pomocí Entity Frameworku. Biometrické algoritmy pro detekci i porovnání využil z existující volně přístupné knihovny OpenCV, respektive jejího .NET wrapperu EmguCV. Základní funkce, které měl program obsahovat implementoval a tuto implementaci i výslednou funkčnost popsal s doplněním obrázků pro ilustrace.

Při implementaci vznikla potřeba další aplikace pro hledání prahové hodnoty a následné testování úspěšnosti biometrických algoritmů. Tato aplikace je popsána v poslední kapitole praktické části, včetně principů, kterých využívají oba režimy aplikace.

# Literatura


[1] RAK, Roman, Václav MATYÁŠ a Zdeněk ŘÍHA. *Biometrie a identita člověka: ve forenzních a komerčních aplikacích*. 1. Grada Publishing, 2008, 664 s. ISBN 8024763923.

[2] JAIN, Anil K., Patrick FLYNN a Arun A. ROSS. *Handbook of biometrics*. New York: Springer, c2008. ISBN 978-038-7710-419.

[3] BOLLE, Ruud M., Jonathan H. CONNELL, Sharath PANKANTI, Nalini K. RATHA a Andrew W. SENIOR. *Guide to Biometrics*. Ilustrované vydání. Springer Science & Business Media, 2013. ISBN 1475740360.

[4] Biometrie otisku prstu. *Biometric Line* [online]. [cit. 2019-03-24]. Dostupné z: http://www.biometricke-ctecky.cz/biometriky/otisk-prstu/

[5] How Facial Recognition Systems Work. *HowStuffWorks* [online]. [cit. 2019-03-24]. Dostupné z: https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/facial-recognition1.htm

[6] PONTIN, Mark Williams. Better Face-Recognition Software. *MIT Technology Review* [online]. 30. 5. 2007, 1 [cit. 2019-03-24]. Dostupné z: https://www.technologyreview.com/s/407976/better-face-recognition-software/

[7] *Biometrie obličeje* [online]. [cit. 2019-03-24]. Dostupné z: http://www.biometricke-ctecky.cz/biometriky/oblicej/

[8] *Army develops face recognition technology that works in the dark* [online]. 2018 [cit. 2019-03-24]. Dostupné z: https://www.arl.army.mil/www/default.cfm?article=3199

[9] KAPUR, Jay P. Face Detection in Color Images. *University of Washington Department of Electrical Engineering* [online]. 1997, , 1 [cit. 2019-03-24]. Dostupné z: http://web.archive.org/web/20090723024922/http:/geocities.com/jaykapur/face.html

[10] ROWLEY, Henry R., Shumeet BALUJA a Takeo KANADE. *Rotation Invariant Neural Network-Based Face Detection* [online]., 1-7 [cit. 2019-04-12]. Dostupné z: https://www.eecs.wsu.edu/~holder/courses/cse6363/fall00/papers/rowley-cvpr98.pdf

[11] FLECK, Margaret M., David A. FORSYTH a Chris BREGLER. *Finding Naked People* [online]. [cit. 2019-04-12]. Dostupné z: http://luthuli.cs.uiuc.edu/~daf/papers/naked.pdf

[12] RASHEED, Asmaa Hameed a Haneen Mohammed HUSSEIN. *Effect of Different Window Size on Median Filter Performance with Variable Noise Densities* [online]. 2017, 1-6 [cit. 2019-04-12]. Dostupné z: https://www.ijcaonline.org/archives/volume178/number2/rasheed-2017-ijca-915732.pdf

[13] FLECK, Margaret M. *Practical Edge Finding with a Robust Estimator* [online]. 1-5 [cit. 2019-04-12]. Dostupné z: https://pdfs.semanticscholar.org/9472/e808e44a53a9f3f011ce926bd380612d556f.pdf

[14] HJELMÅS, Erik. *Motion Detection* [online]. 1998, 1 [cit. 2019-04-12]. Dostupné z: http://web.archive.org/web/20080522171806/http:/www.ansatt.hig.no/erikh/papers/hig98_6/node2.html

[15] REIGNIER, Patrick. *Finding a face by blink detection* [online]. 21. 7. 1995, 1 [cit. 2019-04-12]. Dostupné z: http://www-prima.imag.fr/ECVNet/IRS95/node13.html

[16] *Face Detection & Recognition Homepage* [online]. 2019 [cit. 2019-03-24]. Dostupné z: https://facedetection.com/algorithms/

[17] JESORSKY, Oliver, Klaus J. KIRCHBERG a Robert W. FRISCHHOLZ. Robust Face Detection Using the Hausdorff Distance. *Third International Conference on Audio- and Video-based Biometric Person Authentication* [online]. Halmstad, Sweden, 2001, 1-6 [cit. 2019-03-24]. Dostupné z: https://facedetection.com/wp-content/uploads/AVBPA01BioID.pdf

[18] GRÉGOIRE, Normand a Mikael BOUILLOT. *Hausdorff distance between convex polygons* [online]. 1998, 1 [cit. 2019-04-12]. Dostupné z: http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html

[19] RUCKLIDGE, William J. *Efficiently Locating Objects Using the Hausdorff Distance* [online]. 1996, 1-20 [cit. 2019-04-12]. Dostupné z: https://www.classes.cs.uchicago.edu/archive/2004/fall/35900-2/hausdorff.pdf

[20] SOBEL, Irwin. *An Isotropic 3x3 Image Gradient Operator* [online]. 2014, , 1-5 [cit. 2019-04-12]. Dostupné z: https://www.researchgate.net/publication/239398674_An_Isotropic_3x3_Image_Gradient_Operator

[21] JUNG, Haebichan. *Adaboost for Dummies: Breaking Down the Math (and its Equations) into Simple Terms* [online]. 2018 [cit. 2019-04-12]. Dostupné z: https://towardsdatascience.com/adaboost-for-dummies-breaking-down-the-math-and-its-equations-into-simple-terms-87f439757dcf

[22] VIOLA, Paul a Michael J. JONES. Robust Real-Time Face Detection. *International Journal of Computer Vision* [online]. The Netherlands, 2004, 2004, 2(57), 137-154 [cit. 2019-03-25]. DOI: 10.1.1.93.8268. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.8268&rep=rep1&type=pdf

[23] DALAL, Navneet a Bill TRIGGS. *Histograms of Oriented Gradients for Human Detection* [online]. 2005, 1-8 [cit. 2019-03-25]. Dostupné z: http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

[24] KAZEMI, Vahid a Josephine SULLIVAN. *One Millisecond Face Alignment with an Ensemble of Regression Trees* [online]. 2014, 1-8 [cit. 2019-03-25]. Dostupné z: http://www.csc.kth.se/~vahidk/papers/KazemiCVPR14.pdf

[25] GEITGEY, Adam. Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning. *MEDIUM* [online]. 2016, 24. 7. 2016 [cit. 2019-03-25]. Dostupné z: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

[26] What Are Biometrics? – White Paper. *AWARE* [online]. [cit. 2019-03-26]. Dostupné z: https://www.aware.com/what-are-biometrics/biometric-processes/

[27] FAR and FRR: security level versus user convenience. *Recogtech* [online]. [cit. 2019-03-26]. Dostupné z: https://www.recogtech.com/en/knowledge-base/security-level-versus-user-convenience

[28] ZHANG, Sheng a Matthew TURK. Eigenfaces. *Scholarpedia* [online]. 2008, 3(9) [cit. 2019-03-26]. DOI: 10.4249/scholarpedia.4244. ISSN 1941-6016. Dostupné z: http://www.scholarpedia.org/article/Eigenfaces

[29] SALTON DO PRADO, Kelvin. *Face Recognition: Understanding LBPH Algorithm* [online]. 10. 11. 2017, 1 [cit. 2019-03-26]. Dostupné z: https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b

[30] MARTINEZ, Aleix. Fisherfaces. *Scholarpedia* [online]. 2011, 6(2) [cit. 2019-03-26]. DOI: 10.4249/scholarpedia.4282. ISSN 1941-6016. Dostupné z: http://www.scholarpedia.org/article/Fisherfaces

[31] LINDEBERG, Tony. Scale Invariant Feature Transform. *Scholarpedia* [online]. 2012, 7(5) [cit. 2019-03-27]. DOI: 10.4249/scholarpedia.10491. ISSN 1941-6016. Dostupné z: http://www.scholarpedia.org/article/Scale_Invariant_Feature_Transform

[32] BAY, Herbert, Andreas ESS, Tinne TUYTELAARS a Luc Van GOOL. Speeded-Up Robust Features (SURF). *BIWI* [online]. 1-14 [cit. 2019-03-31]. Dostupné z: file:///D:/Sta%C5%BEen%C3%A9%20soubory/eth_biwi_00517.pdf

[33] LI, Jun-Bao, Shu-Chuan CHU a Jeng-Shyang PAN. *Kernel learning algorithms for face recognition*. New York: Springer, [2014]. ISBN 978-1-4614-0160-5.
[34] WICKLIN, Rick. What is Mahalanobis distance?. *The DO Loop* [online]. 2012, 1 [cit. 2019-04-12]. Dostupné z: https://blogs.sas.com/content/iml/2012/02/15/what-is-mahalanobis-distance.html
[35] SMITH, Lindsay I. *A tutorial on Principal Components Analysis* [online]. 2002 [cit. 2019-04-12]. Dostupné z: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf
[36] KLOUDA, Karel. *Metoda nejmenších čtverců: řešení rovnic, které nemají řešení* [online]. 2017, 1 [cit. 2019-04-12]. Dostupné z: https://marast.fit.cvut.cz/cs/blog_posts/19
[37] HONZÍK, Petr. *Lineární modely, diskriminační analýza a podpůrné vektory* [online]. Fakulta elektrotechniky a komunikačních technologií, 2014 [cit. 2019-04-12]. Dostupné z: http://midas.uamt.feec.vutbr.cz/STU/Lectures/11_Linearni_modely_diskriminacni_analyza_a_SVM.pdf. Odborná přednáška. Vysoké Učení Technické v Brně.
[38] Weisstein, Eric W. "Laplace Transform." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/LaplaceTransform.html
[39] Weisstein, Eric W. "Hessian." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Hessian.html
[40] Weisstein, Eric W. "Gaussian Function." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/GaussianFunction.html
[41] *ICAO MRTD report* [online]. 7. Montréal, Québec: International Civil Aviation Organization, 2015 [cit. 2019-04-13]. ISBN 978-92-9249-798-9. Dostupné z: https://www.icao.int/publications/Documents/9303_p3_cons_en.pdf
[42] MELKUS, Adam. *Čtení elektronického pasu na OS Android* [online]. Brno, 2013 [cit. 2019-04-13]. Dostupné z: https://is.muni.cz/th/hop20/bp.pdf. Bakalářská práce. Masarykova univerzita. Vedoucí práce Mgr. Ing. Zdeněk Říha, Phd.
[43] Czech passport 2006 MRZ data. *Wikipedie* [online]. 2007 [cit. 2019-04-13]. Dostupné z: https://cs.wikipedia.org/wiki/Soubor:Czech_passport_2006_MRZ_data.jpg
[44] CVCA & DVCA. *Cryptomathic* [online]. [cit. 2019-04-13]. Dostupné z: https://www.cryptomathic.com/products/pki-id/cvca-dvca
[45] About (OpenCV). *OpenCV* [online]. [cit. 2019-04-19]. Dostupné z: https://opencv.org/about/

[46] Main Page (EmguCV). *EMGU* [online]. [cit. 2019-04-19]. Dostupné z: http://www.emgu.com/wiki/index.php/Main_Page

[47] GOSSMAN, John. *Introduction to Model/View/ViewModel pattern for building WPF apps* [online]. 2005, 8. 10. 2005, , 1 [cit. 2019-04-19]. Dostupné z: https://blogs.msdn.microsoft.com/johngossman/2005/10/08/introduction-to-modelviewviewmodel-pattern-for-building-wpf-apps/

[48] SMITH, Marvin. *Introduction to OpenCV* [online]. Nevada, Reno [cit. 2019-05-01]. Dostupné z: https://www.cse.unr.edu/~bebis/CS485/Lectures/Intro_OpenCV.pdf. Odborná přednáška. University of Nevada.

[49] MALÝ, Martin. YAML: Serializační formát pro ukládání dat. *Zdroják* [online]. 2009, 1(1), 1 [cit. 2019-05-01]. Dostupné z: https://www.zdrojak.cz/clanky/yaml-serializacni-format-pro-ukladani-dat/

# Přílohy

Následující přílohy jsou obsaženy na CD, které je součástí odevzdávané práce.

- Příloha A – Projekt Enterprise Architect
- Příloha B – Programátorská dokumentace
- Příloha C – Zdrojové kódy
- Příloha D – Testovací fotografie
- Příloha E – Uživatelská příručka