



System pro automatické detekování a identifikaci osob na základě digitálního obrazu obličeje

Bakalářská práce

Studijní program:

B2646 Informační technologie

Studijní obor:

Informační technologie

Autor práce:

Martin Matoušek

Vedoucí práce:

doc. Ing. Josef Chaloupka, Ph.D.

Ústav informačních technologií a elektroniky



Zadání bakalářské práce

System pro automatické detekování a identifikaci osob na základě digitálního obrazu obličeje

Jméno a příjmení: **Martin Matoušek**
Osobní číslo: M16000043
Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie
Zadávací katedra: Ústav informačních technologií a elektroniky
Akademický rok: **2020/2021**

Zásady pro vypracování:

1. Seznamte se s problematikou zpracování a rozpoznávání obrazu, s ohledem na řešenou problematiku detekování a identifikování osob z digitálního obrazu obličeje.
2. Navrhněte a realizujte program pro (polo)automatické stahování fotografií konkrétních osob z webu.
3. Použijte daný program pro vytvoření databáze obličejů osob. V databázi by se mělo vyskytovat alespoň tři sta mužů a stejný počet žen. Každá osoba by měla být v databázi zastoupena alespoň sto různými snímky.
4. Otestujte na vytvořené databázi algoritmy pro detekování a identifikaci osob.
5. Na základě předchozí analýzy (testování), vytvořte komplexní systém, který bude sloužit pro automatické detekování obličejů osob a jejich identifikaci.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

Dle potřeby dokumentace
30-40 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

[1] ŠONKA, M., Hlaváč, V., Boyle, R.: Image processing, analysis, and machine vision. Fourth Edition. Australia: Cengage Learning, ISBN 978-1-133-59369-0, 2015.

[2] GONZALEZ, Rafael C. a Richard E. WOODS. Digital image processing. Global edition. New York: Pearson, ISBN 978-1-292-22304-9, 2017.

[3] HLAVÁČ, V., Sedláček, M.: Zpracování signálů a obrazů. 2. přeprac. vyd. Praha: ČVUT, 255 s. ISBN 978-80-01-03110-0, 2007.

Vedoucí práce:

doc. Ing. Josef Chaloupka, Ph.D.
Ústav informačních technologií a elektroniky

Datum zadání práce:

19. října 2020

Předpokládaný termín odevzdání:

17. května 2021

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

prof. Ing. Ondřej Novák, CSc.
vedoucí ústavu

V Liberci dne 19. října 2020

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

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

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci 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.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

25. února 2021

Martin Matoušek

Poděkování

Děkuji svému vedoucímu bakalářské práce doc. Ing. Josefu Chaloupkovi, Ph.D. za odborné a přínosné konzultace, rady k vypracování práce, a hlavně za vstřícné vedení práce. Dále bych chtěl poděkovat své přítelkyni za podporu, trpělivost a pochopení. A také své rodině za podporu při studii i při bakalářské práci.

Anotace

Tato bakalářská práce se zaměřuje na problematiku zpracování a rozpoznání obrazu, primárně na problematiku detekování a identifikování osob z digitálního obrazu obličeje.

V teoretické části je řešena hlavně analýza hlavních komponent (PCA) a metoda podpůrných vektorů (SVM). U těchto metod je popsána jejich funkce a využití, různé variace těchto metod a obecný postup jejich implementace.

Cílem vlastní praktické části je navrhnout a realizovat aplikaci pro poloautomatické stahování fotografií konkrétních osob z webu a pomocí tohoto programu vytvořit databázi obličejů osob. Tato databáze obsahuje zhruba 300 mužů a 300 žen. Na dané databázi jsou otestovány a porovnány různé algoritmy pro detekování a identifikaci osob. Je testována funkčnost, přesnost a využití jednotlivých metod pro rozpoznání pohlaví, věku a samotných osob, na základě jejich obličejů. Z těchto testů a experimentů jsou vyvozeny porovnání mezi různými metodami. Na různých parametrech je porovnávána efektivita daných algoritmů a jejich vhodnost pro využití v určitých situacích, případně použití pro rozdílná data.

Na základě předchozí analýzy a testování je vytvořena komplexní aplikace, která slouží pro automatické detekování obličejů osob a jejich identifikaci. Aplikace umožňuje na vložených obrazových datech či videích identifikovat obličeje a rozpoznat pohlaví, věk a danou osobu. Aplikace využívá metody a externí knihovny, které jsou v předchozích fázích testování a experimentace vyhodnoceny, jako nejvhodnější pro dané využití a použitá data.

Klíčová slova

zpracování obrazu, rozpoznání obličeje, identifikace osob, databáze osob, analýza hlavních komponent (PCA), metoda podpůrných vektorů (SVM)

Anotation

This bachelor's thesis focuses on the issue of image processing and recognition., primarily on the issue of detection and identification of persons from the digital image of the face.

In the theoretical part, the analysis of the main components (PCA) and the method of supporting vectors (SVM) are addressed. For these methods, their function, usage, various variations of these methods and the general procedure for their implementation is described.

The aim of the practical part is to design and implement an application for semi-automatic downloading of photos of specific persons from the web and using this application to create a database of people's faces. This database contains roughly 300 men and 300 women. Various algorithms for detecting and identifying people are tested and compared on this database. The functionality, accuracy and use of individual methods for recognizing gender, age and the individuals themselves, based on their faces, is tested. From these tests and experiments, comparisons between different methods are derived. The effectiveness of given algorithms and their suitability for use in certain situations, or use for different data are compared for different parameters.

Based on previous analysis and testing, a comprehensive application is created to automatically detect people's faces and identify them. The application allows you to identify faces and identify gender, age and person on the input image data or videos. The application uses methods and external libraries that are evaluated in the previous stages of testing and experimentation as the most suitable for the given use and the data used.

Keywords

image processing, face recognition, person identification, person database, principal component analysis (PCA), support vector machines (SVM)

Obsah

1	Úvod.....	11
2	Využití metody z oblasti zpracování obrazu.....	12
3	Analýza hlavních komponent (Principal Component Analysis)	13
3.1	Škálování proměnných (Variable scaling).....	13
3.1.1	Skupinové škálování (Group scaling).....	14
3.1.2	Blokové škálování (Block scaling).....	14
3.1.3	Škálování logaritmického rozpadu (Log decay scaling)	15
3.1.4	Poissonovo škálování (Poisson scaling).....	15
3.1.5	Paretovo škálování (Pareto scaling).....	15
3.1.6	Odchylkové škálování (Variance scaling).....	15
3.1.7	Centrování a škálování ve třídách (Class centroid centering and scaling).....	16
3.2	Trénování metodou PCA	16
3.2.1	Standardizace dat	16
3.2.2	Výpočet vlastních vektorů a vlastních čísel	16
3.2.3	Výběr hlavních komponent	17
3.2.4	Projekce dat do nového příznakového prostoru	17
4	Metoda podpůrných vektorů (Support Vector Machine).....	18
4.1	Lineární klasifikátory	18
4.2	Nelineární klasifikátory.....	19
4.2.1	Jádrová (kernelová) metoda	20
4.2.2	Hraniční pásmo (margin).....	21
4.3	Hyperparametry SVM	22
4.3.1	Soft-margin.....	22
4.3.2	Parametry jádra	23
5	Detekce obličeje	25
5.1	Kaskádový klasifikátor.....	25
5.1.1	Příznaky.....	26
5.1.2	Integrální obraz	26
5.1.3	Učení klasifikačních funkcí.....	26
5.1.4	Pozorná kaskáda (Attentional Cascade).....	27
5.2	Metoda histogramu orientovaných gradientů	27
5.3	Víceúlohové kaskádované konvoluční síť.....	29
6	Rozpoznání pohlaví a věku.....	30

6.1	Model age-gender-estimation	30
7	Algoritmy PCA a SVM.....	32
7.1	Trénování a testování PCA	32
7.2	Trénování a testování SVM	32
8	Databáze a získávání dat.....	33
8.1	Databáze obrázků	33
8.2	Google Images Download.....	33
8.3	Fatkun Batch Image Download	34
9	Zpracování obrazových dat.....	36
9.1	Zpracování stažených dat.....	36
9.2	Transformace obrázků.....	37
10	Experimenty.....	39
10.1	Detekce obličeje.....	39
10.2	Rozpoznání věku	40
10.3	Rozpoznání pohlaví.....	43
10.4	Rozpoznání osob.....	45
11	System pro detekci a identifikaci osoby	47
12	Závěr	50

Seznam použitých zkratek

Zkratka	Popis
PCA	Principal Component Analysis
SVM	Support Vector Machine
CNN	Cascaded Convolutional Networks
MTCNN	Multitask Cascaded Convolutional Networks
ResNet	Residual neural network
URL	Uniform Resource Locator
RGB	red, green, blue
HOG	Histogram of Oriented Gradients
API	Application Programming Interface

1 Úvod

Různé formy strojového učení se v posledních letech rozšířily do většiny oborů. Jednou z nejdůležitějších součástí se stalo rozpoznávání objektů v obraze. Jedním z nejaktuálnějších využití může být například systém autonomních dopravních prostředků, které musí situaci kolem sebe vyhodnocovat právě pomocí rozpoznávání objektů. Technologii lze také využít k optimalizaci webových obrázkových vyhledávačů, či k rozpoznávání obličejů. Rozpoznávání obličejů je prvním krokem pro další analýzu obličeje. Může jít o rozpoznání věku, pohlaví, případně i emocí na základě mimiky obličeje. To může být využito v mnoha odvětvích. V poslední době tyto algoritmy používají například mobilní telefony při focení a zpracování fotografií. Rozpoznání obličeje je také základem pro automatizované odezírání ze rtů a například i animaci obličejů.

Metody detekce obličeje a rozpoznání jednotlivých osob lze využít pro biometrickou identifikaci. Ta může být uplatněna například pro rozpoznávání osob na bezpečnostních kamerách, což lze využít pro střežení objektů, či například pro porovnání obličejů při řešení trestných činů – záznamy kamer z banky, obchodu, případně pouliční bezpečnostní kamery. Dalším využitím může být také nahrazení různých typů zabezpečení – například dveře či trezory na klíč, kód, přístupovou kartu a podobné. Biometrický senzor se může využít také pro přístup do střežených objektů, nebo odemykání dveří či trezorů s omezeným přístupem. V dnešní době se stává běžným odemknutí mobilních telefonů, případně počítačů za pomoci detekce a identifikace obličeje.

Cílem této bakalářské práce je rozpoznání osobností na obrázcích či videích. Databáze této práce je z důvodu přístupnosti a množství dostupných obrázků online, zaměřena na známé osobnosti, jako například herce, zpěváky, moderátory či politiky. Práce se zaměřuje nejen na rozpoznání osob, ale také jejich pohlaví a věku. Proto je důležitá diverzifikace dat. Databáze tedy obsahuje široké rozpětí různých věkových kategorií, také zastoupení žen i mužů je poměrově vyvážené.

2 Využité metody z oblasti zpracování obrazu

Základním prvkem v problematice detekce a identifikace osob v obraze je nalezení obličeje. Toho může být dosaženo za použití mnoha různých metod. Mezi nejvyužívanější metody patří tzv. HAAR klasifikátory a výpočet histogramu orientovaných gradientů (HOG). HAAR klasifikátory jsou založeny na nalezení haarovských příznaků. Tyto příznaky jsou výrazné komponenty obličeje, jako například oči, nos či pusa. Výpočet histogramu orientovaných gradientů počítá výskyty orientace hran v určité oblasti obrázku. Obrázek je rozdělen do buněk a pro každou z nich jsou vypočítány směry a velikosti gradientu pro všechny tři barevné kanály. Toto umožňuje v obrázku zachytit detaily obličeje, to mohou být například oči, nos, pusa, nebo kraje obličeje.

Další prvek v průběhu identifikace je zpracování dat. To je důležité pro normalizaci datové sady a také rychlost algoritmu. Obrázky obličeje jsou zmenšeny na předem určenou velikost a převedeny na černobílé, což zajišťuje výrazné snížení objemu dat. Zmenšení obrázku snižuje velikost exponenciálně a také nastává trojnásobné snížení velikosti obrázku převodem z barevné na černobílou.

Poslední částí je samotná identifikace, v této práci zaměřená na věk, pohlaví a samotnou osobu. V tomto kroku byly v práci použity a otestovány následující možnosti:

- Analýza hlavních komponent
- Metoda podpůrných vektorů
- Konvoluční neuronová síť

Analýza hlavních komponent je založena na extrakci rysů. Tímto se dosáhne snížení dimenze dat s co nejmenší ztrátou informace. Tyto rysy jsou komponenty s největším vlivem na rozpoznávání jednotlivých kategorií. Poté se pro správnou identifikaci používá metoda nejbližšího souseda. Tato metoda vyhledává z trénovací sady obrázků s minimální vzdáleností od určovaného obrázku a podle natrénovaných kategorií trénovacích dat určuje jeho třídu.

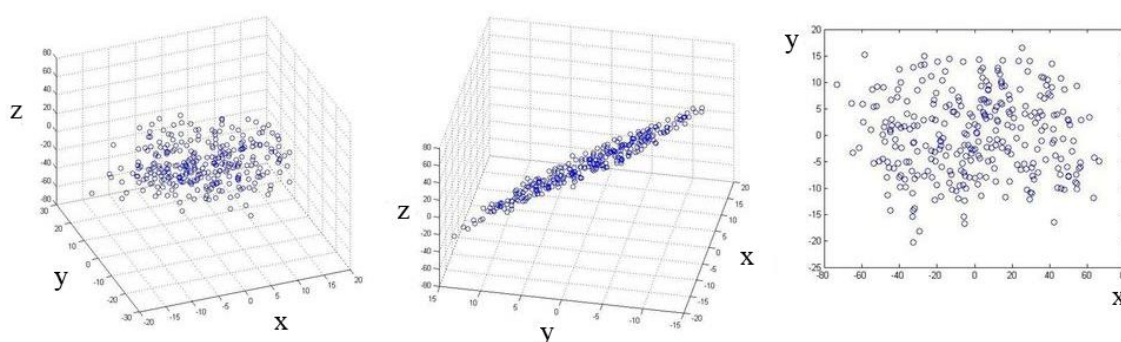
Při metodě podpůrných vektorů se vyhledává nadrovina, která rozděluje daný prostor na 2 poloprostory, kde se nachází majorita případů jedné třídy. Touto nadrovinou je lineární $n-1$ dimenzionální prostor. Tento prostor je vyjádřen lineární diskriminační funkcí, podle které je zjištěna kategorie určovaného obrázku.

Konvoluční neuronové sítě se skládají z vstupní vrstvy, skrytých vrstev a výstupní vrstvy.

3 Analýza hlavních komponent (Principal Component Analysis)

Hlavní myšlenkou analýzy hlavních komponent[1] je snížit rozměr dat, které mají velké množství souvisejících proměnných, ale také zachovat dostatečnou variabilitu v souboru dat. Toho lze dosáhnout extrakcí dominantních vzorů v matici z hlediska důležitosti obrazových dat pro výsledné skóre. Je důležité předem určit jaká data jsou využívána a pro jaké účely jsou zpracována. Musí tedy být předem určeno, jaké vzory se v daných datech hledají a jak jsou tyto prvky důležité. Obrázek 3.1 znázorňuje redukci počtu dimenzí. Za využití PCA je tato množina dat redukována z třídímního prostoru na dvoudímní. Data při tomto procesu přijdou pouze o minimum informací.

Redukce počtu dimenzí pomocí metody PCA



Obrázek 3.1: Redukce počtu dimenzí pomocí metody PCA[2]

Analýza hlavních komponent [3] rozpoznává vzory v datech a jedním z jejích využití je analýza vysokodímního dat, kterým není jednoduché porozumět jen pohledem na tak velké množství dat. Pro analýzu dat je důležité tato data redukovat na menší počet dimenzí a interpretovat výsledky. PCA se využívá pro výpočet vztahů ve velké korelované datové sadě. Cílem algoritmu PCA je redukovat tyto objemné korelované proměnné na menší počet proměnných. Tyto proměnné se nazývají hlavní komponenty. Hlavní motivací je vytvoření matice, která obsahuje maximální počet informací v minimálním počtu proměnných.

3.1 Škálování proměnných (Variable scaling)

Škálování proměnných [4] je běžná metoda používaná v metodách analýzy více proměnných. Tyto techniky předpokládají, že velikost měření je úměrná jeho důležitosti a míra šumu je podobná ve všech proměnných. Pokud tyto proměnné mají velmi odlišné měřítka, způsobené tím,

že jsou v různých jednotkách, velikost hodnot nemusí odpovídat jejich informativnosti. Měřítko může také být problematické, pokud některé proměnné obsahují více šumu než jiné. Pro běžné využití se používá normalizace, která využívá centrování k průměrné hodnotě a následně se každá proměnná dělí směrodatnou odchylkou daného prvku. Ve výsledku nabývá průměr každého prvku hodnoty nula a má směrodatnou odchylku rovnu jedné. Tímto způsobem se zajistí, že všechny proměnné budou mít pro model podobnou váhu. Z tohoto důvodu se při změně velikosti proměnných využívá i přizpůsobitelný parametr odsazení, který se přidává k směrodatné odchylce každého prvku před změnou velikosti. Pokud je odsazení nenulové, posouvá se diagonála matice o danou hodnotu. To způsobí, že hodnoty, které mají směrodatnou odchylku blízkou nule nebudou tak výrazně zvětšeny. Ve výsledku to zajišťuje, že tyto prvky nebudou mít tak velký vliv na model. Tento přístup ke změně velikosti je označován jako automatické škálování.

3.1.1 Skupinové škálování (Group scaling)

Podobně jako automatické škálování, skupinové škálování provádí změnu velikosti založenou na směrodatných odchylkách. Skupinové škálování se často využívá, pokud se data skládají z několika podobně velkých bloků proměnných. Každý blok obsahuje proměnné s určenou měrnou jednotkou, ale každý blok má jednotku jinou. Skupinové škálování se provádí rozdělením proměnných do podobně velkých bloků a každý blok mění velikost podle průměru jeho směrodatných odchylek. Směrodatná odchylka je vypočítána pro každou proměnnou a jejich průměr se používá pro změnu velikostí všech proměnných v bloku.

3.1.2 Blokové škálování (Block scaling)

Při kombinování různých typů proměnných, které nejsou centrované, dochází k tzv. blokovému škálování. Při tomto procesu každý blok mění velikost podle odmocniny sdílené odchylky jeho proměnných.

3.1.3 Škálování logaritmického rozpadu (Log decay scaling)

Škálování logaritmického rozpadu se využívá v hmotnostní spektrometrii a je zásadní alternativou k automatickému škálování pro data hmotností spektrometrie. Toto škálování je založeno na předpokladu, že v některých typech přístrojů je citlivost zhruba logaritmicky úměrná hmotnosti. Škálování logaritmického rozpadu se pokouší opravit chyby způsobené vyšší hmotností tím, že postupně snižuje logaritmickou funkci:

$$s_i = e^{-\frac{i}{n\tau}} \quad (3.1)$$

Proměnná s_i označuje škálování pro proměnnou i (seřazeno pro zvyšující se hmotnost), n je celkový počet proměnných a τ je parametr definující škálování od menších hmotností po vyšší. Čím menší je hodnota τ , tím více jsou prvky s větší hmotností nadhodnoceny s ohledem na prvky s menší hmotností. Oproti metodě automatického škálování, škálování logaritmického rozpadu nezpůsobuje centrování dat.

3.1.4 Poissonovo škálování (Poisson scaling)

V mnoha případech je cílem škálování úprava velikosti každé proměnné, aby míra šumu byla ve všech případech stejná. V případě, že šum je zhruba proporcionální odmocnině signálu v každé proměnné, může být použito Poissonovo škálování (také známo jako škálování průměrem odmocnin). Tato metoda mění velikost každé proměnné odmocninou průměru dané proměnné. Pokud je dominantní zdroj šumu proporční k odmocnině signálu, pomůže to efektivně upravit všechny proměnné na stejnou úroveň šumu. Může se také zavést odsazení, které zajišťuje, že se nadměrně nezdůrazňují proměnné s téměř nulovým průměrem.

3.1.5 Paretovo škálování (Pareto scaling)

Tato metoda je podobná Poissonově metodě škálování. Může být využívána, když se očekává, že šum bude odpovídat odmocnině směrodatné odchylky proměnných. I v tomto případě to znamená, že data nebudou centrována.

3.1.6 Odchytkové škálování (Variance scaling)

Tato metoda je velmi podobná automatickému škálování a používá se v případě, kdy lze očekávat, že šum bude proporční k směrodatné odchylce proměnných. Ale oproti automatickému škálování, tato metoda necentruje data.

3.1.7 Centrování a škálování ve třídách (Class centroid centering and scaling)

Tato metoda je variací na automatické škálování, která se používá v případě, že data jsou v podkategoriích identifikovaných určitou třídou. Data jsou nejprve centrována podle tříd a poté jsou škálována sdílenou směrodatnou odchylkou tříd.

3.2 Trénování metodou PCA

Trénování metodou PCA [5] probíhá v následujících krocích.

3.2.1 Standardizace dat

Standardizace dat je pro PCA velmi důležitá, protože PCA je velmi citlivá na odchylky. To znamená, že pokud existují příliš velké rozdíly měřítek příznaků, tak příznaky s velkým měřítkem budou pro model výrazně dominantní oproti příznakům s malým měřítkem.

Například příznak s rozmezím hodnot od nuly do sta bude mnohem důležitější pro model než příznak s rozmezím od nuly do jedné, a to povede ke zkreslení výsledků. Tomuto problému předchází transformace dat na stejné měřítko. K tomu se využívá standardizace, která upraví příznaky tak, aby jejich průměr byl roven nule a jejich odchylka rovna jedné. Pro tento účel se hodnota každého příznaku vypočítá jako:

$$x_i = \frac{x_i - \bar{x}}{\sigma_x} \quad (3.2)$$

Proměnná x_i označuje hodnotu příznaku, \bar{x} je průměrná hodnota všech příznaků x a σ_x je směrodatná odchylka x .

3.2.2 Výpočet vlastních vektorů a vlastních čísel

Vlastní vektory a vlastní čísla kovarianční matice představují jádro PCA. Vlastní vektory (hlavní komponenty) určují směr nového prostoru příznaků a vlastní čísla určují jejich velikost. Vlastní čísla tedy určují odchylku dat kolem nové osy příznaků. To znamená že korespondující vlastní čísla vyjadřují, jak velké odchylku má transformovaný příznak. Pro získání vlastních čísel a vlastních vektorů musí nejprve být vypočítána kovarianční matice.

Kovarianční matice, je matice, která v každém prvku představuje kovarianci mezi dvěma příznaky. Tato matice se poté rozkládá na její vlastní vektory a vlastní čísla. Součet druhých mocnin všech prvků ve vlastním vektoru musí vždy být roven jedné. Tím lze ověřit korektnost daného vlastního vektoru.

3.2.3 Výběr hlavních komponent

Při PCA dochází k redukci dimenzí původního prostoru příznaků tak, že se promítá do menšího podprostoru, kde tvoří osy vypočítané vlastní vektory. Vlastní vektory určují pouze směr. Pro vlastní čísla ale platí, že vlastní vektory s nejnižšími vlastními čísly nesou nejméně informací o distribuci dat. Tyto vlastní vektory tedy mohou být zanedbány. Proto je k nalezení nového seznamu hlavních komponent potřeba sestupně seřadit vlastní čísla a vybrat vlastní vektory korespondující k největším vlastním číslům.

Hlavním důvodem k výběru několika hlavních příznaků je, že jejich rozptyl je dostatečný k popisu datové sady. Ztráta ostatních příznaků v tom případě ovlivňuje přesnost modelu jen v minimální míře, anebo je pro model výrazně důležitější získaná rychlost než ztráta určitého procenta přesnosti.

Po seřazení vlastních vektorů podle vlastních čísel je potřeba určit, kolik hlavních komponent je potřeba vybrat pro dostatečnou přesnost modelu. To lze vypočítat jako podíl vlastního čísla k sumě vlastních čísel. Z tohoto výpočtu vystupuje procentuální hodnota, která určuje, jakým dílem daný vlastní vektor ovlivňuje daný příznak.

3.2.4 Projekce dat do nového příznakového prostoru

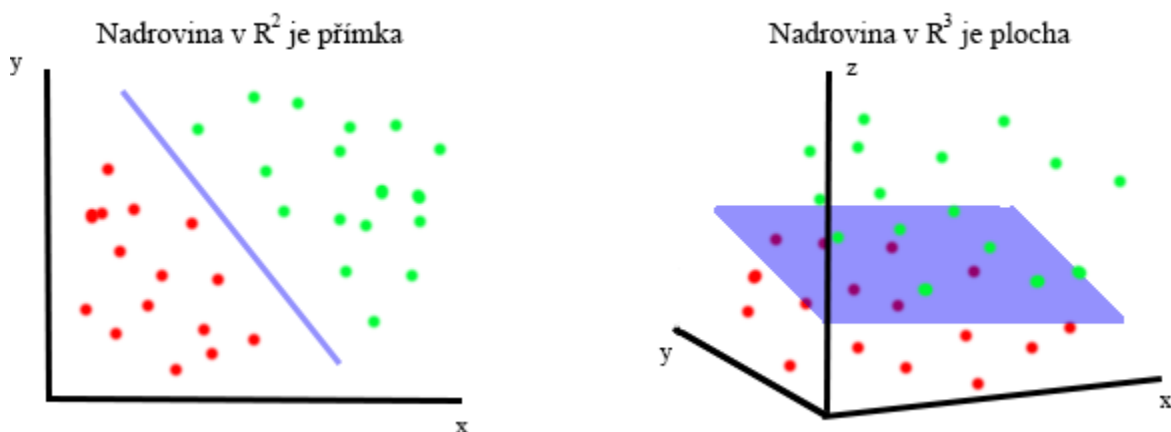
Vlastní prostor obsahuje vybrané vlastní vektory a používá se k transformaci vstupních dat do nového příznakového prostoru s redukováným počtem dimenzí. Tato transformace se provede vynásobením matice vstupních dat s vlastním prostorem.

$$Y = X * W \quad (3.3)$$

Proměnná Y označuje výslednou matici v novém příznakovém prostoru, X je matice vstupních dat a W je vlastní prostor. Po této operaci má každý prvek počet příznaků určený velikostí vlastního prostoru.

4 Metoda podpůrných vektorů (Support Vector Machine)

Support vector machines (SVM) [6] neboli metoda podpůrných vektorů je metoda strojového učení, která rozděluje data nejčastěji do 2 tříd. Toho lze dosáhnout lineárním rozdělením dat, kde hledáme takovou nadrovinu, která nejlépe oddělí data 2 daných tříd. Tou je $n-1$ dimenzi-onální prostor který rozděluje prostor na 2 poloprostory.



Obrázek 4.1: Nadroviny ve dvourozměrném a třírozměrném prostoru

SVM je využíváno pro svojí vysokou přesnost, schopnost pracovat s daty s vysokým dimen-zemi a flexibilitu v práci s různými typy dat. SVM se řadí do kategorie kernelových metod. Jádrové metody jsou algoritmy, které závisí na datech pouze přes skalární součin. Skalární součin může v tom případě být nahrazen jádrovou funkcí, která počítá skalární součin v něja-kém prostoru příznaků (často ve vyšší dimenzi). Tento přístup má výhodu v tom, že může ge-nerovat nelineární rozhodovací meze za využití metod navržených pro lineární klasifikátory. Využití jádrové funkce také umožňuje použití klasifikátoru na data, která nemají žádnou jasnou reprezentaci ve vektorovém prostoru s fixní dimenzí.

K efektivnímu využití SVM je zapotřebí porozumění jejich funkčnosti. Při trénování SVM je tedy nutné rozhodnout několik rozhodujících faktorů: jak předzpracovat vstupní data, jaké po-užít jádro (kernel) a velmi důležité jsou i hyperparametry SVM i jádra. Tato rozhodnutí mohou výrazně pozitivně, ale i negativně ovlivnit výkon.

4.1 Lineární klasifikátory

Metoda SVM je příkladem lineárního, dvoutřídního klasifikátoru. Data pro dvoutřídní učební problematiku se skládají z objektů, které jsou označeny jedním z dvou označení korespondující

ke dvěma třídám – pro jednoduchost lze tyto třídy vnímat jako pozitivní a negativní příklady, značené jako +1 a -1. Vstupní data se skládají ze dvou množin, kde množina X se skládá z vektorů rozpoznávaných vzorů a množina Y obsahuje označení třídy přidružené k vektorům z množiny X.

Hlavním pojmem pro definici lineárního klasifikátoru je skalární součin mezi dvěma vektory, který je definován jako:

$$w^T x = \sum_i w_i x_i \quad (4.1)$$

Lineární klasifikátor je založen na lineární diskriminační funkci:

$$f(x) = w^T x + b \quad (4.2)$$

Vektor w označuje váhový vektor a b označuje bias. Za předpokladu, že bias je roven nule, všechna x , pro která platí, že $w^T x = 0$, prochází počáteční nadrovinou. Bias b posouvá tuto nadrovinu o danou hodnotu od počátku. Tato nadrovinu rozděluje prostor na dva prostory na základě diskriminační funkce. Mez mezi prostory klasifikovanými jako pozitivní a negativní se nazývá rozhodovací mez. Rozhodovací mez je definována nadrovinou, která je lineární. Pokud je data v prostoru potřeba rozdělit pomocí nelineární funkce, označuje se tento klasifikátor jako nelineární.

4.2 Nelineární klasifikátory

Pro data, která nelze rozdělit lineárně se zavádí nelineární SVM. Nelineární SVM řeší to, že nelineární klasifikátory jsou významně náročnější než lineární. Základní rozdíl oproti lineárnímu SVM je převod prostoru příznaků do vyšší dimenze, ve které ho lineárně rozdělit lze. Toho lze dosáhnout zavedením nelineární funkce ϕ . V prostoru příznaků vyšší dimenze je poté diskriminační funkce definována jako:

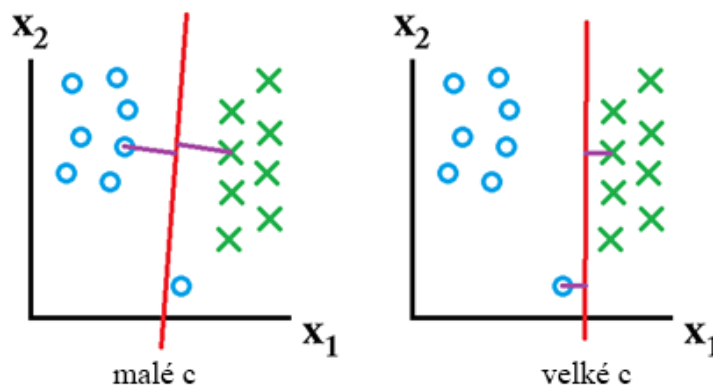
$$f(x) = w^T \phi(x) + b \quad (4.3)$$

Tento způsob explicitního výpočtu nelineárních příznaků ale neškáluje dobře s počtem vstupních příznaků. Pokud se tento způsob zavádí pro vstupní hodnoty, kterými jsou vektory, jejich

dimenze příznaků je v novém prostoru příznaků kvadratická. Z toho plynou i kvadratické nároky na paměť a kvadratická časová náročnost výpočtu diskriminační funkce klasifikátoru. Proto se pro data s vyšší dimenzí využívá jádrový (kernelový) přístup.

4.2.1 Jádrová (kernelová) metoda

Jádro je definováno jako $K(x, x') = (x^T x' + c)^d$, kde x a x' jsou vstupní vektory trénovacích či testovacích dat, d je parametr, který představuje stupeň polynomu a c je parametr určující velikost hraničního pásma, který pro větší hodnoty způsobuje přetrénování na trénovací sadě a při menších hodnotách maximalizuje hraniční pásmo mezi dvěma třídami.



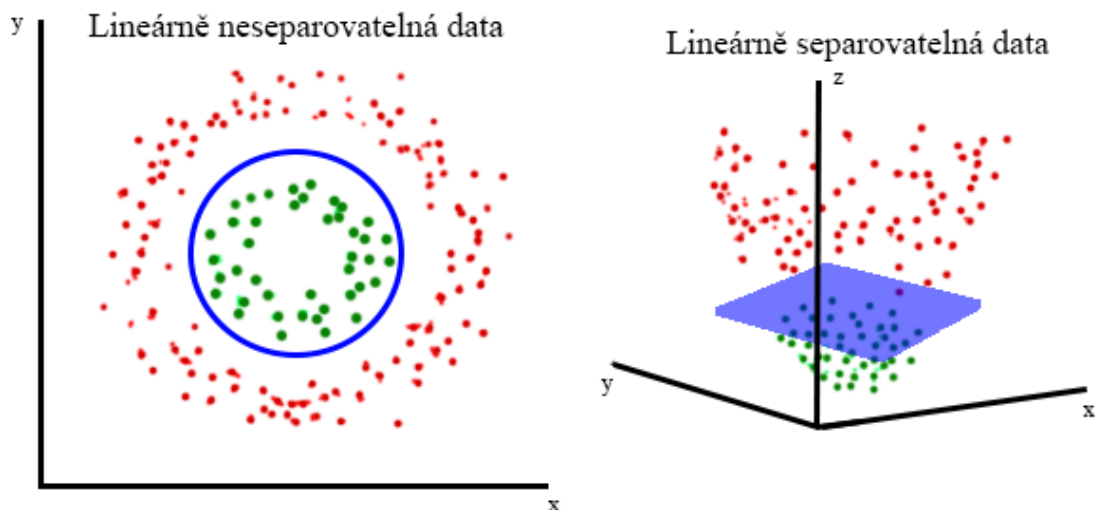
Obrázek 4.2: Parametr velikosti hraničního pásma

Dalším využívaným jádrem je Gaussovské jádro, definováno jako:

$$k(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (4.4)$$

Proměnná γ je parametr, který určuje šířku normálního (Gaussovského) rozdělení a $\|x\|$ je normalizované x vypočítáno jako $\sqrt{x^T x}$.

Po zavedení jádra je vytvořena nadrovina vyšší dimenze, ve které jsou třídy lineárně separovatelné. Tato nadrovina je v původní dimenzi ekvivalentní k polynomiální nadrovině.



Obrázek 1.3: Převod dat do prostoru s vyšší dimenzí

4.2.2 Hraniční pásmo (margin)

Pro danou nadrovinu je určen okraj či hraniční pásmo (margin), které ohraničuje nejbližší body k nadrovině od pozitivních a negativních prvků. Z čistě geometrického hlediska je hraniční pásmo nadrovinu určeno váhovým vektorem w s ohledem na množinu dat D .

$$m_D(w) = \frac{1}{2} \hat{w}^T (x_+ - x_-) \quad (4.4)$$

V tomto vzorci je \hat{w} jednotkový vektor ve směru w a lze předpokládat že x_+ a x_- jsou ve stejné vzdálenosti od rozhodovací meze.

$$f(x_+) = w^T x_+ + b = a \quad (4.5)$$

$$f(x_-) = w^T x_- + b = -a \quad (4.6)$$

Konstanta a musí být větší než nula. Pro smysluplné využití geometrických hraničních mezí je vhodné zafixování hodnoty diskriminační funkce a určení parametru a rovno jedné. Za těchto podmínek funkce pro výpočet hraniční meze nabývá tohoto tvaru:

$$m_D(w) = \frac{1}{2} \hat{w}^T (x_+ - x_-) = \frac{1}{\|w\|} \quad (4.7)$$

Pevné hraniční pásmo (hard-margin) se využívá pro lineárně separovatelnou množinu dat. Klasifikátor maximálního hraničního pásma je diskriminační funkce, která maximalizuje geometrické hraniční pásmo $\frac{1}{\|w\|}$ a minimalizuje $\|w\|^2$. Optimalizace je tedy ve tvaru:

$$\frac{1}{2} \|w\|^2 \quad (4.8)$$

Tato optimalizace zajišťuje, že klasifikátor klasifikuje každý prvek správně. To je možné pouze za předpokladu, že data jsou lineárně separovatelná. V praktickém využití většinou data nejsou kompletně lineárně separovatelná a i v případě že jsou, je možné dosáhnout lepších výsledků povolením nesprávné klasifikace pro určité body. Pro povolení chyb v klasifikaci je potřeba změnit omezení nerovnosti v rovnici:

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad (4.9)$$

ξ_i označuje volné proměnné, které umožňují prvkům být v oblasti hraničního pásma (také zvané margin error, $(1 \geq \xi_i \geq 0)$), nebo i chybně klasifikovány do nesprávné třídy ($\xi_i \geq 1$). Prvek je tedy chybně klasifikován, pokud je hodnota jeho volné proměnné větší než jedna, to znamená, že součet volných proměnných je vázán na počet nesprávně určených prvků. Optimalizace je tedy rozšířena o $C \sum_i \xi_i$, aby penalizovala chybné klasifikace a margin error. Optimalizace se tedy upraví na tvar:

$$\frac{1}{2} \|w\|^2 + C \sum_i \xi_i \quad (4.10)$$

Konstanta C , která musí nabývat kladné hodnoty nastavuje relativní důležitost maximalizace hraničního pásma a minimalizace volnosti (počet volných proměnných neboli slack). Tato formulace je nazývána soft-margin SVM.

4.3 Hyperparametry SVM

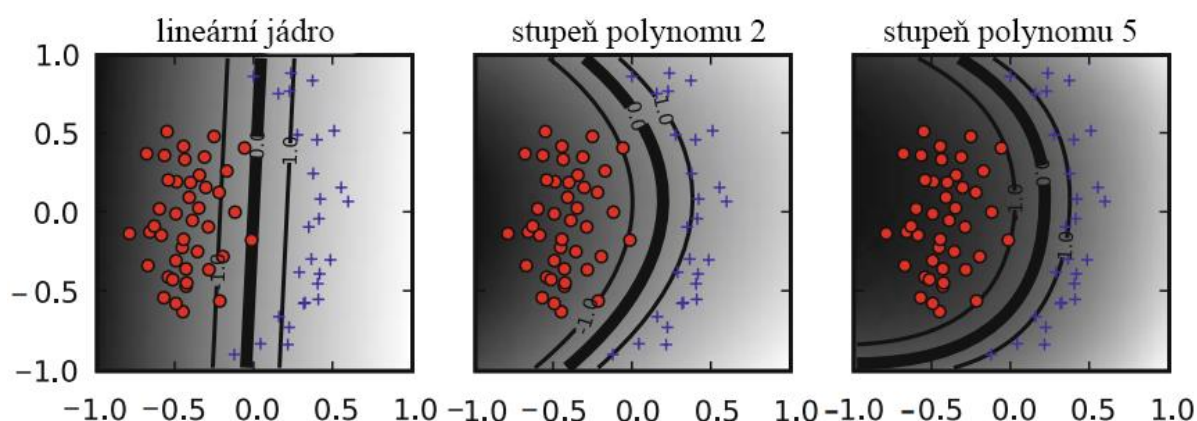
4.3.1 Soft-margin

Změnou velikosti této konstanty lze ovlivnit vliv chybně klasifikovaných prvků a chyby hraniční meze (viz Obrázek 2). Malá hodnota konstanty C , zvětšuje se tolerovaný počet chybně určených prvků a prvků v hraničním pásmu. To znamená, že model je robustnější a není tolik ovlivněn prvky, které se umístěním vymykají normě jejich třídy (outliers). Naopak při zvýšení

hodnoty C je výrazně zvýšen důraz na penalizaci chybných klasifikací a prvků v hraničním pásmu. Tudiž může docházet k přetrénování modelu, který je poté příliš ovlivněn výjimkami v trénovací množině dat.

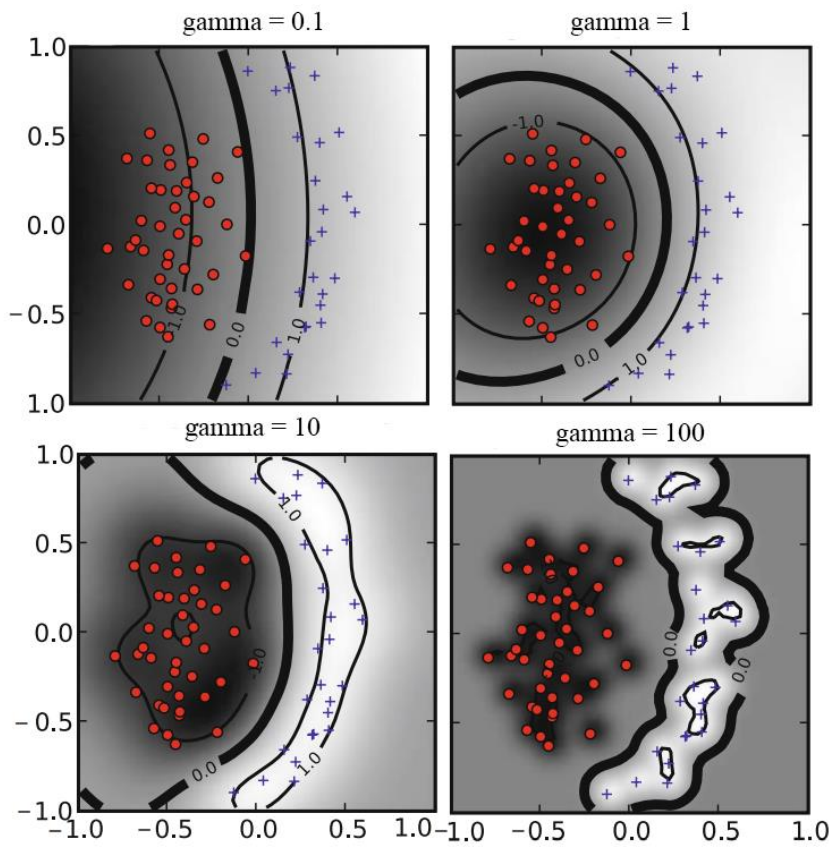
4.3.2 Parametry jádra

Tyto parametry mají významný efekt na hranici rozhodování (decision boundary). Stupeň polynomiálního jádra a šířka Gaussovského jádra kontrolují flexibilitu výsledného klasifikátoru. Nejnižší stupeň polynomu je lineární. Ten není vhodný, pokud existují nelineární vztahy mezi příznaky. Na obrázku 4.4 lze vidět, že zakřivení hranice rozhodování může výrazně ovlivnit přesnost klasifikátoru, a proto je nutné určit nejlepší možný stupeň polynomu pro daná data.



Obrázek 4.4: Efekt stupně polynomu na jádro[6]

Šířka Gaussovského jádra je označována symbolem γ a určuje vliv hodnoty $\|x - x'\|^2$ na hranici rozhodování. Při menších hodnotách γ je hranice rozhodování blízká lineární. Při zvyšování hodnoty γ dochází k nárůstu zakřivení hranice rozhodování. Pokud je γ příliš vysoké, dochází k tomu, že hodnoty diskriminační funkce jsou skoro konstantní mimo nejbližšího okolí koncentrace prvků. Obrázek 4.5 ukazuje, že při nižších hodnotách se hranice rozhodování blíží lineární nadrovině a při příliš vysoké hodnotě γ dochází kvůli přílišné přesnosti k významnému přetrénování.



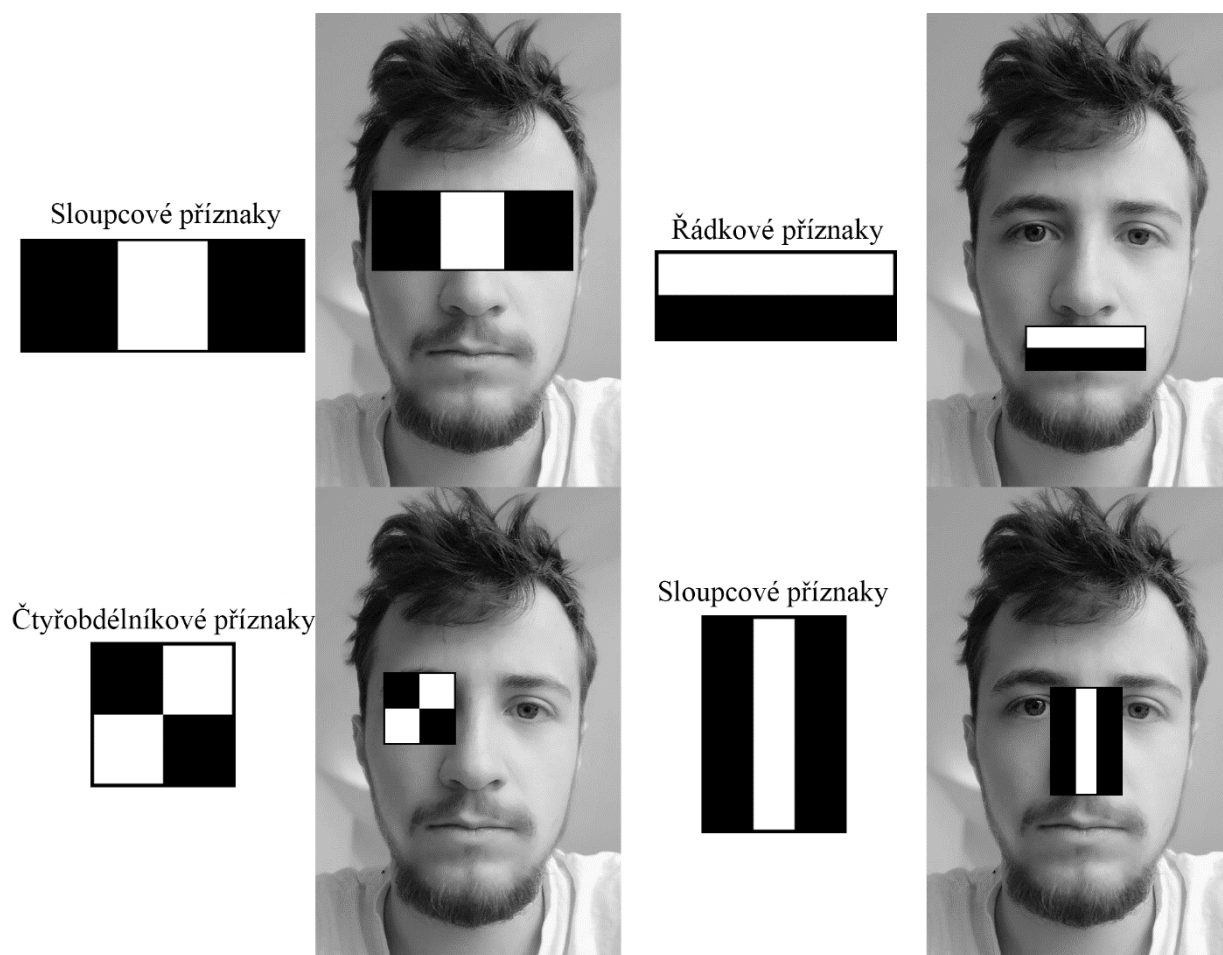
Obrázek 4.5: Parametr šířky pro Gaussovské jádro[6]

5 Detekce obličeje

5.1 Kaskádový klasifikátor

Kaskádový klasifikátor (Haar Cascade Classifier) [7] je metoda rozpoznání objektů. Navrhli ji Paul Viola a Michael Jones v roce 2001. Kaskádová funkce je trénovaná na vysokém počtu pozitivních a negativních obrázků. Poté je využívána pro rozpoznání natrénovaných objektů v dalších obrázcích.

Positivní obrázky jsou v tomto případě obrázky obličejů, negativní naopak jakékoliv obrázky bez obličejů. Klasifikátor se učí rozpoznávat, zda obrázek obsahuje obličej. Poté přichází na řadu extrakce příznaků z těchto obrázků. K tomu se využívají haarovské příznaky (viz Obrázek 5.1).



Obrázek 5.1: Příznaky kaskádového klasifikátoru

5.1.1 Příznaky

Detekce objektů je v kaskádovém klasifikátoru prováděna na základě jednoduchých příznaků. Nejvýznamnějším důvodem, proč používat příznaky namísto pixelů je schopnost příznaků natrénovat přímo na specifickou kategorii. Druhým kritickým důvodem je, že systém příznaků pracuje výrazně rychleji než systém založený na pixelech. Příznaky se dělí do tří kategorií. Hodnota dvou-obdélníkových příznaků je rozdíl mezi součty dvou obdélníkových oblastí. Tyto oblasti mají stejnou velikost a tvar a horizontálně či vertikálně sousedí. Tří-obdélníkové příznaky počítají součet dvou venkovních obdélníků odečtený od součtu vnitřního obdélníku. Čtyř-obdélníkový počítá rozdíl součtů mezi diagonálními páry obdélníků.

Tímto způsobem jsou všechny možné velikosti a pozice každého jádra využity k výpočtu obrovského množství příznaků. Pro zrychlení výpočtů se využívají integrální obrazy.

5.1.2 Integrální obraz

Integrální obraz je způsob reprezentace obrazu, kde každý pixel nabývá hodnoty součtu pixelů vlevo a nahoře od jeho pozice:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (5.1)$$

Hodnota $ii(x, y)$ označuje integrální obraz a $i(x, y)$ je původní obraz. Tento způsob zajistí výrazné zrychlení výpočtů.

5.1.3 Učení klasifikačních funkcí

S danou množinou příznaků a trénovací sadou dat pozitivních a negativních obrázků, může být každá hodnota, které algoritmus dosáhne, použita jako klasifikační funkce. V kaskádovém klasifikátoru se tedy používá AdaBoost pro výběr menšího počtu příznaků a pro trénování klasifikátoru. Pro dostatečně přesný klasifikátor v teorii stačí malé množství příznaků. Ty je ale potřeba identifikovat. AdaBoost nachází optimální příznak pro separaci obrázků na pozitivní a negativní, to jsou příznaky s nejmenší chybovostí na tréninkových datech. Pro každý příznak je poté nalezen optimální práh, který minimalizuje počet chybných klasifikací. Finální klasifikátor je váženým součtem těchto slabých klasifikátorů, protože samostatně tyto klasifikátory nemohou správně klasifikovat obrázek. Pro maximální efektivitu se při klasifikaci využívá konceptu kaskády klasifikátorů.

5.1.4 Pozorná kaskáda (Attentional Cascade)

Pro zvýšení výkonu detekce a výrazné zrychlení času výpočtů se využívá tzv. pozorná kaskáda. Ta zajišťuje, že jednoduché (a rychlé) klasifikátory odmítnou velké množství negativních částí obrázku, ale přijímají naprostou většinu pozitivních. Toho lze dosáhnout posunutím rozhodovacího prahu tak, aby počet nesprávně chybných detekcí byl minimální. Tyto jednodušší klasifikátory tedy odstraní velké množství částí obrázku, než přijdou na řadu komplexnější, výpočetně náročné klasifikátory, které zajišťují minimalizaci počtu nesprávně pozitivních detekcí. V prvních úrovních klasifikace se většinou používá nejnižší počet klasifikátorů pro maximalizaci rychlosti. Pokud část obrazu selže při jakémkoli z průchodů, je vyřazena z klasifikace. Jen části obrazu, které projdou všemi fázemi klasifikace jsou klasifikovány jako obličejová oblast.

Struktura kaskády reaguje na fakt, že v obrázku je naprostá většina částí negativních. A proto se kaskáda snaží vyřadit co největší množství negativ v raných fázích algoritmu. Pozitivní detekce v kterékoli fázi postupně způsobují průchod dalšími klasifikátory. Všechny následující klasifikátory pracují pouze s daty, které projdou prvními fázemi kaskády. To pro model znamená, že do nich přichází data s náročnější detekční problematikou. Čím hlouběji v modelu se klasifikátory nachází, tím preciznější musí být. Z toho důvodu mají také vyšší počet chybných pozitiv, aby se předešlo chybnému vyřazení pozitivních prvků.

5.2 Metoda histogramu orientovaných gradientů

Algoritmus pro získávání histogramu orientovaných gradientů [8] počítá výskyty orientace hran v určité oblasti obrázku. Obrázek je nejprve rozdělen do malých, spojených částí, zvaných buňky. Pro každou buňku je vypočítán histogram orientací hran. Směry histogramu jsou rovnoměrně rozprostřeny přes 0–180 stupňů nebo 0–360 stupňů. Počty histogramů jsou normalizovány pro kompenzaci osvětlení. Toho lze dosáhnout akumulací míry energie lokálního histogramu ve větších spojených oblastech a využití výsledku pro normalizace všech buněk v daném bloku. Kombinace těchto histogramů znázorňuje finální deskriptor histogramu orientovaných gradientů.

Výpočet histogramu orientovaných gradientů (HOG).

- Při předzpracování obrazových dat může probíhat například oříznutí nebo změna velikosti.

- Pro výpočet HOG deskriptoru je potřeba vypočítat horizontální a vertikální gradienty. Toho lze dosáhnout filtrováním obrázků pomocí těchto jader (kernelů):

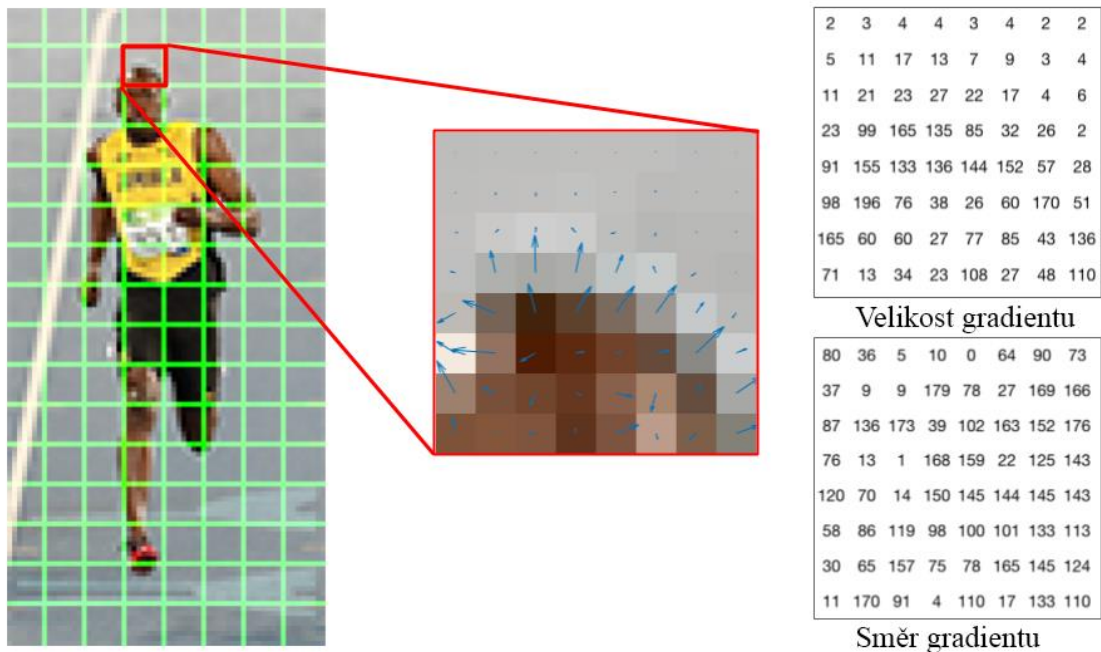
$$\begin{array}{ccc} -1 & 0 & 1 \\ & -1 & \\ & 0 & \\ & & 1 \end{array}$$

- Poté je nalezena velikost a směr gradientů, pomocí těchto vzorců:

$$g = \sqrt{g_x^2 + g_y^2} \quad (5.2)$$

$$\theta = \arctan \frac{g_y}{g_x} \quad (5.3)$$

- Každý pixel má velikost a směr gradientu. Pro barevné obrázky jsou vyhodnocovány gradienty pro všechny tři barevné kanály. V tom případě je velikost gradientu maximum těchto tří hodnot a směr je určen maximálním gradientem.
- Obrázek je rozdělen do buněk vybrané velikosti a histogram gradientů je vypočítán pro každou z nich. Tím je možné snížit šum dat a umožňuje zachytit detailnější vlastnosti obrázku. V rozpoznávání obličeje to můžou být například oči, nos, ústa nebo kraje obličeje.



Obrázek 5.2: Histogram orientovaných gradientů

- Histogram je vytvořen jako zásobníky, které shlukují směry do určitého počtu kategorií. Tyto kategorie určují směr, například 0, 20, 40, ... 340 stupňů. Do těchto zásobníků se ukládají hodnoty velikosti, poměrně rozdělené podle jejich směru.
- Histogramy jednotlivých buněk jsou poté normalizovány, aby nebyly výrazně ovlivněny variací osvětlení.
- Pro výsledný vektor pro celý obrázek jsou všechny histogramy jednotlivých buněk spojeny do jednoho a vzniká jeden obrovský vektor příznaků.[9]

5.3 Víceúlohové kaskádované konvoluční síť

Víceúlohové kaskádované konvoluční síť neboli MTCNN (Multitask Cascaded Convolutional Networks)[10] využívají inherentní korelaci mezi detekcí a zarovnáním pro zvýšení jejich účinnosti. Tento aplikační rámec vyzdvihuje kaskádovou architekturu se třemi úrovněmi hlubokých konvolučních sítí pro nalezení obličeje a jeho umístění za postupného zvyšování rozlišení vstupních obrazových dat.

MTCNN spojuje různé způsoby zpracování dat, pro zvýšení rychlosti i účinnosti:

- Kaskádový detektor obličeje, který navrhli Viola a Jones, využívající AdaBoost dosahuje dobrých výsledků v reálném čase. Může však degradovat při reálném použití, kde mohou různé podmínky ovlivnit účinnost daných klasifikátorů.
- Hluboké kaskádové konvoluční síť dosahují pozoruhodných výsledků v oblasti klasifikace obrázků. Vzhledem k těmto úspěchům jsou kaskádové konvoluční síť využívány i pro rozpoznávání obličejů. Avšak, pro jejich komplexní strukturu, je tento přístup pro praktické využití velmi časově náročný.

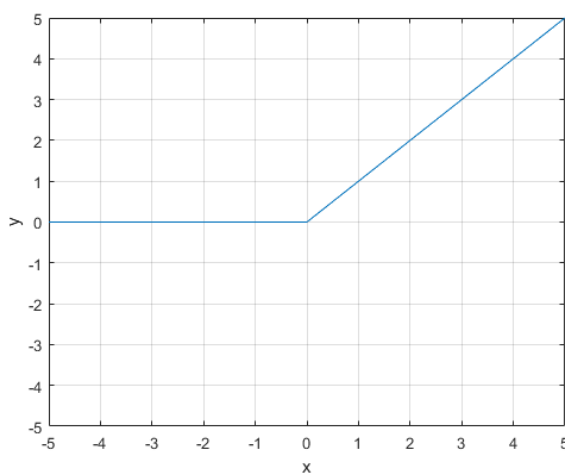
Mnoho metod pro rozpoznání a nalezení umístění obličeje nevyužívá jejich korelaci. Tento aplikační rámec integruje tyto dvě úlohy a využívá sjednocené kaskádové konvoluční síť pomocí víceúlohového učení. Tato konvoluční síť se skládá z tří fází. V první fázi, za využití jednoduché rychlé konvoluční síť, produkuje kandidátní okénka. V druhé fázi, je velké množství okýnek neobsahující obličeje odmítnuto komplexnější konvoluční sítí. A konečně využívá účinnější konvoluční síť pro vylepšení výsledků a jako výstup vrací pět obličejových orientačních bodů. Za pomoci těchto metod, je výrazně zlepšen výkon algoritmu.

6 Rozpoznání pohlaví a věku

6.1 Model age-gender-estimation

Tento model je kerasovou implementací konvoluční neuronové sítě. Keras je knihovna pro práci s neuronovými sítěmi napsaná v pythonu. Byla vyvinuta se zaměřením na rychlé a jednoduché experimenty s neuronovými sítěmi. Model využívá architekturu Wide ResNet, která obsahuje tyto vrstvy [11]:

- Conv2D je konvoluční vrstva pro práci s obrazovými daty. Do této vrstvy vstupuje pole o rozměrech (počet obrázků) x (šířka obrázku) x (výška obrázku) x (hloubka obrázku) - počet bitů, které určují barvu bodu. Toto pole prochází určitý filtr, ten je definován svojí velikostí a váhami na jednotlivých místech filtru. Po průchodu konvoluční vrstvou jsou data transformována na mapu příznaků. Snižuje se tak počet parametrů, s kterými model pracuje. To způsobuje snížení nároků na paměť.
- Aktivační vrstva ReLU je typ aktivační funkce, která je matematicky definována jako $y = \max(0, x)$. ReLU je lineární pro pozitivní hodnoty a 0 pro negativní hodnoty. To znamená, že je jednoduchý a rychlý pro trénování i běh aplikace.



Graf 6.1: Funkce ReLU

- AveragePooling2D je vrstva snižující velikost výstupních dat. Ve většině případů snižuje velikost každé mapy příznaků na polovinu ve všech dimenzích. Jako filtr se ve většině případů používá blok 2x2 s krokem dvou pixelů. Average Pooling vypočte průměrnou hodnotu pro každý takový blok mapy příznaků. Z této vrstvy vystupuje minimalizovaná verze příznaků detekovaných ve vstupu.

- Softmax je aktivační vrstva, kterou tento model využívá při běhu aplikace. Jako vstupní hodnoty vstupují do softmaxu skóre, která aplikace získá po průchodu vstupních dat modelem. Softmax z těchto hodnot získává pravděpodobnosti pro každou z natrénovaných tříd. Součet těchto pravděpodobností musí být vždy 0. Výstupem je poté třída s nejvyšší pravděpodobností, která je potom využita jako predikce daného obrázku.

$$p(y = j|x) = \frac{e^{(w_j^T x + b_j)}}{\sum_{k \in K} e^{(w_k^T x + b_k)}} \quad (6.1)$$

7 Algoritmy PCA a SVM

7.1 Trénování a testování PCA

Do PCA vstupuje vektor n proměnných, v tomto případě obrázek převedený na jednorozměrný vektor. Tím se získává matice vektorů o rozměru (počet pixelů) \times (počet obrázků). Tato matice je normalizována odečtením vektoru průměrných hodnot. Vynásobením této matice s maticí transponovanou se vytvoří kovarianční matice o rozměru (počet obrázků) \times (počet obrázků). Z kovarianční matice se spočítají vlastní vektory. Vynásobením matice vektorů s vlastními vektory vznikne vlastní prostor o rozměru (počet pixelů) \times (počet obrázků). Poté se matice vektorů promítá do vlastního prostoru a vzniká výsledný vektor vah.

Pro testovací data se vytváří matice vektorů, která je promítnuta do natrénovaného vlastního prostoru. Vzniká vektor vah testovacích dat. K nalezení správné třídy se poté pro každý testovací vektor hledá natrénovaný obrázek s nejmenší vzdáleností. Pro hledání nejmenší vzdálenosti se používá například Euklidovská vzdálenost.

$$d_E(s_r, s_i) = \sqrt{\sum_{m=1}^M (s_r(m) - s_i(m))^2} \quad (7.1)$$

7.2 Trénování a testování SVM

Je zavedena matice vah o rozměru (počet pixelů) \times (počet tříd) a matice biasu o rozměru (počet tříd). Pro předem určený počet průchodů je model trénován na trénovací sadě. Trénovací data jsou při vstupu do modelu normalizována ze stupňů šedi (0 - 255) na hodnoty od nuly do jedné. Tyto hodnoty jsou poté násobeny maticí vah a je přičtena matice biasu. Tím je získáváno skóre. Skóre je porovnáno s maticí tříd, aby se minimalizoval tzv. hinge loss. Výstup funkce je matice obsahující -1 pro správnou třídu a 1 pro nesprávnou třídu, v případě, že $(1 + \text{skóre nesprávné třídy} - \text{skóre správné třídy})$ je větší než 0. Pokud tato podmínka není splněna, výstupem jsou nuly pro obě třídy. Výsledná matice je rovna gradientu na matici biasu a gradient na matici vah se získává násobením normalizovaných trénovacích dat a výsledné matice.

Testovací funkce je výrazně jednodušší. Do funkce vstupují testovací data. Ta jsou normalizována na hodnoty od 0 do 1. Tato matice je vynásobena natrénovanou maticí vah a je k ní přičtena natrénovaná matice biasu. Pokud maximální hodnota skóre odpovídá správné třídě daného obrázku, obrázek je vyhodnocen jako úspěšný.

8 Databáze a získávání dat

8.1 Databáze obrázků

V databázi osob se nachází přes 600 osob. Pro každou osobu databáze obsahuje umístění obrázků, jméno dané osoby, rozmezí věku, do kterého patří a pohlaví v tomto formátu:

```
{
  "people": [
    {
      "file": "D:\\BP downloads\\Adam_Vojtech",
      "name": "Adam Vojtěch",
      "age": "30-39",
      "gender": "male"
    },
    {
      "file": "D:\\BP downloads\\Adela_Gondikova",
      "name": "Adéla Gondíková",
      "age": "40-49",
      "gender": "female"
    },
    {
      "file": "D:\\BP downloads\\Agata_Pracharova",
      "name": "Agáta Prachařová",
      "age": "30-39",
      "gender": "female"
    },
  ],
}
```

V databázi se nachází výběr celebrit a veřejných osobností, převážně to jsou herci, ale také politici a zpěváci, případně také moderátoři. Tyto osoby byly vybrány primárně z důvodu množství dostupných fotografií na internetu. Z toho důvodu je pro tyto osoby jednoduché vytvořit relativně objemnou a různorodou množinu obrazových dat pro každou osobu v databázi. Seznam osob obsahuje vyvážený počet mužů a žen a také široké rozpětí věkových kategorií. Vzhledem k povaze seznamu osob, převažují osoby ve věku 20 až 50, ale databáze obsahuje i dostatek starších osob.

8.2 Google Images Download

Obrázky jsou stahovány pomocí knihovny `google_images_download`, která je stažena za pomoci instalátoru `pip` z repozitáře `python package index`. Je to program v `pythonu` pro příkazovou řádku, který vyhledává klíčová slova či klíčové fráze a stahuje nalezené obrázky do počítače. Pro stahování více než 100 obrázků, je zapotřebí nainstalovat knihovnu `Selenium`. `Selenium` je knihovna používaná pro tazkvaný *web crawling*, neboli automatické získávání dat

z online zdroje. Web crawling je jednoduchou cestou, jak získat obrovské množství dat v relativně krátkém čase. Ve větší míře, je z důvodu zátěže, na mnoha serverech zakázáno. Také je potřeba stáhnout program Chromedriver. Webdriver je open source nástroj určený primárně pro automatizované testování webových aplikací napříč mnoha prohlížeči. Také ale umožňuje automatizaci procházení webových stránek, uživatelských vstupů nebo spuštění javascriptu. Chromedriver je jeho varianta pro Google Chrome. Tento program je kompatibilní s pythonem 2 i 3. Lze jednoduše spustit v příkazovém řádku se specifikovanými parametry.

```
$ python google_images_download.py [Arguments...]
```

Také je možné implementovat do kódu v pythonu.

```
from google_images_download import google_images_download
response = google_images_download.googleimagesdownload()
absolute_image_paths = response.download({<Arguments...>})
```

Mezi používané argumenty patří:

Keywords - Obsahuje klíčová slova, či fráze, které poté program vyhledává. Pro více klíčových slov lze zadat vstup jako pole.

Limit - Určuje počet stahovaných souborů pro každé klíčové slovo.

Output_directory - Specifikuje hlavní adresář, kam se ukládají stažené obrázky. Celá struktura je následující `<output_directory><klíčové slovo><název obrázku>`.

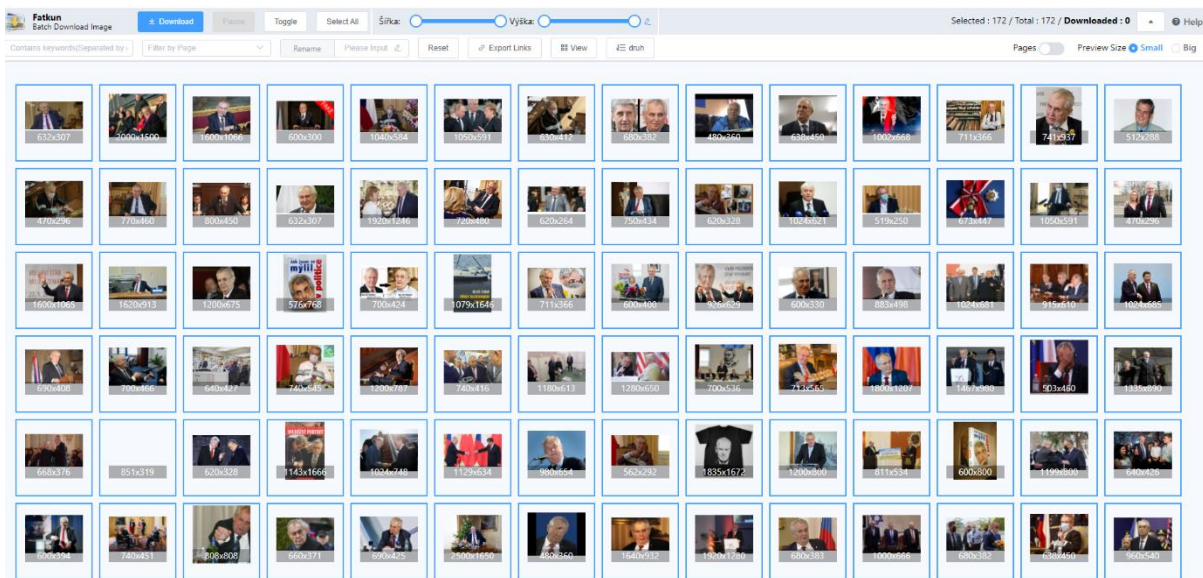
Chromedriver - Do tohoto argumentu se vkládá cesta k chromedriver.exe, který je třeba ke stažení většího počtu obrázků.

Pro každou osobu je tedy staženo prvních 200 obrázků z Google.

8.3 Fatkun Batch Image Download

Vzhledem k tomu, že Google změnil strukturu a odstranil `rg_metadata` (tyto data využíval `google_image_download` pro zpracování obrázků přímo z html), tak jej nadále nebylo možné využít pro stahování obrázků. Nyní už Google pouze odkazuje na URL licencovaných obrázků.

Proto bylo nutné využít jiného programu pro stahování velkého objemu obrázků, který využívá jiných metod pro zpracování a stažení obrázků. Fatkun batch image download je rozšíření do prohlížeče Google Chrome, které načte všechny URL z dané stránky. Proto je nutné na googlu načíst dostatečný počet obrázků, které potom program zobrazí v nově otevřeném okně prohlížeče. V tomto okně je možné vybrat jednotlivé obrázky, či například filtrovat podle velikosti obrázku.



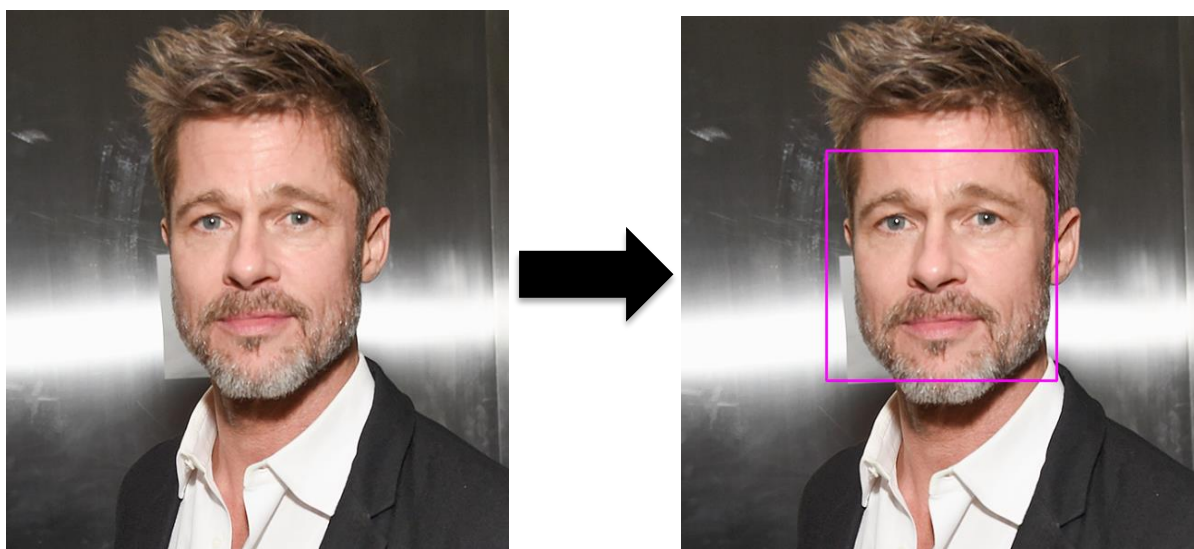
Obrázek 8.1: Fatkun Batch Image Downloader

9 Zpracování obrazových dat

9.1 Zpracování stažených dat

Po stažení jsou obrázky přejmenovány a jména, pohlaví, věk a umístění souboru zapsány do textového souboru, který slouží jako databáze jmen spojených s obrázky a informacemi o jednotlivých osobnostech. Také se pozmění název souboru z původního formátu *Jméno Příjmení* s diakritikou na *Jmeno_Prijmeni* bez diakritiky pro jednodušší práci s daty.

Knihovna dlib je sada nástrojů, psaná v C++, která obsahuje velké množství algoritmů strojového učení a metody zpracování obrazových dat. Dlib je open source a využívá se jak v počítačových a mobilních programech, tak v robotice a to v akademickém i průmyslovém využití. Na obrázcích jsou za využití knihovny dlib a její metody *get_frontal_face_detector* nalezeny obličej. Tato metoda nalezne na obrázku obdélníky, ve kterých se nachází obličej, jež jsou následně vyříznuty a uloženy. (Obrázky v plné velikosti jsou ponechány v samostatném adresáři pro testovací účely)



Obrázek 9.12: Detekce obličeje

Z takto zpracovaných obrázků jsou, za použití knihovny *face_recognition* a ručně vybraného vzorového obrázku, vybrány pouze obličej patřící dané osobě. Toho je dosaženo využitím metody *compare_faces*. Do této metody vstupuje výstup funkce *face_encodings*, která z obrazových dat vrátí 128 dimenzionální kódování obličeje pro každý obličej nalezený na obrázku. Do této funkce vstupuje obrázek, který obsahuje obličej a vrací kódování. Metoda *compare_faces* poté porovnává kódování známého obličeje (v tomto případě ručně vybraný obličej,

který jistě zobrazuje danou osobu) a kódování obličejů, u kterých je potřeba ověřit, zda patří stejné osobě. V této metodě je možné nastavit i toleranci, to znamená, jak velká vektorová vzdálenost mezi obličejí je stále považována za shodu.

Z dotazu na Google je možné obdržet nedostatek použitelných obrázků, to může znamenat špatnou kvalitu jednotlivých obrázků, na obrázcích jsou nesprávné osoby (například shoda jmen, případně to mohou být osoby příbuzné či související). Také je zde možnost, že obličej je na obrázku příliš natočený, nebo špatně viditelný.

Vzhledem k této nejednotnosti kvality stažených obrazových dat i jednotlivých obrázků se v databázi vyskytuje velká variabilita počtu obličejů u jednotlivých osob. To může způsobit nepřesnost u některých testů na detekci a rozpoznání daných osob.

9.2 Transformace obrázků

Pro využití rozpoznávací algoritmy je nutná standardizace vstupních dat. Při práci s obrázky obličejů je možné výrazně zmenšit velikost dat, jelikož se hledá kompromis mezi rychlostí a přesností algoritmu. Pro daná data je v mnoha případech dostačující i výrazně menší velikost obrázků. To je způsobeno hlavně dostupností vysoce kvalitních fotografií online, kde mohou fotky obličejů dosahovat rozlišení výrazně nad hranici při které by algoritmus pracoval dostačující rychlostí. Proto se před trénováním nastavují určité předem definované vlastnosti. U většiny algoritmů je obrázek převeden na černobílý, což způsobí jak zmenšení obsahu dat, tak také zjednoduší práci s daty, protože se počet procházených pixelů redukuje na třetinu a je odstraněna další variabilita dat (například nasvícení). Také se mění vstupní obrázky na čtvercové, u většiny algoritmů na 64x64 pixelů. To v důsledku znamená, že například z obrázku o velikosti



Obrázek 9.2: Transformace obrázku

256x256 v barevném modelu RGB s výslednou velikostí 256x256x3, se získá matice o velikosti 64x64. Tato matice již zabere mnohem méně prostoru v paměti a algoritmus ji zpracuje mnohonásobně rychleji.

Tento způsob je ale vhodné využívat jen u obrázků, které obsahují pouze obličej a také pouze na algoritmy, u kterých je toto zpracování dat vhodné. Při práci s celými obrázky, které obličej obsahují, je zapotřebí zachovat poměry stran jak trénovacích, tak testovacích dat. Testovací data mohou být na šířku i na výšku, takže převádění na jednu fixní velikost by mohlo způsobit problémy v algoritmu. Testovací obrázky by také měly zůstat výrazně větší, pro případ že jsou na nich vyobrazené obličej z větší vzdálenosti. Z těchto důvodů se pro úpravu testovacích obrázků určuje pouze velikost jedné strany a druhá strana je dopočítána, aby poměrově odpovídala původní velikosti. V tomto případě je to 600 pixelů na šířku a odpovídající hodnota na výšku. Pro trénovací data, je vhodné využít také poměrově odpovídající šířku a výšku. Tyto data však mohou být menší, například 64 nebo 128 pixelů na šířku s odpovídající výškou.

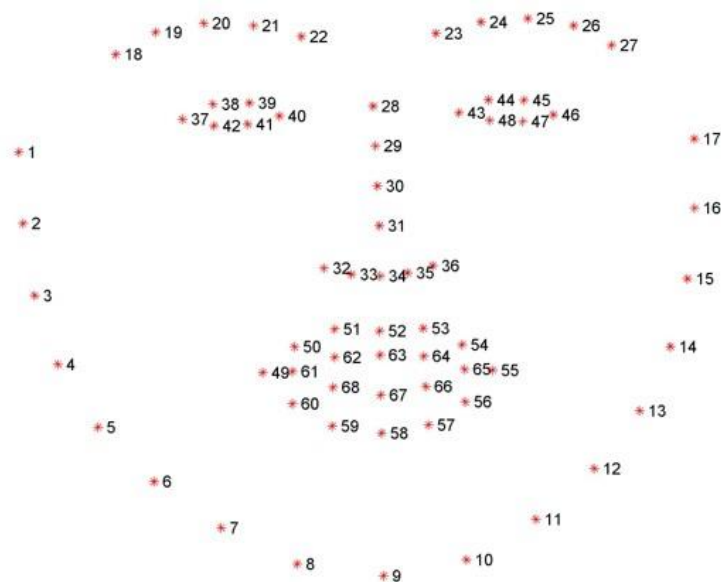
Zvolená velikost obrázků je vždy volena podle prioritizace rychlosti a náročnosti na paměť či přesnosti algoritmu. To určuje hlavně dostupný hardware, objem dat, potřebná rychlost a přesnost. Tyto hodnoty můžou být velmi různé pro různorodá využití.

10 Experimenty

10.1 Detekce obličeje

V tomto experimentu se porovnává úspěšnost nalezených obličejů při použití metody *get_frontal_face_detector* z knihovny *dlib* a kaskádového klasifikátoru (Haar Cascade Classifier) z knihovny *cv2*.

Knihovna *dlib* využívá orientační body obličeje. Metoda *get_frontal_face_detector* je založena na histogramu orientovaných gradientů (HOG). HOG počítá počet výskytů orientovaných gradientů v lokalizovaných částech obrázku. Nalezené orientační body ohraničují nejunikátnější části obličeje. To jsou například oči, nos, obočí, ústa a čelist. Pomocí těchto bodů je poté velmi rychle a přesně nalezena část obrázku, která obsahuje obličej, případně vyloučeno, že obrázek obsahuje obličej. Z této metody se získávají souřadnice, ve kterých se obličej nachází.[12]



Obrázek 10.1: Orientační body obličeje

Knihovna *cv2* využívá kaskádový klasifikátor. Tento klasifikátor se učí na velkém množství pozitivních (obrázky s obličejí) a negativních obrázků (obrázky bez obličejů). Z těchto obrázků je poté zapotřebí získat příznaky. Pro to se používají haarovské příznaky. Pro každou kalkulaci těchto příznaků je zapotřebí spočítat sumu pixelů pod černými a bílými částmi těchto příznaků.

Tento algoritmus za pomoci naučených dat nalézá, kde se na obrázku nachází určité části obličeje. Například oči, nos, ústa a podobně.



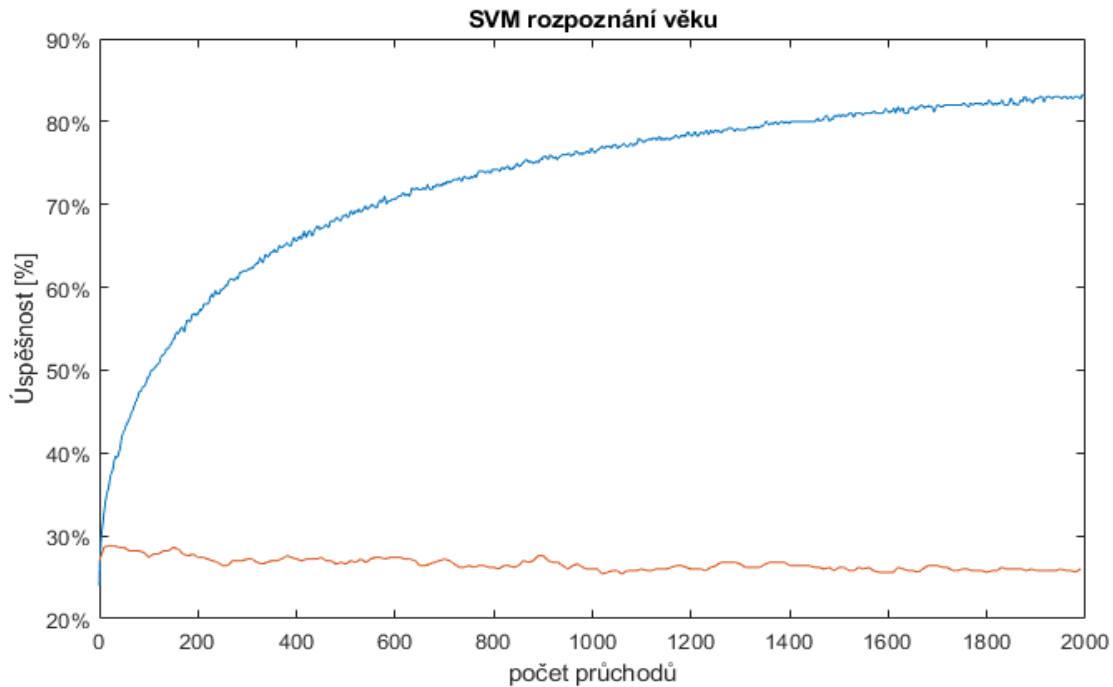
Obrázek 10.2: Haarovské příznaky

Při využití knihovny dlib bylo na použité databázi detekováno až o 20 % více obličejů, a to s velmi nízkým výskytem chybně pozitivních výsledků. Během pokusů kaskádový klasifikátor nebyl schopen detekovat obličeje, které byly příliš natočeny do strany a také obličeje, které byly z části zakryté, například vlasy, případně se nacházely na kraji obrázku a z malé části byly oříznuty.

10.2 Rozpoznání věku

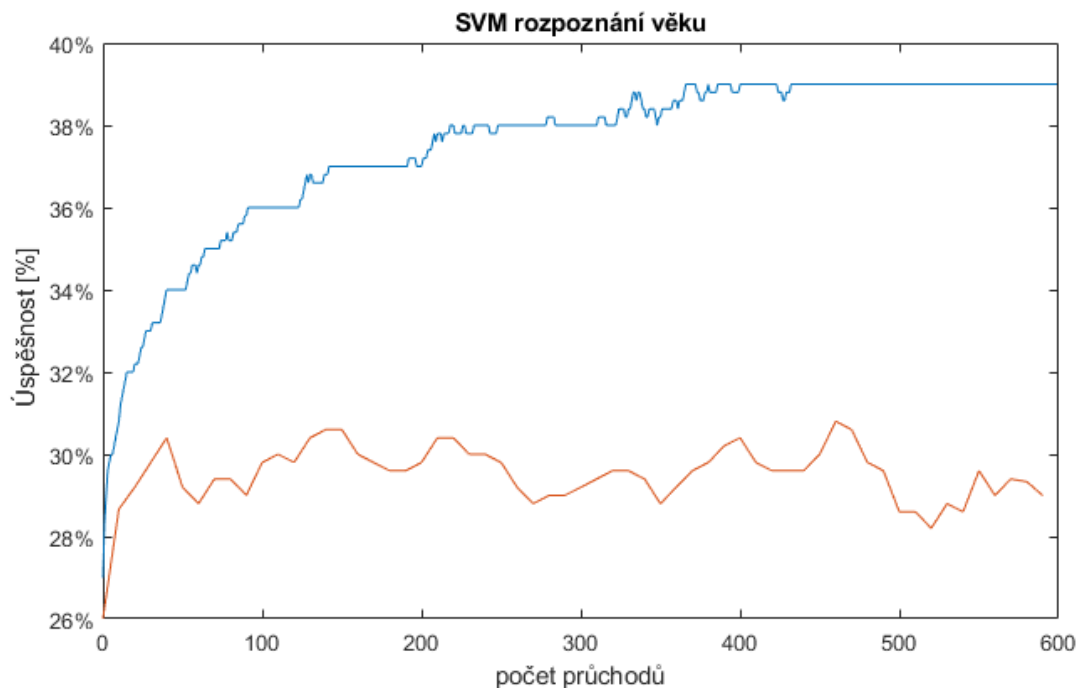
Pro rozpoznávání věku při prvním experimentu byla použita metoda SVM. Byla použita míra učení (learning rate) 0.0001, původní parametr váhy W nastaven jako vektor náhodných čísel generovaných metodou *random.rand* z knihovny numpy zmenšené stonásobně a parametr bias b jako vektor nul. U všech testů byly použity věkové kategorie 20-29, 30-39, 40-49, 50-59, 60-69 a 70 a více let. Za správně určené tedy bylo považováno rozpoznání faktu, že člověk na obrázku patří do dané věkové kategorie.

Při prvním testu bylo použito kolem 50 000 obrázků jako trénovací data a otestováno na 5 000 obrázcích z testovací sady. Během tohoto testu se každým průchodem postupně zvyšovala úspěšnost rozpoznání věku na trénovacích datech. Naopak úspěšnost rozpoznání věku na testovacích datech dosahovala téměř maximálních hodnot 27 % již po deseti průchodech. Po třiceti průchodech bylo dosaženo maximální hodnoty 30 %. Poté úspěšnost pouze kolísala mezi 25 % - 30 %. Toho bylo dosaženo i přes to, že na trénovacích datech po deseti průchodech byla také pouze 30% úspěšnost. Po třiceti průchodech dosahovala už 40 % úspěšnosti. Přesnost na trénovacích datech se ze začátku zvyšovala relativně rychle až do 500 průchodů na 70 %. Poté se rychlost učení začala postupně ale výrazně snižovat a během dalších 1500 průchodů se zvýšila pouze o dalších 12 %.



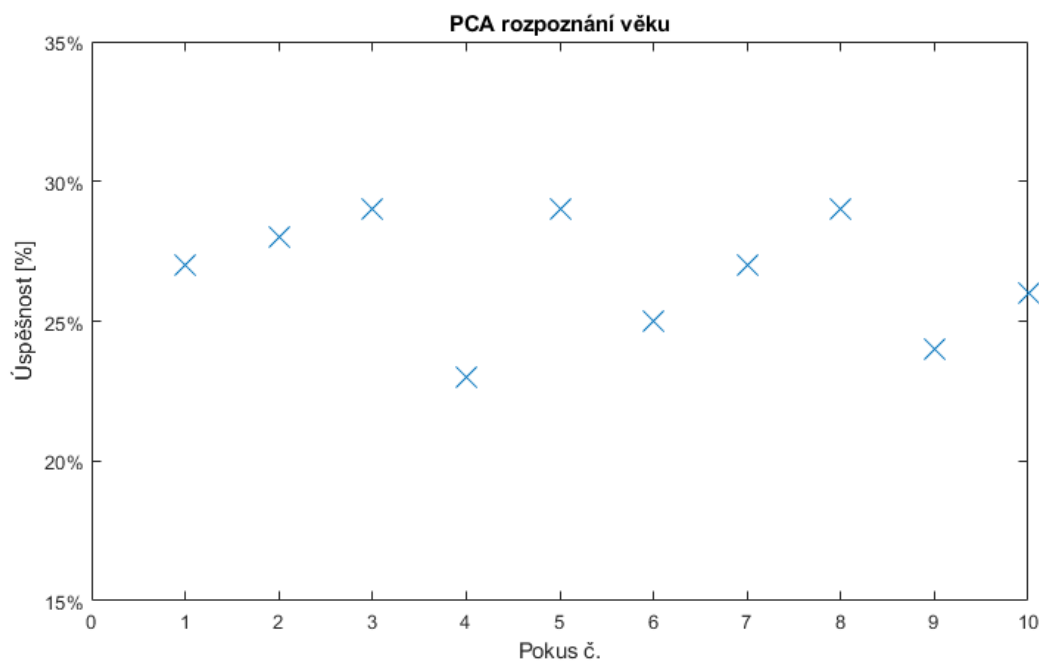
Graf 10.1: rozpoznání věku pomocí SVM

Při druhém testu metodou SVM bylo pro trénování použito 5 000 obrázků a 50 000 obrázků použito jako testovací data. Během tohoto testu postupovala úspěšnost rozpoznávání věků na trénovacích datech výrazně pomaleji, než při menším počtu trénovacích dat. Také se postup zvyšování úspěšnosti zastavil na 39 %. Při testu také proběhlo 2 000 průchodů, ale vzhledem k zastavení učení toto již není zaneseno do grafu, ale je pouze zobrazena stagnace úspěšnosti. Přesnost rozpoznání na testovacích datech je obdobná jako při předchozím testu. Po deseti průchodech dosáhla 27 %, po třiceti průchodech 32 % a poté začala také kolísat. Úspěšnost kolísala mezi 28 % a 32 %, takže byl experiment mírně úspěšnější i přes to, že přesnost na trénovacích datech dosáhla maximálně 39 %.



Graf 10.2: rozpoznání věku pomocí SVM

Pro třetí test byla zvolena metoda PCA. Jako trénovací data bylo zvoleno 50 000 obrázků. Z důvodu časové náročnosti PCA byl pro testování použit menší počet testovacích dat. Při jednotlivých experimentech se úspěšnost pohybovala vždy mezi 23 % -29 % v závislosti na náhodně zvolených obrázcích.

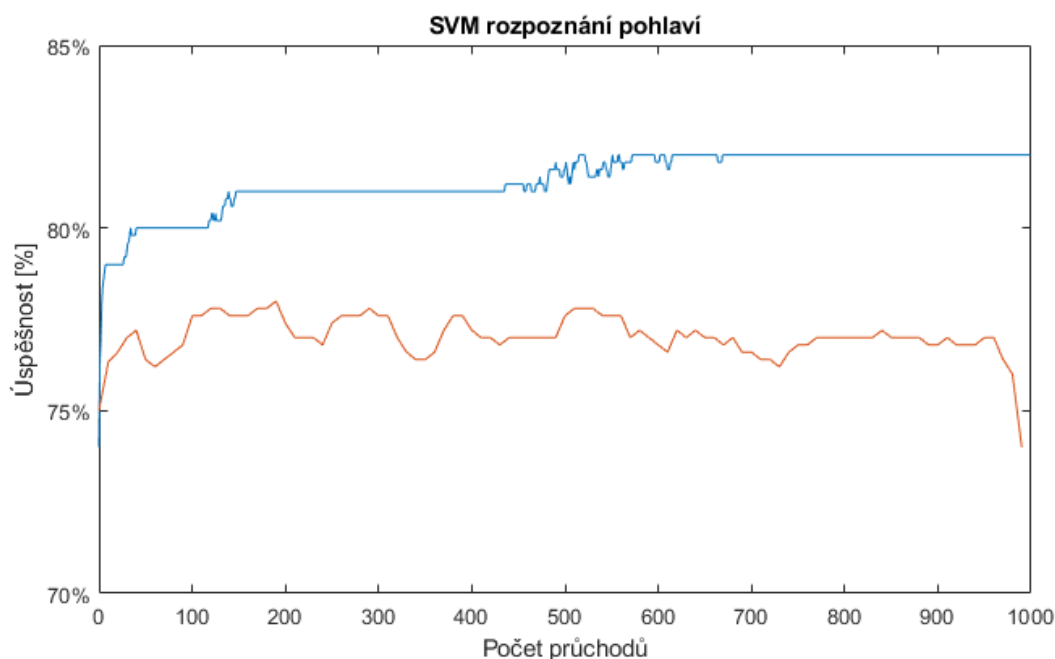


Graf 10.3: rozpoznání věku pomocí PCA

10.3 Rozpoznání pohlaví

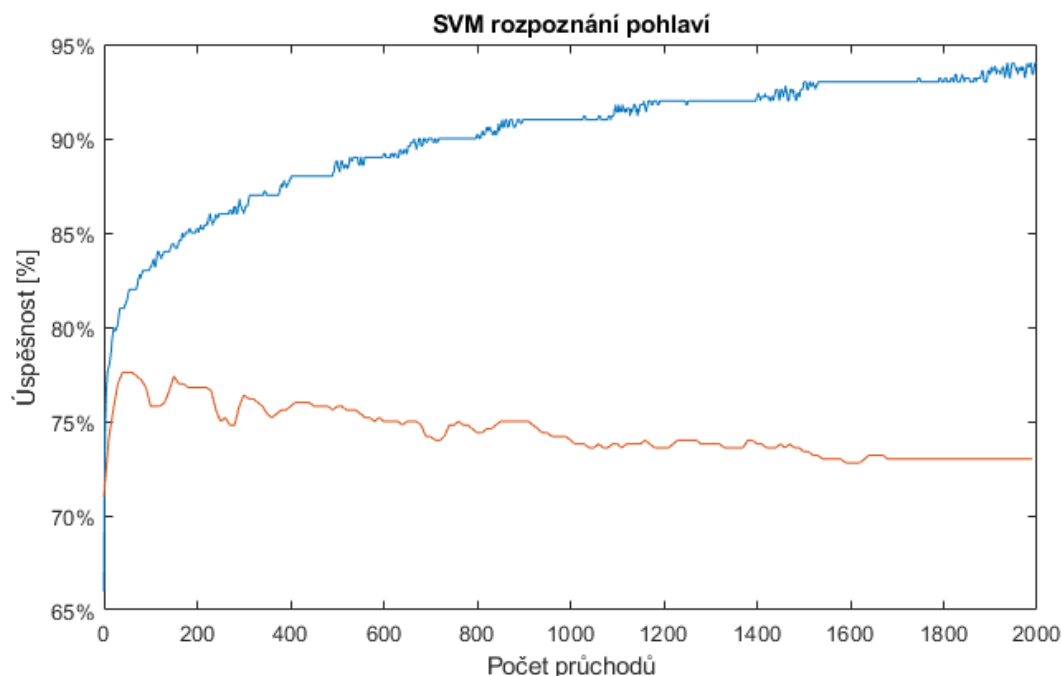
Pro první část experimentu na rozpoznání pohlaví byla zvolena metoda SVM, míra učení 0.0001, původní parametr váhy W nastaven jako vektor náhodných čísel zmenšených stonásobně a parametr biasu jako vektor nul.

Při prvním testu bylo použito 50 000 obrázků jako trénovací data a součástí testovací sady bylo 5 000 obrázků. Již od začátku trénování byla míra učení velmi pomalá, na trénovacích datech dosáhla již po prvním průchodu 74 % přesnosti. Po druhém průchodu dosáhla 77 % a poté se jen velmi pomalu zvyšovala a kolísala od 78 % do 82 %. Na této úspěšnosti se i učení zastavilo. Úspěšnost na testovacích datech byla zjišťována po každých deseti průchodech testovacích dat. Po deseti průchodech na trénovacích datech bylo na testovací sadě dosaženo 75% přesnosti a poté pouze kolísala mezi 74 % a 79 %.



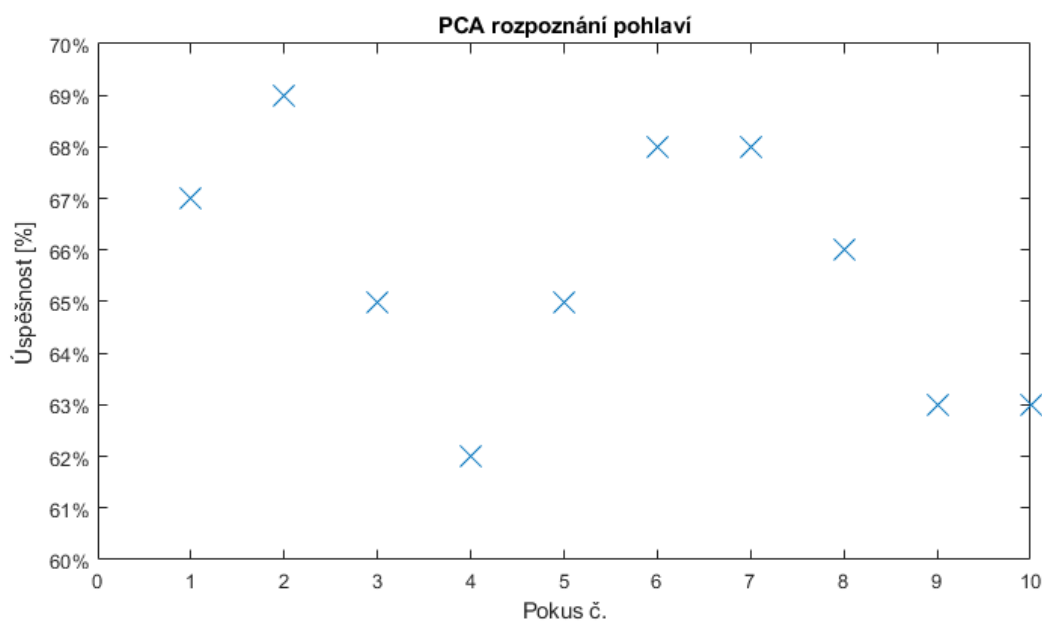
Graf 10.4: rozpoznání pohlaví pomocí SVM

Pro druhý test bylo vybráno 5 000 obrázků jako trénovací sada a testovací sada obsahovala 50 000 obrázků. Rychlost učení na trénovacích datech byla výrazně lepší a dosáhla i výrazně vyšší přesnosti. Po prvním průchodu dosáhla sice pouze 66 %, ale postupně se zvyšovala. Do 85 % se přesnost zvyšovala poměrně rychle a poté začala rychlost učení klesat, ale během testu nezačala kolísat a dosáhla až 94 % se stále stoupající tendencí. Přesnost na testovacích datech byla po deseti průchodech 71 %, po dvaceti průchodech 73 %. Několik dalších desítek průchodů se přesnost udržovala na 78 %, ale poté začala klesat a stagnovat mezi 73 % až 77 %.



Graf 10.5: rozpoznávání pohlaví pomocí SVM

Pro druhou část experimentu byla využita metoda PCA. Bylo provedeno 10 pokusů s náhodnými trénovacími a testovacími daty. Jako trénovací data bylo zvoleno 50 000 obrázků. Z důvodu časové náročnosti PCA byl pro testování použit menší počet testovacích dat. Proběhlo celkem 10 testů. Při těchto testech byla vyšší variabilita výsledných úspěšností a celkově nižší přesnost než u metody SVM. Úspěšnost se zde pohybovala mezi 62 až 69 % v závislosti na náhodně zvolených testovacích obrázcích.



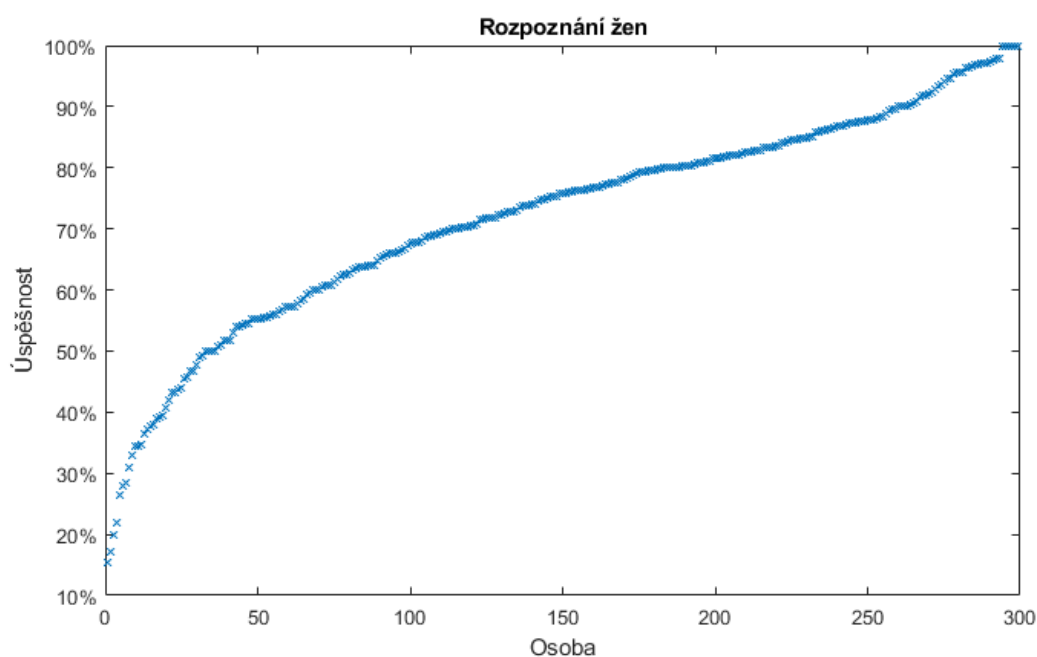
Graf 10.6: rozpoznávání pohlaví pomocí PCA

10.4 Rozpoznání osob

Pro rozpoznání osob byla použita knihovna `face_recognition` a její metoda `compare_faces`. Pro každou osobu byl zvolen jeden trénovací obrázek a různé množství testovacích obrázků, kvůli různému počtu obrázků u každé osoby. Tento test tedy testuje pozitivní případy a jako úspěch je započítán obličej u kterého je rozpoznáno, že patří stejné osobě jako v trénovacím obrázku.

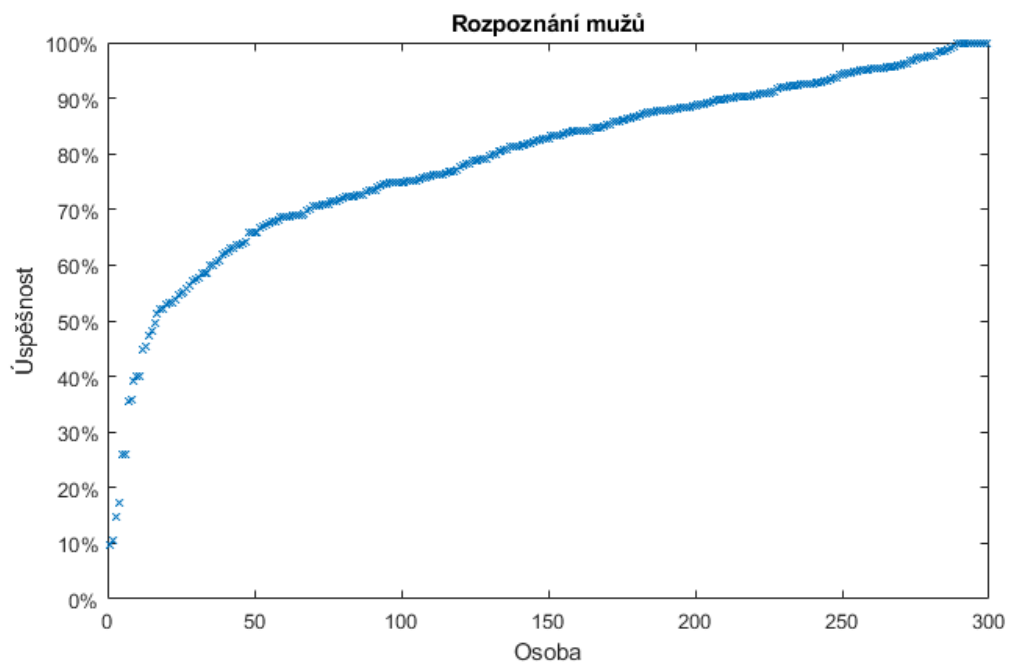
Výsledky experimentu se významně lišily pro každou osobu a pohybovaly se od 9,5 % až po 100 %. Průměrnou úspěšností pro muže i ženy bylo 75,5 %

Nižší celkové úspěšnosti rozpoznání bylo dosaženo při rozpoznávání žen. Při tomto experimentu byla celková přesnost 72 % s rozsahem od 15 % až do 100 %.



Graf 10.7: Rozpoznání žen metodou `compare_faces` knihovny `face_recognition`

Na datech s mužskými obličejemi bylo dosaženo významně vyšší úspěšnosti. Celková přesnost byla 79 % a to hlavně z toho důvodu, že se zde výrazně snížil počet osob s velmi nízkou úspěšností rozpoznání. A to i přes to, že nejnižší skóre při testování rozpoznání mužů bylo 9,5 %. Významně však oproti ženským datům narostl počet osob se skóre 100 % a stabilita výsledných přesností.

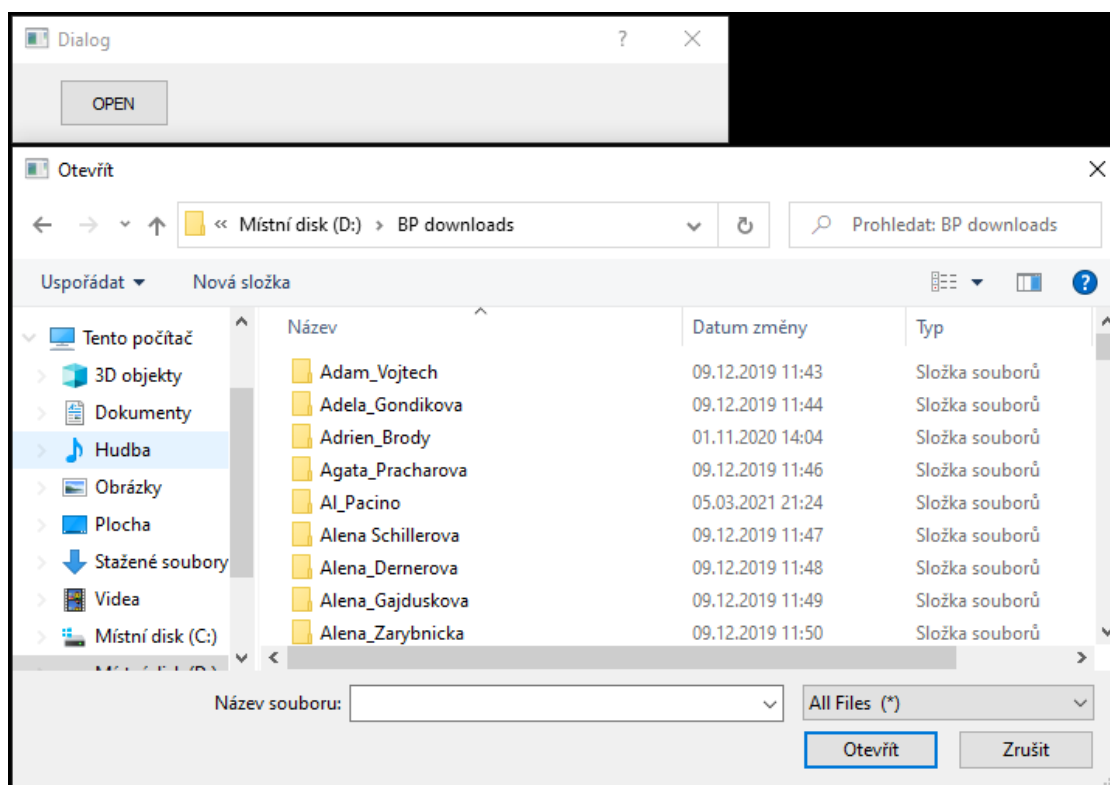


Graf 10.8: Rozpoznání mužů metodou `compare_faces` knihovny `face_recognition`

11 Systém pro detekci a identifikaci osoby

Finální aplikace pro nalezení a rozpoznání obličeje byla napsána v jazyce python a její grafické rozhraní bylo vytvořeno v programu *PyQT*. Tato aplikace dokáže ve vybraném obrázku či videu najít obličeje, rozpoznat věk, pohlaví a danou osobu, pokud se nachází v databázi.

Nejprve je v aplikaci nutné vybrat obrázek či video. Toho lze dosáhnout stisknutím tlačítka *OPEN*. Stisknutí tlačítka zobrazí výběr souboru, jak je možné vidět na obrázku číslo 14. Poté je vybrán obrázek či video, program určí, který z těchto typů souboru to je a podle toho se dále přizpůsobuje.



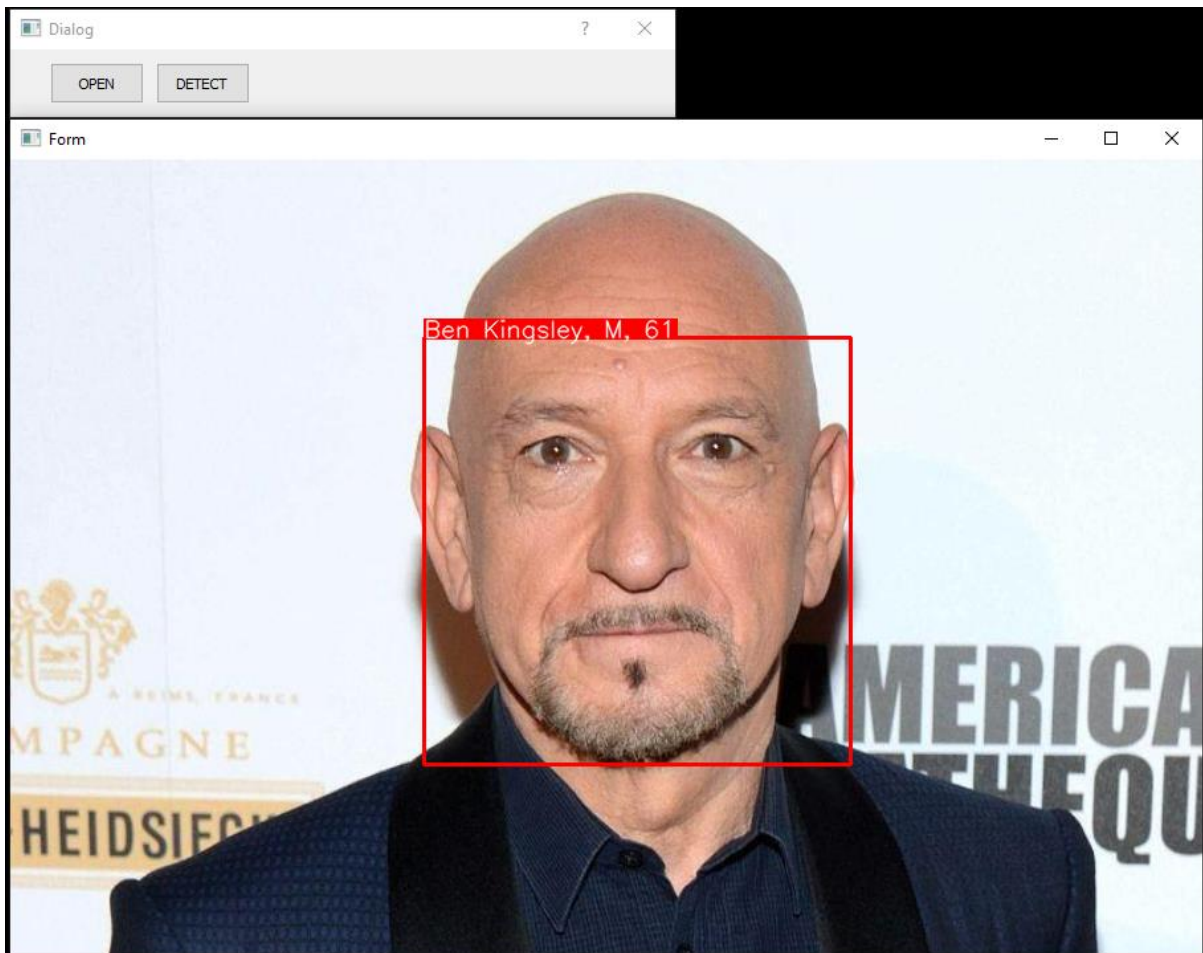
Obrázek 11.1: Ukázka výběru souboru v aplikaci

Pokud je zvolen obrázek, otevře se do nového okna. Po stisknutí tlačítka *DETECT* probíhá nejprve nalezení obličejů na obrázku. K tomu je využit detektor *frontal_face_detector* z knihovny *dlib*. Výstupem tohoto detektoru je seznam souřadnic všech obličejů nacházejících se na obrázku. Podle těchto souřadnic jsou potom vykresleny obdélníky, které obepínají obličeje.

Jako další se identifikuje odhadovaný věk a pohlaví nalezených obličejů. K tomu byl využit model *WideResNet* s šestnácti vrstvami, velikostí vstupních obrázků 64x64 pixelů a předtrénovaný vektor vah z modelu *age_gender_estimation*.

Poté je na řadě identifikace dané osoby, k tomu je využívána knihovna *face_recognition*. Nejdříve je nutné ze vstupního obrázku pomocí metody *face_encodings* získat dekodované části obrazů, které obsahují obličej. Dále si program z json souboru načítá dekodované obličej všech osob, které se nacházejí v databázi. Tyto dekodované obličej jsou poté porovnány v metodě *compare_faces*.

Tento postup je proveden pro všechny nalezené obličej v obrázku, které jsou potom označeny jménem osoby, pohlavím a odhadovaným věkem, jak je možné vidět na obrázku číslo 15.



Obrázek 11.2: Detekovaný a identifikovaný obličej

Pro video obsahuje grafické rozhraní navíc spuštění, pozastavení, posunutí času videa a také nastavení času mezi jednotlivými snímky. Pomocí knihovny *cv2*, její metody *VideoCapture* je procházeno video. Za využití metody *read* se z videa získávají jednotlivé snímky. Poté nastává proces detekce a identifikace obličej. Ten probíhá stejným způsobem jako u obrázku. Snímky videa se střídají po čase určeném výběrem v uživatelském rozhraní (v sekundách, případně desetinách sekund). Nalezené a identifikované obličej z videa jsou zobrazeny na obrázku číslo 16.



Obrázek 11.33: Detekované a rozpoznané obličeje ve videu

12 Závěr

V rámci bakalářské práce byl navržen systém pro detekování obličejů osob a jejich identifikaci. Toho bylo dosaženo za využití dříve otestovaných metod a algoritmů. Tato aplikace umí z obrázku či videa detekovat obličej, odhadnout věk a pohlaví a identifikovat osobu, pokud se nachází v databázi. Nejdůležitější pro tuto aplikaci bylo vybalancovat rychlost a přesnost detekce i identifikace obličeje. Proto tvorba aplikace vycházela z výsledků experimentů a následného ladění její funkčnosti, rychlosti a úspěšnosti.

V teoretické části jsou popsány základní principy pro zpracování a rozpoznání obrazu. Dále jsou zde popsány metody, algoritmy i modely, které byly v dalších částech práce testovány, či využity ve výsledné aplikaci. (SVM, PCA, kaskádový klasifikátor, histogram orientovaných gradientů i předtrénovaný model *age_gender_estimation*).

Jedním z dílčích cílů bylo vytvořit databázi 600 osob, obsahující fotografie obličejů daných osob. Pro variabilitu bylo důležité rovnoměrné rozdělení žen a mužů. Proto je v databázi zastoupeno 300 žen a 300 mužů. Databáze také obsahuje široké rozpětí věkových kategorií od 20 až po 80 a více let. Pro každou osobu bylo staženo 200 prvních obrázků z vyhledávání vyhledávače Google. Obrázky byly z části stahovány za využití knihovny *google_images_download*, která ale z důvodu změny *Google API* během zpracování bakalářské práce přestala fungovat. Proto bylo nutné pro stažení zbylých obrázků využít rozšíření do prohlížeče *fatkun batch downloader*. Z těchto obrázků byly poté za využití knihovny *dlib* oříznuty pouze obličejové a pro vyfiltrování nesprávných osob byla použita knihovna *face_recognition*, která podle vybraného obličeje určovala, zda ostatní obrázky obličejů patří stejné osobě.

Další částí práce bylo na této databázi otestovat algoritmy pro detekování a identifikaci osob. Pro detekci obličeje byl testován kaskádový klasifikátor a histogram orientovaných gradientů. Obě tyto metody detekují výrazné rysy obličeje, pro identifikaci umístění obličeje v obrazu. Při využití histogramu orientovaných gradientů bylo při experimentech dosaženo výrazně lepších výsledků a byl využit pro oříznutí obrázků při tvorbě databáze a také ve finální aplikaci.

Pro rozpoznání pohlaví a věku byly provedeny experimenty s metodami SVM a PCA. Při testování se v obou případech SVM prokázalo jako přesnější, spolehlivější a také významně rychlejší řešení. Za využití metody SVM bylo pro rozpoznání věku dosaženo úspěšnosti 30 % a pro rozpoznání pohlaví úspěšnosti až 79 %.

V oblasti rozpoznání osoby byla testována knihovna *face_recognition*, která je současně nejmodernější a nejlepší knihovnou v oblasti rozpoznávání osob. Tato knihovna dosahovala velmi dobrých výsledků, a proto byla využita jak při tvorbě databáze, tak i v aplikaci. Průměrnou úspěšností pro všechny osoby v databázi bylo 75,5 %, přičemž vyšší úspěšnosti bylo dosaženo u mužů a to 79 %.

- [1] WOLD, Svante, Kim ESBENSEN and Paul GELADI. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* [online]. 1987, **2**(1–3), 37–52 [accessed. 2020-04-21]. ISSN 01697439. Available at: doi:10.1016/0169-7439(87)80084-9
- [2] SADEGHI, Fereshteh. *Dimensionality Reduction*. no date.
- [3] SEHGAL, Shruti, Harpreet SINGH, Mohit AGARWAL, V. BHASKER and SHANTANU. Data analysis using principal component analysis. In: *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems, MedCom 2014* [online]. B.m.: Institute of Electrical and Electronics Engineers Inc., 2014, p. 45–48 [accessed. 2020-04-21]. ISBN 9781479950973. Available at: doi:10.1109/MedCom.2014.7005973
- [4] *Advanced Preprocessing: Variable Scaling - Eigenvector Research Documentation Wiki* [online]. [accessed. 2020-04-21]. Available at: http://wiki.eigenvector.com/index.php?title=Advanced_Preprocessing:_Variable_Scaling#Group_Scale
- [5] *A Complete Guide to Principal Component Analysis – PCA in Machine Learning – Data Science, Machine Learning & Artificial Intelligence* [online]. [accessed. 2020-04-23]. Available at: <https://ashutoshtripathi.com/2019/07/11/a-complete-guide-to-principal-component-analysis-pca-in-machine-learning/>
- [6] BEN-HUR, Asa and Jason WESTON. A user’s guide to support vector machines. *Methods in molecular biology (Clifton, N.J.)* [online]. 2010, **609**, 223–239 [accessed. 2020-04-24]. ISSN 19406029. Available at: doi:10.1007/978-1-60327-241-4_13
- [7] *Face Detection using Haar Cascades — OpenCV-Python Tutorials 1 documentation* [online]. [accessed. 2020-04-19]. Available at: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html
- [8] DÉNIZ, O., G. BUENO, J. SALIDO and F. DE LA TORRE. Face recognition using Histograms of Oriented Gradients. *Pattern Recognition Letters* [online]. 2011, **32**(12), 1598–1603 [accessed. 2020-04-19]. ISSN 01678655. Available at: doi:10.1016/j.patrec.2011.01.004
- [9] *Histogram of Oriented Gradients | Learn OpenCV* [online]. [accessed. 2020-04-19]. Available at: <https://www.learnopencv.com/histogram-of-oriented-gradients/>
- [10] ZHANG, Kaipeng, Zhanpeng ZHANG, Zhifeng LI and Yu QIAO. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal*

Processing Letters [online]. 2016, **23**(10), 1499–1503 [accessed. 2020-04-18]. ISSN 10709908. Available at: doi:10.1109/LSP.2016.2603342

[11] *A Gentle Introduction to Pooling Layers for Convolutional Neural Networks* [online]. [accessed. 2020-03-29]. Available at: <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>

[12] *#009 How to detect facial landmarks using DLIB and OpenCV* [online]. [accessed. 2021-03-05]. Available at: <http://datahacker.rs/009-how-to-detect-facial-landmarks-using-dlib-and-opencv/>