

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2017

Bc. Andrea Rozhoňová



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ**

DEPARTMENT OF BIOMEDICAL ENGINEERING

**DETEKCE A ROZPOZNÁVANÍ TVÁŘE S VYUŽITÍM  
PLATFORMY RASPBERRY PI**

FACE DETECTION AND RECOGNITION WITH USE OF RASPBERRY PI

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Andrea Rozhoňová**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Branislav Hesko**

**BRNO 2017**



# Bakalářská práce

bakalářský studijní obor **Biomedicínská technika a bioinformatika**

Ústav biomedicínského inženýrství

**Studentka:** Bc. Andrea Rozhoňová

**ID:** 174526

**Ročník:** 3

**Akademický rok:** 2016/17

## NÁZEV TÉMATU:

### Detekce a rozpoznávání tváře s využitím platformy Raspberry Pi

#### POKYNY PRO VYPRACOVÁNÍ:

1) Nastudujte problematiku detekce tváře v obraze. 2) Seznamte se a vytvořte přehled metod používaných k rozpoznávání lidské tváře. 3) Vhodně zvolte libovolnou metodu detekce tváře a implementujte ji v programovacím jazyku Python, s použitím modulu PiCamera. 4) Vyberte a implementujte klasifikátor pro rozpoznávání tváře. 5) Vytvořený program pro detekci a rozpoznávání tváře na platformě Raspberry Pi otestujte na vzorcích alespoň pěti osob a dosažené výsledky popište.

#### DOPORUČENÁ LITERATURA:

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

[2] GONG, Shaogang., Stephen J. MCKENNA a Alexandra. PSARROU. Dynamic vision: from images to face recognition. London: Imperial College Press, 2000. ISBN 1860941818.

**Termín zadání:** 6.2.2017

**Termín odevzdání:** 2.6.2017

**Vedoucí práce:** Ing. Branislav Hesko

**Konzultant:**

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

#### UPOZORNĚNÍ:

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

## **ABSTRAKT**

Následující bakalářská práce se věnuje problematice detekce a rozpoznání tváří v obraze. Teoretická část rozděluje metody detekce a rozpoznání obličejů do několika skupin, které jsou poté blíže popsány a vysvětleny. Na konci teoretické části je shrnuto současné využití rozpoznání osob na základě tváří v praxi. V praktické části je implementovaná metoda detekce obličejů jako kombinace přístupu, který využívá haarovy příznaky a přístupu s vyhledáváním pomocí šablony oka. Následné rozpoznání zajišťuje konvoluční neuronová síť. Závěrem jsou shrnuty zásady a problémy spojené s implementací na mikropočítač Raspberry Pi.

## **KLÍČOVÁ SLOVA**

obličej, předzpracování, obraz, metody, detekce, rozpoznání, příznaky, šablony, openCV, TensorFlow, Keras, Raspberry Pi, Pi Camera

## **ABSTRACT**

The following bachelor thesis is focused on the face detection and recognition in an image. The theoretical part divides methods of detection and recognition into several groups and there is better description and explanation of these methods in this part. At the end of the theoretical part is summarized the current utilization of person recognition on the bases of its face in practice. In the practical part is first implemented method for face detection. It is combination of two approaches - approach using haar features and approach using templates of eye. The face recognition is provided by the convolutional neural network. In conclusion there are summarized principles and problems associated with implementation on microcomputer Raspberry Pi and there is also evaluated the success of implemented methods.

## **KEYWORDS**

face, preprocessing, image, methods, detection, recognition, features, templates, OpenCV, TensorFlow, Keras Raspberry Pi, Pi Camera

ROZHOŇOVÁ, Andrea *Detekce a rozpoznání obličejů s využitím platformy Raspberry Pi*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2017. 56 s. Vedoucí práce byl Ing. Branislav Hesko

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Detekce a rozpoznání obličejů s využitím platformy Raspberry Pi“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu diplomové práce panu Ing. Branislavu Heskovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora(-ky)

# OBSAH

Úvod	10
<b>1 Teoretická část</b>	<b>11</b>
1.1 Obličej a jeho detekce	11
1.1.1 Vývoj v oblasti detekce obličeje	11
1.1.2 Obecný postup detekce	12
1.2 Metody detekce obličeje	12
1.2.1 Metody založené na znalostech	13
1.2.2 Metody založené na neměnných rysech	14
1.2.3 Metody založené na porovnávání šablon	18
1.2.4 Metody založené na vzhledu	20
1.3 Rozpoznání obličeje	24
1.3.1 Obecný postup rozpoznání	25
1.4 Metody rozpoznání obličeje	26
1.4.1 Geometrické metody	26
1.4.2 Holistické metody	29
1.5 Využití detekce a rozpoznání obličeje	35
<b>2 Praktická část</b>	<b>37</b>
2.1 Použitá zařízení a nástroje	37
2.1.1 Raspberry Pi	37
2.1.2 Modul Pi Camera	38
2.1.3 Python	39
2.2 Detekce tváře v obraze	40
2.2.1 Implementovaná metoda pro detekci tváře	40
2.3 Rozpoznání detekované tváře	43
2.3.1 Metoda využívající konvoluční neuronovou síť	43
2.4 Implementace na mikropočítač Raspberry Pi	46
2.5 Porovnání a zhodnocení metod	47
2.5.1 Detekce tváře	47
2.5.2 Rozpoznání tváře	49
<b>3 Závěr</b>	<b>52</b>
Literatura	53
Seznam symbolů, veličin a zkratk	56

# SEZNAM OBRÁZKŮ

1.1	Abstraktní model obličeje [13]. . . . .	13
1.2	Originální obraz a odpovídající obrazy se snižovaným rozlišením (stupeň šedi v každé buňce odpovídá průměru z pixelů, které jsou v ní obsaženy [30]. . . . .	14
1.3	Zobrazení hran u vzorové tváře [18] . . . . .	16
1.4	Vzory barev kůže[28]. . . . .	16
1.5	Pixely pleťové barvy v modelu YCbCr [12]. . . . .	17
1.6	Barevný model HSV [12]. . . . .	17
1.7	Příklad šablony pro detekci tváře, kde lze nalézt 16 oblastí a 23 vzájemných vztahů mezi nimi [30]. . . . .	19
1.8	Deformovatelná šablona pro detekci oka [31] . . . . .	20
1.9	Schéma nejčastějšího postupu detekce tváře [1]. . . . .	21
1.10	Ukázky některých využívaných obdélníkových haarových příznaků. A je příznak pro hrany, B představuje příznak středový, D pro linii a C ukazuje další možnou kombinaci příznaku [27]. . . . .	22
1.11	Integrální obraz - výpočet sumy libovolného obdélníku v obraze. . . . .	23
1.12	Výběr vhodných haarových příznaků [27]. . . . .	23
1.13	Princip funkce kaskádových klasifikátorů [27]. . . . .	24
1.14	Obecné schéma rozpoznání obličeje [1]. . . . .	25
1.15	Vertikální a horizontální projekce zobrazení hran (postupně) [3]. . . . .	26
1.16	35 porovnávaných obličejových znaků [9]. . . . .	28
1.17	Postup získání shlukového grafu z obličeje [29]. . . . .	28
1.18	Grafické znázornění hlavních komponent při použití PCA [15] . . . . .	30
1.19	Výskyt odpovídajících bodů [2] . . . . .	31
1.20	Schéma dvouvrstvého perceptronu [14]. . . . .	32
1.21	Proces konvoluce v konvoluční vrstvě [5]. . . . .	33
1.22	Proces sdružování ve sdružovací vrstvě [5]. . . . .	33
1.23	Architektura popisované konvoluční sítě [11]. . . . .	35
2.1	Mikropočítač Raspberry Pi 2 Model B . . . . .	37
2.2	Modul Pi Camera V2.1. . . . .	38
2.3	Ohraničení vyhledané tváře. . . . .	41
2.4	Hledání oblasti s okem pomocí šablon. . . . .	42
2.5	Tvorba trénovací databáze pro jednu osobu. . . . .	44
2.6	Postup klasifikace testovaného obličeje. . . . .	45
2.7	Původní obraz a výsledný detekovaný obličej po využití metody s haarovými příznaky. . . . .	47



2.8	Původní obraz a výsledný detekovaný obličej po využití metody srovnávání šablon. . . . .	47
2.9	Problematika detekce v obraze s více tvářemi. . . . .	48
2.10	Úspěšnost rozpoznání v různých pozicích tváře. . . . .	49
2.11	Úspěšnost rozpoznání s brýlemi a zavřenýma očima. . . . .	50
2.12	Úspěšnost rozpoznání tváře s úšklebky. . . . .	50
2.13	Úspěšnost rozpoznání částečně zakryté tváře. . . . .	51

## SEZNAM TABULEK

2.1	Procentuální úspěšnost v diskutovaných metodách detekce . . . . .	48
2.2	Procentuální úspěšnost rozpoznání v různých situacích . . . . .	49

# ÚVOD

Cílem bakalářské práce bylo implementovat metodu detekce a rozpoznání obličeje v obraze v programovacím jazyce Python za pomoci platformy Raspberry Pi.

Problematika detekce tváře spolu s jejím následným rozpoznáním je v popředí zájmu již od 70. let 20. století. Detekce je využívána zejména jako základní krok při rozpoznání obličeje, zatímco rozpoznání samotné lze využít v mnoha oblastech, od ověření totožnosti přes kontrolu přístupu k utajeným informacím až po herní průmysl. Největší výhody, které zajišťuje implementace na platformě Raspberry Pi, jsou malé rozměry mikropočítače a tudíž snadná manipulovatelnost s přístrojem a jeho relativně nízká cena.

Bakalářská práce je členěna na dvě části - část teoretickou a část praktickou. V teoretickém oddíle je nejdříve popsána detekce tváře, která je považována za stěžejní krok při jejím následném rozpoznání. Po základním uvedení do problematiky jsou zde metody detekce tváře rozděleny do čtyř hlavních skupin. Ty jsou dále blíže popsány a vysvětleny. Následuje podobný postup pro navazující rozpoznání detekované tváře. Metody rozpoznání jsou opět rozděleny, nyní do dvou hlavních skupin. I do nich řadíme několik odlišných postupů, které jsou rovněž stručně shrnuty a objasněny.

Struktura praktické části navazuje na strukturu v části teoretické. Nejprve je podrobně objasněn návrh metody detekce tváře, která je v této práci použita. Tato metoda kombinuje algoritmus navržený Paulem Violou a Michaellem Jonesem založený na využití haarových příznaků a vyhledávání tváře pomocí šablony oka. Stejně je vysvětleno i následující rozpoznání obličeje, které pro svou funkci využívá konvoluční neuronové sítě. Rozpoznání probíhá ve výřezu získaném po úspěšném detekování tváře.

Důležité je zpracování programu pro spuštění na mikropočítači Raspberry Pi. Obraz, ve kterém probíhá detekce a rozpoznání, je vytvořen pomocí připojeného modulu Pi Camera.

Nedílnou součástí práce je její zhodnocení, které vysvětluje, proč je použita daná metoda detekce a také popisuje okolnosti, které nejvíce ovlivňují úspěšnost rozpoznání osoby v testovaném obraze.

# 1 TEORETICKÁ ČÁST

## 1.1 Obličej a jeho detekce

Tváře hrají jednu z nejdůležitějších rolí v našem životě, a to zejména proto, že na nich nalezneme různé smyslové orgány zprostředkovávající nám kontakt s okolním světem. Nejvýznamnějším smyslovým orgánem je pro člověka oko, díky němuž může vnímat svět kolem. Naopak ke sdělování informací okolí používají lidé nejčastěji komunikování skrze řeč. Mnoho dalších podstatných informací jako například náladu vyjadřuje samotný vzhled tváře za pomoci pohybu a mimiky obličeje. [7]

Vyjma samotné komunikace nám obličej poskytuje i trvalejší vizuální znaky, díky kterým lze odhadnout mimo jiné pohlaví, věk, etnický původ a v neposlední řadě identitu, které se bude věnovat následující práce. Na tyto faktory bylo vhodné poukázat zejména proto, že samotná identifikace je úzce spjatá s dalšími funkcemi obličeje a s variabilitami v jeho vzhledu. [7]

Obličej je unikátní zejména v tom, že všechny nesou stejné běžné znaky, navíc vždy v podobném uspořádání. Rozpoznání osoby podle tváře je přesto velmi pozoruhodné zejména po uvážení, kolik různých lidí nesoucích tyto podobné znaky je nutné odlišit, a to navíc v neustále se měnících podmínkách okolního prostředí a osvětlení. [7]

### 1.1.1 Vývoj v oblasti detekce obličeje

Počátky výzkumů v oblasti detekce obličeje sahají do 70. let 20. století, kdy se nejvíce využívalo metod heuristických a antropometrických. Tyto techniky ovšem nepočítaly s mnoha proměnlivostmi, které detekci obličeje v různých situacích komplikují, a bylo možné je využít jen u přesně daných obrazů. I přes obrovský zájem o výzkum v této oblasti se však udržely v popředí až do let devadesátých. Největší rozmach nastal během posledních deseti let, kdy bylo vyřešeno mnoho dosud nerozluštitelných problémů a překlenuto velké množství překážek. Vědci a inženýři pracovali na tvorbě a vývoji strojů schopných automatického rozpoznání, začaly být navrhovány systémy, které uměly zohlednit různé odstíny pleti nebo pohyb obličeje. Pro detekci tváří ze scénérií se složitějším proměnlivým pozadím se začaly čteně využívat statistické metody a neuronové sítě. [8] Neuronové sítě spadají do oblasti umělé inteligence, jež můžeme definovat jako obor, do kterého lze zařadit velké množství disciplín počínaje počítačovými vědami, přes neurologii, psychologii, až po robotiku či lingvistiku. Její výrazný rozmach neodmyslitelně souvisí s aktuálním rozvojem právě v oboru počítačových technologií, které se detekcí tváří zabývají. Cílem umělé inteligence je do jisté míry duplikovat mozkovou aktivitu člověka spolu

s jeho chováním. [14] Počítačové vidění snažící se co nejpřesněji napodobit lidský zrak je jednou z jejích velmi důležitých aplikací. S tím souvisí i velmi dobře osvědčené metody umělé inteligence uplatňující se v problematice zpracování obrazu, detekce a následného rozpoznání tváře. [8]

### 1.1.2 Obecný postup detekce

Samotná lokalizace obličeje je základním kamenem, na kterém stojí úspěch celého následného zpracování. Tento krok ovšem není zdaleka tak jednoduchý, jak se může jevit, jelikož vzhled obrazu, ze kterého potřebujeme obličej detekovat, může být do značné míry variabilní. Snadněji se bude hledat obraz u testovací množiny obrazů s ideálně umístěným obličejem, neutrálním výrazem a jednobarevným podkladem. Změny, které oproti tomu ve většině případů nalezneme v reálném obraze, mohou být způsobeny vnějšími nebo vnitřními faktory. Odlišnosti ovlivněné vnitřními faktory jsou nezávislé na pozorovateli a vznikají čistě v důsledku fyzikální podstaty obličeje. Mezi faktory vnitřní spadá například věk, pohlaví, etnická příslušnost, tvar obličeje nebo jeho výraz a mimika. Na druhou stranu k faktorům vnějším řadíme například interakci světla s obličejem, dále úhel pohledu neboli pozici, ze které nahlížíme na hledaný obličej a osvětlení při tvorbě obrazu. Patří mezi ně i parametry zařízení, kterým obličej snímáme. Mezi ně řadíme například zaostření, rozlišení, šum a další. V neposlední řadě jsou vnějšími faktory i jiné objekty a obličeje způsobující například stíny ve tváři, její překrytí či další, nepřímé osvětlení. [7, 8, 12]

Jak vyplývá z předchozího textu, v každé vizuální scéně tedy nalezneme velké množství informací, které je nutné zpracovat. Základem detekce je proto upravit obraz do takové podoby, aby bylo možné najít oblast v zorném poli, ve které se bude hledaná tvář pravděpodobně nacházet. Konečným cílem detekce obličeje je poznatek, zda se v obraze nachází žádný, jeden, případně více obličejů. Pokud jsou opravdu v obraze nějaké tváře přítomny, je úkolem detekce vrátit jejich pozice a rozměry. [30]

## 1.2 Metody detekce obličeje

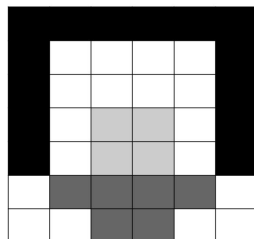
Na problematiku rozdělení algoritmů pro detekování obličeje hledí různí autoři odlišně. Většina metod se všelijak mísí a překrývá a tudíž neexistuje žádné oficiální rozřazení. Dobře přijaté bylo následující rozdělení metod do čtyř větších kategorií. I zde ovšem platí, že některé postupy se různě překrývají a mohou se řadit do více z nich. [30]

### 1.2.1 Metody založené na znalostech

Jedná se o metody, které jsou založeny na pravidlech a zachycují naše znalosti o obličejích. Využívá se zde tzv. typické tváře, která je vyhledávána v obraze. Tvář je v obraze lokalizována za pomoci typických vztahů mezi různými rysy a obličejem. Jako příklad můžeme uvést, že ve většině obrazů má obličej dvě oči, které jsou navzájem symetrické a v určité vzdálenosti od sebe. Okolí oka je zastíněné a tudíž tmavší než vystouplé tváře. Právě oči bývají při těchto metodách využívány k detekci asi nejčastěji. Důležitá je přesná lokalizace a kvalitní popis jednotlivých vztahů. Pokud jsou pravidla pro nalezení obličeje předdefinována příliš obecně, nastává časté zaměnění s jinými strukturami a naopak při příliš přísných pravidlech je možnost, že budou některé tváře přehlédnuty. Aby bylo dosaženo požadovaných výsledků, jsou tedy nutné poněkud složitější postupy. [30, 19]

#### Hierarchické metody

Vhodné je použití hierarchických metod, kdy se v systému nachází pravidla na třech úrovních [30]. Na nejnižší úrovni zpracování jsou nalezeny pravděpodobné oblasti



Obr. 1.1: Abstraktní model obličeje [13].

výskytu obličejů, které jsou dále zpracovávány na dalších úrovních. Mezi znaky nastavené pro vymezení těchto oblastí patří různé části obličeje. Stupeň šedi v každé buňce na obrázku 1.1 udává průměrnou hodnotu ze všech pixelů obsažených v dané buňce. Hlavní - střední část obličeje, odpovídá na obrázku velikosti  $4 \times 4$  buňky s označením "x". Světle šedá oblast uvnitř hlavní části obličeje blíže vymezuje část tváře s výraznými rysy, mezi které patří oči, nos nebo pusa. [13] Tmavší buňky v okolí středové části obličeje abstraktního modelu dále ukazují oblast možného výskytu vousů, černé buňky představují vlasovou linii. Velmi důležité jsou vzájemné rozdíly mezi stupni šedi v jednotlivých oblastech. Takové rozložení odstínů šedi je následně vyhledáváno v předzpracovaném obraze, příklad vidíme na obrázku 1.2.

Takto vymezené oblasti pravděpodobného výskytu obličeje dále podstupují druhou úroveň detekce, kdy je prováděna lokální ekvalizace histogramu následovaná



Obr. 1.2: Originální obraz a odpovídající obrazy se snižovaným rozlišením (stupeň šedi v každé buňce odpovídá průměru z pixelů, které jsou v ní obsaženy [30].

detekcí hran. Při detekci hran musí být hrany správně označeny a přizpůsobeny modelu obličeje, aby bylo možné správnou detekci ověřit. Jednou z metod nalezení obličeje je označení hrany jako levé strany obličeje, vlasové linie nebo pravé strany tváře. Následně porovnáme oblast vymezenou hranami s modelem zlatého poměru pro ideální tvář:

$$\frac{\text{výška}}{\text{šířka}} \equiv \frac{1 + \sqrt{5}}{2}. \quad (1.1)$$

Výsledkem druhé úrovně je tedy nalezení přibližných obrysů možného obličeje v oblastí vymezené první úrovní. [8]

Na úrovni tři jsou následně u dosud nalezených možných tváří hledány v odpovídající oblasti výrazné obličejové struktury jako jsou oči, obočí, nos či ústa. [30, 13]

Základní krok při lokalizaci očí a obočí je nalezení řádku nejjasnějších buněk v horní části oblasti vymezené pro výrazné obličejové struktury. Nalezením lokálních minim ve vyšetřovaném řádku jsou následně detekovány oči. Pokud se v řádku nachází více lokálních maxim, je důležité zvolit maxima s dostatečnou vzdáleností od sebe a bez signifikatního rozdílu v jejich stupních šedi. [13]

Detekce nosu a nosních dírek se provádí ve sloupcích mezi očima a pod nimi. Vyhledávají se opět lokální extrémy, které se shodují ve svislém i vodorovném směru. Buňky, které disponují nalezenými lokálními minimy jsou označeny jako oblasti výskytu nozder a buňky s lokálními maximy označují špičku nosu, kam dopadá nejvíce světla. [13]

Jako poslední jsou k završení a potvrzení správnosti detekce lokalizována ústa jako homogenní oblast 2 nebo 3 buněk nacházejících se několik řad pod nosem. [13]

### 1.2.2 Metody založené na neměnných rysech

Postupy podobné předchozím metodám založené na neměnných rysech, mají za úkol najít takové obličejové rysy a charakteristiky, které zůstávají neměnné i v případě, že je obličej například jinak natočený, podléhá různým světelným podmínkám či

jiným vlivům okolí. Na základě tohoto předpokladu bylo navrženo mnoho metod, které nejprve detekují obličejové rysy a z nich následně odvozují přítomnost obličeje. Mezi zmiňované rysy můžeme zařadit například barvu obličeje, obličejové struktury jako oči, obočí a nos či texturu obličeje. Pro jejich detekci je nutno ověřit, zda platí vzájemné popsané geometrické vztahy mezi pravděpodobnými místy výskytu, čímž přítomnosti obličeje potvrdíme nebo vyvrátíme. Strukturami, na kterých bývá tato metoda nejčastěji založena, jsou oči. Velkou výhodou je při použití metod založených na neměnných rysech jejich poměrně snadná implementace při dosažení přijatelných výsledků. Postupů, které vycházejí z neměnných rysů, je mnoho. V následujících podkapitolách jsou shrnuty některé z nich. [30]

### Obličejové znaky

Metody pro nalezení výrazných obličejových znaků bývají často založeny na antropometrických měřeních. Velmi jednoduchým příkladem založeným na antropometrickém měření je systém pro detekci obličeje, kde prvním krokem je lokalizace možného umístění očí v předzpracovaném binarizovaném obraze. Požadovaný obraz získáme po převedení původního obrazu do stupňů šedi a zobrazení histogramu. Podle prominentního píku v histogramu je určena prahová hodnota a obraz je převeden na černobílý. Pro každou detekovanou potenciální dvojici očí z takového obrazu jsou v požadované konfiguraci dále vyhledávány významné obličejové rysy jako nos, ústa a obočí. Každý vyhledávaný rys má přiřazenou svou váženou vyhodnocovací funkci podle důležitosti 1.2, která je následně využita k určení nejpravděpodobnějších oblastí výskytu obličeje. [8]

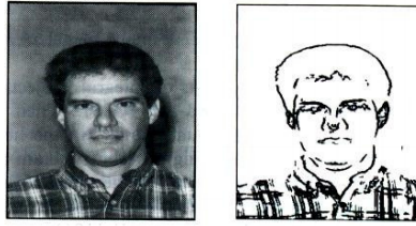
$$E = 0.5E_{oko} + 0.2E_{ústa} + 0.1E_{obočí} + 0.1E_{nos} \quad (1.2)$$

Algoritmus další metody začíná vyhledáním hypotetického vrcholu hlavy. Předzpracovaný obraz je postupně procházen směrem dolů a vyhledává se oční rovina charakteristická prudkým nárůstem množství hran. Zobrazené hrany můžeme vidět na 1.3. Jako referenční délka se využívá vzdálenost mezi vrcholem obličeje a linií očí. Tato oblast je naopak charakterizována nepřítomností hran. Za pomoci zjištěné referenční délky lze v předpokládaných oblastech tváře vyhledávat jednotlivé obličejové rysy pomocí předdefinovaných vzorů. [18]

### Barva pleti

Mnoho studií dokazuje, že rozdíly mezi barvami pleti jsou do velké míry způsobeny spíše rozdílnou intenzitou barev než odlišnostmi v odstínu, proto se v metodách pro detekci obličeje barva lidské kůže ukázala jako účinný prvek. [30]





Obr. 1.3: Zobrazení hran u vzorové tváře [18]

Zatímco vstupní obraz je ve většině případů ve formátu RGB (Red Green Blue), techniky založené na barvě pleti spíše využívají barevné prostory HSV (Hue Saturation Value) či YCbCr (Luminence, Chromatic Blue, Chromatic Red). To zejména proto, že obraz ve formátu RGB podléhá vlivům světelných podmínek. [12]

Pro vytvoření modelu barvy pleti bylo navrženo mnoho metod. Jedna z nejjednodušších metod je založená na zmiňovaném formátu YCbCr. Hodnota jasu je obsažena ve složce  $Y$  a barevnost ve složkách  $Cr$  a  $Cb$ , přičemž  $Cr$  značí stupeň červené v obraze a  $Cb$  stupeň modré. Jednotlivé složky z formátu RGB jsou konvertovány do tohoto formátu následovně: [12]

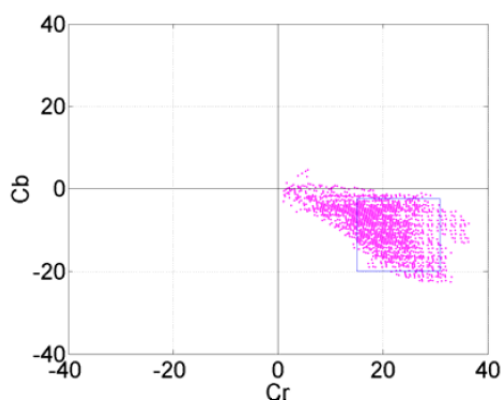
$$\begin{aligned}
 Y &= 0.299R + 0.587G + 0.114B \\
 Cb &= -0.169R - 0.332G + 0.500B \\
 Cr &= 0.500R + 0.419G - 0.081B.
 \end{aligned}
 \tag{1.3}$$

Při použití této metody je každý pixel označený jako pixel pletové barvy či naopak pixel barvy nepletové na základě hodnot složek  $Cr$  a  $Cb$ . Pixely jsou klasifikovány na základě nastavených prahových hodnot. Prahové hodnoty nastavíme



Obr. 1.4: Vzory barev kůže[28].

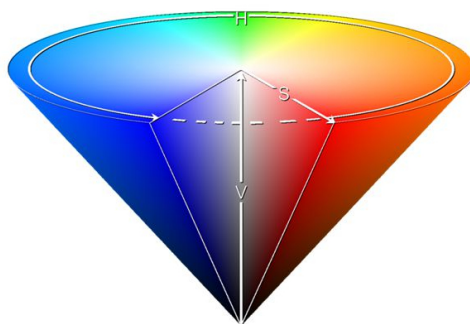
porovnáním různých testovacích vzorů barev kůže, které můžeme vidět na obrázku 1.4. [12, 28]



Obr. 1.5: Pixely pletové barvy v modelu YCbCr [12].

Na obrázku 1.5 můžeme vidět jaké je rozložení pixelů pletové barvy v barevném prostoru formátu YCbCr [12].

Jiným postupem pro detekci oblasti s barvou pleti může být metoda iterační identifikace pokožky. Tato metoda využívá průsečíku v histogramu barevného prostoru HSV [12]. Tento barevný model, zobrazený na obrázku 1.6, se skládá ze tří složek - H = odstín, s = sytost a V = jas. Iniciační pole s pixely barvy pleti je



Obr. 1.6: Barevný model HSV [12].

voleno uživatelem. Pro detekci požadovaných oblastí se po obraze se posouvá maska a v každé oblasti se porovnává iniciační pole s aktuálním. Pro iniciační i aktuální pole je zobrazen histogram barevných odstínů a pokud je hodnota na ose y v oblasti průsečíku větší než stanovená prahová hodnota, aktuální pole se stává novou oblastí,

která nejlíže odpovídá barvě lidské pleti a podle které jsou hledány tváře ve zbytku obrazu. [30]

## **Textura**

Mimo výše zmiňované metody mohou být použity i algoritmy založené na texturní analýze. Ty vycházejí z předpokladu, že se textura obličeje odlišuje od jiných objektů v obraze. Textura obličeje je detekována na základě jeho tří významných znaků: vlasů, tváří a zbytku. Pro uskupení jednotlivých textur do tříd je vhodné použít neuronovou síť. Následné zhodnocení přítomnosti obličeje je doporučeno na základě přítomnosti textury pleti a vlasů. K rozřídění obrazu podle textur a následnému nalezení obličeje může výrazně pomoci začlenění barevné informace do algoritmu. [30]

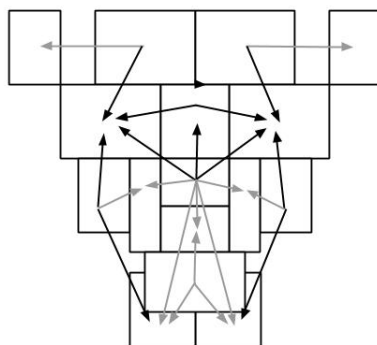
Většina používaných metod založených na invariantních rysech samozřejmě v současnosti vychází při detekci obličeje z kombinování výše zmiňovaných postupů a bere ohled současně na výrazné obličejové struktury, barvu pleti i texturu obličeje. [30]

### **1.2.3 Metody založené na porovnávání šablon**

Metody využívají předdefinovaných standardních vzorů buďto celého obličeje, nebo zvlášt některého z obličejových rysů. Následně je za účelem detekce počítána korelace mezi vstupním obrazem a uloženými šablonami. Právě nutnost tvorby a předdefinování jednotlivých šablon dělá tyto metody zdoluhavými a zároveň složitými. Nevýhodou jsou také nedostatečné výsledky, pokud dojde v obraze k změně pozice, změně tvaru nebo překryvu obličeje. V poslední době byly nicméně navrženy i deformovatelné šablony pro ošetření těchto úskalí. [30, 28, 19]

#### **Předdefinované šablony**

Jedním z prvních algoritmů založených na předdefinovaných šablonách je detekování tváře z předního pohledu za použití několika „podšablon“. Aplikování podšablon slouží k detekci samotných očí, nosu, úst a obrysu obličeje a jsou navrženy pomocí množství úseček, které udávají vztahy mezi strukturami. Příklad je uveden na obrázku 1.19. Podšablony jsou následně srovnávány se vstupním obrazem a jsou počítány vzájemné korelace. Pokud dojde k objevení hledaných obličejových znaků, jsou následně vzájemně spojovány. Při nalezení obličejových znaků ve správném geometrickém rozložení je detekován obličej. Některé metody jsou založeny na opačném postupu, kdy nejprve nacházíme podle šablony tvar obličeje a následně jsou v této detekované oblasti vyhledávány výrazné obličejové rysy podle podšablon jako oči, nos nebo ústa. Porovnávání s šablonami bývá nejčastěji založeno na detekci hran. [30]



Obr. 1.7: Příklad šablony pro detekci tváře, kde lze nalézt 16 oblastí a 23 vzájemných vztahů mezi nimi [30].

Jindy jsou jako templaty pro lokalizaci obličeje použity různé siluety. Sadu základních obličejových siluet můžeme získat zredukováním dimenze dat pomocí PCA (Principal Component Analysis) neboli analýzy hlavních komponent na příkladech obličejů, kde jsou siluety zastoupeny bitovým polem. Získaná sada siluet je následně využívána k lokalizaci tváří pomocí zobecněné Houghovy transformace, což je transformace sloužící k určení pozic libovolných tvarů v obraze. [8]

Další postup založený na předdefinovaných šablonách vychází z předpokladu, že zatímco například osvětlení různých struktur v obličeji se může lišit, relativní jas těchto struktur zůstává po většinu času stejný. Na známých tvářích jsou pozorovány pravidelnosti ve změně jasu u jednotlivých výrazných obličejových struktur a jejich okolí. Po ustálení těchto pravidel jsou v obraze vyhledávány tváře na základě oblastí vykazujících stejné rozdíly ve stupních jasu. [30]

### Deformovatelné šablony

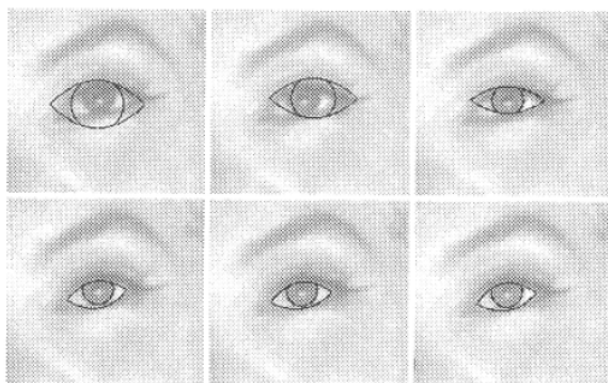
Jelikož předdefinované šablony jsou vhodné pouze pro nalezení obličeje z předního pohledu, začínají se v poslední době velmi často objevovat i šablony deformovatelné. Vznikají postupným přizpůsobováním šablon předdefinovaných podle obličejů v obraze a snaží se odstranit jejich úskalí. Deformovatelná šablona je definovaná její parametrickou přímkou a plochou, jejichž funkcí je modelovat hlavní charakteristiky tváře. Nejčastěji se šablona postupně posouvá po obraze a její tvar se přizpůsobuje pomocí zmiňovaných parametrů. Dokud měnící se hodnoty parametrů nepřekročí určitou mez, je oblast považována za oblast s nalezeným obličejem. Při překročení stanovené meze už je oblast moc odlišná na to, aby se jednalo o tvář. [31]

Často využívaná je například šablona oka, kterou můžeme vidět na 1.8. Je určena několika parametry a jakmile se dostane do blízkosti hledaného znaku - oka, začne

se pomalu deformovat podle jeho optimální hranice. Vývoje šablony oka je dosaženo minimalizací funkce energie:  $E$ .

$$E_{celková} = E_{vnitřní} + E_{vnější}, \quad (1.4)$$

kde  $E_{vnitřní}$  je vnitřní energetická funkce, která definuje přirozený vývoj šablony - typicky je to její rozšiřování a smršťování. Naopak  $E_{vnější}$  působí proti vnitřní energii a umožňuje odchýlení šablony od přirozeného tvaru po působení vnějších podnětů. [31]



Obr. 1.8: Deformovatelná šablona pro detekci oka [31]

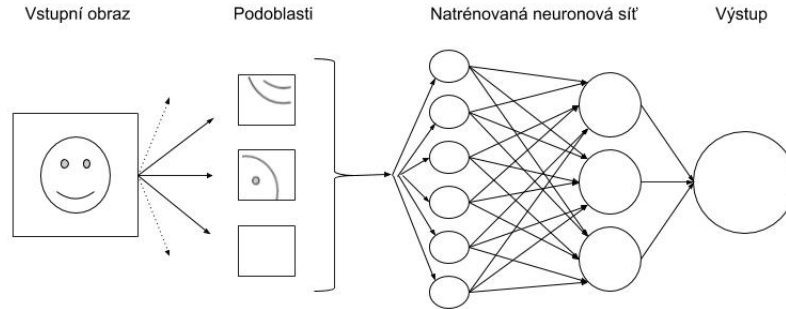
### 1.2.4 Metody založené na vzhledu

Do poslední, čtvrté skupiny, spadají metody založené na vzhledu. Základním rozdílem mezi těmito metodami a metodami zmiňovanými výše, které porovnávají obrazy s šablonami, je trénovací množina. V ní nalezneme velké množství vzorů tváří reprezentujících variabilitu obličejů v různých situacích. Zjednodušeně lze říci, že pomocí trénovací množiny učíme systém jak takový obličej vypadá.

#### Minimalizace diskriminační funkce

Tyto metody řadí se většinou do oblasti umělé inteligence do velké míry odstraňují nevýhody v podobě nepředvídatelnosti vzhledu obličeje a podmínek okolí, kterým je potřeba čelit v případě porovnávání s šablonami. Přístupy mohou být založeny na minimalizaci tzv. diskriminační funkce. Tu lze definovat jako práh mezi třídou, kde je zařazen obličej a třídou, kde je zařazena oblast bez něj. Návrh diskriminační funkce pro klasifikaci je velmi obtížný a běžně jej získáváme za pomoci mnohovrstvých neuronových sítí. [1] Nutná je tvorba a definování modelu obličeje a dále

vhodné nastavení procesu učení. Základem trénovací množiny pro učení sítí je velké množství dat - oblasti obličejové i bez obličeje. Porovnáním těchto oblastí s histogramem vstupního vzorku lze následně nalézt místa pravděpodobného výskytu obličeje. Pomocí popisovaných postupů je možné dosáhnout při správném nastavení velmi kvalitních výsledků. [30, 28, 19, 8]



Obr. 1.9: Schéma nejčastějšího postupu detekce tváře [1].

Vstupní obraz bývá nejčastěji rozdělen na množství následně analyzovaných podoblastí. K tomu je možné využít ověřenou Analýzu hlavních komponent. Každá podoblast je dále rozčleněna na části, které jsou posílány do zvoleného typu natrénované neuronové sítě. Výstupem naučené sítě je detekovaný obličej pomocí rozpoznávaných obličejových oblastí. Na obrázku 1.9 můžeme vidět nastíněný postup. [1]

## Metody pravděpodobnostní

Jinou možností je realizace těchto metod jako metod pravděpodobnostních. Ty nevyužívají umělé inteligence, každý obraz je definovaný pomocí náhodné proměnné  $x$ . Za pomoci pravděpodobnostní funkce je následně možné definovat proměnou  $x$  pro část obrazu, kde se obličej nachází  $p(x/face)$  a pro část, kde se nenachází  $p(x/nonface)$ . Pro klasifikaci, zda se v dané části obrazu obličej nachází či ne je používán Bayesův vzorec nebo maximální pravděpodobnost.

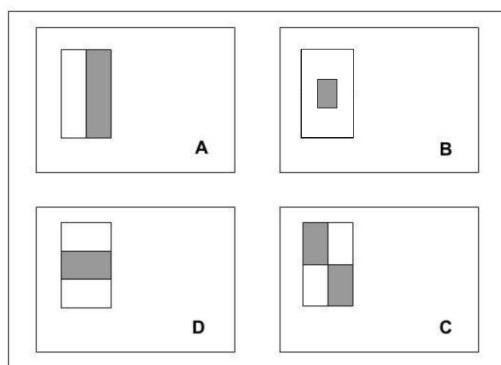
$$\frac{P(x|face)}{P(x|nonface)} > \frac{P(face)}{P(nonface)} \quad (1.5)$$

Pokud hodnota poměru pravděpodobností na levé straně rovnice 1.5 je vyšší než hodnota poměru na straně pravé, je na aktuálním místě přítomna tvář. Pokud jsou hodnoty v rovnici přesné, je použití Bayesova vzorce považováno za ideální. Nicméně proměnná  $x$  je mnohorozměrná a  $p(x/face)$  a  $p(x/nonface)$  jsou multimodální, tudíž je ve většině případů nemožné Bayesův vzorec implementovat jednoduchým přepisem. Studie, zabývající se popsaným postupem, jsou proto převážně prováděny

na základě empiricky ověřených parametrů pravděpodobnostních funkcí  $p(x/face)$  na  $p(x/nonface)$ . [8]

### Metody využívající haarovy příznaky

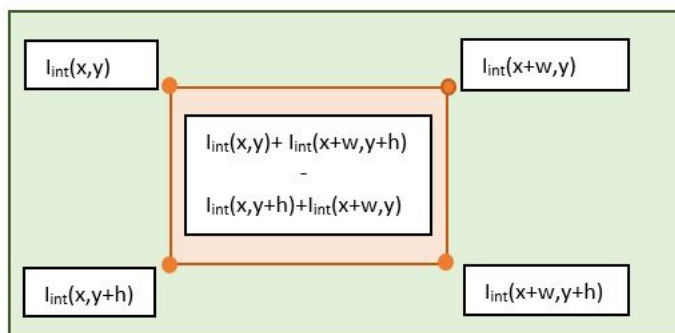
Další metodou, která je založená ne vzhledu obličeje, je detekce pomocí haarových příznaků. Celý přístup založený na strojovém učení, kdy je klasifikátor trénovaný mnoha snímky s detekovanými tvářemi i bez nich navrhli v roce 2001 ve své studii „Rapid Object Detection using a Boosted Cascade of Simple Features” Paul Viola a Michael Jones. Jejich studie je zaměřená obecněji na detekci objektů v obraze, je možné jí ovšem jednoduše využít pro požadované hledání tváří. Metoda využívá



Obr. 1.10: Ukázky některých využívaných obdélníkových haarových příznaků. A je příznak pro hrany, B představuje příznak středový, D pro linii a C ukazuje další možnou kombinaci příznaku [27].

rozřazení na základě kaskádových klasifikátorů. Ty jsou postaveny na reprezentaci obličejů tzv. haarovými příznaky zobrazenými na 1.10. Haarovy příznaky se odlišují podle informace, kterou v obraze mají detekovat. Můžeme rozlišovat příznaky reprezentující hrany, linie, případně příznaky středové. Pro důkladnější reprezentaci obrazu existuje podobných haarových příznaků a jejich dalších různých variací několik desítek tisíc. Haarovy příznaky jsou postupně horizontálně i vertikálně posouvány po vstupním obraze. Odezva aktuálně zkoumané oblasti vstupního obrazu na haarovy příznaky je počítána následovně. Od součtu pixelů v bílých obdélnících jsou odečítány součty pixelů v obdélnících šedých. Takto jsou pomocí haarových příznaků a jejich vhodného rozložení v obraze jsou vyhledávány významné, pro tvář charakteristické, struktury. Pro účely detekce bývá obraz rozdělený na  $24 \times 24$  oblastí, přimčež bývá použita sada 180 000 haarových příznaků. Nastává tedy problém

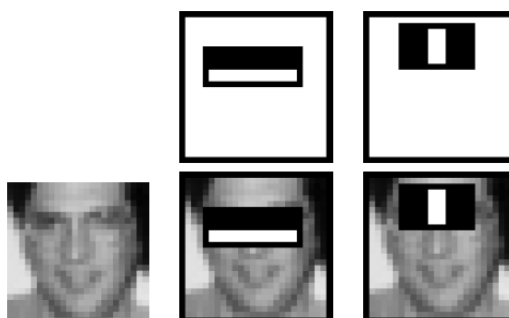
obrovské výpočetní náročnosti, kterou představuje vypočítávání odezvy příznaků pro celý obraz. Ten je do značné míry zjednodušený a urychlený zavedením integrálních obrazů [20]. Princip jejich výpočtu je znázorněn na 1.11.



Obr. 1.11: Integrální obraz - výpočet sumy libovolného obdélníku v obraze.

Základem metody je velké množství vstupních dat pro učení klasifikátoru, jehož úkolem je následná detekce obličeje. Důležité je do vstupních dat zahrnout dostatečné množství obrazů, na kterých se obličej nachází, i dostatečné množství obrazů bez obličejů.

Protože je definováno 180 000 příznaků, bylo by při samotné detekci nadbytečné a výpočetně náročné vyhledávat ve všech oblastech vložených obrazů každý zavedený příznak. Na obrázku 1.12 můžeme vidět vhodně umístěné haarovy příznaky v obličeji. Využití stejných příznaků například na tváři by ovšem bylo zbytečné. Při učení



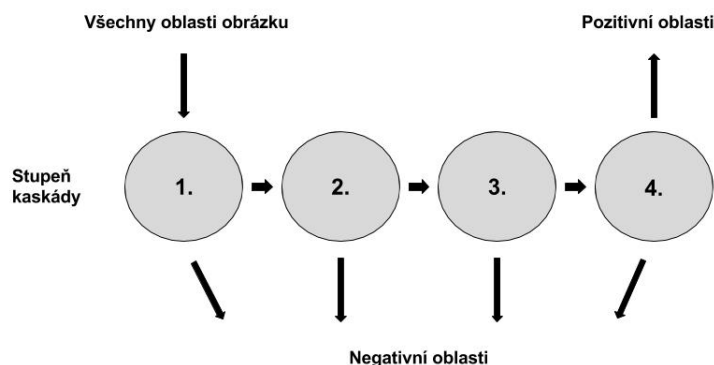
Obr. 1.12: Výběr vhodných haarových příznaků [27].

proto aplikujeme nejprve všechny příznaky na trénovací obrázky. Pro každý příznak je tak v dané oblasti nalezen práh, který určuje, zda se zde vyhledávaná struktura nachází nebo ne. Jelikož samozřejmě dochází k různým chybám a špatnému zařazení, vybírají se příznaky, u kterých je na konci procesu zjištěna chybovost nejmenší.



Tímto se každému příznaku přiřadí váha, která určuje jejich použití při detekování tváře v testovaném obraze. Pomocí popsaného procesu se příznaky zredukuje ze 180 000 na přibližně 6 000 [20].

Pro další redukcí výpočetní náročnosti je velmi důležité také zavedení kaskádových klasifikátorů, díky kterým není nutné každé okno v obraze procházet všemi 6 000 příznaky. Na všechny oblasti je nejprve aplikováno jen několik základních příznaků, které mohou přítomnost obličeje vyloučit. Pokud přítomnost vyloučena není, jsou aplikovány další příznaky. V případě, že jsou v okně vyhledávány postupně všechny příznaky a ani v jednom kroku nedojde k vyloučení přítomnosti obličeje, jedná se pravděpodobně o oblast, kde se nachází vyhledávaná obličejová struktura [20].



Obr. 1.13: Princip funkce kaskádových klasifikátorů [27].

### 1.3 Rozpoznání obličeje

Rozpoznání a odlišení obličejů je považováno za pravděpodobně jednu z nejpozoruhodnějších schopností zprostředkovaných lidskými smysly. Ještě přibližně před dvaceti lety bylo rozpoznání tváře považováno za jednu z nejsložitějších úloh, které lze zajistit pomocí počítačového vidění v oblasti umělé inteligence, což samozřejmě do jisté míry souvisí i se slabším výkonem tehdejších počítačů. V posledních letech byla však vyvinuta řada metod a přístupů, které tento úkol do jisté míry zvládají. Dnes dokonce existují společnosti prodávající velmi spolehlivé programy pro rozpoznání společně i s databázemi fotografií lidí. [22]

Pro rozpoznání a identifikaci je v současnosti využíváno velké množství technologií, zejména díky komerčnímu využití. Velkému zájmu se v poslední době těší zejména biometrická identifikace, která vychází z identifikace na základě jedinečné

fyziologické charakteristiky každého člověka. Při biometrickém rozpoznání obličeje je ovšem nutné, aby byl vyšetřovaný jedinec v přesném postavení a vzdálenosti oproti snímacímu zařízení a na určitou dobu zůstal nehybný. Biometrie je proto například hojně využívána při ochraně vysoce zabezpečených aplikací, nicméně pro většinu jiných využití se začínají rozvíjet nové metody rozpoznání tváře. [22]

### 1.3.1 Obecný postup rozpoznání

Obecně lze rozpoznání obličeje rozdělit do následujících kroků. Základem je videozáznam či statický snímek, ve kterém chceme tvář rozpoznávat. Dalším krokem ověření přítomnosti tváře a její následná detekce, které je věnována první část bakalářské práce. Z detekovaného obličeje jsou následně extrahovány důležité znaky potřebné pro následné porovnání. V mnoha případech, je extrakce znaků používána současně k detekci tváře a tyto dva kroky jsou proto prováděny současně. [19] Extrakce



Obr. 1.14: Obecné schéma rozpoznání obličeje [1].

znaků bývá často prováděna pomocí integrální projekce. Obraz, z kterého extrahujeme znaky označíme jako  $O(x,y)$ . Vertikální integrální projekce je poté definována jako: [3]

$$V(x) = \sum_{y=y_1}^{y_2} O(x, y) \quad (1.6)$$

a horizontální jako:

$$H(y) = \sum_{x=x_1}^{x_2} O(x, y) \quad (1.7)$$

Tyto projekce mohou být velmi nápomocné zejména při určování pozice obličejových znaků. Na obrázku 1.15 můžeme například vidět, že oči a ústa jsou dobře patrné při projekci horizontální, nos při projekci vertikální.

Předzpracovaná a detekovaná tvář je nakonec porovnávána a vyhledávána pomocí některé z metod, které jsou popsány níže. [19]



Obr. 1.15: Vertikální a horizontální projekce zobrazení hran (postupně) [3].

## 1.4 Metody rozpoznání obličeje

Z obecnějšího hlediska lze dnes metody pro rozpoznání obličeje rozdělit do dvou širších skupin. Do první skupiny řadíme přístupy založené na porovnávání s co největší sadou šablon, obrazů či různých modelů. Právě sada šablon tvořená pomocí různých nástrojů je pro tyto metody velmi důležitá. [19, 9] Do druhé skupiny poté řadíme metody geometrické neboli metody založené na výrazných obličejových znacích, které mají v každé tváři unikátní geometrii neboli vzájemné postavení. Stejně jako některé metody detekce vycházejí z výrazných struktur obličeje jako jsou oči, obočí, ústa a nos. Jejich úkolem je tyto struktury analyzovat a mapovat vztahy mezi nimi. V dnešní době je tato skupina metod využívána o něco méně než skupina zmiňovaná dříve, stále však často dochází ke kombinaci obou těchto přístupů. [19]

### 1.4.1 Geometrické metody

Jak již bylo zmíněno, tyto metody vycházejí z jedinečné geometrie a vzájemného postavení významných struktur v každé tváři. Celkové rozložení výrazných znaků v obličejí lze popsat číselným vektorem, který nese informace o jejich pozici a velikosti. Pro přesnější klasifikaci je vhodné tyto informace doplnit o tvar popisovaného obličeje při frontálním pohledu. Pro spolehlivost metody je ovšem důležité dodržet několik základních podmínek jako je co nejjednodušší odhad, malá závislost na drobných změnách ve výrazu tváře a na světelných podmínkách a zároveň co nejvyšší informační hodnota. [3]

Důležitým a často velmi záludným krokem při využití geometrických metod je normalizace neboli zajištění, že extrahované znaky budou nezávislé na pozici, měřítku či rotaci obličeje. Základem je stanovit počáteční souřadnice detekovaného

obličejů. Následně je nastavena vzdálenost mezi očima spolu se směrem osy, která obě oči spojuje. Pomocí zjištění tohoto směru lze eliminovat závislost rotace. [3]

Velkou výhodou geometrických metod může být jejich tolerance ke změnám ve vstupním obraze, nicméně jejich největší klad se nachází v podobě možnosti velmi rychlého porovnání a zařazení vstupního obličejů. Naopak velkou nevýhodu přináší obtížné rozhodování, které znaky jsou významné a také jejich následná automatická detekce. [9]

### **Metoda porovnávací obličejové vektory**

Na geometrických metodách byly založeny nejstarší metody využívané k rozpoznání tváře. Jednou z prvních byla metoda, kterou vyvinul Takeo Kanade v roce 1973. V ní se po zpracování obrazu získal z obličejů vektor šestnácti obličejových parametrů. Mezi nimi byly zahrnuty poměry vzdáleností obličejových znaků, jejich umístění v obličejích a úhly, které tyto znaky navzájem svírají. Pro srovnání bylo použito počítání prosté Euklidovské vzdálenosti. Metoda dosahovala úspěšnosti až 75 % při využití databáze s dvaceti různými jedinci, kdy vždy jeden obraz osoby byl použit jako referenční a jeden jako testovací. Tato metoda byla v následujících letech zdokonalována přidáváním měřených parametrů a zvětšováním databází, čímž docházelo zároveň ke zvyšování její úspěšnosti. [9, 17]

Přesnost můžeme zvýšit metody založené na deformovatelných šablonách nebo na Houghově transformaci a dalších. Všechny tyto metody jsou však do velké míry závislé na heuristice, proto musí být nastavena při porovnávání poměrně velká míra tolerance. Ta však může nepříznivě ovlivnit přesnost experimentu. [17, 9]

Pravděpodobně nejvyšší úspěšnosti 95 % bylo dosaženo při využití databáze se 685 obrazy (každý jeden pro jednoho jedince) a znakového vektoru odvozeného z třiceti pěti obličejových znaků, znázorněných na obrázku 1.16. Velké úskalí této metody nicméně spočívá v manuální extrakci jednotlivých porovnávaných znaků [9].

### **Metoda shlukových grafů**

Poměrně často využívané bývá také porovnávání shlukových grafů založené na Gaborové vlnkové transformaci. Výhodou přístupu je zejména rozlišení tváře i při změnách podmínek okolí nebo samotného obrazu, což je zajištěno právě využitím vlnek. V této metodě jsou obličejové struktury reprezentovány grafy, kde uzly odpovídají pozicím významných struktur v tváři a hrany tyto uzly propojují. Každý uzel charakterizuje sada Gaborových vlnkových koeficientů o různých orientacích a frekvencích, které jsou běžně označovány jako „jety“. Popisují vlastnosti významných obličejových struktur, jež jsou dále porovnávány [17]. Jeta znázorňuje malou část šedé hodnoty v obraze  $I(\mathbf{x})$  okolo pixelu  $\mathbf{x} = (x, y)$  a je získána konvolucí vstupního

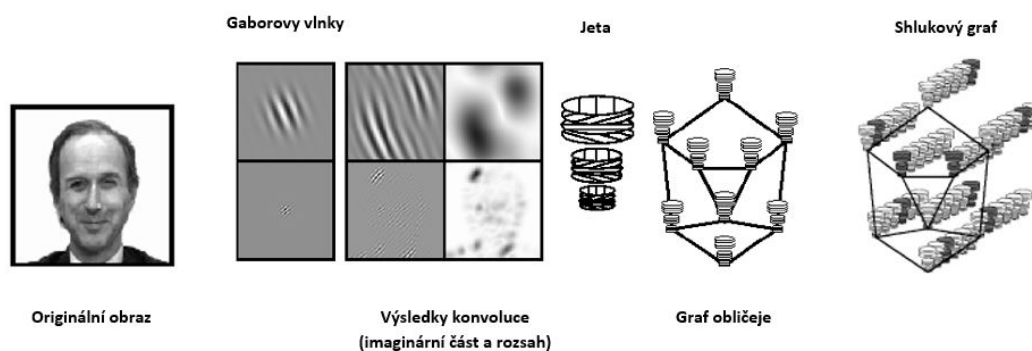


Obr. 1.16: 35 porovnávaných obličejových znakov [9].

obrazu s Gaborovými vlnkami  $\Psi_j$  reprezentujícími různé pozice a polohy obličeje v obraze: [29]

$$J_j = \int I(\mathbf{x})\Psi_j(\mathbf{x} - \mathbf{x}')d^2\mathbf{x}' \quad (1.8)$$

Hrany spojující tyto uzly oproti tomu udávají vzájemné postavení a vzdálenost jednotlivých struktur neboli celkovou geometrii obličeje. Z těchto uzlů a hran jsou dále zobrazovány obrazové grafy [17]. Shlukový graf je tvořen obrazovými grafy po-



Obr. 1.17: Postup získání shlukového grafu z obličeje [29].

rovnávané osoby v různých pozicích a světelných podmínkách. Na obrázku 1.17 je znázorněn proces jeho získání od konvoluce s Gaborovými vlnkami po tvorbu samotného shlukového grafu. Pomocí porovnání shlukových grafů může být obličej přiřazen odpovídající osobě. [19]

## 1.4.2 Holistické metody

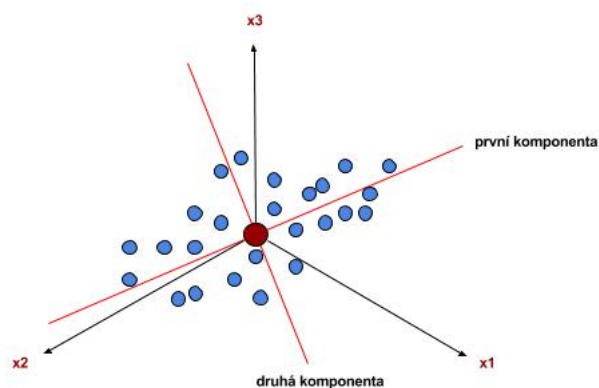
Jak již bylo zmíněno, holistické metody jsou založeny na porovnávání obličeje jako celku bod po bodu. Jejich pravděpodobně největší výhodou je, že je porovnáván celý obličej a není tudíž opomíjena žádná část, která by mohla neočekávaně pomoci správné klasifikaci. Tato jejich výhoda je však zároveň i jejich největším problémem. Nelze porovnávat obrazy pixel po pixelu se stejnou výpočetní rychlostí, jako když jsou porovnávány pouze některé jejich části. V současnosti je však i toto úskalí do velké míry ošetřeno a některé metody jsou zdokonaleny do velmi spolehlivé a rychlé podoby. [9]

### Metody statistické

Ve statistických metodách je každý obraz znázorněn  $d$  příznaky a je na něj tedy nazíráno jako na datový bod (případně vektor) v  $d$ -rozměrném prostoru. Množství informací, které by bylo potřebné k tomu, aby bylo možné určit souřadnice a vlastnosti tohoto bodu, je obrovské. Právě statistické nástroje lze dobře využít k potřebné extrakci a následné analýze požadovaných příznaků. Nástroje musí vhodně definovat obličejový prostor v obraze a extrahovat z něj důležité základní funkce. Po zpracování je možné identifikovat linie, ohyby, roviny a oddělit tváře spadající do jiných tříd. [19]

Jednou z nejpoužívanějších statistických metod je **Analýza hlavních komponent - PCA**, která slouží k redukci rozměrů mnohorozměrných dat. Za tímto účelem využívá, jak vyplývá z jejího názvu, extrahování požadovaného množství hlavních komponent, což jsou lineární kombinace původních proměnných. [17] První hlavní komponenta je obvykle lineární kombinace z originálních rozměrů s nejvyšší odchylkou. Druhá komponenta je kolmá na první a popisuje rozptyl, který nezahrnuje komponenta první. Třetí je kolmá na první a druhou a popisuje rozptyl, který není zahrnutý ani v první ani v druhé komponentě.  $N$ -tá hlavní komponenta je s maximální odchylkou kolmá na  $n-1$  komponentu. Hlavní komponenty jsou seřazeny podle důležitosti, největší rozptyl a tedy variabilita v obraze vždy zatěžuje první hlavní komponentu. [15] Výsledkem analýzy hlavních komponent je řada, ve které jsou komponenty seřazeny se snižujícím se rozptylem. Pro snížení dimenzí můžeme některé komponenty s nízkými rozptyly, které jsou na rozdíl od vstupních komponent nekorelované, odebrat. Na základě rozdílů rozptylů u ponechaných komponent lze poté rozpoznat a identifikovat tváře v obraze. Nikdy nicméně nemáme jistotu, že právě nejvýznamnější komponenty přispějí ke správné klasifikaci do tříd. [15]

Statistických metod existuje velké množství, mezi další patří například **Diskrétní kosinová transformace** nebo **Goborova vlnková transformace**. [19]



Obr. 1.18: Grafické znázornění hlavních komponent při použití PCA [15] .

### Metody založené na porovnávání s šablonami

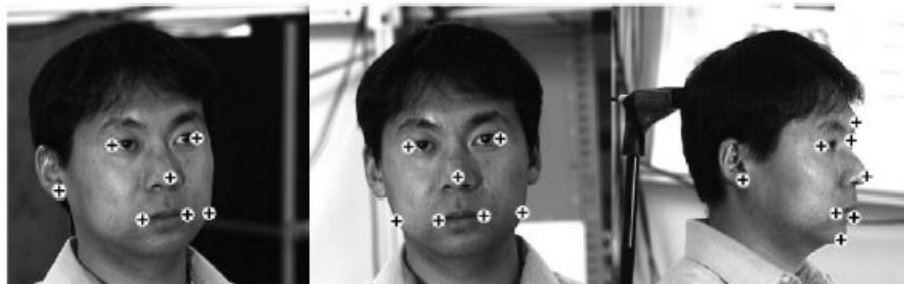
Technik, které využívají porovnávání vstupního obrazu s určitou šablonou, je mnoho. Tyto přístupy využívají jako srovnávací znaky jednotlivé pixely, komplexnější vzorky, celé modely nebo texturu oblastí v obraze. Stejně jako při detekci obličejů je u metod založených na porovnání šablon prováděn výpočet korelace, případně jsou počítány odchylky (vzdálenosti) vstupního obrazu od definovaných vzorů. Velmi hojně se dnes využívá 3D šablon, jelikož nejsou do velké míry citlivé na změnu pozice tváře a stejně tak na změnu osvětlení. Problém ovšem je, že u subjektu, který by měl být rozpoznán, je v tomto případě nutná spolupráce, což je nemyslitelné například u sledovacích systémů. Proto jsou při použití rozpoznání tváře často využity techniky, které převádí 2D data na 3D model. [19]

Princip některých metod je stejný jako jejich využití při detekování tváře. Jako příklad jiné, pokročilejší metody můžeme uvést metodu založenou na proměnlivém modelu 3D obličejů. Vychází z vektorového prostoru, který reprezentuje tváře. Různé kombinace vektorů  $S_i$  a  $T_i$ , znázorňující tvar a strukturu obličejů (postupně) popisují reálnou lidskou tvář.

$$\begin{aligned}
 S &= \sum_{i=1}^m a_i S_i, \\
 T &= \sum_{i=1}^m b_i T_i.
 \end{aligned}
 \tag{1.9}$$

U modelu dochází ke kontinuálním změnám ve tvaru buď pomocí změny parametru  $a_i$ , který ovlivňuje tvar šablony nebo pomocí změn parametru  $b_i$ , který ovlivňuje změny v textuře. Podstatou přístupu je, aby se v proměnlivé šabloně vyskytovaly

zadané důležité body označující výrazné struktury jako například špička nosu nebo koutky úst na obrázku 1.19. Určení vektorů tvaru a struktury vychází ze stanove-



Obr. 1.19: Výskyt odpovídajících bodů [2]

ného referenčního obličejce. Vektor tvaru  $S_0$  je určen kartézskou soustavou souřadnic, zatímco vektor struktury  $T_0$  je definován v RGB barevném prostoru.

$$\begin{aligned} S_0 &= (x_1, y_1, z_1, x_2, \dots, x_n, y_n, z_n)^T, \\ T_0 &= (R_1, G_1, B_1, R_2, \dots, R_n, G_n, B_n)^T. \end{aligned} \quad (1.10)$$

Ve většině případů následuje analýza hlavních komponent zmiňovaná již v části věnované detekci obličejce a v části popisující statistické metody. Tato analýza je aplikována zvláště na sady vektorů tvaru a zvláště na vektory struktury, ze všech vzorových obličejů. Po případném snížení přebytečného množství informace v testovaném obraze jsou za pomoci sad vektorů hledány požadované body a tváře jsou přiřazovány odpovídajícím osobám. [19, 2]

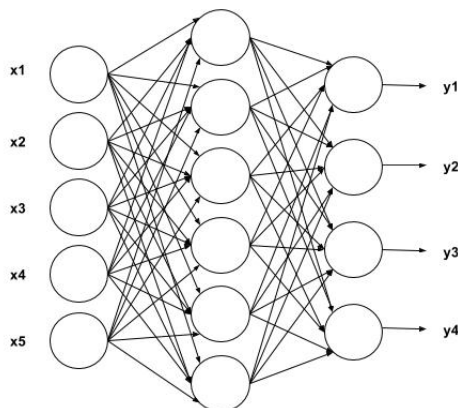
### Metody založené na využití neuronových sítí

Metody, které k rozpoznání tváře využívají strojového učení a umělých neuronových sítí jsou dnes asi nejpobulárnější. Jako první využil umělé neuronové sítě pro rozpoznání tváří ve své práci Kohonen, nicméně od té doby začaly být využívány stále hojněji. Do velké míry jsou tyto metody kombinovány s metodami statistickými.

V jedné z nejjednodušších sítí bylo 50 hlavních komponent za použití Analýzy hlavních komponent redukováno do 5 rozměrů, přičemž byl použitý obyčejný **mnohvrstevný perceptron**. V takovém případě je síť složena z jednoduchých perceptronů, které jsou propojeny podle určitých pravidel. Ze všech neuronů v předchozí vrstvě vede s určenou váhou spojení ke všem neuronům ve vrstvě následující. Takovou neuronovou síť vždy tvoří stála vstupní a výstupní vrstva a jedna či více



vrstev skrytých s různým počtem uzlů. Přes tuto síť prochází signál postupně a přímo, na obrázku 1.20 zleva doprava. Popsané uspořádání se nazývá dopředná síť - síť bez zpětných vazeb. [14] Rovněž testovací a trénovací databáze byly jednodu-



Obr. 1.20: Schéma dvouvrstvého perceptronu [14].

ché, obsahovaly manuálně zarovnané obrazy 20 lidí, všechny se stejným osvětlením, naklopením a rotací. Proto i výsledná klasifikace byla velmi spolehlivá. [9]

U vícevrstvých sítí přesnost a spolehlivost velmi ovlivňuje proces jejich učení. V dnešní době se velmi často využívá tzv. **metoda zpětného šíření chyby**. Cílem této metody je upravovat váhy jednotlivých vstupů tak, aby co nejvíce minimalizovaly chybovou funkci  $\varepsilon$ .

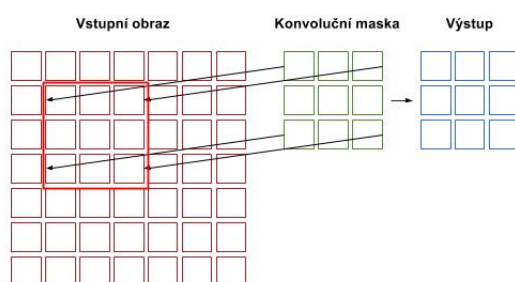
$$\varepsilon = \sum_{j=1}^M (d_j - y_j)^2 \quad (1.11)$$

Hodnota  $d_j$  značí  $j$ -tý požadovaný výstupní vektor odpovídající učebnímu vstupnímu vektoru a hodnoty  $y_j$  představují  $j$ -tý reálný výstup sítě.  $M$  udává počet výstupů. [14] Po každém průchodu sítí je počítána odchylka reálného výstupu od požadovaného, která se zpětně přepočítává do předchozích vrstev a poté se díky ní upraví hodnoty vah. Další průchod sítí již probíhá s upravenými váhami. [26] Proces učení probíhá v tzv. epochách, kdy jsou síti ukázány všechny dvojice z učební posloupnosti  $(\vec{x}, \vec{d})$ . [14]

Zdokonalením mnohvrstevných perceptronů jsou **konvoluční neuronové sítě**. Ty by měly o něco lépe řešit problémy v rozdílném natočení, zkreslení či posunutí vstupních obličejů. Využívají skutečnosti, že na vstupu jsou obrazy a jejich neurony jsou proto uspořádány do tří vrstev - výška, šířka a hloubka. Příkladem může být obrazy v barevném prostoru RGB s velikostí  $(32 \times 32 \times 3)$ . Konvoluční síť jsou

dopředné sítě schopné bez předchozího předzpracování extrahovat topologické vlastnosti ze vstupního obrazu. Extrakce příznaků je u těchto sítí tedy součástí jejího trénování. Konvoluční neuronové sítě proto s velkou úspěšností rozpoznávají i tváře s obrovskou variabilitou. [16]

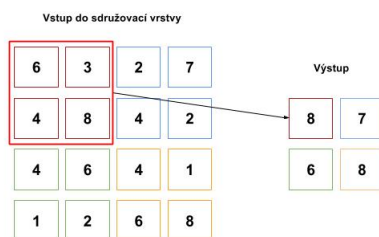
Pro sestavování konvolučních sítí jsou použity tři typy vrstev - konvoluční, sdružovací a plně propojená vrstva. Konvoluční vrstva se skládá z množiny malých filtrů. Ty postupně posouváme po obraze a v každé oblasti provádíme konvoluci s ohraničenou oblastí vstupu. Postup je znázorněný na obrázku 1.21. Filtrů bývá ve většině



Obr. 1.21: Proces konvoluce v konvoluční vrstvě [5].

případů větší množství. Všechny hodnoty u jednoho výstupu konvoluce mají sdílené váhy, což je hlavní rozdíl oproti mnohovorstevnému perceptronu. [5]

Po konvoluční vrstvě následuje vrstva sdružovací, která z jednotlivých bloků výstupů vytvoří vždy jednu hodnotu. Jako výstupní hodnota bývá nejčastěji brána maximální hodnota z bloku výstupu. Proces je znázorněný na 1.22. Tato vrstva



Obr. 1.22: Proces sdružování ve sdružovací vrstvě [5].

výrazně snižuje množství výpočtů v síti.

Dostatečně zredukovaný obsah přichází následně do plně propojené vrstvy, kde jsou všechny neurony propojeny s předchozí vrstvou. Dochází zde k nastavení aktivity. Tato vrstva má tedy v konvolučních sítích úlohu klasifikátoru. [5]

Následuje zpětné šíření chyby. Zde se oproti mnohvrstevným perceptronům zavádí výpočet chybové funkce, kterou se snažíme učením minimalizovat. Využívá se optimalizačního algoritmu zvaného gradient sestupu. Výpočet chybové funkce v předešlých vrstvách je vzhledem k objektivě vhodné provést pomocí parciálních derivací tzv. řetězovým pravidlem. Předpokladem je, že známe chybovou funkci  $E$  na výstupu a počítáme chybu v předchozí vrstvě -  $(\frac{\delta E}{\delta y_{ij}^l})$ . Prvním krokem je zjistit, jaké jsou složky gradientu u jednotlivých vah. [4]

$$\left(\frac{\delta E}{\delta w_{ab}}\right) = \sum_{i=1}^{N-m} \sum_{j=0}^{N-m} \left(\frac{\delta E}{\delta x_{ij}^l}\right) \left(\frac{\delta x_{ij}^l}{\delta w_{ab}}\right) \quad (1.12)$$

Hodnoty  $x_{ij}^l$  jsou hodnoty vstupů,  $w_{ab}$  jsou hodnoty vah. Díky sdíleným vahám musíme sečíst všechny výrazy  $x_{ij}^l$ , ve kterých se vyskytuje  $w_{ab}$ . Pro výpočet gradientu je nutné znát hodnoty  $(\frac{\delta E}{\delta x_{ij}^l})$ , nazývané delta. I jejich hodnoty lze snadno vypočítat použitím řetězového pravidla. [4]

$$\left(\frac{\delta E}{\delta x_{ij}^l}\right) = \left(\frac{\delta E}{\delta y_{ij}^l}\right) \left(\frac{\delta y_{ij}^l}{\delta x_{ij}^l}\right) = \left(\frac{\delta E}{\delta y_{ij}^l}\right) \sigma'(x_{ij}^l) \quad (1.13)$$

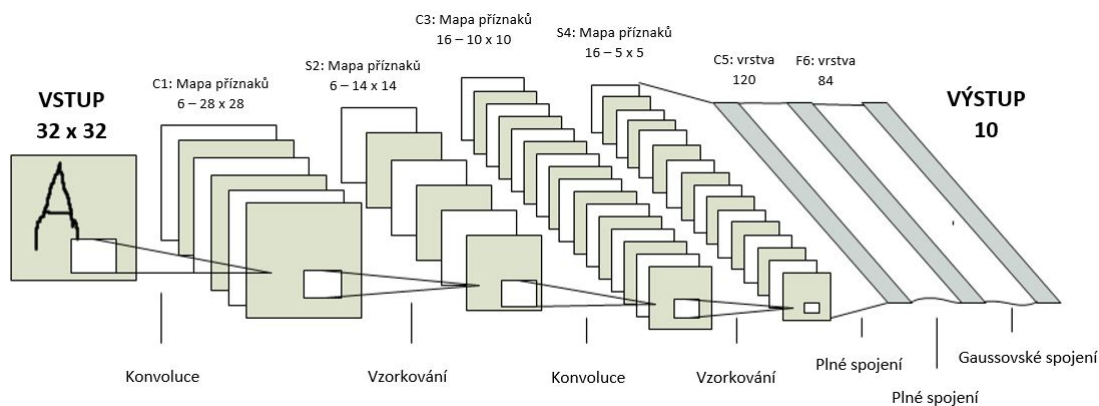
Hodnoty  $y_{ij}^l$  značí ve vzorci hodnoty výstupu a  $\sigma'(x_{ij}^l)$  použitou aktivační funkci. Vztah ukazuje, že deltu lze snadno spočítat derivací aktivační funkce  $\sigma'(x)$ . S ohledem na váhy v konvoluční vrstvě lze snadno spočítat gradient, pokud už známe hodnotu chyby v současné vrstvě. Ten je tvořen všemi parciálními derivacemi dohromady. Síť se zdokonaluje pokud dochází k pohybu proti směru gradientu, jelikož se snižuje hodnota chybové funkce. [4]

Pro šíření chyby dál do předchozí vrstvy je nutné opět využít řetězového pravidla.

$$\left(\frac{\delta E}{\delta y_{ab}^{l-1}}\right) = \sum_{a=1}^{m-1} \sum_{b=0}^{m-1} \left(\frac{\delta E}{\delta x_{(i-a)(j-b)}^l}\right) \left(\frac{\delta x_{(i-a)(j-b)}^l}{\delta y_{ij}^{l-1}}\right) = \sum_{a=1}^{m-1} \sum_{b=0}^{m-1} \left(\frac{\delta E}{\delta x_{(i-a)(j-b)}^l}\right) w_{ab} \quad (1.14)$$

Z tohoto vztahu si můžeme všimnout, že jde o jednoduchou konvoluci s použitím  $w$  převrácené kolem obou os. [4]

Příklad architektury celé konvoluční neuronové sítě je znázorněn na obrázku 1.23. Do sítě se sedmi vrstvami vstupuje nezpracovaný obraz o velikosti  $32 \times 32$  pixelů. Síť je složena ze 3 konvolučních vrstev, 2 sdružovacích, 1 plně propojené a 1 výstupní vrstvy. [11]



Obr. 1.23: Architektura popisované konvoluční sítě [11].

## 1.5 Využití detekce a rozpoznání obličeje

Detekce a rozpoznání tváří v obraze je v dnešní době velmi studovanou problematikou, která nachází uplatnění v řadě odvětví. V praxi se využívá nejčastěji dvojitým způsobem. Prvním způsobem je ověření, zda se jedná o stejnou osobu. Při druhém způsobu porovnáme získanou tvář neznámé osoby s obličejem v určité databázi. Dále jsou uvedeny některé konkrétnější případy využití. [9]

- Obecné ověření totožnosti - biometrie

Biometrie je metoda, která slouží k rozpoznání jedinečných biologických vlastností každé osoby. Systémy pro rozpoznání tváře se na základě biometrických metod někdy mohou využívat pro obecné ověření totožnosti například při volební registraci, v bankovníctví, k identifikaci novorozenců, u občanských průkazů a pasů nebo k identifikaci zaměstnanců ve firmách. [21]

- Bezpečnost

Jelikož v dnešní době je potřeba dbát na bezpečnost jednou z nejdůležitějších, byly například na mnoha letištích po celém světě do letištních ochranných systémů implementovány technologie rozpoznání obličeje při letištních kontrolách. Příkladem může být mezinárodní letiště v Kalifornii Fresno Yosemite, kde technologie zavedená roku 2001 upozorní zaměstnance bezpečnostní složky vždy, když bezpečnostní kontrolou prochází osoba podobající se vzhledem někomu z databáze lidí podezřelých z terorismu. [21]

- Databáze osob pro vyšetřování

Databáze fotografií se mohou využívat pro identifikaci při různých typech vyšetřování. V databázích lze vyhledávat i licencované řidiče, pohřešované děti,

příjemce dávek, přistěhovalce a další. [21]

- Dohled

Využití za účelem dohledu je podobné jako u hlídání bezpečnosti. Technologie pro rozpoznání obličeje nainstalovány na veřejných místech, kde slouží ke snižování trestné činnosti. Například v londýnské čtvrti Newham po nainstalování 300 kamer podle městské rady pokleslo množství spáchaných trestných činů až o 34 %. [21]

- Kontrola přístupu

V mnoha situacích je nutno zajistit, aby k počítačům či určitým aplikacím měly přístup jen oprávněné osoby. V praxi bývá přístup kontrolován například přístupem do samotné kanceláře nebo přihlášením do počítače. Technologie rozpoznání obličeje může také fungovat na principu nepřetržitého sledování, kdo se vyskytuje před hlídaným zařízením. V případě, že uživatel odejde bez odhlášení a zavření skrývaných souborů, uzamkne systém po stanovené době klávesnici a myš a spustí spořič obrazovky. Po návratu oprávněného uživatele systém opět rozpozná jeho přítomnost a jedinec může pokračovat tam, kde skončil. Jakékoli jiné osobě bez oprávnění je přístup naopak odepřen a se zařízením pracovat nemůže. [21]

- Zábava

Nelze samozřejmě zapomenout na odvětví, které detekci obličeje dnes rovněž využívá v hojné míře, a tím je zábava. Existuje řada webových či mobilních aplikací spojených s web kamerou nebo fotoaparátem, které pracují s vyhledáváním tváří. Stejně tak detekci využívají domácí systémy ke hraní video her. [19]

## 2 PRAKTICKÁ ČÁST

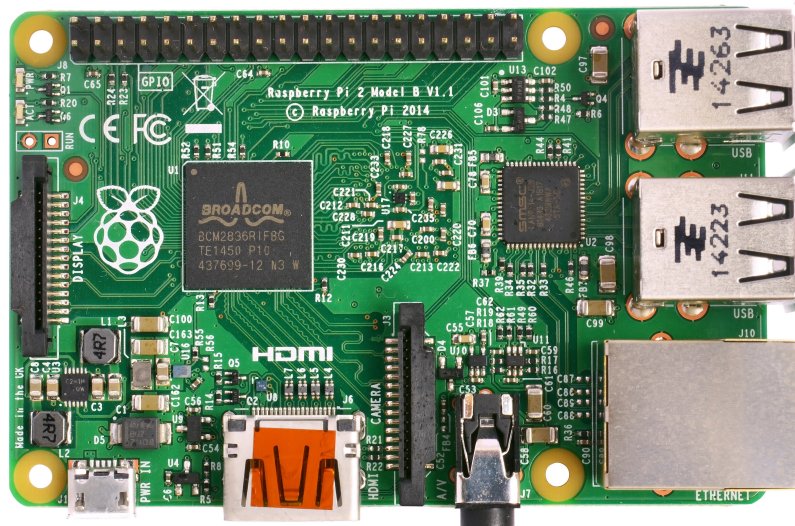
### 2.1 Použitá zařízení a nástroje

V této části bude stručně popsáno a přiblíženo zařízení, za pomoci kterého byla praktická část práce provedena. Kapitola obsahuje také několik slov o použitém programovacím jazyce a nejdůležitějších nástrojích potřebných pro její zrealizování.

#### 2.1.1 Raspberry Pi

Raspberry Pi je mikropočítač s deskou plošných spojů, který dosahuje přibližně rozměrů kreditní karty. Byl vyvinutý v roce 2012 ve Velké Británii za účelem podpořit výuku programování pro děti a studenty na školách a zároveň je seznámit s možností ovládání jiných zařízení pomocí počítačů. Tento počítač zvládá i řadu funkcí, které jsou běžné na stolním počítači či notebooku, jako je zpracování dat, načtení internetových stránek, hraní her nebo multimediální přehrávání videí s vysokým rozlišením a hudby. Lze ho zároveň použít k přímo k vývoji aplikací [24].

Používaným operačním systémem zařízení jsou nejčastěji různé distribuce firmy Linux, případně Windows 10 IoT (Internet of Things) od firmy Microsoft. Samotný počítač obsahuje několik vývodů pro připojení koncových zařízení jako monitor, klávesnice, myš, případně vývod pro připojení modulu Pi Camera a dále konektor pro připojení síťového kabelu. V dnešní době již existuje několik verzí lišících se především výkonem a předpokládaným použitím [24].



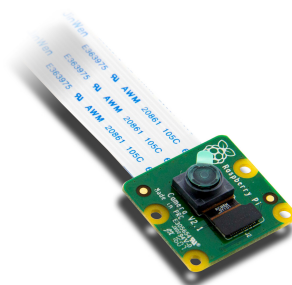
Obr. 2.1: Mikropočítač Raspberry Pi 2 Model B

Pro mou práci byla použita verze Raspberry Pi 2, která se na trhu objevila na počátku roku 2015. Operačním systémem využívaného zařízení je Raspbian Jessie. Použitý je čtyřjádrový mikroprocesor s taktem 900 MHz z rodiny ARM (Advanced Risc Machine) vyvinutý firmou Broadcom. Dále zařízení disponuje s 1 GB paměti RAM (Random Access Memory) a grafickým procesorem VideoCoreIV podporujícím fullHD (full High-definition) zobrazení. Na počítači nalezneme také slot pro mikroSD (mikroSecure Digital) kartu, která je nedílnou součástí zařízení, jelikož samotné Raspberry Pi nemá zabudované žádné interní úložisko dat. Přítomné jsou rovněž čtyři porty USB 2.0 (Universal Serial Bus), síťový adaptér 10/100 Mbps a GPIO (General Purpose Input/Output) vývody. Pro mou práci je velmi důležitý vývod pro připojení modulu Pi Camera [24].

## 2.1.2 Modul Pi Camera

Modul Pi Camera je používán za účelem tvorby videí nebo fotografií s vysokým rozlišením. Jeho použití je vhodné pro začátečníky, nicméně i pokročilejší uživatel může za pomoci modulu rozšířit své znalosti. Mnoho uživatelů využívá časosběrného režimu, zpomalení pohybu a jiných možností. Je možné také používat knihovny, které jsou s kamerou svázány [6].

Modul Pi Camera je uzpůsobený pro připojení k zařízení Raspberry Pi verze 1 a 2, pomocí kterého jej můžeme ovládat. Spojení Pi Camery s mikropočítačem je realizováno prostřednictvím plochého flex kabelu připojeného do portu CSI na Raspberry Pi. K zařízení je možno přistupovat za pomoci rozhraní API (Application Programming Interface) MMAL (Multi-Media Abstraction Layer) a V4L (Video for Linux) nebo příslušných knihoven. Knihovna, která umožňuje práci s modulem přímo v jazyce Python využívaném v následující práci, se nazývá Picamera. Rozlišení modulu je 5 Mpx, umožňuje snímat videa ve fullHD v rozlišení 1080p s 30 snímkovou snímací frekvencí za sekundu, stejně jako verze používaného počítače Raspberry Pi 2 [6].



Obr. 2.2: Modul Pi Camera V2.1.

### 2.1.3 Python

Python je snadno naučitelný programovací jazyk, jehož kód je pro uživatele srozumitelný a stručný. Programovací jazyk Python je multiplatformní. To lze vysvětlit tak, že program napsaný v tomto jazyce lze spustit po jednoduchém zkopírování souboru tvořícího program bez složité kompilace na zařízení s operačním systémem Windows, Linux, Mac OS X nebo BSD (Berkeley Software Distribution). V jádru je Python objektově orientovaným jazykem, nicméně stejně tak jej lze používat pro programování ve stylu procedurálním nebo funkcionálním [23].

K velkým výhodám Pythonu patří, že v základu obsahuje kompletní standardní knihovnu, která uživateli umožňuje funkce jako například stahování souborů z internetu či rozbalování zkomprimovaných souborů pomocí velmi krátkého kódu. Vyjma zmiňované knihovny je v Pythonu k dispozici mnoho dalších knihoven, které jsou často velmi specializované a ve standardní verzi je nenalezneme. Nicméně většina knihoven pro Python se nachází v seznamu balíčků pro jazyk Python (`pypi.python.org`). Právě knihovny jsou pro tuto práci velmi důležité. Nejvýznamnější z použitých jsou například knihovny OpenCV (Open Source Computer Vision), NumPy (Numerical Python) nebo PiCamera [23].

Používaná verze programovacího jazyka Python byla v mém případě Python 3.4 [23].

#### PiCamera

Knihovnu pro umožnění práce s modulem Pi Camera vyvinul a uvedl Dave Jones. S modulem lze pomocí knihovny navržené pro Python verze 2.7 a novější pracovat skrze mikropočítač Raspberry Pi [6].

#### OpenCV

Knihovna OpenCV, která je dnes velmi populární pro realizaci počítačového vidění, byla přestavena společností Intel v roce 1999. Knihovna se v první chvíli zaměřovala na zpracování obrazu v reálném čase a byly v ní zahrnuty té doby nejnovější algoritmy pro počítačové vidění. Willow Garag, který převzal nad knihovnou kontrolu v roce 2008, přišel s programovacím rozhraním pro jazyk C, C++, Python a Javu. Šířena je pod licencí BSD a používá se mimo jiné v akademických projektech či komerčních produktech [20]. Pro tuto práci je knihovna OpenCV jednou z nejdůležitější, jelikož umožňuje kvalitní předzpracování obrazu. Použitá verze je OpenCV 2.4.



## TensorFlow

TensorFlow je knihovna s volně šiřitelným zdrojovým kódem pro numerické výpočty pomocí grafů znázorňujících datový tok. Původně byla vyvinuta pro vědce a inženýry pracující v oblasti strojového učení a navrhování hlubokých neuronových sítí. Díky širokému záběru ji ovšem lze využít i v řadě jiných oblastí. V případě této bakalářské práce byla využita jako nástroj běžící na pozadí, umožňující fungovat dále popsané knihovna Keras. [25] Použitá verze je TensorFlow 1.0.1.

## Keras

Patrně nejdůležitější knihovnou využívanou v této bakalářské práci pro rozpoznání obličeje je knihovna Keras pro hluboké učení. Knihovna napsaná v programovacím jazyce Python je zaměřena na neuronové sítě a byla vyvinuta za účelem umožnění rychlého experimentování. Jak již bylo zmíněno, pro svou funkci potřebuje na pozadí spuštěný výše zmiňovaný nástroj TensorFlow, případně Theano. Hlavním autorem a vlastníkem knihovny je Francois Chollet pracující jako inženýr ve společnosti Google. [10] Verze, která byla využita pro zpracování neuronových sítí v této práci, je Keras 2.0.4.

## 2.2 Detekce tváře v obraze

### 2.2.1 Implementovaná metoda pro detekci tváře

Metoda implementována v rámci této práce se inspiruje dvěma typy metod popsanými v teoretické části a kombinuje je dohromady. První z nich je metoda, založená na vyhledávání tváře pomocí haarových příznaků, druhá využívá pro nalezení obličeje posouvání šablony oka. V následující části se nachází popis její implementace.

Úvodním krokem je tvorba a načtení snímku pomocí modulu Pi Camera, na kterém je detekce prováděna. První část detekce pak probíhá pomocí výše popsaných haarových příznaků a kaskádových klasifikátorů. Do programu jsou nejprve načteny předem natrénované klasifikátory ve formátu XML (Extensible Markup Language). Při práci s mikropočítačem Raspberry Pi je pro správnou funkci vždy nutné zadat absolutní cestu k souborům.

Následuje předzpracování obrazu, které začíná převedením barevné fotografie do odstínů šedi, za využití metody z knihovny openCV - *cvtColor*.

Dále už je detekován obličej za pomoci načtených natrénovaných klasifikátorů metodou *detectMultiScale* s parametry určenými přímo pro natrénovaná data z knihovny OpenCV. Pokud je na fotografii tvář nalezena, jsou vráceny parametry obdélníku  $x$ ,  $y$ ,  $w$ ,  $h$ , který ji ohraničuje. Zde  $x$  a  $y$  reprezentují kartézské souřadnice

levého horního rohu nalezeného ohraničení tváře,  $w$  představuje šířku obdélníku a  $h$  jeho výšku. Tento obdélník je následně libovolnou barvu v obraze vykreslen a původní obraz je dále oříznutý podle jeho hranic.

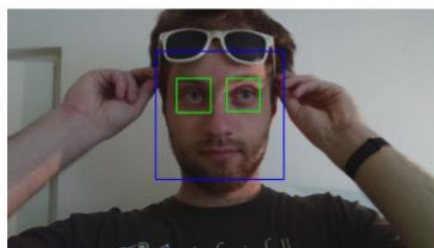
Výpis 2.1: Detekce výřezu tváře.

```
face_class = cv2.CascadeClassifier(path_xml) 1
grey = cv2.cvtColor(load_img, cv2.COLOR_BGR2GRAY) 2
points = face_class.detectMultiScale(grey) 3
4
for(x,y,w,h) in points: 5
    cv2.rectangle(grey, (x,y), (x+300,y+400), (255,0,0), 2) 6
    x,y = points [0] [:2] 7
    detection = grey[y: y + 400, x: x + 300] 8
```

Na vyhledání tváře může stejným způsobem navazovat vyhledávání očí v detekovaném výřezu. Vyhledání tváře s očima i bez nich je znázorněno na obrázku 2.3.



Ohraničení vyhledané tváře



Ohraničení vyhledané tváře i s očima

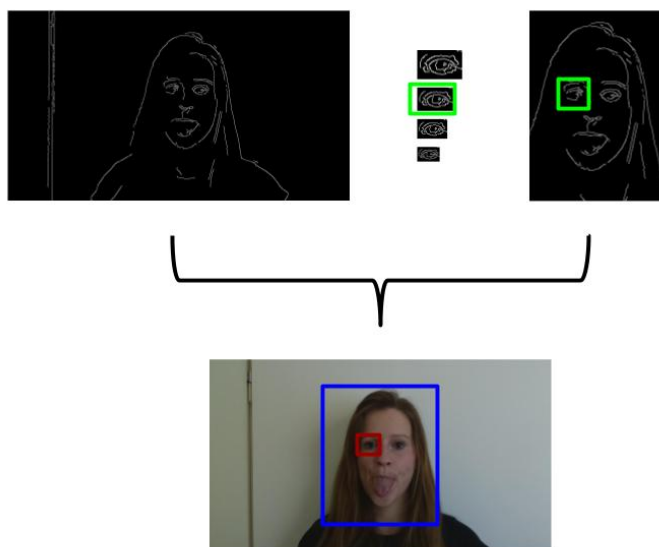
Obr. 2.3: Ohraničení vyhledané tváře.

V případě implementované metody jsou však oči v obličejí vyhledávány pomocí posouvání co největšího možného počtu šablon očí po obraze a vyhledávání místa s největší korelací. Pokud je dosaženo určité míry podobnosti, jsou nejpodobnější oblasti označeny jako místa výskytu očí.

Je proto opět nutné výřez s pravděpodobným výskytem tváře předzpracovat do vhodné podoby. Protože výřez je možné uložit už v odstínech šedi, lze pokračovat přímo aplikací Gaussova filtru pro vyhlazení obrazu. Využita je metoda z knihovny OpenCV - *GaussianBlur*, která provádí Gaussovou filtraci jako konvoluci původního obrazu s Gaussovou maskou, čímž dojde k vyhlazení a zanedbání náhodného šumu, který by jinak mohl škodit následné detekci. Do této funkce vstupuje šedotónový

obraz, rozměry masky a odchylky masky ve směru X a Y. Na zobrazení hran ve zpracovávaném obraze je následně použitý Cannyho hranový detektor pomocí funkce *Canny*. Funkce vyhledáváním lokálních extrémů zobrazuje hrany v místech největšího gradientu podle prahových hodnot zjištěných z histogramu. Do funkce vstupuje vyhlazený šedotónový obraz a prahové hodnoty.

Dále jsou do programu načteny připravené šablony očí. Šablony jsou pomocí vytvořené funkce *change\_size* zpracovány stejným způsobem. Zároveň pro větší přesnost prochází navíc každá šablona obrazem v několika různých velikostech. Vstupy funkce *change\_size* jsou cesta k načítané šabloně, velikost nejmenší a velikost největší šablony, krok se kterým je změna velikosti prováděna a pole, kam jsou všechny výsledné šablony ukládány.



Obr. 2.4: Hledání oblasti s okem pomocí šablon.

Samotná korelace šablon s jednotlivými oblastmi v obraze je prováděna ve *for cyklu* pro jednotlivé připravené šablony očí. Po obraze se horizontálně a vertikálně posouvá jedna šablona oka za druhou s nastaveným krokem. Původně použité porovnávání velkého množství šablon na základě hodnot pixelů se všemi oblastmi v obraze bylo velmi výpočetně náročné. Pro nalezení výskytu oka je proto v každé oblasti obrazu využíván výpočet korelační funkce, který celý proces výrazně urychluje. Pro výpočet korelace se využívá další metoda z knihovny OpenCV, *matchTemplate*. Jejími vstupními parametry jsou obraz, šablony očí se zobrazenými hranami a v mém případě parametr *TM-CCORR-NORMED*, který značí, že hodnoty korelace budou normalizované v intervalu hodnot od 0 do 1. Pomocí metody *minMaxLoc*

jsou z těchto korelací nalezena maxima a jejich pozice. Pokud je úplně nejvyšší maximum ze všech šablon v celém obraze nad empiricky stanoveným prahem, je potvrzen výskyt tváře v oblasti vymezené použitím metody s haarovými příznaky.

Výpis 2.2: Zpracování obrazu a výpočet korelace

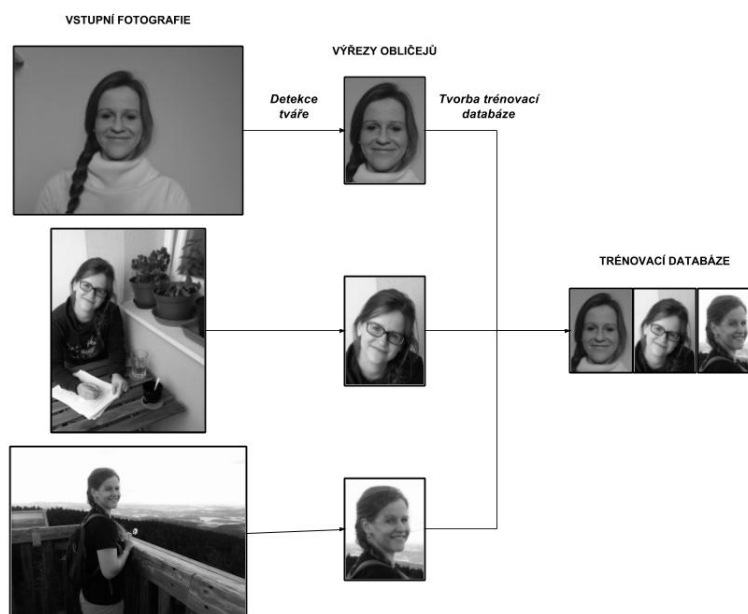
```
detection = cv2.GaussianBlur(detection, (5, 5), 0) 1
edges = cv2.Canny(detection, 50, 150) 2
3
for ind_temp in range(0, len(template)): 4
    templ = template[ind_temp] 5
    correlation = cv2.matchTemplate(edges, templ, 6
    cv2.TM_CCORR_NORMED) 7
    min, max, min_loc, max_loc = cv2.minMaxLoc(correlation) 8
    maximum.append(max) 9
    same = np.amax(maximum) 10
11
    if same >= TRESHOLD: 12
        crop.append(detection) 13
```

## 2.3 Rozpoznání detekované tváře

### 2.3.1 Metoda využívající konvoluční neuronovou síť

Po úspěšném detekování obličejů v obraze provedeném v prvním kroku následuje rozpoznání osoby. V bakalářské práci je za tímto účelem implementována metoda, která pro svou funkci využívá konvoluční neuronovou síť. Její princip je opět podrobněji vysvětlený v teoretické části práce.

Pro svoji funkci potřebuje konvoluční síť rozsáhlou trénovací databázi rozpoznávaných osob. Každá osoba by se v databázi měla vyskytovat z několika úhlů pohledu, při různém osvětlení. Využito může být i prostředků částečně zakrývajících obličej jako jsou brýle nebo čepice. Na fotografiích v této databázi je vyhledán obličej algoritmem popsaným výše a model sítě už je učen pouze výřezy vyhledaných tváří. Načtení fotografií ze složek a jejich následné oříznutí podle nalezené tváře je prováděno zavedenou funkcí *loading*. Prvním vstupem do funkce jsou cesty ke složkám, kde každá složka obsahuje trénovací fotografie pro jednu osobu. Dalšími vstupy jsou načtené šablony používané k detekci obličejů, číslo třídy, která bude reprezentovat danou osobu a cesta k souboru s daty natrénovanými pomocí haarových příznaků, které jsou uloženy ve formátu XML. Poslední dva vstupy jsou prázdná pole *crop* a *labels*. Do nich se po každém zavolání funkce uloží další a další výřezy s detekovanými tvářemi spolu s příslušnou třídou, do které spadají.



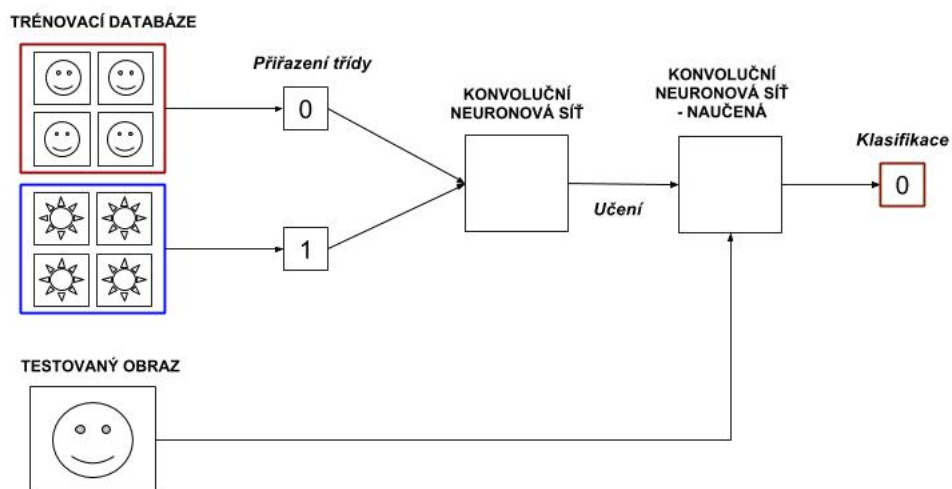
Obr. 2.5: Tvorba trénovací databáze pro jednu osobu.

Následuje krok rozdělení vytvořené databáze na trénovací část, sloužící k učení a testovací část, pomocí které se ověřuje přesnost rozpoznání osob. To je provedeno metodou *train\_test\_split*, do níž vstupují výřezy detekovaných tváří všech osob spolu s odpovídajícími třídami a v neposlední řadě parametr *train\_size*, který udává procentuální zastoupení trénovacích dat z celkové databáze. Trénovací i testovací data jsou následně zpracována a převedena do datového typu vhodného pro vstup do konvoluční neuronové sítě. V případě této bakalářské práce je to typ *float32*.

Při učení sítě má každá osoba přiřazenou třídu. Při průchodu obrazů sítí jsou sbírány informace o charakteristickém vzhledu a rozložení znaků v tvářích jednotlivých jedinců. Po ukončení učení je na základě těchto informací naučená konvoluční síť schopná predikovat, do které třídy má neznámou osobu v testovacím obraze zařadit. Postup klasifikace tváře je znázorněný na obrázku 2.6.

Důležitým krokem je definování konvoluční neuronové sítě. V této práci se skládá ze dvou vrstev konvolučních - *Conv2D*, dvou vrstev sdružovacích - *MaxPooling2D* a vrstev plně propojených - *Dense*. V konvolučních vrstvách dochází ke konvoluci vstupujícího obrazu s filtry, přizpůsobenými pro vyhledávání znaků v obličeji. Vrstvy sdružovací slouží k redukci rozměrů obrazu vybíráním maxima či průměru z oblasti ohraničené nastavenou maskou. Poslední plně propojené vrstvy určují aktivační funkci a jsou proto zodpovědné za zařazení osob na vstupních obrazech do tříd.

První konvoluční vrstva obsahuje 12 masek pro vyhledávání obličejových znaků



Obr. 2.6: Postup klasifikace testovaného obličej.

s rozměry  $5 \times 5$  a aktivační funkcí *relu*. Tato rektifikační lineární funkce je zdola omezená, aktivace neuronu tudíž probíhá pouze pokud je jeho vstup vyšší než nula. Následuje sdružovací vrstva s oknem velikosti  $2 \times 2$ , kde je pro redukci rozměrů vybíráno maximum v aktuálním okně. Další konvoluční vrstva obsahuje 25 masek, opět s velikostí  $5 \times 5$  a je následována opět vrstvou sdružovací s oknem o rozměrech  $2 \times 2$ . Výstup je vstupem do několika plně propojených vrstev, které postupně snižují množství tříd, do kterých jsou data rozřazována. Nejprve opět pomocí aktivační funkce *relu*, definitivní klasifikace probíhá potom aktivační funkcí *softmax*. Funkční hodnotou této aktivační funkce je pravděpodobnost, že výstup přísluší do odpovídající třídy.

Výpis 2.3: Architektura použité konvoluční sítě

```

model = Sequential()
1
2
model.add(Conv2D(12, 5, 5, activation = 'relu',
3
4
5
6
7
8
9
10
11
12
model.add(Conv2D(25, 5, 5, activation = 'relu',
input_shape=(dimensions), init='he_normal'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(25, 5, 5, activation = 'relu',
init='he_normal'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(180, activation = 'relu', init='he_normal'))
model.add(Dropout(0.5))
model.add(Dense(100, activation = 'relu', init='he_normal'))

```

<code>model.add(Dropout(0.5))</code>	13
<code>model.add(Dense(classes_num, activation='softmax',</code>	14
<code>init='he_normal'))</code>	15

Následuje provedení kompilace modelu pomocí metody *compile*, do které vstupuje parametr udávající, že výsledná data budou řazena do několika kategorií - *categorical\_crossentropy*, dále postup výpočtu zpětného šíření chyby - *SDG* (Stochastic Gradient Descent) a jednotky, ve kterých je udávána funkčnost sítě - *accuracy*.

Následuje učení sítě metodou *fit*, do které vstupují trénovací data spolu s příslušnou třídou zařazení, parametr *batch\_size*, který udává po kolika vzorcích bude síť trénována - v mém případě je roven 64, parametr *nb\_epoch* specifikující v kolika epochách bude učení probíhat - v mém případě 50, a testovací data - *validation\_data* pro spočítání přesnosti fungování navržené sítě. Jejich množství závisí na procentuálním rozdělení databáze na data trénovací a testovací.

Natrénovaná síť je závěrem uložena do formátu JSON (JavaScript Object Notation) a její váhy jsou uloženy zvlášť ve formátu HDF (Hierarchical Data Format).

Celý proces návržení, trénování a testování konvoluční neuronové sítě je zproštěnkovaný vytvořenou funkcí *network*.

## 2.4 Implementace na mikropočítač Raspberry Pi

Program je navržen tak, aby bylo možné naučenou síť načíst a spustit na platformě Raspberry Pi. Pomocí připojeného modulu Pi Camera je následně udělán snímek rozpoznávané osoby. Zpracování snímku a detekce tváře probíhá stejným způsobem jako při detekci využívané pro tvorbu učební databáze. Tvorba trénovací databáze byla dokonce ve velké míře tvořena právě pomocí modulu Pi Camera, kde lze nastavit požadované rozlišení s vhodnou sekvencí snímání. Samotné učení však vinou nižší operační paměti neprobíhá přímo na přístroji.

Po tvorbě testovaného snímku a detekci tváře je výřez obličeje poslán do načtené naučené sítě a pomocí metody *predict\_classes* je přiřazen odpovídající osobě. Implementace na mikropočítač Raspberry Pi umožňuje díky malým rozměrům zařízení snadné nasazení na nejrůznější místa, a to za cenu relativně malých pořizovacích nákladů.

## 2.5 Porovnání a zhodnocení metod

### 2.5.1 Detekce tváře

Metoda implementovaná přímo z knihovny OpenCV pracující na principu kaskádových klasifikátorů pomocí haarových příznaků vykazuje velmi dobrou úspěšnost, porovnání s další metodou následuje v tabulce 2.1. Není příliš výpočetně náročná a bylo by možné ji použít i pro detekci obličeje v reálném čase. Poměrně vysoké úspěšnosti dosahuje i v případě, že se tvář vyskytuje v komplexním pozadí. Mírný nedostatek lze vyzorovat v případě, že se v obraze nachází více obličejů, přičemž i takovou situaci by měla tato metoda být schopná úspěšně vyhodnotit.



Obr. 2.7: Původní obraz a výsledný detekovaný obličej po využití metody s haarovými příznaky.

Druhá implementovaná metoda pracuje na principu vyhledávání podle šablon očí, což je struktura, která by v obličeji neměla nikdy chybět. Důležité je však



Obr. 2.8: Původní obraz a výsledný detekovaný obličej po využití metody srovnávání šablon.

optimalizovat banku používaných šablon tak, aby jich bylo dostatek pro úspěšnou detekci, nicméně aby jich nebylo moc kvůli následné výpočetní náročnosti programu. Zrealizovaná metoda je přesto poměrně výpočetně náročná, zejména při spouštění



přes mikropočítač Raspberry Pi. I tato metoda ovšem dosahuje dobré úspěšnosti zhodnocené opět v tabulce 2.1. Úspěšnost nicméně nepatrně klesá při výskytu výraznějších objektů v pozadí obrazu.

Tabulka 2.1 ukazuje porovnání úspěšnosti obou metod. Je nutné ovšem podotknout, že testování probíhalo na dobře detekovatelných obličejích a sloužilo pro vzájemné zhodnocení obou metod.

Tab. 2.1: Procentuální úspěšnost v diskutovaných metodách detekce

Metoda	Absolutní úspěšnost	Relativní úspěšnost [%]
S haarovými příznaky	9/10	90
S šablonami oka	8/10	80

Jejich omezení, která přichází se složitějším obrazem, můžeme vidět na obrázku 2.9. V levé části obrázku jsou detekované tváře při využití metody založené na haarových příznacích. Je patrné, že metoda není schopná správně detekovat všechny tváře obraze. Největší problém nastává v přítomnosti brýlí. Přítomná je i falešně pozitivní detekce. Zobrazení hran v pravé části naopak ukazuje úskalí metody využívající šablony oka. Je zde vidět, že jednoduché předzpracování nestačí pro vhodné zobrazení hran a umožnění vyhledávat pomocí šablony oka. Metoda s šablonou oka navíc není pro větší množství tváří v obraze přizpůsobena.



Obr. 2.9: Problematika detekce v obraze s více tvářemi.

Zkombinování obou metod potlačuje vyhledávání falešně pozitivních detekcí první metody a snižuje výpočetní náročnost druhé metody. Zároveň omezuje její potřebu složitějšího předzpracování obrazu a potlačuje i následnou neschopnost detekovat více tváří v obraze. Korelace s šablonou oka může být prováděna ve všech výřezech nalezených metodou, která využívá haarovy příznaky.

## 2.5.2 Rozpoznání tváře

Pro rozpoznání tváří bylo využito konvoluční neuronové sítě. Jejím největším úskalím je relativně dlouhá doba učení sítě odvíjející se od velikosti trénovací databáze. V praxi je nicméně učení sítě od jejího samotného využívání ve většině případů odděleno. Oproti tomu testování probíhá v řádu několika sekund.

Tab. 2.2: Procentuální úspěšnost rozpoznání v různých situacích

Varianta	Relativní úspěšnost	Absolutní úspěšnost [%]
Frontální pohled	20/20	100
Natočení tváře	14/20	70
Brýle, zavřené oči	16/20	80
Grimasy	17/20	85
Zakrytá tvář	10/20	50
Náhodný výběr	15/20	75

Databáze využívaná v této bakalářské práci obsahuje přibližně 500 fotografií pro 12 rozpoznávaných osob. Následující tabulka 2.2 shrnuje úspěšnosti rozpoznání v různých podmínkách. Testování pro výpočet úspěšnosti bylo prováděno vždy na 20 tvářích v různých situacích.

Z tabulky je zřejmé, že nejvyšší úspěšnosti bylo dosaženo při testování obličejů z frontálního pohledu. Při této variantě byly testovány téměř ideální tváře a proto je dosaženo úspěšnosti až 100 %.

Při testování různě natočené tváře na obraze klesla úspěšnost až o 30 %. Největší problémy nastávaly při rozpoznání obličejů vyfocených z profilu, kde je tvář z velké části odkloněna. V případě natočení při stálém frontálním pohledu však bylo



Obr. 2.10: Úspěšnost rozpoznání v různých pozicích tváře.

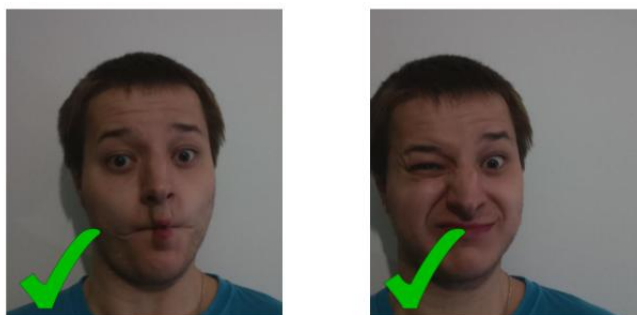
rozpoznání ve většině případů úspěšné, pravděpodobně díky přítomnosti většiny charakteristických rysů. Porovnání je zobrazeno na obrázku 2.10.

Testování obličejů s dioptrickými brýlemi, případně se zavřenýma očima vykazovalo úspěšnost 80 %. Zde hrály největší roli výrazné brýle, brýle s jemnými obroučkami ve většině případů nepředstavovaly problém, stejně jako zavřené oči.



Obr. 2.11: Úspěšnost rozpoznání s brýlemi a zavřenýma očima.

Různé grimasy a úšklebky představovaly pro navrženou síť asi nejmenší problém. Zde nelze objektivně zhodnotit jaké pozice tváře byly nejvíce problematické.



Obr. 2.12: Úspěšnost rozpoznání tváře s úšklebkou.

Naopak částečně zakrytá tvář měla úspěšnost pouhých 50 %. Zde je vidět, že vzájemné uspořádání výrazných znaků v obličejí hraje při jeho rozpoznání důležitou roli.

Úspěšnost náhodného výběru dosahující hodnoty 75 % má asi nejvyšší výpovědní hodnotu, jelikož dochází k testování různých variant tváří jako v reálných situacích.

Celkově je úspěšnost rozpoznání asi v nejvyšší míře ovlivněna velikostí a kvalitou trénovací databáze. Je důležité, aby obsahovala dostatečné množství snímků všech



Obr. 2.13: Úspěšnost rozpoznání částečně zakryté tváře.

klasifikovaných osob v různých podmínkách a byla je následně schopna rozpoznat v odlišných situacích.

### 3 ZÁVĚR

Bakalářská práce se věnuje studiu problematiky detekce a rozpoznání tváře v obraze. Metody k detekci jsou rozděleny do čtyř hlavních skupin, ve kterých jsou poté konkrétněji vysvětleny různé navržené algoritmy nalezené v literatuře. Jelikož detekce je základním krokem pro přiřazování které osobě nalezená tvář patří, druhá část je zaměřena právě na rozpoznání tváří. Zde jsou metody rozděleny do 2 hlavních skupin a opět podrobněji přiblíženy.

V části praktické je nejprve popsána implementovaná metoda pro detekci tváře. Je založena na kombinaci metody využívající haarových příznaků a metody hledající v obraze oko na základě banky šablon. Kombinace vznikla zejména z důvodu zvýšení úspěšnosti a snížení výpočetní náročnosti.

Detekce pomocí implementované metody je pro následné rozpoznání tváří stěží. To je prováděno na detekovaných výřezech. Je implementována navržená konvoluční neuronová síť učená vytvořenou databází obsahující 500 fotografií celkem 12 osob. Celá síť je natrénována a uložena. Důležité je přizpůsobení programu pro načtení naučené sítě na platformě Raspberry Pi, kdy je rozpoznávána fotografie vytvořená modulem Pi Camera.

Nakonec je provedeno zhodnocení metod, kde je nejprve vysvětleno použití implementované metody detekce. Následně je zhodnocena a popsána úspěšnost rozpoznání obličejů v různých situacích.

Úspěšná detekce spolu s následným rozpoznáním tváře je v současnosti velmi studovaným odvětvím výzkumu. Zejména v oblasti bezpečnosti a kontroly přístupu je automatické rozpoznávání osob využíváno stále hojněji. Implementace na mikropočítač Raspberry Pi, ke kterému je připojený modul Pi Camera pro tvorbu snímků, nabízí díky malým rozměrům zařízení snadné nasazení na různých místech s relativně nízkými pořizovacími náklady. Je však žádoucí optimalizovat program pro fungování na zařízení, jelikož nemá tak výkonný procesor a zpracování probíhá pomaleji. V některých případech může být řešením paralelizace zpracování, kdy je využíváno více než jedno jádro procesoru. Dalším úskalím, které je nutné obejít, je nízká operační paměť mikropočítače.

## LITERATURA

- [1] AL-ALLAF, Omaila L. A. Review of Face Detection Szstems Based Artificial Neural Networks Algorithms. *The International Journal of Multimedia and Its Applications*. 2014, **1**(6).
- [2] BLANZ, Volker a Thomas VETTER. Face Recognition Based on Fitting a 3D Morphable Model. *Transaction on Pattern Analysis and Machine Intelligence*. 2003, **25**(9).
- [3] BRUNELLI, Roberto a Tomaso POGGIO. Face Recognition: Features versus Templates. *Transactions on Pattern Analysis and Machine Intelligence*. 1993, **15**(10), 1042 - 1052.
- [4] Convolutional Neural Networks - Andrew Gibiansky. *Andrew Gibiansky* [online]. [cit. 2017-05-17]. Dostupné z: <http://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/>
- [5] Convolutional Neural Networks for Visual Recognition. *CS231nConvolutional Neural Networks for Visual Recognition* [online]. Stanford, 2017 [cit. 2017-05-17]. Dostupné z: <http://cs231n.github.io/convolutional-networks/>
- [6] Getting started with Pi Camera. In: *Raspberry Pi - Teach, Learn and Make with Raspberry Pi* [online]. [cit. 2016-11-19]. Dostupné z: [www.raspberrypi.org/learning/getting-started-with-picamera](http://www.raspberrypi.org/learning/getting-started-with-picamera)
- [7] GONG, Shaogang., Stephen J. MCKENNA a Alexandra. PSARROU. *Dynamic vision: From images to face recognition*. 2 nd. London: Imperial College Press, 2000. ISBN 18-609-4181-8.
- [8] HJELMAS, Eric a Boon Kee LOW. Face Detection: A Survey. *Computer Vision and Image Understanding*. 2001, **83**(3), 236-274. DOI: 10.1006/cviu.2001.0921. ISSN 10773142.
- [9] JAFRI, Rabia a Hamid R. ARABNIA. A Survey of Face Recognition Techniques. *Journal of Information Processing Systems*. 2009, **5**(2), 41-68. DOI: 10.3745/JIPS.2009.5.2.041. ISSN 1976-913x. Dostupné také z: <http://koreascience.or.kr/journal/view.jsp?kj=E1JBB0>.
- [10] *Keras Documentation* [online]. MkDocs [cit. 2017-04-25]. Dostupné z: <https://keras.io/>

- [11] KHALAJZADEH, Hurieh, Mohammad MANSOURI a Mohammad TESHNEHLAB. Face Recognition using Convolutional Neural Network and Simple Logistic Classifier. *Online Conference on Soft Computing in Industrial Applications Anywhere on Earth*. 2012
- [12] KIM, Inseong, Joon Hyung SHIM a Jinkyu YANG. *Face detection*. 2016. Stanford University.
- [13] KONTROPOULOS, Constantine a Ioannis PITAS. *Rule-based Face Detection in Frontal Views*. Greece, 2016. Aristotle University of Thessaloniki.
- [14] KOZUMPLÍK, Jiří a Ivo PROVAZNÍK. *Umělá inteligence v medicíně*. Brno, 2007. Vysoké učení technické v Brně.
- [15] KOZUMPLÍK, Jiří. Výběr příznaků. In: *Umělá inteligence v medicíně* [přednáška]. Vysoké učení technické v Brně, 2016.
- [16] LAWRENCE, Steve, C. Lee GILES, Ah Chung TSOI a Andrew D. BACK. Face Recognition: A Convolutional Neural-Network Approach. *Transactions on Neural Networks*. 1997, **8**(1), 98 - 113.
- [17] LI, Stan Z. a Anil K. JAIN. *Handbook of Face Recognition*. Second Edition. London: Springer, 2011. ISBN 978-0-85729-931-4.
- [18] LIYANAGE, De Silva, Aizawa KIYOHARU a Hatori MITSUTOSHI. Detection and Tracking of Facial Features by Using Edge Pixel Counting and Deformable Circular Template Matching. *Ieice Transactions on Information and Systems*. 1995, **E78-D**(9), 1195-1206.
- [19] MARQUÉS, Ion. *Face Recognition Algorithms*. Leioa, 2010. Universidad del País Vasco. Vedoucí práce Manuel Grana.
- [20] OpenCV. *OpenCV documentation* [online]. [cit. 2016-12-04]. Dostupné z: [opencv.org](http://opencv.org)
- [21] PARMAR, Divyarajsinh N. a Brijesh B. MEHTA. Face Recognition Methods and Applications. *Int. J. Computer Technology and Applications*. 2013, **4**(1), 84-86. ISSN 2229-6093.
- [22] PENTLAND, Alex a Tanzeem CHOUDHURY. Face Recognition for Smart Enviroments. *Computer*. 2000, , 50 - 55.
- [23] SUMMERFIELD, Mark. *Python 3: Výukový kurz*. Brno: Computer Press, 2013. ISBN 978-80-251-2737-7.

- [24] *Raspberry Pi - Teach, Learn and Make with Raspberry Pi* [online]. [cit. 2016-12-04]. Dostupné z: [www.raspberrypi.org](http://www.raspberrypi.org)
- [25] *TensorFlow* [online]. 2014 [cit. 2017-04-25]. Dostupné z: <https://www.tensorflow.org/>
- [26] TUČKOVÁ, Jana. *Vybrané aplikace umělých neuronových sítí při zpracování signálů*. Praha: České vysoké učení technické v Praze, 2009. ISBN 978-80-01-04229-8.
- [27] VIOLA, P. a M. JONES. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 2001, 511-518.
- [28] VLACH, Jan a Jiří PŘINOSIL. Lokalizace obličeje v obraze s komplexním pozadím. *Elektrorevue*. 2007, (12), 1-12. ISSN 1213-1539.
- [29] WISKOTT, Laurenz, Jean-Marc FELLOUS, Norbert KRÜGER a Christoph von der MALSBERG. Face Recognition by Elastic Bunch Graph Matching. *Intelligent biometric techniques in fingerprint and face recognition*. Boca Raton: CRC Press, 1999, 355 - 396. ISBN 0-8493-2055-0.
- [30] YANG, Ming-Hsuan, David J. KRIEGMAN a Narendra AHUJA. Detecting Faces in Images: A Survey. *Transactions on pattern analysis and machine intelligence*. 2002, **24**(1), 34-58.
- [31] YUILLE, Alan L., Peter W. HALLINAN a David S. COHEN. Feature Extractions from Faces Using Deformable Templates. *International Journal of Computer Vision*. 1992, **8:2**, 99-111.



## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

RGB	Red, Green, Blue
HSV	Hue, Saturation, Value
YCbCr	Luminence, Chromatic Blue, Chromatic Red
PCA	Principal Component Analysis
IoT	Internet of Things
ARM	Advanced Risc Machine
RAM	Random Access Memory
fullHD	full High-definition
microSD	microSecure Digital
USB	Universal Serial Bus
GPIO	General Purpose Input/Output
API	Application Programming Interface
MMAL	Multi-Media Abstraction Layer
V4L	Video for Linux
BSD	Berkeley Software Distribution
OpenCV	Open Source Computer Vision
NumPy	Numerical Python
XML	Extensible Markup Language
SDG	Stochastic Gradient Descent
JSON	JavaScript Object Notation
HDF	Hierarchical Data Format