



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

<b>Název:</b>	Rozpoznávání pohlaví, emocí a stáří osob z obrázků obličejů pro robota NAO
<b>Student:</b>	Eliška Vondráčková
<b>Vedoucí:</b>	Ing. Miroslav Skrbek, Ph.D.
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Znalostní inženýrství
<b>Katedra:</b>	Katedra aplikované matematiky
<b>Platnost zadání:</b>	Do konce letního semestru 2018/19

### Pokyny pro vypracování

Provedte rešerši metod a jejich dostupných implementací pro rozpoznávání pohlaví, emocí a stáří osob z obrázků obličejů. Zaměřte se jak na klasická, tak neuronová řešení. Na základě rešerše, kde zohledníte jednak použitelnost, nároky na výkon počítače, přesnost klasifikace a možnost integrace do jedné aplikace, vyberte vhodné řešení pro robota NAO. Pro výběr řešení proveďte potřebné úpravy existujících implementací a vyhodnocovací experimenty. Vybrané řešení integrujte s programovým vybavením robota NAO a vytvořte aplikaci, kde bude robot NAO reagovat na obličej, které se objeví před ním. Rozsah práce upřesněte po dohodě s vedoucím práce.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Karel Klouda, Ph.D.  
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
děkan

V Praze dne 17. prosince 2017





**FAKULTA  
INFORMAČNÍCH  
TECHNOLÓGIÍ  
ČVUT V PRAZE**

Bakalářská práce

## **Rozpoznávání pohlaví, emocí a stáří osob z obrázků obličejů pro robota NAO**

*Eliška Vondráčková*

Katedra teoretické informatiky

Vedoucí práce: Ing. Miroslav Skrbek, Ph.D.

14. května 2018



---

## Poděkování

Děkuji vedoucímu mé bakalářské práce, panu Ing. Miroslavu Skrbkovi, Ph.D. za pomoc při realizaci práce. Hlavně však musím poděkovat celé mé rodině a přátelům za podporu nejen při psaní této práce, ale po celou dobu studia.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 14. května 2018

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2018 Eliška Vondráčková. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Vondráčková, Eliška. *Rozpoznávání pohlaví, emocí a stáří osob z obrázků obličejů pro robota NAO*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.



---

# Abstrakt

Tato práce se zaměřuje na rozpoznání pohlaví, věku a emocí podle obličeje snímaného robotem NAO. V první části se zaměřím na rešerši metod rozpoznání obličeje. V části druhé potom na existující řešení tohoto problému a výběr jednoho vhodného z nich. Dále analyzuji možnosti a prostředky pro práci s robotem NAO. V poslední části potom vybrané řešení upravím pro robota a provedu testování.

**Klíčová slova** aplikace, rozpoznání obličeje, robot NAO, umělá intelligence, JAVA, Microsoft Azure

---

# Abstract

The aim of this work is to integrate gender, age and emotion recognition from face images in robot NAO. In first part, I will make a research of face recognition methods. In second part I will analyze existing implementations and choose the most suitable one. And I will analyze resources for work with robot NAO. In the last part I will modify selected implementation for robot and I will test it.

**Keywords** application, face recognition, robot NAO, artificial intelligence, JAVA, Microsoft Azure

---

# Obsah

Úvod	1
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Robot NAO</b>	<b>5</b>
<b>3 Algoritmy rozpoznání obličeje</b>	<b>7</b>
3.1 PCA (Principal Component Analysis)	7
3.2 ICA (Independent Component Analysis)	8
3.3 LDA (Linear Discriminant Analysis)	8
3.4 Evolutionary Pursuit	8
3.5 Elastic Bunch Graph Matching	9
3.6 Trace transform	9
3.7 3D Morfovateľný model	9
3.8 Bayesovské rozpoznání obličeje	10
3.9 Boosting a Ensemble metody	10
3.10 Deep Learning	10
3.11 Neuronové sítě	11
<b>4 Analýza existujících implementací</b>	<b>15</b>
4.1 OpenCV	15
4.2 OpenFace	16
4.3 Google Cloud Vision API	17
4.4 Kairos	17
4.5 Microsoft Azure	18
4.6 Výběr vhodné implementace	19
4.7 Face API	19
<b>5 NAOqi Framework</b>	<b>23</b>
5.1 Jak funguje NAOqi	23

5.2	NAOqi API . . . . .	24
<b>6</b>	<b>Realizace</b>	<b>27</b>
6.1	Vývoj aplikace . . . . .	28
6.2	Popis aplikace . . . . .	28
6.3	Návrhy na zlepšení . . . . .	30
6.4	Instalační příručka . . . . .	30
6.5	Uživatelská příručka . . . . .	31
<b>7</b>	<b>Testování</b>	<b>33</b>
7.1	Testování API samostatně . . . . .	34
7.2	Testování s robotem . . . . .	34
	<b>Závěr</b>	<b>37</b>
	<b>Literatura</b>	<b>39</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>41</b>
<b>B</b>	<b>Obsah příloženého CD</b>	<b>43</b>

---

## Seznam obrázků

2.1	NAO robot . . . . .	6
3.1	Histogram orientovaných gradientů (převzato z [1]) . . . . .	11
3.2	Orientační body obličeje (převzato z [1]) . . . . .	12
3.3	Neuronová síť . . . . .	13
5.1	Schéma načítání modulů a jejich metod (převzato z dokumentace [2]) . . . . .	24
6.1	Schéma komunikace s Face API a robotem . . . . .	27
6.2	Diagram tříd a jejich komunikace . . . . .	29
7.1	Ukázka testovacího GUI . . . . .	33



---

## Seznam tabulek

4.1	Srovnání Microsoft Azure a Kairos . . . . .	19
7.1	Úspěšnost určování jednotlivých atributů . . . . .	34
7.2	Úspěšnost určování jednotlivých atributů . . . . .	35





---

# Úvod

Téma rozpoznání obličeje je velmi aktuální a mnoho firem a odvětví se rozhodlo této technologii využít. Jedná se především o odvětví bezpečnost, marketing a zdraví. V oblasti bezpečnosti pomáhá rozpoznání obličeje k odhalení podvodů, je prostředkem k ověření identity uživatele, k monitorování davů lidí, v kriminalistice k identifikaci pachatele. V marketingu především k cílené reklamě a ve zdravotnictví k odhalení bolestí pacienta podle výrazu tváře, sledování konzumace léčiv.

Práce je určena pro robota NAO Fakulty informačních technologií ČVUT. Ten je prostředkem pro přiblížení technologií dětem a k představení výuky například zájemcům o studium.

Mojí motivací je přiblížení technologií lidem mimo obor, přilákání mladých lidí a dětí k zájmu o robotiku a k informačním technologiím obecně. Myslím si, že je podstatné tento obor propagovat, protože stále není dostatek odborníků v této oblasti.

V mojí bakalářské práci se budu zabývat tvorbou aplikace pro robota NAO, která bude umět rozpoznat obličeje, konkrétně určit pohlaví, věk a emoce. Použila jsem pro to Face API od Microsoft Azure.

V první části práce se budu zaměřovat na výzkum v této oblasti, jaké jsou aktuální trendy, jaké algoritmy jsou využívány. Ve druhé části provedu analýzu již existujících řešení a také analýzu možností robota. Ve třetí části budu již implementovat vlastní aplikaci, která bude integrovat jedno z vhodných řešení rozpoznání obličeje s robotem NAO.



## Cíl práce

Cílem rešeršní části práce je analýza metod rozpoznání obličeje, především určení pohlaví, věku a emocí. Dále rozbor již existujících implementací problému. Součástí bude též výzkum možností robota NAO. Cílem praktické části je výběr vhodné implementace a její úprava pro integraci s robotem. A dále pak komunikace robota s člověkem nacházejícím se před ním, založená na získané informaci o jeho věku, pohlaví a emocích.



## Robot NAO

Robot NAO je humanoidní robot, kterého na trh uvedla francouzská společnost Aldebaran Robotics. Tato poměrně mladá firma, založená v roce 2005 uvedla NAO poprvé na trh u příležitosti soutěže RoboCup, mezinárodní turnaj robotů ve fotbale, v roce 2008. Od tohoto roku bylo uvedeno několik nových verzí a v současné době je software pro vývoj robota open-source. Již v roce 2008 byla uvedena The NAO Academics Edition pro laboratoře a univerzity, od té doby využívá robota přes 200 akademických institucí po celém světě. NAO je používán pro přiblížení světa robotiky menším dětem, byl například také darován pro výuku autistických dětí v Británii, protože děti mu důvěřují více než dospělým. Další úspěch NAO sklidil při tanci na EXPO v Šanghaji. Od roku 2015 je Aldebaran Robotics pod japonskou společností SoftBank Group.

Jak již bylo zmíněno, NAO je humanoidní robot, snaží se tedy svými dovednostmi přiblížit člověku. Dokáže komunikovat, pohybovat se, reagovat na dotek, hlas nebo lidský obličej. Shrňme si zde jeho technické parametry.

Senzory, které NAO má, jsou: 2 kamery, 4 mikrofony, sonar pro měření vzdálenosti, 2 infračervené zářiče a přijímače, 9 dotekových a 8 tlakových senzorů a interciální deska.

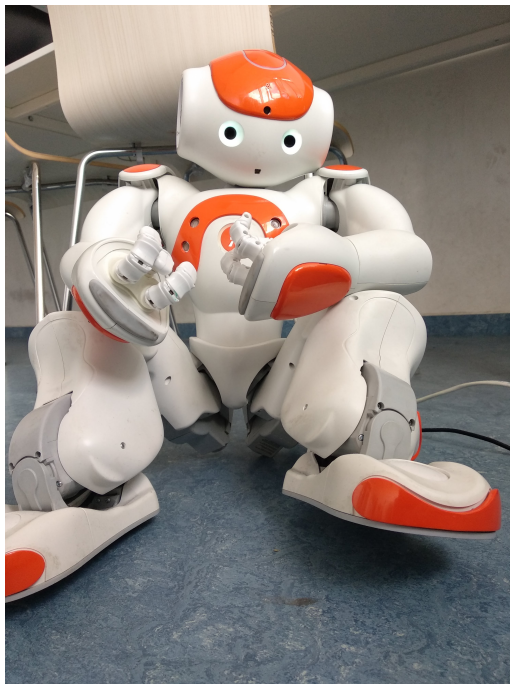
Dále má NAO elektrické motory sloužící k pohybu, syntetizátor hlasu, reproduktory a LED světla. V hlavě má umístěn procesor Intel Atom (1,6 GHz). [3]

Vizuální a zvukové dovednosti:

- Kamera - pohyb hlavou až o 239 stupňů, až 30 snímků za sekundu v HD kvalitě.
- Rozpoznání objektu - rozpoznání velkého množství objektů včetně jejich uložení k opětovné identifikaci

## 2. ROBOT NAO

---



Obrázek 2.1: NAO robot

- Detekce a rozpoznání obličeje - je schopen detekovat před sebou obličej a zapamatovat si ho
- Mluvení (Text to Speech) - dokáže mluvit 19 jazyky (včetně češtiny)
- Rozpoznání mluveného slova - schopnost rozpoznat větu nebo jednotlivá slova věty ze vzdálenosti 2 metrů
- Detekce a lokalizace zvuku - po celé hlavě rozmístěné mikrofony zajišťují detekci zvuku z okolního prostředí

### Pohybové reflexy

- Koordinace pádu - při pádu se chrání rukama a dokáže se postavit zpět na nohy
- Ochrana proti kolizi jednotlivých hybných částí sám se sebou - stále si hlídá pozici hlavy, rukou, trupu a nohou, aby do sebe tyto části nenarazily
- Koordinace příkazů - dokáže přerušit nebo upravit své chování, sjednotit konfliktní příkazy

## Algoritmy rozpoznání obličeje

Rozpoznání obličeje je lidská schopnost, kterou bychom rádi zvládli i strojově, tedy bez lidského přičinění. Motivací je vzrůstající poptávka po této schopnosti a to nejlépe v reálném čase. Je třeba lokalizovat obličej na jakémkoliv pozadí, identifikovat a verifikovat ho.

V této části představím základní algoritmy a přístupy k detekci a rozpoznání obličeje. Popíšu základní principy a myšlenky, ze kterých tyto algoritmy vychází.

Nezákladnější je Analýza hlavních komponent (PCA), ze které často vychází i další metody, známý je jako Eigenface algoritmus. Dalším důležitým přístupem jsou neuronové sítě, které jsou v současnosti stále více využívány i pro rozpoznání obličeje. V následujících odstavcích popíši jednotlivé metody.

### 3.1 PCA (Principal Component Analysis)

PCA, Analýza hlavních komponent, je statistická metoda, používaná zejména k redukci dimenze dat tak, aby byla ztráta informace co nejmenší. Je základem a startovním bodem algoritmů k rozpoznání obličeje. Je totiž jednoduchá na implementaci a má dobrý výkon.

V kontextu rozpoznání obličeje se dá PCA využít také. Obličej reprezentujeme jako  $\mathbf{n}$  dimenzionální vektor, který PCA transformuje do  $\mathbf{k}$  dimenzionálního podprostoru, jehož základní vektory odpovídají směru maximálního rozptylu v originálním prostoru. Z principu PCA platí, že  $\mathbf{k}$  musí být mnohem menší než  $\mathbf{n}$ .

Vstupem je trénovací set obrázků, kde každý obrázek je reprezentován jako bod v prostoru. PCA najde optimální lineární reprezentaci v prostoru (zachovávajícím varianci) metodou nejmenších čtverců. PCA reprezentace je charakteristická souborem vlastních vektorů a vlastních čísel. Každý řádek obrázku je uložen jako vektor. Vlastní vektory jsou nejprve normalizovány, aby byly ortogonální a uspořádány podle jejich vlastních čísel. Vlastní čísla jsou rovna rozptylu projekce trénovacího setu do vektoru. První seřazený vek-

tor má velký rozptyl od trénovacího setu, naopak výše zařazený vektor má rozptyl menší. Horní část s malým rozptylem můžeme odseknout, protože obrazy nejlépe charakterizují vektory seřazené na začátek. Vzniká rozptylu má exponenciální trend, tedy 90% rozptylu je obsaženo v prvních 5% – 10% vektorů. Tím vznikne nový prostor o menším počtu vektorů, někdy nazývaný “facespace”. [4]

## 3.2 ICA (Independent Component Analysis)

ICA, Analýza nezávislých komponent, hledá lineární transformaci, která vyjádří soubor náhodných proměnných jako lineární kombinaci statisticky nezávislých proměnných. Kritériem k vyhledání této transformace zahrnuje minimalizaci vzájemných informací vyjádřených funkcí vyšších řádů jednotek. Pokud porovnáme PCA a ICA, tak PCA uvažuje pouze dva řády jednotek, zatímco ICA počítá vyšší řády. Tím identifikuje nezávislé komponenty z jejich lineárních směrů. ICA tedy poskytuje lepší data pro reprezentaci obličejů. [5]

## 3.3 LDA (Linear Discriminant Analysis)

LDA, Analýza lineárních diskriminantů, najde vektory, které nejlépe rozlišují jednotlivé třídy. Pro všechny vzorky ze všech tříd jsou definovány dvě matice. První matice je pro rozptyl mezi třídami a druhá pro rozptyl uvnitř třídy. Cílem je maximalizovat rozptyl mezi třídami a současně minimalizovat rozptyl uvnitř jednotlivých tříd. Lidský obličej má charakteristické prvky, podle kterých se dá v různé míře rozdělovat obličej do tříd (například podle pohlaví). Míry obličejů v tomto případě určují význam různých atributů podle toho, jak dobře odlišují různé třídy od sebe. [6]

## 3.4 Evolutionary Pursuit

Generický algoritmus, který se ve vztahu k problému rozpoznání obličejů dá využít ke zjištění optimální báze zakódování obličejů, rotuje základní vektory v prostoru redukovaném PCA. Evoluce je řízena fitness funkcí, která vyjadřuje přesnost a index rozptylu. Přesnost vyjadřuje, do jaké míry bylo učení úspěšné, index rozptylu udává předpokládanou fitness na budoucích pokusech (schopnost generalizace). Pokud je index moc velký je vhodné další učení, aby současná přesnost byla dobrým indikátorem budoucího výkonu. Predikce budoucího výkonu mohou ovlivňovat míru učení nebo učení úplně zastavit.

Výsledkem je lepší výkon a generalizace v porovnání s PCA. [7]



### 3.5 Elastic Bunch Graph Matching

Tato metoda slibuje lepší výkon než neuronové sítě. Snaží se vyhnout statistickým metodám a mít spíše jednoduchý, přirozený přístup. Využívá tzv. graf obličejů. Výrazné body obličejů (oči, ústa) jsou znázorněny jako uzly grafu a hrany mezi nimi jsou 2-D vektory. Každý uzel je popsán vlnkami, kterým se říká “jets” a vektory jsou popsány jejich vzdáleností. Rozpoznání je založeno na porovnávání těchto grafů. Je odolná vůči rotacím, deformacím obrázků a problémům s osvětlením, díky vlnkám. Pro každou pózu (systém používá asi 70 různých póz) jsou umístěny vlnky na již zmíněné výrazné body obličejů, tím vznikne shlukový graf. Shlukové grafy jsou brány jako kombinační entity, kde pro každý výrazný bod, může být vybrána kterákoliv vlnka z jednotlivých póz. Tím vzniká vysoce přizpůsobivý model, který spolehlivě najde výrazné body obličejů.[8]

### 3.6 Trace transform

Trace transform, v překladu trasová transformace, se používá k rozpoznání objektu pod různými transformacemi (rotace, deformace, škálování). Nejdříve je nutné spočítat funkci podle trasovacích linií obrazu. Každá linie je charakterizována vzdáleností od středu os a úhlem kolmice, kterou má linie vzhledem k referencovanému směru. Trasová transformace je tedy 2D funkce s těmito dvěma parametry. Trasová transformace dobře vyzdvihuje tvar a strukturu charakteristickou pro objekt, který popisuje. Pro použití rozpoznání obličejů se pracuje s maskou obličejů ve tvaru elipsy. Jsou definovány osy se středem ve středu elipsy a obdélník opisující elipsu, u kterého jsou označeny pixely náležící do elipsy. Trasovací linie jsou definovány pouze pro označené pixely v obdélníku, neoznačené jsou ignorovány. Posouváním os je dosaženo odstranění neoznačených pixelů a zároveň zajištění, že tvar elipsy nebude mít vliv a jen podstatné informace budou použity k výpočtům. Pozadí kazí strukturu i tvar, takže je dobré se ho tímto způsobem zbavit. K určení rotovaného obrázku se využívá úhel z trasové transformace. Porovnává se každý sloupec originálního a rotovaného obrázku a podle největší shody se sloupec posune.[9]

### 3.7 3D Morfovateľný model

Tato metoda se používá pro generování 3D modelu obličejů ze tří obrázků osoby. Jako tři vstupní obrázky pro model jsou vzaty fotky zepředu, z polovičního profilu a z úplného profilu. Modely jsou renderovány z různých úhlů póz a za různých světelných podmínek. Z modelů jsou vybudovány velké sady umělých obrázků, podle kterých je poté trénován některý ze systémů založených na analýze komponent. Jde o to, aby byla vybudována dostatečně velká

databáze modelů obličeje tak, že jakýkoliv obličej bude moci být vygenerován morfováním obrázků v databázi.[10]

### 3.8 Bayesovské rozpoznání obličeje

Bayesovská metoda využívá známou pravděpodobnostní funkci k určení míry podobnosti. Rozlišuje dvě variace člověka. První je intrapersonální, která může být způsobena například různými výrazy jednoho člověka. Druhá variace je extrapersonální, to je rozdíl mezi jednotlivými lidmi. Znamená to tedy, že rozdíl mezi dvěma obličejí můžeme vyjádřit pomocí pravděpodobnosti rovnicí 3.1.

$$S(I_1, I_2) = P(\Delta \in \Omega_I) = P(\Omega_I|\Delta) \quad (3.1)$$

$S(I_1, I_2)$  je podobnost obrázku  $I_1$  a  $I_2$ ,  
 $\Delta$  je rozdíl mezi obrázky  $I_1$  a  $I_2$  ( $\Delta = I_1 - I_2$ ),  
 $\Omega_I$  je intrapersonální variace.[11]

### 3.9 Boosting a Ensemble metody

Boosting je algoritmus strojového učení, který je založen na tom, že jsou za sebou zapojené tzv. weak learners (špatně učící se jednotky) schopny dohromady být dobře se učící jednotkou. Weak learner je algoritmus, který je schopen klasifikace alespoň o něco málo lepší než náhodná volba. V kontextu rozpoznání obličeje je využíván především algoritmus AdaBoost. AdaBoost kombinuje výstupy jednotlivých učících jednotek podle vah do finální sumy, která je potom výstupem. Viola a Jones tento algoritmus využili ve své práci Robust Real-time face recognition. Tento algoritmus je použit k vybrání malého množství důležitých vizuálních prvků z velkého množství potencionálních prvků. Učící jednotky jsou spojeny kaskádovitě, což umožňuje snadno odstranit pozadí obrázků.[12]

### 3.10 Deep Learning

V současné době se k rozpoznání obličeje používají metody deep learningu, tedy hlubokého učení. Hluboké učení je jednou z částí strojového učení, jež je založeno na naučení se reprezentovat data. Učení může být s učitelem, bez učitele a posilované učení. Je nutné vyřešit několik problémů. Prvním je, jak obličej najít, dále jak rozeznat obličej z různých úhlů a jak obličej zakódovat.

HOG – Histogram of Oriented Gradients je způsob jak obličej najít. Obrázek je transformován do černobílé podoby. Poté pro každý pixel se podívá na jeho sousední pixely a nakreslí šipku, podle toho, kterým směrem se pixely ztmavují. Tímto je každý pixel nahrazen šipkou, které se říká gradient (znázorňuje sklon v daném místě obrázku). Obrázek rozdělíme na menší čtverce



Obrázek 3.1: Histogram orientovaných gradientů (převzato z [1])

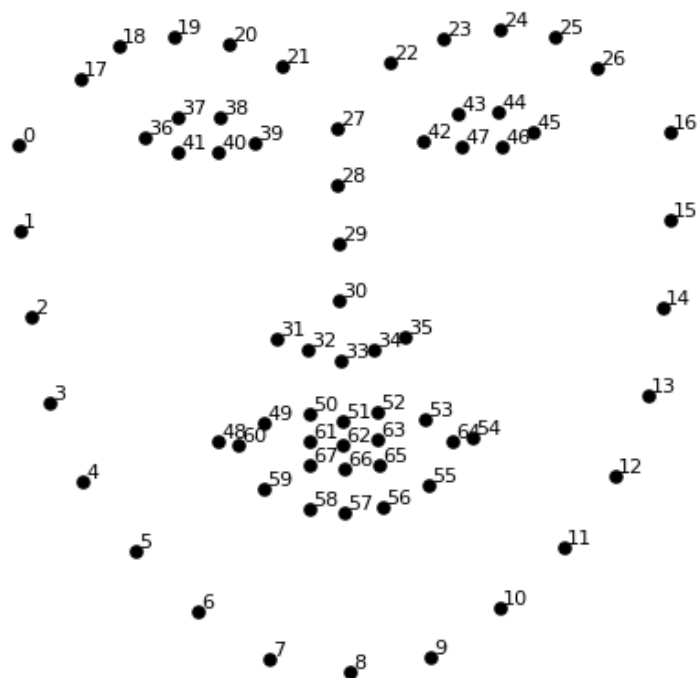
(16\*16 pixelu) a v každém čtverci spočítáme zastoupení směrů jednotlivých gradientů (šipek). Daný čtverec potom bude nahrazen jednou šipkou a to tou, která má největší zastoupení. V takto vzniklém obrázku hledáme oblast podobnou vzoru obličeje (viz obrázek 3.1).

Face Landmark Estimation – Odhad orientačních bodů obličeje je metoda k rozeznání obličeje z různých úhlů. Landmark je v překladu orientační bod. V této metodě je těmito orientačními body 68 specifických bodů obličeje (obrysy očí, nosu a úst, obočí a kraje tváří). Podle těchto bodů dokážeme lokalizovat obličej (viz obrázek 3.2). Natrénuje neuronovou síť (nebo jiný machine learningový algoritmus), která bude schopna tyto body najít na jakémkoliv obličejí. Každý obličej se potom pomocí rotování, ořezávání atp. budeme snažit dostat do co nejlepší polohy vzhledem k těmto bodům (vycentrovat je).[1]

Posledním problémem je zakódování obličeje. K tomu se využívají konvoluční neuronové sítě, které generují nejlepší (nejvíce charakteristické) míry obličeje. Neuronovým sítím a jejich aplikaci k rozpoznání obličeje je věnována další kapitola

### 3.11 Neuronové sítě

Neuronové sítě jsou velmi silnou klasifikační metodou, která nejenže dokáže roztrždit známá data, ale také ta neznámá. Využívá se ve spoustě oblastí kromě počítačového vidění je to třeba ještě interpretace mluveného slova. Jsou inspirovány fungováním nervového systému v těle. Skládají se z umělých neuronů, které jsou navzájem propojeny a předávají si mezi sebou signály. Jsou tři druhy neuronů: vstupní, skryté a výstupní. Každé spojení má svojí váhu v intervalu



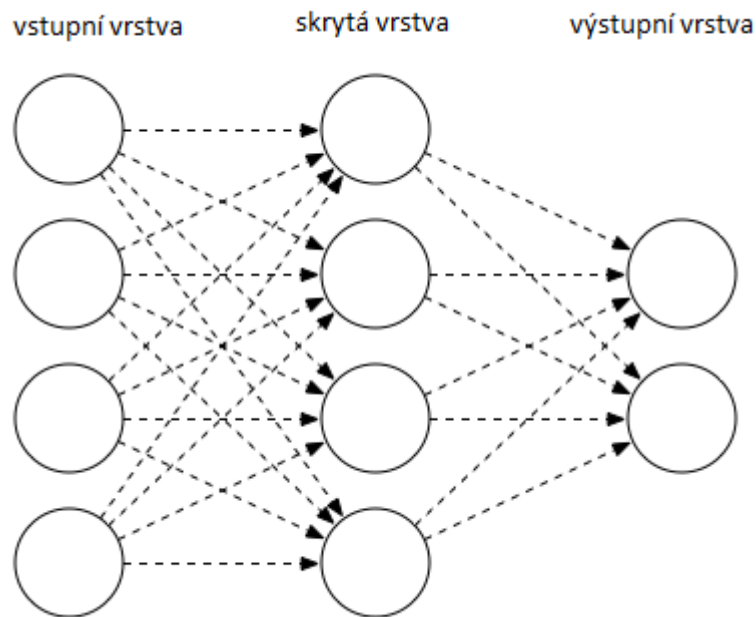
Obrázek 3.2: Orientační body obličeje (převzato z [1])

od -1 do 1. Informace vstupuje do neuronové sítě vstupními neurony jako numerický výraz, který se nazývá aktivační funkce. V každém uzlu (neuronu) je uloženo číslo  $a$ , pokud je aktivační funkce vyšší, je předána dále do sítě. Aktivační funkce závisí na vstupu, váhách spojení, přenosové funkci. Každý uzel přijme aktivační funkce z jednotlivých vstupů, sečte je a upraví výstup podle své přenosové funkce. Přes skryté vrstvy se aktivační funkce postupně dostane až k výstupním uzlům sítě.[13]

Existuje spousta druhů sítí. Já se zde budu věnovat třema případům užití. První je jednoduché navázání na PCA, které dobře popisuje fungování neuronových sítí. Další dvě sítě jsou složitější a prokázaly dobrý výkon při rozpoznávání obličeje.

### 3.11.1 PCA s umělou neuronovou sítí

PCA je zde použito k redukci dimenze obrázku obličeje a získání hlavních rozdíků v datasetu. Spočítá vektory pro jednotlivé body obličeje a z těchto vektorů vytvoří sloupcovou matici. Po každém spočítání vektoru je vypočten průměr obličeje, který se použije k normalizování každého vstupního obrázku. Nejprve je od vstupu odečten průměr a vytvořena kovarianční matice. Dále jsou spočítány vlastní čísla této matice a uložena pouze ta největší, protože nejlépe



Obrázek 3.3: Neuronová síť

charakterizují obličej. Následně se spočítají vlastní vektory kovarianční matice, tím vznikne tzv. eigenface, který má nejlepší vypočítací hodnotu o obličej. Takto jsou z vysoké dimenze extrahována data do nízké dimenze. Zachovány zůstanou pouze ty informace, které jsou nejvíce charakteristické, výrazné.

Umělé neuronové sítě jsou potom použity pro trénování a testování. Pro testování se jako vstup použijí již zmíněné eigenface pro natrénovanou síť, ta potom najde největší shodu. Pro trénování se používá umělá neuronová síť se zpětnou propagací, kde vstupem jsou přímo obrázky obličeje. V neuronové síti jsou nastaveny iniciální parametry - váhy. Vzdálenost jednotlivých eigenface je kritériem pro určení správnosti výsledků z neuronové sítě. Stejně eigenfaces by měly mít vzdálenost 0. Pokud nastane chyba, tak je z výstupní vrstvy poslána zpět do předchozích vrstev a na těchto vrstvách jsou upraveny váhy tak, aby byla chyba minimalizována.

### 3.11.2 RBF síť

RBF síť (Radial Basis Function Network) jsou sítě, které mají ve skryté vrstvě tzv. kruhové báze funkce a výstup sítě je potom lineární kombinací této báze funkce vstupu a vah spojení sítě. K redukci dimenzionality může být opět použito PCA. RBF síť se potom postará o nalezení unikátního vzoru v obličej každého člověka. Architektura sítě se skládá ze tří částí: podmínková fáze, závěrečná fáze a důsledková fáze. Tyto fáze jsou založeny na fuzzy pravidlech,

kteře mají většinou podobu if-then. V první podmínkové části fuzzy pravidel je vstupní prostor rozdělen pomocí metody k-means. Ve druhé závěrečné části jsou váhy spojení neuronové sítě reprezentovány polynomy, nejvýše kvadratickými. Výstup sítě je vytvořen poslední důsledkovou částí fuzzy pravidel. RBF se používá pro svoji kompaktnost a rychlost učení.

#### 3.11.3 Kombinace BPN a RBF sítí

Ukázalo se jako velmi efektivní zkombinovat síť se zpětnou propagací (BPN) a RBF síť. Rozpoznání je docíleno porovnáním charakteristik nového obličeje se známými obličeji.

V této metodě je zahrnuta i detekce. A to tak, že jsou nalezena ústa a oči a je spočítána vzdálenost mezi nimi. Rozpoznání je provedeno BPN a RBF sítí. BPN obsahuje jednu vstupní, jednu skrytou a jednu výstupní vrstvu. Vstupy jsou vektory, které jsou získány z detekce (vzdálenosti důležitých částí obličeje). Zpětná propagace má tři části: předání vstupních trénovacích dat, zpětná propagace chyby a přenastavení vah spojení. V první části dostane každý vstupní neuron vstup a tento je dále rozeslán do všech skrytých neuronů. Skryté neurony počítají aktivační funkci a předávají ji dále do výstupních neuronů, jež znovu spočítají aktivační funkci, která je již výstupem. Při trénování je tento výstup porovnán s očekávanou hodnotou a určí se chyba, která je zpětně propagována do skryté vrstvy. Podle této chyby jsou upraveny váhy spojení tak, aby byla chyba minimalizována. Výstupní vrstva obsahuje pouze jeden vektor a výstup z této vrstvy je použit jako vstup pro RBF síť. Ta používá gaussovu funkci jako odhad výstupu BPF. Tato metoda pracuje s přesností 98,88 %, je tedy jednou z nejlepších.[14]

# Analýza existujících implementací

V této části práce představím několik možných řešení, které jsem našla. Následně tato řešení porovnám a vyberu jedno, jež použiji k implementaci mé aplikace.

## 4.1 OpenCV

OpenCV je populární knihovna pro počítačové vidění (computer vision), kterou založil Intel již v roce 1999. Používá ji přes 47 tisíc lidí. Má velmi široké použití od robotiky, detekci min až po interaktivní umění. Zaměřuje se především na zpracování obrazu v reálném čase a na nejnovější algoritmy s tím související. Je opensource, proto se často využívá pro akademické a komerční účely. Má interface pro C++, Python, Javu, podporuje Linux, Windows, Mac OS, iOS i Android. Byl vytvořen pro vysokou výpočetní efektivitu a pro real-time aplikace. Napsán je v C++, má výhodu, že je multiplatformní a podporuje vícejádrové procesování.

OpenCV2.4 má třídu FaceRecognizer na rozpoznání obličeje. K dispozici jsou algoritmy Eigenface i Fisherface. Pro určení věku, pohlaví a emocí v této knihovně řešení není, ale nabízí se možnost natrénovat model, který to bude umět. FaceRecognizer podporuje trénink modelu, predikcy nebo označování fotografií popisky (to se hodí při trénování modelu). Trénování lze provést pomocí algoritmu Fisherface, který umí dobře určit rozdíly mezi třídami.[15]

Pro toto řešení by tedy bylo nutné:

- najít dostatečně velké množství fotografií
- označit jejich klasifikaci labely
- natrénovat FaceRecognizer a to hned třikrát (pro pohlaví, věk, emoce)

Trénink pro emoci by byl poměrně složitý, protože narozdíl od pohlaví je emoci velká škála a bylo by nutné velké množství obrázků různých emoci (tříd).

### 4.1.1 Shrnutí výhod a nevýhod řešení

Výhody:

- open-source, možnost jakékoliv úpravy
- poměrně dobrý základ pro trénování vlastních modelů
- není třeba připojení k síti

Nevýhody:

- potřebuji klasifikovat podle více kritérií a v případě emoci také do více tříd
- řešení by bylo zbytečně zdlouhavé a složité
- s touto knihovnou lze pracovat pouze v Pythonu

## 4.2 OpenFace

OpenFace je interface pro rozpoznání obličeje pomocí hluboké neuronové sítě napsaný v Pythonu. K dispozici jsou předem natrénované modely od OpenCV a dlib. Obrázek je transformován do verze pro neuronovou síť. Je tedy rotován a centrován tak, aby se oči a ústa nacházely vždy ve stejných místech na každém obrázku. Dále je použita hluboká neuronová síť, která obličej zakóduje a určí jeho míry, podle kterých se dá později vyhodnotit podobnost obličejů.[16]

### 4.2.1 Shrnutí výhod a nevýhod řešení

Výhody:

- open-source, možnost jakékoliv úpravy
- dobrá dokumentace
- není třeba připojení k síti

Nevýhody:



- neposkytuje hotové řešení pro určení atributů obličeje
- řešení by bylo zbytečně zdlouhavé a složité
- s touto knihovnou lze pracovat pouze v Pythonu

## 4.3 Google Cloud Vision API

Google vyvinul REST API, které slouží k analýze obrázků. Umí detekovat předměty a obličeje, klasifikovat do kategorií a přečíst text na obrázku. Podporuje také detekci obličeje včetně emocí: radost, smutek, zlost a překvapení. Tyto však neurčuje, v jaké míře se vyskytují, ale s jakou pravděpodobností (likelihood). Věk a pohlaví určit neumí. Po nahrání zkušebního obrázku trvala odpověď velmi dlouho.[17]

### 4.3.1 Shrnutí výhod a nevýhod

Výhody:

- do 1000 obrázků za měsíc služba zdarma
- spolehlivá společnost, v současnosti velký rozmach a oblíbenost
- detekuje všechny obličeje na fotce

Nevýhody:

- věk a pohlaví neumí určit
- pomalá odpověď
- nutné neustálé připojení k síti kvůli komunikaci přes API

## 4.4 Kairos

Kairos se zabývá strojovým učením, především pak rozpoznáním obličeje. Cílí především na větší firmy, které mohou jejich řešení aplikovat v marketingu a ke svým komerčním účelům. Od toho se také odvíjí, že pouze malé množství služeb je od této společnosti zdarma. Poskytuje nejen API pro přístup k rozpoznání obličeje, ale také offline SDK. Svoji platformu nazývají Human Analytics, tato platforma dokáže obličej najít, identifikovat, verifikovat a to i více obličejů na jednom obrázku. Navíc dokáže určit emoce: radost, překvapení, smutek, strach, zlost a znechucení, dále zvládá klasifikovat etnikum, věk, pohlaví, brýle a další. Tato společnost se zabývá pouze touto problematikou, proto je jejich řešení komplexní a kvalitní.[18]

### 4.4.1 Shrnutí výhod a nevýhod řešení

Výhody:

- velká škála emocí, které umí určit
- klasifikace věku a pohlaví
- společnost specializující se pouze na tuto oblast

Nevýhody:

- nutné stálé připojení k síti kvůli komunikaci API
- omezení transakcí při použití bezplatné verze
- nutná registrace k získání API klíče

## 4.5 Microsoft Azure

Microsoft Azure je cloudové řešení k hostování a běhu aplikací. Využívá data-centra po celém světě, v Evropě jsou dvě, jedno v Dublinu a druhé v Amsterdamu. Jeho velkou výhodou je, že podporuje mnoho programovacích jazyků a frameworků. Azure poskytuje velké množství služeb. Jednou z nich jsou Cognitive Services, tedy poznávací služby. Poskytuje inteligentní algoritmy k vidění, poslechu, mluvení, porozumění mluvenému slovu a jeho interpretaci. Jednou ze služeb k vidění je Face API. Umožňuje detekovat lidský obličej a porovnat ho s podobnými obličejí, sdružovat obličej do skupin podle jejich podobnosti a také identifikovat již uložený obličej. Detekce obličeje obsahuje: obdélník, který určí, v jakém místě se obličej nachází, dále pohlaví, věk, emoce, vlasy, úsměv, pozici hlavy, vousy, brýle a dokonce i make up. U emocí určí, v jaké míře jsou jednotlivé emoce zastoupeny ve škále 0-1. Škála emocí je také velká, konkrétně: vztek, pohrdání, znechucení, strach, spokojenost, neutralita, smutek, překvapení. Na webových stránkách je k dispozici rozsáhlá dokumentace včetně návodů jak začít. Podporuje jazyky C#, Java, JavaScript, PHP, Python a Ruby.[19]

### 4.5.1 Shrnutí výhod a nevýhod řešení

Výhody:

- široká škála emocí, které umí určit
- další velké množství parametrů včetně věku a pohlaví

- podpora mnoha programovacích jazyků (včetně Javy)

Nevýhody:

- nutné připojení sítě kvůli komunikaci s API
- registrace a klíč
- omezení na 30 000 transakcí za 20 minut

## 4.6 Výběr vhodné implementace

Po porovnání možných implementací přichází v úvahu řešení Kairos nebo Microsoft Azure. Ostatní řešení jsou nedostatečná (nepokrývají všechny potřebné atributy obličeje) nebo příliš složitá (především OpenCV kvůli nutnosti trénování vlastního modelu).

Tabulka 4.1: Srovnání Microsoft Azure a Kairos

název	emoce	registrace	dokumentace	odezva
MicroSoft Azure	ANO	ANO	4	5s
Kairos	ANO	ANO	2	10s

Vybrala jsem implementaci Microsoft Azure Face API. A to z důvodu, že má dobrou dokumentaci a ukázky práce s tímto API v Javě. Dalším důvodem je můj subjektivní názor, že Kairos je příliš komerční.

## 4.7 Face API

Zvolenou implementací rozpoznání obličeje je Face API od Microsoft Azure. K používání tohoto API je nutné se registrovat na stránkách MicroSoft Azure (<https://azure.microsoft.com/cs-cz/try/cognitive-services/>). Při registraci dostane uživatel klíč, díky kterému je potom možné toto API používat. Tento účet je zdarma, ale transakce jsou omezeny na 30 000 za 20 minut. Tento limit tuto práci nijak neomezí. Toto API zvládá jak detekci, tak rozpoznání obličeje. V případě detekce stačí zadat URL adresu obrázku nebo obrázek samotný jako bytové pole. Obličej je lokalizován obdélníkem a jsou určeny zvolené atributy (mezi nimi i věk, pohlaví a emoce).

Budu používat pouze detekci, ale pro úplnost uvedu, jaké základní věci u rozpoznání toto API dokáže.

- verifikovat obličej - verifikuje, zda dva obrázky obličeje jsou stejného člověka nebo podle jednoho obrázku obličeje určí, zda se jedná o určitou osobu

## 4. ANALÝZA EXISTUJÍCÍCH IMPLEMENTACÍ

---

- najít podobný obličej - podle cílového obrázku dokáže z několika obrázků najít ten nejpodobnější
- seskupovat obličeje - rozdělí obličeje do tříd podle podobnosti
- identifikovat osobu - na základě detekovaného obličeje a obličeje uloženého v databázi

Microsoft navíc poskytuje Face Storage, datábázi pro ukládání obrázků, tato služba je ale placená.

### 4.7.1 Omezení

Detekce obličeje akceptuje obrázky ve formátu jpg, png, bmp, gif a velikost je omezena na interval 1KB - 4MB. Aby byl obličej identifikován, musí mít velikost od 36x36 do 4096x4096 pixelů. Maximálně detekuje 64 obličejů na obrázku a to v pořadí od největšího po nejmenší.

### 4.7.2 Formát požadavku

Komunikace s API probíhá přes POST metodu v HTTP požadavku.

Požadavek se skládá z:

- URL
- parametrů
- hlavičky
- těla

**URL** požadavku má formát závislý na lokaci uživatele, každá část světa má svoji URL. Pro Evropu je to West Europe (západní) a North Europe (severní). Já jsem použila URL pro západní Evropu.

**Parametry** požadavku:

- id obličeje - vrátí ID detekovaného obličeje, vhodné pokud je např. obličej ukládán do databáze
- výrazné body obličeje (landmarks)
- atributy obličeje - zde lze specifikovat, které atributy jsou požadovány, aby byly vyhodnoceny a vráceny v odpovědi

V **hlavičce** se uvádí klíč, který byl získán při registraci. A dále se specifikuje, jaký obsah má tělo. V případě zadání URL adresy obrázku je tělo *application/json*, v případě binárních dat *application/octet-stream*.

V **těle** je potom poslán obrázek samotný. Lze uvést URL obrázku a nebo ho předat jako binární data.

Pokud nenastane žádná chyba, odpověď přijde v podobě JSON a měla by vypadat takto (v tomto případě bylo mezi atributy k vyhodnocení uvedeno pouze: emoce, úsměv, věk a pohlaví):

```
[{
  "faceRectangle": {
    "top": 46,
    "left": 382,
    "width": 515,
    "height": 515
  },
  "faceAttributes": {
    "emotion": {
      "contempt": 0.057,
      "surprise": 0,
      "happiness": 0,
      "neutral": 0.019,
      "sadness": 0.001,
      "disgust": 0.401,
      "anger": 0.524,
      "fear": 0
    },
    "gender": "female",
    "age": 30.3,
    "smile": 0
  }
}]
```

Atributy, které je možné určit jsou:

- věk - na který osoba na obrázku vypadá
- pohlaví
- úsměv - míra úsměvu v rozmezí od 0 do 1
- vousy - určí míru, v jaké se vousy vyskytují (a to na 3 místech: knír, brada, tváře)

#### 4. ANALÝZA EXISTUJÍCÍCH IMPLEMENTACÍ

---

- pozice hlavy
- brýle - žádné, na čtení, sluneční, plavecké
- emoce - v jaké míře se vyskytuje opovržení, neutrální, radost, smutek, překvapení, strach, vztek, znechucení
- vlasy - barva vlasů a zda jsou viditelné
- makeup - zda je oblast očí a úst upravena
- doplňky - brýle, pokrývka hlavy
- dále je možné určit míru rozmazání, expozice a šumu

Chyby, které mohou nastat při odeslání požadavku:

- nevalidní parametry nebo tělo požadavku
- neplatný klíč
- vyčerpání povolených transakcí
- timeout
- nevalidní typ obrázku
- překročení limitu transakcí za 20 min.

---

# NAOqi Framewrok

NAOqi je software, který běží uvnitř robota NAO a řídí ho. Framework NAOqi se používá k programování robota. Je multiplatformní, to znamená, že je možné vyvíjet na kterémkoliv operačním systému. Je schopný monitorovat všechny funkce API (instrospection). Podporuje primárně jazyky Python a C++, ale má SDK pro Javu, .Net a Matlab. Přidávat nové moduly lze pouze v C++ a Pythonu. Volání metod však díky API funguje ve všech jazycích stejně.

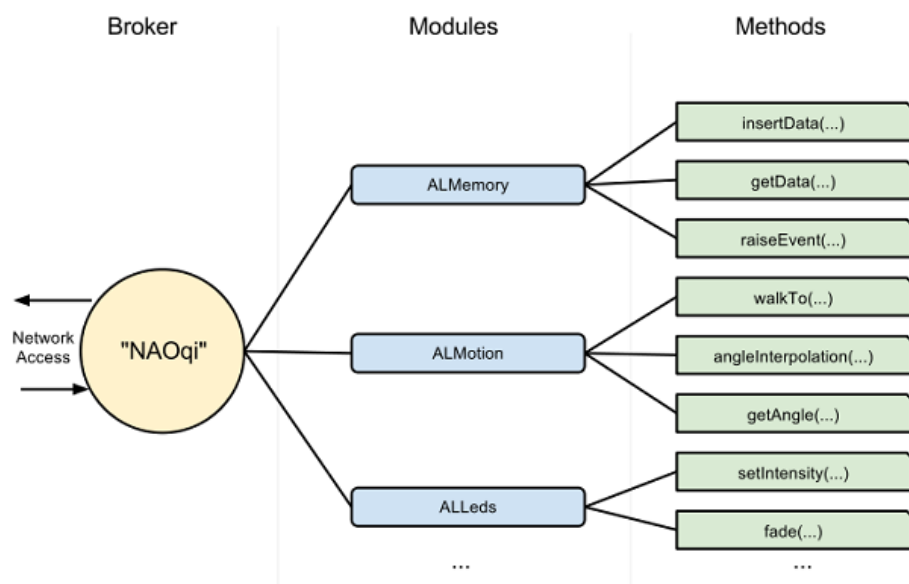
Pro jazyk Java existuje JNAOqi, což je opensource projekt, který umožňuje volat všechny moduly z již zmíněného NAOqi a to v Javě. Toho jsem se rozhodla využít pro svou práci. Komunikace JNAOqi a NAOqi probíhá pomocí ALProxy objektů. NAO v sobě nemá vlastní JVM, je tedy nutné, aby aplikace běžela na PC a s robotem bude komunikovat přes síť (Ethernet nebo Wifi).

## 5.1 Jak funguje NAOqi

NAO má v sobě NAOqi executable, který je tzv. broker (zprostředkovatel). Při spuštění je zavolán soubor autoload.ini, který definuje knihovny k načtení. Každá knihovna obsahuje moduly, které zprostředkovatel používá. Zprostředkovatel poskytuje adresářové služby, tedy říká, kde se moduly a jejich metody nachází. A dále poskytuje síťové připojení, tím umožňuje metodám, aby byly volány mimo proces.

Dalším objektem je proxy. Proxy vždy reprezentuje nějaký modul a podle něho se také chová a obsahuje metody modulu. Modul je třída v knihovně. Když je knihovna načtena z autoload.ini, tak je třída modulu inicializována. Některé moduly vystavují události. To znamená, že z jiného modulu je možné událost odebírat a vytvořit callback, který bude metodou modulu odebírajícího událost.

Objekt ALMemory je paměť robota. Všechny moduly do ní mohou zapisovat nebo z ní číst, odebírat události, to znamená, že budou volány, pokud



Obrázek 5.1: Schéma načítání modulů a jejich metod (převzato z dokumentace [2])

bude vyvolána událost. Obsahuje data ze senzorů a událostí.[2]

## 5.2 NAOqi API

V této sekci představím moduly (třídy), které jsou pro tuto práci relevantní. V JNAOqi jsou tyto třídy k dispozici přes proxy.

### ALMemory

Z tohoto modulu použijte metodu *subscribeOnEvent*, která začne odebírat událost, která je mu zadána jako parametr. Automaticky nahraje modul, který tuto událost vystavuje. Je nutné definovat *callBack* metodu, tedy co se má stát, když je tato událost vyvolána. Tyto metody použijte k odběru událostí, které vyvolají začátek komunikace robota, především událost vyvolaná na dotyk hlavy.

### ALTextToSpeech

Tento modul zajišťuje mluvení robota. Metoda *say(String stringToSay)* se postará o to, aby robot řekl text zadaný jako parametr. Dále je dobré ještě před tím, než bude robot mluvit, nastavit jazyk (podporuje i češtinu) a to metodou *setLanguage(String language)*. Tento modul použijte pro mluvení robota na základě zjištěných informací o člověku.



### **ALTouch**

Kontroluje dotyky na senzorech (rukách, nohách, hlavě a uprostřed trupu). Umí vyvolat události na dotyk jednotlivých senzorů. Použijí událost *FrontTactilTouched*, která je vyvolána při dotyku hlavy robota, tímto je zahájena komunikace s robotem.

### **ALVideoDevice**

Modul zajišťuje pořizování obrázků z videokamer robota a dokáže také nahrát video. Metodou *subscribeCamera* se připojí ke kameře a metodou *getImageRemote*, případně *getImageLocal*, pokud je modul lokální, udělá fotografii. Když už kamera není potřeba, pomocí *unsubscribeCamera* se modul odpojí. V této aplikaci použijí kameru pro vytvoření fotografie člověka před robotem. Tuto fotografii dále použijí pro klasifikaci emocí, věku a pohlaví.

Tyto dva moduly jsem nakonec nepoužila, v případě rozšiřování aplikace je ale jistě vhodné je použít.

### **ALFaceDected**

Modul, který se stará o to, že NAO reaguje na obličej před sebou. Vystavuje událost *FaceDected*, na kterou lze patřičným způsobem reagovat a která vrací souřadnice obličeje a další údaje.

### **ALPeoplePerception**

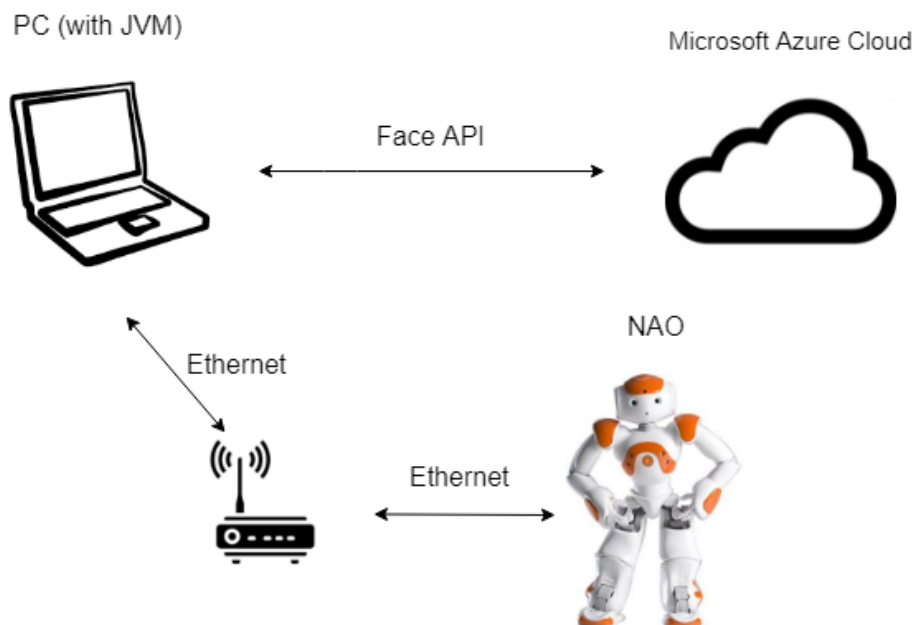
Modul, který si uchovává informace o lidech kolem robota a sleduje je.



## Realizace

Cílem praktické části této práce je vytvořit aplikaci, která integruje rozpoznání věku, emocí a pohlaví do robota NAO. NAO tak bude reagovat na obličeje před sebou, podle atributů, které o nich zjistí.

Samotné rozpoznání je realizováno přes Face API, blíže popsané v kapitole 4. Komunikace s robotem bude probíhat přes framework NAOqi, tento je popsán v předešlé kapitole. Aplikace je napsána v Javě, a protože NAO v sobě nemá JVM, bude připojen k počítači, ve kterém aplikace poběží. Navíc vývoj probíhal na OS Windows, takže spojení počítače s robotem je nutné provést přes router, robot má totiž OS Linux. Schéma zapojení na obrázku 6.1.



Obrázek 6.1: Schéma komunikace s Face API a robotem

## 6.1 Vývoj aplikace

Přesto, že primárními jazyky pro práci s robotem jsou Python a C++, vybrala jsem si Javu. Důvodem je lepší práce s pamětí a moje větší zkušenosti s tímto jazykem. Tento jazyk je společností Aldebaran podporován a to díky projektu JNAOqi popsaného v kapitole 5. Vývoj probíhal v IntelliJ IDEA, prostředí pro vývoj aplikací v Javě od společnosti Jet Brains. Návod na spuštění tedy bude v tomto prostředí. Bylo nutné stáhnout SDK pro verzi 32-bit, protože nativní knihovna pro robota NAO je na OS Windows kompatibilní pouze s 32-bit verzí Javy.

## 6.2 Popis aplikace

Aplikace bude z následujících navzájem komunikujících modulů (znázorněny na obr. 6.2):

- FaceDetect - komunikuje s Face API a určí atributy obrázku
- ReactToEvent - hlavní třída, reaguje na obličej před kamerou, poté zavolá detekci a mluvení robota
- NAO Speak - komunikace robota podle zjištěných atributů

### 6.2.1 ReactToEvent třída

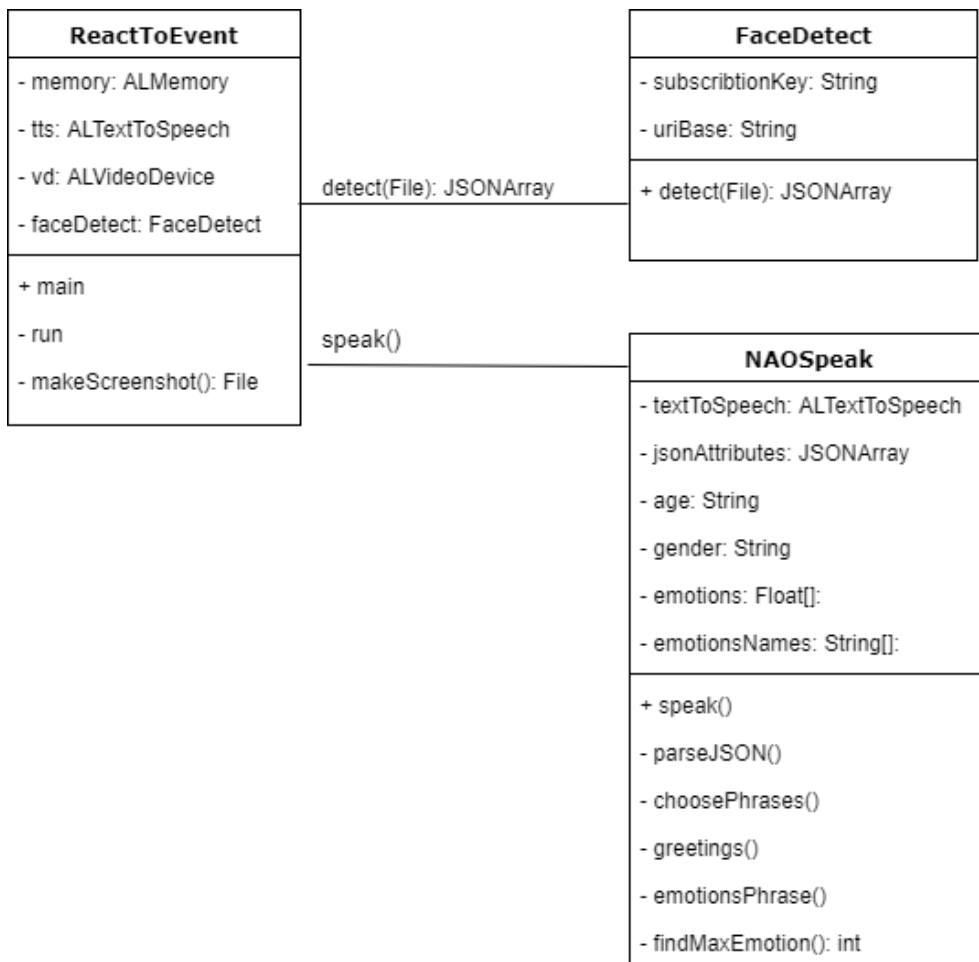
Tato třída se stará o chod celé aplikace. Má tři metody: spustitelnou *main* a dále *run* a *makeScreenshot*.

#### **main**

Touto metodou lze celou aplikaci spustit. Nejdříve naváže spojení s robotem a zahájí session.

#### **run**

Zde si nainicializuje moduly *ALMemory*, *ALTextToSpeech*, *ALVideoDevice*. *ALMemory* použije k odebírání událostí, *ALTextToSpeech* použije k mluvení robota a *ALVideoDevice* k pořízení snímku z kamery. Po inicializaci těchto modulů začne odebírat událost *FrontTactilTouched* (dotek hlavy). V případě vyvolání této události je použit callback. Zde se zavolá metoda *makeScreenshot*, ze které získá soubor s fotografií. Dále je tento soubor předán instancí třídy *FaceDetect*, konkrétně její metodě *detect*. Tato zpět předá JSON pole se zjištěnými atributy obličeje. Pokud byl před robotem zjištěn obličej, je vytvořena instance třídy *NAOSpeak* a zavolána její metoda *speak*, díky které NAO osloví člověka před sebou.



Obrázek 6.2: Diagram tříd a jejich komunikace

**makeScreenshot**

Pomocí modulu `ALVideoDevice` začne odebírat kameru robota a pořídí z ní snímek. Z toho snímku potom vytvoří soubor, který je dále použit ke klasifikaci atributů obličeje.

**6.2.2 FaceDetect třída**

Tato třída se stará o komunikaci s Face API. Má dva důležité atributy: `subscriptionKey` a `uriBase`. První atribut je klíč, který je přidělen k používanému účtu u Microsoft Azure. Druhý atribut je adresa, kam bude poslán HTTP požadavek. Tato třída má pouze jednu metodu `detect`.

**detect**

Pro komunikaci je použit `HttpClient` a `HttpPost`. V požadavku, který má

formu `HttpPost`, se specifikuje, jaké atributy požadujeme. V mém případě jsem zadala věk, pohlaví a emoce. Dále je nastavena hlavička požadavku. A to jakého typu je obrázek a klíč. Já předávám obrázek jako binární data, proto do hlavičky zadám *application/octet-stream*. A jako poslední věc je do požadavku přidán obrázek. Tatovýho požadavek je poté odeslán pomocí `HttpClienta`. Odpověď je přijata jako `HttpResponse`, ve které jsou určené atributy obličeje ve formátu JSON. Tento JSON je vrácen jako výsledek této metody. Pokud na fotografii nebyl identifikován žádný obličej, je vrácen `null`.

### 6.2.3 NAOSpeak třída

Tato třída zajišťuje komunikaci robota s člověkem před ním na základě zjištěných informací.

#### **speak**

Jediná veřejná metoda této třídy. Nejdříve volá *parseJSON* ke zpracování zjištěných atributů a poté *choosePhrases* k samotné komunikaci.

#### **parseJSON**

Z přijatého JSON souboru získá věk, pohlaví a zastoupení jednotlivých emocí.

#### **choosePhrases**

Nastaví jazyk na češtinu. Zavolá metodu **greetings**, která podle věku a pohlaví vybere vhodný pozdrav. A dále metodu **emotionsPhrase**, která použije hlášku podle toho, která emoce převažuje.

## 6.3 Návrhy na zlepšení

Hlavní nevýhodou aplikace je, že je nutné k začátku komunikace dotknout se hlavy robota. Původní záměr byl, aby robot sám reagoval na obličej před sebou, ale tato funkcionality v Javě bohužel zatím není dostupná. Proto toto je můj hlavní návrh na zlepšení.

Další věcí je komunikace robota. Bylo by dobré nějakým lepším způsobem využít jeho získané znalosti o člověku před ním.

## 6.4 Instalační příručka

### 6.4.1 Vývojové prostředí a JRE

V IntelliJ IDEA je nutné projektu nastavit JRE. Jak již bylo zmíněno, nativní knihovna je kompatibilní pouze s 32-bit verzí Javy, je tedy nutné nastavit JRE pro Windows x86. JRE je ke stažení na oficiálních stránkách Oracle (<http://www.oracle.com/technetwork/java/javase/downloads/>). Toto stažené JRE potom v IDEI nastavíme v Run > Edit Configurations a do sekce JRE.

### 6.4.2 Knihovna JNAOqi

Na stránkách <https://community.aldebaran-robotics.com/> je třeba si založit účet a v sekci Resources > Software si podle platformy stáhnout příslušnou verzi JNAOqi.

### 6.4.3 Knihovny pro Face API

Pro Face API je nutné si stáhnout následující knihovny z Maven Repository (<https://mvnrepository.com/>):

**org.apache.httpcomponents:httpclient:4.2.4**

**org.json:json:20170516**

**org.apache.httpcomponents:httpcore:4.2.4**

**commons-logging:1.2**

### 6.4.4 Import knihoven v IntelliJ IDEA

Všechny stažené knihovny je třeba importovat do projektu. V sekci File > Project Structure > Libraries lze tak učinit.

### 6.4.5 Git repozitář

Celý projekt jsem umístila na gitlab na adresu <https://gitlab.fit.cvut.cz/vondreli/bc-code/>. Odtud je možné si repozitář clonovat a spustit. Adresářová struktura je připravena na spuštění v IntelliJ IDEA.

## 6.5 Uživatelská příručka

Po importu projektu (clonování repozitáře) a stažení všech potřebných knihoven je aplikace připravena ke spuštění.

Samozřejmě je nutné mít u sebe robota NAO. Dále pro připojení k počítači jsou vyžadovány dva síťové kabely a router. Zapojení je znázorněno na obrázku 6.1.

Nyní by měl být robot připojen k počítači. Lze otestovat připojení pomocí příkazové řádky na vašem počítači. A to pomocí příkazu:

```
$ ping IP_ADRESA_ROBOTA
```

IP adresa robota by měla být 192.168.0.101 (případně lze robota požádat o sdělení IP adresy). V *main* ve třídě *ReactToEvent* je nastavena IP adresa v proměnné *robotUrl*. Po spuštění *main* je pro zahájení komunikace nutný dotyk robota na hlavě. Po tomto robot osloví člověka před sebou.

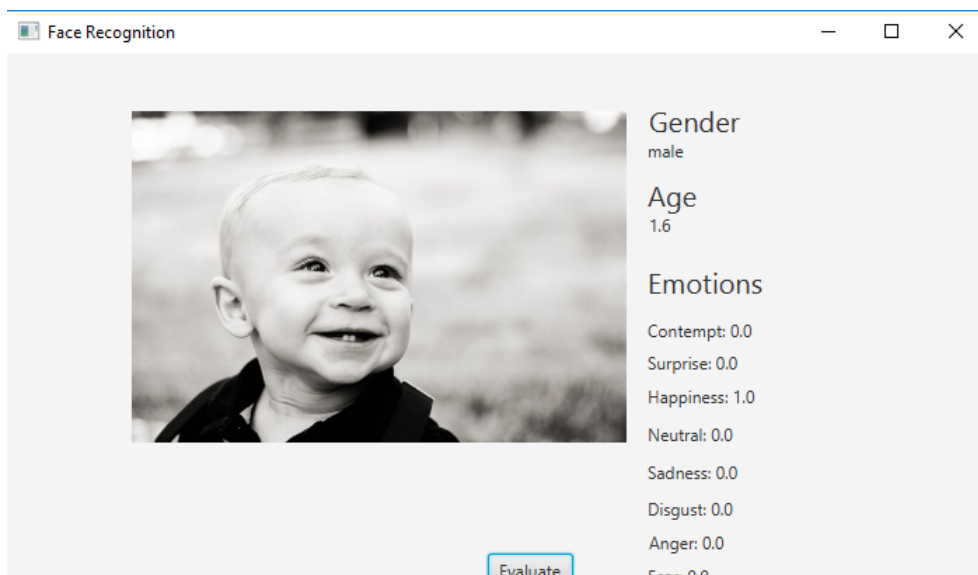




## Testování

Testování probíhalo ve dvou fázích. Nejdříve jsem testovala samotné Face API a poté robota i s reálnými uživateli.

Na testování samotného Face API jsem vytvořila jednoduchou JavaFX aplikaci s GUI, které přehledně zobrazí zjištěné atributy spolu s klasifikovaným obrázkem obličeje (obr. 7.1 Ukázka testovacího GUI)



Obrázek 7.1: Ukázka testovacího GUI

### 7.1 Testování API samostatně

V tomto testování jsem použila obrázky obličejů stažené z internetu. Zaměřila jsem se především na emoce, protože to je nejsložitější úkol pro API. Také jsem vybrala různé věkové kategorie.

Výsledky testů:

Tabulka 7.1: Úspěšnost určování jednotlivých atributů

atribut	úspěšnost
pohlaví	100%
věk	88%
opovržení	10%
překvapení	60%
šťěstí	100%
neutrální	100%
smutek	100%
znechucení	20%
hněv	90%
strach	80%

Pohlaví bylo určeno vždy správně. Věk byl řádově určen správně. Při projevení silných emocí se pravděpodobně kvůli více vráskám v obličejí zvyšuje odhadovaný věk. Například dívka, které je 10 let, ale její výraz byl kvalifikován jako hněv a znechucení, byl odhadnut věk na 20.

Kvalifikování emocí bylo velmi dobré v případě štěstí, neutrální, smutku, hněvu a strachu. Opovržení a znechucení se nepovedlo plně nasimulovat ani jednou.

### 7.2 Testování s robotem

Druhá část testování probíhala již s robotem. Scénář byl takový, že člověk si sedl před robota, pohládl ho po hlavě a zatvářil se mnou požadovaným způsobem. Využila jsem 3 testovací osoby, které se pokusily nasimulovat zkoumané emoce.

Popis osob:

U1 - muž, 21 let

U2 - žena, 21 let

U3 - žena, 21 let

Výsledky byly následující:

Tabulka 7.2: Úspěšnost určování jednotlivých atributů

atribut	U1	U2	U3
pohlaví	100%	100%	100%
věk	70%	90%	90%
opovržení	5%	30%	10%
překvapení	30%	80%	60%
šťěstí	100%	100%	100%
neutrální	70%	95%	90%
smutek	15%	100%	60%
znechucení	5%	10%	10%
hněv	10%	30%	35%
strach	5%	25%	20%

Celková úspěšnost je horší a to z důvodu menší kvality fotek pořízených robotem v porovnání s fotkami z internetu. Pohlaví bylo opět určeno vždy správně. Věk vždy o něco vyšší než reálně. Emoce štěstí, smutek a neutralita byly určeny s téměř 100% úspěšností. Nejhorší naopak opět opovržení. Face API hodnotím jako velmi povedlé, určení pohlaví a věku je velmi přesné a emoce jsou odhadovány ve vysoké míře správně.



---

## Závěr

Prvním z cílů této bakalářské práce bylo udělat rešerši metod rozpoznání obličeje. Existuje spousta odborných článků o jednotlivých metodách a jejich výzkumech v angličtině. Proto jsem udělala souhrn těchto metod a jejich hlavních myšlenek a to včetně metod využívajících neuronové sítě.

Dalším cílem bylo najít existující implementace pro rozpoznání pohlaví, emocí a věku podle obličeje. Zjistila jsem, že pro pokrytí všech těchto atributů velmi mnoho řešení není. Udělala jsem shrnutí řešení, která se tímto tématem zabývají a vybrala jsem to, které nejvíce vyhovovalo požadavkům. Vybraným řešením je Face API od Microsoft Azure.

Dále jsem se seznámila s frameworkem pro práci s robotem NAO, který se nazývá NAOqi framework. V první fázi jsem řešila připojení k robotovi a pořízení snímku z kamery. V další fázi jsem se zaměřila na to, jak bude robot reagovat na lidi před ním. Nakonec jsem zvolila, že bude reagovat na dotyk hlavy. V poslední fázi jsem integrovala rozpoznání Face API do reakce na dotyk hlavy a na závěr provedla potřebné testování.

Testování proběhlo ve dvou fázích. První testování bylo po výběru Face API jako řešení problému rozpoznávání obličeje. Ukázalo se, že API je velmi kvalitní a mohu ho integrovat do robota. Druhá fáze testování přišla po integraci rozpoznání obličeje do robota. Pozvala jsem 3 osoby, které nasimulovaly požadované emoce. Výsledek byl podobný jako v první fázi, projevil se pouze zvýšení věku a to z důvodu horší kvality snímků.

Do budoucna by bylo dobré upravit aplikaci tak, aby nebyl nutný dotek hlavy, ale aby robot hned reagoval na obličej před sebou. Dále je možné lepší využití získaných informací o člověku, který stojí před robotem, k lepší komunikaci mezi nimi.

Pro mně je cenná zkušenost práce s robotem, protože jsem nikdy před tím neměla šanci si něco podobného vyzkoušet. Myslím si, že robotika má velkou budoucnost a tato zkušenost se mi bude hodit.



---

## Literatura

- [1] Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning. *Medium [online]*, červenec 2016, [cit. 2018-03-38]. Dostupné z: <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>
- [2] SoftBank Robotics: *NAOqi Documentation [online]*. [cit. 2018-03-28]. Dostupné z: [http://doc.aldebaran.com/2-4/index\\_dev\\_guide.html](http://doc.aldebaran.com/2-4/index_dev_guide.html)
- [3] SoftBank Robotics: *NAO Documentation [online]*. [cit. 2018-03-28]. Dostupné z: [http://doc.aldebaran.com/2-4/home\\_ao.html](http://doc.aldebaran.com/2-4/home_ao.html)
- [4] M. Turk, A. P.: *Eigenfaces for recognition*. Journal of Cognitive Neuroscience, 1991. Dostupné z: <http://www.face-rec.org/algorithms/PCA/jcn.pdf>
- [5] C. Liu, H. W.: *Comparative Assessment of Independent Component Analysis (ICA) for Face Recognition*. Department of Computer Science, George Mason University, 1999. Dostupné z: <http://www.face-rec.org/algorithms/ICA/liu99comparative.pdf>
- [6] K. Etemad, R. C.: *Discriminant Analysis for Recognition of Human Face Images*. Journal of the Optical Society of America A, 1997. Dostupné z: <http://www.face-rec.org/algorithms/LDA/discriminant-analysis-for-recognition.pdf>
- [7] C. Liu, H. W.: *Face Recognition Using Evolutionary Pursuit*. Department of Computer Science, George Mason University, 1998. Dostupné z: <http://www.face-rec.org/algorithms/EP/liu98face.pdf>
- [8] L. Wiskott, N. K. C. v. d. M., J.-M. Fellous: *Face Recognition by Elastic Bunch Graph Matching*. CRC Press, 1999. Dostupné z: <http://www.face-rec.org/algorithms/EBGM/WisFelKrue99-FaceRecognition-JainBook.pdf>

- [9] Srisuk, S.; Kurutach, W.: *Face Recognition using a New Texture Representation of Face Images*. Proceedings of Electrical Engineering Conference, 2003. Dostupné z: <http://www.face-rec.org/algorithms/Trace/eecon03.pdf>
- [10] J. Huang, V. B., B. Heisele: *Component-based Face Recognition with 3D Morphable Models*. 4th International Conference on Audio- and Video-Based Biometric Person Authentication, 2003. Dostupné z: [http://www.face-rec.org/algorithms/3D\\_Morph/avbpa2003.pdf](http://www.face-rec.org/algorithms/3D_Morph/avbpa2003.pdf)
- [11] B. Moghaddam, A. P., T. Jebara: *Bayesian Face Recognition, Pattern Recognition*. Mitsubishi Electric Research Laboratories, 2003. Dostupné z: <http://www.face-rec.org/algorithms/Bayes/TR2000-42.pdf>
- [12] P. Viola, M. J.: *Robust Real-Time Face Detection*. International Journal of Computer Vision, 2004. Dostupné z: <http://www.face-rec.org/algorithms/Boosting-Ensemble/16981346.pdf>
- [13] Skyminid team Chris V. Nicholson: *Introduction to Deep Neural Networks (Deep Learning) - Deeplearning4j: Open-source, Distributed Deep Learning for the JVM*. [cit. 2018-04-15]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.440.9695&rep=rep1&type=pdf#page=788>
- [14] M.Nandini, G. S., P.Bhargavi: *Face Recognition Using Neural Networks*. Department of EConE, Sree Vidyanikethan Engineering College, 2013. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.440.9695&rep=rep1&type=pdf#page=788>
- [15] OpenCV: *OpenCV library [online]*. [cit. 2018-04-15]. Dostupné z: <https://opencv.org/>
- [16] Open Face: *Open Face API Documentation [online]*. [cit. 2018-04-15]. Dostupné z: <http://openface-api.readthedocs.io/en/latest/>
- [17] Google: *Cloud Vision API [online]*. [cit. 2018-04-15]. Dostupné z: <https://cloud.google.com/vision/>
- [18] Kairos: *Kairos [online]*. [cit. 2018-04-15]. Dostupné z: <https://www.kairos.com/>
- [19] Microsoft: *Microsoft Azure Cognitive Services [online]*. [cit. 2018-04-15]. Dostupné z: <https://azure.microsoft.com/en-us/overview/what-is-azure/>



## Seznam použitých zkratk

**GUI** Graphical user interface

**HTTP** HyperText Transport Protokol

**JSON** JavaScript Object Notation

**API** Application Programming Interface

**SDK** Software Development Kit

**URL** Uniform Resource Locator



---

## Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
├─ impl.....	zdrojové kódy implementace
├─ thesis.....	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
├─ reactFaceDetect.zip.....	projekt prostředí IntelliJ IDEA
text.....	text práce
├─ BP_Vondrackova_Eliska_2018.pdf.....	text práce ve formátu PDF