GYMNÁZIUM JÍROVCOVA 8

MATURITNÍ PRÁCE

Rozpoznávání obličeje pomocí strojového učení

Lukáš Lacina
vedoucí práce: Dr. rer. nat. Michal Kočer

V Českých Budějovicích 14. února 2025 **2024/2025**

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně s vyznačením všech použitých pramenů.

V Českých Budějovicích dne ..... podpis .....

Lukáš Lacina

Abstrakt

Klíčová slova

Poděkování

Obsah

I Rozpoznávání obličeje a strojové učení **2**

1 Úvod do rozpoznávání obličeje **3**

1.1 Definice technologie rozpoznávání obličeje 3

1.2 Způsob fungování 3

1.3 Význam a využití 4

1.4 Zneužití technologie rozpoznávání obličeje 4

2 Historie a vývoj rozpoznávání obličeje **6**

2.1 Počátky technologie rozpoznávání obličeje 6

2.2 Automatizace a První algoritmy 6

2.2.1 Metoda Eigenfaces a PCA 6

2.2.2 Projekt FERET 7

2.3 Strojové učení a jeho vliv v technologii rozpoznávání obličeje 7

2.3.1 Hluboké učení a konvoluční neuronové sítě 8

2.3.2 CNN modely 8

2.4 iPhone X a Face ID 8

2.5 Technologie rozpoznávání obličeje dnes 9

3 Umělá inteligence a strojové učení **10**

3.1 Umělá inteligence 10

3.2 Strojové učení 10

3.2.1 Učení s učitelem (Supervised learning) 11

3.2.2 Učení bez učitele (Unsupervised learning) 11

3.2.3 Učení posilováním (Reinforcement learning) 11

4 Předzpracování dat a datasetu **13**

4.1 Předzpracování dat pro modely strojového učení 13

4.1.1 Detekce obličeje 13

4.1.2 Změna velikosti a normalizace dat 14

4.1.3 Augmentace dat 15

4.2 Vstupní data pro inferenci 15

4.3 Dataset 15

4.3.1 Datasety pro identifikaci obličejů 16

4.3.2 Rozdělení na sady 16

5 Algoritmy pro rozpoznávání obličeje **18**

5.1 Fungování algoritmu 18

5.2 Algoritmy strojového učení 18

5.3 Rozpoznávání pomocí konvolučních neuronových sítí (CNN) 19

5.3.1 Základy neuronových sítí 19

5.3.2 Konvoluční neuronová síť 20

5.3.3 Konvoluční a pooling vrstva 21

5.3.4 Plně propojené vrstvy 22

5.3.5 Konvoluční neuronové sítě a rozpoznávání obličeje 22

5.4 Hodnocení výkonu algoritmu 23

5.4.1 Metriky pro hodnocení výkonu 23

5.4.2 Optimalizace hyperparametrů 23

6 Problematika **25**

6.1 Ochrana soukromí 25

6.2 Diskriminace a zaujatost 26

II Implementace programu pro rozpoznávání obličeje **27**

7 Cíl práce a použité technologie **28**

8 Struktura programu **29**

9 Předzpracování dat **30**

9.1 Inicializace třídy `DataPreprocessing` 30

9.2 Detekce obličejů . . . 30
9.3 Ořezávání obličejových oblastí . . . 31
9.4 Změna velikosti a normalizace . . . 32
9.5 Kompletní předzpracování . . . 33
9.6 Předzpracování datasetu . . . 33

10 Vytvoření vlastního modelu a jeho trénování 34
10.1 Architektura modelu . . . 34
10.2 Ztrátová funkce triplet loss . . . 35
10.3 Generování trénovacích dat . . . 36
10.4 Trénování modelu . . . 37
10.5 Uložení modelu . . . 38

11 Funkce rozpoznávání a databáze 39
11.1 Načtení modelu a generování embeddingů . . . 39
11.1.1 Načtení modelu . . . 39
11.1.2 Generování embeddingů . . . 40
11.2 Správa databází . . . 41
11.2.1 Vytvoření databáze a FAISS indexu . . . 41
11.2.2 Přidání osoby do databáze . . . 42
11.3 Porovnávání osob . . . 43
11.3.1 Vyhledávání ve FAISS indexu . . . 43
11.3.2 Získání jména na základě indexu . . . 43
11.3.3 Identifikace osob . . . 44

12 Uživatelské rozhraní 46

Bibliografie 51

Zkratky 52

Přílohy 55

A Fotky z pokusů 56

B Příloha další 57

Úvod

Rozpoznávání obličejů se v dnešní době stalo běžnou součástí každodenního života. Tato technologie, která dokáže analyzovat a identifikovat lidské tváře z digitálních obrazů, nachází využití v mnoha oblastech – od zabezpečení a autentizace až po pokročilé systémy sledování a analýzy.

Základním principem rozpoznávání obličejů je schopnost algoritmů strojového učení zpracovávat obrazová data a vytvářet unikátní otisky obličejů, které lze porovnávat s databázemi známých identit. Tato technologie se vyvíjí již několik desetiletí, přičemž moderní metody, jako jsou konvoluční neuronové sítě (CNN), přinesly zásadní pokrok v přesnosti a rychlosti rozpoznávání.

V této práci popisuji vývoj a současný stav technologie rozpoznávání obličejů. Zabývám se jak teoretickými základy a historickým kontextem, tak praktickým využitím algoritmů. Praktická část je věnována vytvoření modelu pro rozpoznávání obličejů a jeho aplikaci v reálných podmínkách.

Část I

Rozpoznávání obličeje a strojové učení

1 Úvod do rozpoznávání obličeje

1.1 Definice technologie rozpoznávání obličeje

Technologie rozpoznávání obličeje je technologií, která dokáže detekovat a extrahovat lidskou tvář z digitálního obrazu a poté porovnat tuto tvář s databází předem identifikovaných tváří. Tato technologie se obecně rozděluje na 3 formy dle její funkce: [1, 26]

- **One-to-One Identification**: Tato forma rozpoznávání obličeje porovnává obličej jedné osoby s předem identifikovanou tváří v databázi. Nejčastěji se využívá pro autentizaci uživatelů, například při odemykání mobilních zařízení. [26]
- **One-to-Many Identification**: Tato forma umožňuje technologie identifikovat konkrétní osobu mezi mnoha dalšími tím, že porovná obličej s rozsáhlou databází identit. Používá se zejména k masovému sledování. [26]
- **Emotion and Demographic Recognition**: Tato forma zpracování obličeje je nejpokročilejší a zaměřuje se na analýzu rysů obličeje tak, aby následovně odhadla demografické charakteristiky, jako je věk, pohlaví nebo emoční stav konkrétní osoby. [26]

1.2 Způsob fungování

Rozpoznávání obličeje je složitý proces, který zahrnuje několik klíčových kroků, které lze shrnout následovně: [1, 19]

1. **Shromažďování dat:** Prvním krokem je shromáždění dat z digitálních snímků. Tyto snímky mohou pocházet z různých zdrojů, jako jsou fotografie, soukromá videa nebo bezpečnostní kamery. [1, 19]

2. **Detekce obličeje:** V této fázi dochází za pomoci složitých algoritmů k identifikaci a lokalizaci obličeje v obraze. Tento krok zahrnuje analýzu obrazu za účelem určení oblastí, kde se nacházejí obličeje. [1, 19]

3. **Extrakce rysů:** Jakmile je obličej detekován, následuje extrakce klíčových rysů obličeje, jako jsou vzdálenost mezi očima, tvar lícních kostí a délka čelisti. Tyto rysy se dále převádějí do matematické reprezentace, která umožňuje jejich snadnější analýzu a porovnání. [1, 19]

4. **Porovnání:** Extrahované rysy se poté porovnávají s uloženými daty v databázi obličejů. Tato fáze zahrnuje hodnocení podobnosti mezi jednotlivými obličeji, což je klíčové pro určení identity osob. [1, 19]

1.3 Význam a využití

Význam technologie rozpoznávání obličeje je v dnešní době větší, než si většina lidí uvědomuje. Nejčastěji se využívá **v bezpečnosti**, kdy tato technologie umí ověřovat identitu jednotlivců, například při přihlašování k různým službám či zařízením. Také však hledá pohřešované osoby nebo odhaluje osoby podezřelé. Newyorská policie například uvedla, že díky technologii rozpoznání obličeje dokázala chytit pachatele do 24 hodin od útoku. Bezpečnost je hlavní důvod, proč se technologie rozpoznávání obličeje stává populární, avšak poskytuje mnohem více přínosů. [23]

Rozpoznávání obličeje se používá **v marketingu** pro zefektivnění různých procesů. Například v obchodě Amazon Go stačí zákazníkovi k nákupu pouze projít obchodem, vzít si zboží a odejít. Dále se používá **v sociálních médiích**, kde jsou platformy schopny automaticky označit uživatele na fotografiích. V posledních pár letech se však tato technologie objevuje i **ve zdravotnictví**, kde je schopna sledovat zdravotní stav pacientů a tím zvýšit efektivitu poskytování zdravotní péče. [23]

1.4 Zneužití technologie rozpoznávání obličeje

Ačkoli má technologie rozpoznávání obličeje mnoho přínosů, může být také zneužita. Jeden z nejzásadnějších problémů je **nelegální sledování osob**. Například některé autoritářské

režimy využívají tuto technologii k monitorování a potlačování opozičních skupin, což je narušení základních lidských práv a svobod. [10]

Dalším velkým problémem jsou **kybernetické útoky**, kdy hackeři využívají data o obličeji, aby mohli provádět různé podvody, například odemykání zařízení nebo bankovních účtů. Jelikož změnit biometrické údaje je v podstatě nemožné, jejich odcizení může mít závažné následky. Tyto útoky často vedou k ztrátě finančních prostředků nebo ke ztrátě soukromí. [10]

Z těchto důvodů jsou technologie rozpoznávání obličeje regulovány tak, aby se co nejvíce zamezilo zmíněným problémům. Zavádí se právní rámce a etické standardy pro jejich používání. Nicméně bez těchto regulací může tato technologie způsobit více škody než užitku. [23]

2 Historie a vývoj rozpoznávání obličeje

2.1 Počátky technologie rozpoznávání obličeje

Snaha o vytvoření technologie rozpoznávání obličeje sahá již do roku **1964**, kdy americký vědec **Woodrow Wilson Bledsoe** přišel s myšlenkou vytvořit stroj, který bude rozpoznávat lidské tváře. A tak Bledsoe společně s Helen Chan Wolfovou a Charlesem Bissonem **vytvořil poloautomatickou metodu na rozpoznávání obličeje**. Bledsoeova metoda rozpoznávala rysy manuálně, vědci museli identifikovat celkově 20 různých měřítek, jako je vzdálenost mezi očima, délka nosu nebo šířka rtů. Tyto parametry byly poté využity k vytvoření vektorové reprezentace obličeje, kterou počítač porovnával. V roce **1977** byl systém vylepšen o dalších 21 parametrů pro zlepšení přesnosti. [8, 30]

Bledsoeova metoda byla sice průlomová, avšak měla spousty výrazných omezení jako její **nízkou rychlost či častou nepřesnost**. Nepřesnost byla způsobována variabilitou osvětlení, úhly pohledu a individuálními rysy obličejů, což omezovalo její praktické užití. [8, 30]

2.2 Automatizace a První algoritmy

2.2.1 Metoda Eigenfaces a PCA

V roce **1988** se začala uplatňovat statistická metoda **PCA (Principal component analysis)** v oblasti počítačového vidění. Tato metoda byla schopna redukovat potřebné množství dat a extrahovat klíčové rysy v obličeji, které do této doby určovali vědci manuálně. [1, 8]

Díky metodě PCA vyvinul **Matthew Turk a Alex Pentland** v roce **1991** novou revoluční metodu rozpoznávání obličeje - **metodu Eigenfaces**. Tato metoda byla schopna detekovat obličeje na snímcích, což vedlo k **první automatické technologii rozpoznávání obličeje**. Tato metoda byla jednoduchá a měla relativně vysokou přesnost, což vedlo

k jejímu rychlému rozšíření. Nicméně tomuto průlomu bránily technologické a environmentální faktory, jako například omezený výpočetní výkon a kvalita databází, ze kterých se algoritmus učil. [2, 8]

2.2.2 Projekt FERET

V roce **1993** přišla agentura **DARPA** (Defense Advanced Research Projects Agency) s programem **FERET** (Facial Recognition Technology), který měl za cíl vyvinout **rozsáhlou standardizovanou databázi obličejů**. Tento projekt vznikl jako reakce na rostoucí zájem o technologie rozpoznávání obličeje a jejich aplikace v bezpečnosti a identifikaci. [7, 8, 31]

Databáze FERET obsahovala přes **14 000 snímků obličejů**, které byly foceny **v rozdílných podmínkách**, včetně osvětlení, úhlů a výrazů. Tato rozmanitá databáze umožnila testování a trénování různých algoritmů rozpoznávání obličejů na kvalitním datasetu, který je klíčový pro spolehlivou funkčnost algoritmu. Projekt FERET také přinesl **standardizované metodologie pro hodnocení výkonu algoritmů**, což umožnilo porovnávat efektivitu různých přístupů. [7, 31]

Díky tomuto projektu se zlepšila efektivita a přesnost technologií rozpoznávání obličeje a FERET se stal základem pro budoucí aplikace. [7]

2.3 Strojové učení a jeho vliv v technologii rozpoznávání obličeje

Strojové učení je podmnožina umělé inteligence, která má schopnost se učit nebo predikovat žádané stavy. Strojové učení se začíná v oblasti rozpoznávání obličejů uplatňovat **již v 90. letech** - poprvé se objevilo **v metodě Eigenfaces**. Strojové učení umožňovalo algoritmům učit se z velkého množství dat, což způsobovalo výrazné zlepšení přesnosti a spolehlivosti rozpoznávání obličeje. [11, 15]

Avšak k významnému pokroku došlo až **na začátku 21. století**, kdy se začíná aplikovat technika **hlubokého učení**. [11, 15]

2.3.1 Hluboké učení a konvoluční neuronové sítě

Hluboké učení je technika strojového učení, která používá **konvoluční neuronové sítě (CNN)**. CNN je umělá neuronová síť, která je navržena tak, aby efektivně zpracovávala a analyzovala obrazová data. Tato technika se začala v rozpoznávání obličejů používat **v roce 2012**, kdy **Alex Krizhevsky a jeho tým vytvořili síť AlexNet**. AlexNet vyvolal velký příliv zájmu a investic do metod hlubokého učení a díky tomu vznikly během následujících 4 let nové revoluční algoritmy (modely) - **DeepFace, VGGFace, DeepID a FaceNet**, které posunuly technologii rozpoznávání obličeje na novou úroveň. [11, 15]

2.3.2 CNN modely

Tyto modely byly trénovány na několika milionech snímků, měly vysokou rychlost zpracování a také vysokou úspěšnost. Například model **DeepFace** od společnosti Meta měl úspěšnost správného rozpoznání obličeje **přes 97 %**, což je stejná přesnost, jakou má člověk. [15, 29]

2.4 iPhone X a Face ID

Technologie rozpoznávání obličeje se začala používáním hlubokého učení rychle zdokonalovat a **12. září 2017 představila společnost Apple iPhone X, první iPhone s funkcí Face ID**. Zařízení bylo uvedeno na trh 3. listopadu téhož roku. **Face ID** je biometrická metoda, která pomocí **3D skenování** obličeje umožňuje odemknout iPhone a nabízí tak vysokou úroveň bezpečnosti. Poprvé v historii přineslo Face ID pokročilou technologii 3D rozpoznávání obličeje přímo do spotřebitelského zařízení, což nastavilo nový standard v oblasti zabezpečení. Před Face ID bylo rozpoznávání obličeje u spotřebitelských zařízení převážně pouze 2D, a tedy ne moc bezpečné, jelikož ho bylo možné oklamat například fotografií. [8, 24]

V dnešní době je Face ID tak dokonalé, že rozpozná obličej přes čepici, šátky, optické brýle, kontaktní čočky, roušku či růst vousů. **Úspěšnost Face ID je 99,99 %** a kromě odemykání zařízení umí také ověřovat platby, přihlašovat uživatele do aplikací nebo vytvářet animované emoji (Animoji a Memoji) na základě pohybů obličeje. [24]

2.5 Technologie rozpoznávání obličeje dnes

Dnes se technologie rozpoznávání obličeje stále více opírá o pokroky v hlubokém učení a konvolučních neuronových sítích. Moderní algoritmy, jako jsou DeepFace, FaceNet a Face ID, dosahují vysoké přesnosti rozpoznávání obličeje i v extrémních podmínkách, což posunulo technologii na novou úroveň. Tyto modely jsou schopny v některých aspektech překonávat lidské schopnosti, avšak stále narážejí na problémy, jako je variabilita osvětlení, úhly pohledu a odlišnosti v obličejových výrazech. [26]

3 Umělá inteligence a strojové učení

Předtím, než se podrobně zaměříme na technologii rozpoznávání obličeje, je klíčové si vysvětlit základní principy umělé inteligence a strojového učení, na kterých tato technologie staví a které jí umožňují dosahovat vysoké přesnosti a spolehlivosti při identifikaci obličejů.

3.1 Umělá inteligence

Umělá inteligence (AI, artificial intelligence) je oblast informatiky, která se zabývá vytvářením systémů, které mají napodobovat lidský mozek a být tedy schopné vykonávat úkoly, které vyžadují lidskou inteligenci. Tyto úkoly zahrnují rozpoznávání a používání řeči, plánování tras, řešení problémů, rozpoznávání obrazů atd. Umělá inteligence se dělí na několik podkategorií, z nichž strojové učení je jednou z nejvýznamnějších. [11]

Hlavním cílem umělé inteligence je vyvinout systémy, které se dokážou samostatně učit a adaptovat na nové situace. To zahrnuje schopnost analyzovat velké množství dat, vyvozovat z nich závěry a zrychlovat se. V kontextu rozpoznávání obličeje se umělá inteligence využívá k trénování modelů, které dokážou efektivně rozpoznávat rysy v obličeji a vytvářet z nich unikátní otisk obličeje. [11]

3.2 Strojové učení

Strojové učení (ML, machine learning) je podmnožina umělé inteligence, která se zabývá návrhem metod a algoritmů, které umožňují systémům se učit z předchozích zkušeností, tedy z dat. Data jsou základem správného fungování strojového učení a je nutné, aby byla správně zvolená a předzpracovaná, aby algoritmy mohly efektivně fungovat. Například při analýze počasí je potřeba mít dostatek dat, která obsahují minimální a maximální teploty, rychlost větru, objem srážek atd. Na základě takto zpracovaných dat algoritmus identifikuje vzorce a vztahy, podle kterých predikuje výsledky. [11]

Hlavní výhodou strojového učení je, že algoritmy mohou postupně zlepšovat svou přesnost bez toho, aby bylo nutné je ručně upravovat. Tento proces zlepšování je možný díky adaptivnímu učení, kdy model zohledňuje své chyby v dalších predikcích. S rostoucím množstvím dat a trénováním modelu tedy strojové učení dosahuje stále vyšší přesnosti. [11]

Strojové učení je díky svému autonomnímu zlepšování zásadní v mnoha moderních aplikacích, jako je rozpoznávání obrazu, zpracování přirozeného jazyka, automatizované rozhodovací systémy, diagnostika onemocnění nebo personalizace obsahu na internetu. [11]

Strojové učení lze rozdělit na tři základní typy, z nichž každý má své specifické fungování a algoritmy.

3.2.1 Učení s učitelem (Supervised learning)

Učení s učitelem je metoda, při které je algoritmus trénován na datech obsahujících jak vstupy, tak i odpovídající výstupy. Během trénování model porovnává své predikce s reálnými výstupy a na základě rozdílu upravuje své parametry tak, aby se zvýšila přesnost predikce. Tento typ učení se využívá v úlohách **regrese** (kdy je cílová proměnná spojitého typu, například při odhadování tržní ceny domu) a **klasifikace** (kdy je cílová proměnná kategoriálního typu, například přiřazení obličeje z fotografie ke konkrétní osobě). Tento typ učení se využívá právě u technologie rozpoznávání obličeje, kde se často používají konvoluční neuronové sítě (CNN). Podrobnější informace o CNN se nachází v kapitole 5.3. [11]

3.2.2 Učení bez učitele (Unsupervised learning)

Učení bez učitele, oproti učení s učitelem, neobsahuje žádné informace o struktuře dat. Cílem těchto metod je právě strukturu v datech identifikovat. Struktura se identifikuje především pomocí **shlukování** (kdy se dělí data podle jejich podobnosti) nebo **redukce dimenzionality** (kdy se data přetransformují tak, aby byla zachována jejich nejdůležitější struktura a vzory, což zjednodušuje jejich reprezentaci). Algoritmy učené bez učitele se uplatňují tam, kde nejsou dostupná označená data, například analýza chování zákazníků nebo v doporučovacích systémech, kde se identifikují skupiny s podobnými preferencemi. [11]

3.2.3 Učení posilováním (Reinforcement learning)

Učení posilováním je metoda, při které se algoritmus učí **pomocí zpětné vazby (odměny)**. Algoritmus přijímá pouze dvě formy zpětné vazby - kladné odměny a záporné odměny. Učí

se průběžně a upravuje své parametry díky využívání zpětných vazeb z okolí, a postupně se optimalizuje jeho chování. Algoritmus využívá **Q-učení** (vyhledává optimální predikci tím, že maximalizuje očekávanou hodnotu odměny v dalších krocích) a **TD-učení** (stroj se učí tak, že zkouší určité akce a upravuje predikce podle dat z minulosti). Učení posilováním se využívá v situacích, kde je klíčová schopnost rozhodování, například hraní her (šachy nebo hra go) či autonomní ovládání systémů. [11]

4 Předzpracování dat a datasetu

Aby algoritmus pro rozpoznávání obličeje správně fungoval a dosahoval vysoké přesnosti, je nezbytné správně předzpracovat jak vstupní data, která slouží přímo k rozpoznávání obličejů, tak dataset, na kterém se model trénuje. Tato kapitola se tedy zaměří na procesy předzpracování dat, včetně úprav vstupních dat a přípravy datasetu pro efektivní trénink modelu.

4.1 Předzpracování dat pro modely strojového učení

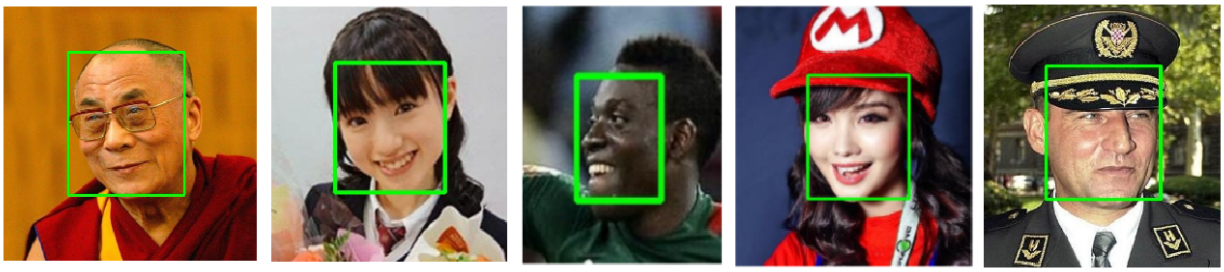
Předzpracování dat je klíčové pro dosažení vysoké přesnosti při zachování co nejkratší doby zpracování. Tento proces zajišťuje, že data jsou konzistentní, správně strukturovaná a optimálně připravená, což minimalizuje riziko chyb a zvyšuje efektivitu modelu.

4.1.1 Detekce obličeje

Prvním krokem předzpracování dat je identifikace oblastí, kde se vyskytuje obličej. Pro tento proces se používají různé metody detekce, které se liší svým přístupem a složitostí. Mezi nejpoužívanější metody patří:

- Haarovy kaskády (Haar Cascades)
- Histogram orientovaných gradientů (HOG)
- DNN (Deep Neutral Networks)
- YOLO (You Only LOOk Once)
- RetinaFace?
- MTCNN

Každá z těchto metod má své výhody a nevýhody a její použití závisí na konkrétních požadavcích aplikace, jako je její rychlost, přesnost nebo odolnost vůči okolnímu šumu.



Obrázek 4.1: Detekované obličeje z obrázků z datasetu WGGFace2

4.1.2 Změna velikosti a normalizace dat

Po detekci oblastí, kde se obličeje nacházejí, je dalším krokem jejich oříznutí, zmenšení a normalizace. Pro zajištění konzistence dat je nutné zajistit, aby každý vyříznutý obličej měl stejný poměr stran a rozlišení. Nejprve se obraz ořízne tak, aby obsahoval pouze oblast s detekovaným obličejem. Přičemž je potřeba zachovat konzistentní poměr stran u všech oříznutých snímků. Oříznutím se zajistí, že se model soustředí výhradně na relevantní část obrazu.

Následně se změní velikost obrázku na jednotnou velikost, například 224×224 pixelů. Tento rozměr je používán jako standardní vstup pro mnoho modelů strojového učení, protože umožňuje optimalizované zpracování a jeho trénink. Také se však stará o kompatibilitu se standardními architekturami hlubokých neuronových sítí, jako jsou ResNet, VGG nebo MobileNet.

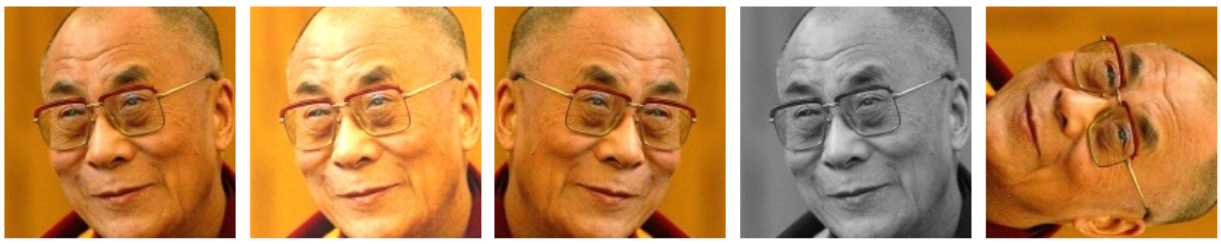
Posledním důležitým krokem je normalizace hodnot pixelů. Většina obrázků je reprezentována hodnotami pixelů v rozsahu 0 až 255, což odpovídá intenzitě jasu jednotlivých bodů. Normalizace tyto hodnoty převede do rozmezí 0 až 1. Tato úprava dat snižuje vliv rozdílů v jasu a kontrastu mezi jednotlivými snímky, což opět napomáhá modelu pracovat s větší přesností.

Tato kombinace kroků, oříznutí, změna velikosti a normalizace, zajišťuje, že data vstupující do modelu jsou konzistentní a připravená pro optimální tréninkový proces a rozpoznávání dat.

Obrázek 4.2: Předzpracované obličeje z obrázků z datasetu WGGFace2

4.1.3 Augmentace dat

Augmentace je technika, která se využívá pro zvýšení rozmanitosti trénovacího datasetu. Tento proces zahrnuje užití různých transformací na původní snímky, čímž se vytvoří nové variace vstupních dat. Augmentace zlepšuje robustnost modelu a jeho schopnost generalizace na nové příklady. Mezi typické metody augmentace dat patří rotace, změna jasu, kontrastu a sytosti, převrácení, geometrická deformace atd.



Obrázek 4.3: Augmentace obličeje z obrázku z datasetu WGGFace2

4.2 Vstupní data pro inferenci

Vstupní data jsou data, která vstupují do již přetrénovaného modelu s účelem rozpoznání osob. Tato data musí vstupovat do modelu již předzpracovaná tak, jak bylo zmíněno: detekce obličeje, jeho oříznutí, změna velikosti a normalizace. Vstupní data lze získat dvěma způsoby, vložením obrázku nebo snímáním obrazu z živé kamery.

4.3 Dataset

Dataset je soubor dat obsahující obrázky obličejů a informace o nich. Účelem datasetu je trénovat a optimalizovat algoritmy rozpoznávání obličejů. Dataset je obvykle organizován do tříd, kdy každé třídě většinou odpovídá identita osoby, avšak mohou jí odpovídat i jiné

charakteristické vlastnosti, jako je věk, pohlaví či emoce. Každá třída poté obsahuje různé variace obličejů, například různé úhly pohledu, osvětlení, výrazy v obličeji nebo zakrytí části obličeje.

Pro efektivní využití datasetu při trénování modelu je nutné jej opět předzpracovat. Tento proces zahrnuje detekci obličeje, jeho oříznutí, změnu velikosti, normalizaci a augmentaci dat. Takto připravený dataset je poté využit k trénování a testování modelu rozpoznávání obličejů.

4.3.1 Datasety pro identifikaci obličejů

Pro trénování modelů zaměřených na identifikaci obličejů se používají různé datasety. Mezi nejznámější datasety patří:

- **LFW (Labeled Faces in the Wild)** - Tento dataset obsahuje přes 13 000 obrázků obličejů 5 749 různých osob. Dataset je vhodný pro testování základních modelů a pro porovnání jejich výkonu v jednoduchých podmínkách, avšak svou velikostí je na trénování robustních modelů pro praktické užití nedostatečný. [13]

- **VGGFace2** - Tento dataset má přes 3 miliony obrázků obličejů 9 131 osob a je velmi populární pro svou velkou variabilitu podmínek, jako jsou různé výrazy a úhly pohledu. Je to jeden z nejrozsáhlejších datasetů pro identifikaci obličejů. [20]

- **Casia-WebFace** - Obsahuje více než 500 000 obrázků obličejů 10 575 osob. Je známý svou rozmanitostí, ale obrázky jsou převážně pořízeny z internetových zdrojů, což může znamenat nižší kvalitu a nedostatečně rozmanité variace dat. [5]

- **MS-Celeb-1M** - Tento dataset je největší svého druhu s více než 10 miliony obrázků obličejů 100 000 osob. Obsahuje obrázky veřejně známých osob a to ve velmi různorodých podmínkách. Díky své velikosti je vhodný pro trénování robustních modelů. [16]

4.3.2 Rozdělení na sady

Dataset pro rozpoznávání obličejů se zpravidla dělí na dvě hlavní sady: **trénovací a testovací**. Trénovací sada (80 % dat) slouží k natrénování modelu, zatímco testovací sada (20 % dat) se používá k hodnocení jeho výkonu na neznámých datech. V některých případech

může být trénovací sada dále rozdělena na **validační sadu**, která se používá k ladění hyperparametrů a prevenci přetrénování. V praxi se však často používá pouze dělení na trénovací a testovací sadu, pokud není potřeba specifické ladění modelu.

5 Algoritmy pro rozpoznávání obličeje

Algoritmus pro rozpoznávání obličeje je jádrem celého rozpoznávacího procesu. Jeho hlavním úkolem je převést vstupní obraz na strukturovanou reprezentaci rysů, nazývanou otisk obličeje, kterou lze snadno porovnávat s reprezentací rysů jiných osob a samotnou klasifikaci získaných dat s databází známých osob. Tato kapitola se zaměřuje na fungování algoritmů, na moderní algoritmy, zejména na konvoluční neuronové sítě, a také na hodnocení výkonu algoritmu.

5.1 Fungování algoritmu

Zásadní proces, který algoritmus provádí, je **extrakce rysů**. Z fotografie se extrahují jedinečné charakteristiky, které odlišují konkrétní obličej od ostatních. Existuje mnoho přístupů k extrakci rysů, avšak v moderních aplikacích dominují metody založené na hlubokém učení. Tyto metody využívají konvoluční neuronové sítě, které se samy učí rozpoznávat důležité rysy obličeje na základě analýzy velkého množství dat. [25]

Výsledkem extrakce rysů je **vektor čísel**, který reprezentuje obličej. Tento vektor je následně porovnáván s vektory ostatních obličejů uložených v databázi, kde se hledá shoda. Čím více se vektory shodují, tím vyšší je pravděpodobnost, že se jedná o stejnou osobu. [25]

5.2 Algoritmy strojového učení

V oblasti rozpoznávání obličejů existuje mnoho různých typů algoritmů, avšak v současné době dominují metody založené na strojovém učení. Tyto algoritmy jsou schopny efektivně analyzovat a porovnávat složité vizuální informace díky své schopnosti učit se z dat. [11, 25]

Zejména metody využívající hluboké učení, jako jsou konvoluční neuronové sítě (CNN, Convolutional Neural Networks), se staly standardem v moderních aplikacích. CNN mají schopnost automaticky se učit extrahovat klíčové rysy obličeje přímo z obrazových dat, bez

nutnosti manuální definice těchto rysů. Díky své architektuře jsou CNN schopny zpracovávat velké množství obrazových dat, identifikovat složité vzory a generalizovat své učení na nové vstupy. [11, 25]

5.3 Rozpoznávání pomocí konvolučních neuronových sítí (CNN)

Konvoluční neuronové sítě jsou jedním z nejvýznamnějších pokroků v oblasti hlubokého učení, zejména v oblasti rozpoznávání obrazu. Konvoluční neuronová síť byla navržena speciálně pro analýzu obrazových dat, kde se klade důraz na klasifikaci objektů. [11]

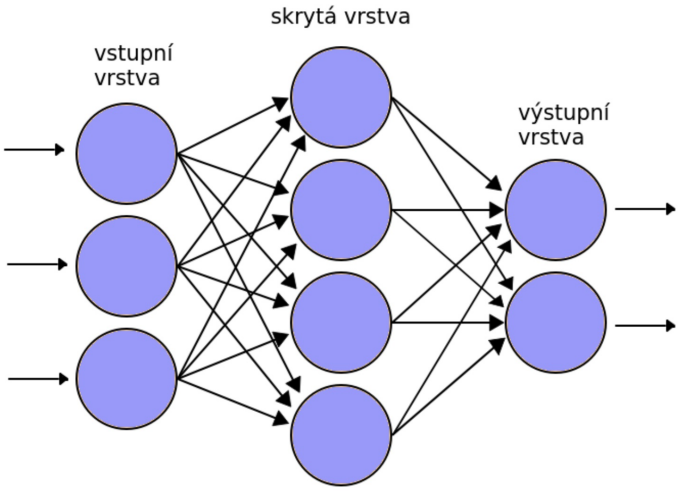
5.3.1 Základy neuronových sítí

Konvoluční neuronové sítě vycházejí z architektury klasických **neuronových sítí (ANN, Artificial Neural Networks)**. ANN jsou složeny z propojených vrstev neuronů, které napodobují činnost biologických neuronů. Díky tomu jsou schopny se přizpůsobit nové situaci. [11]

ANN mají podobnou strukturu a činnost jako lidský mozek. Každá ANN zpracovává daný vstup, například obrázek nebo text, což vede k určitému výstupu, například rozpoznání objektu na obrázku. Podobně jako v mozku jsou ANN tvořeny propojenými umělými neurony, označovanými jako uzly. Tyto uzly spolu komunikují prostřednictvím spojení, která se nazývají hrany sítě. [11]

Jedním z klíčových aspektů fungování neuronových sítí jsou váhy, které reprezentují vztahy mezi jednotlivými uzly. Váhy určují, jak silný vliv má daný vstup na výstup neuronů. Během procesu trénování jsou hodnoty těchto vah upravovány tak, aby model co nejlépe zachytil vztahy ve vstupních datech. Tím se síť postupně přizpůsobuje a zlepšuje svou schopnost správně rozpoznávat vzory a klasifikovat objekty. [11]

V CNN jsou uzly uspořádány do vrstev: do vstupní vrstvy, která přijímá data, jedné nebo více skrytých vrstev, které zpracovávají data, a výstupní vrstvy, která poskytuje výsledek, například určí, co se nachází na obrázku. [11]



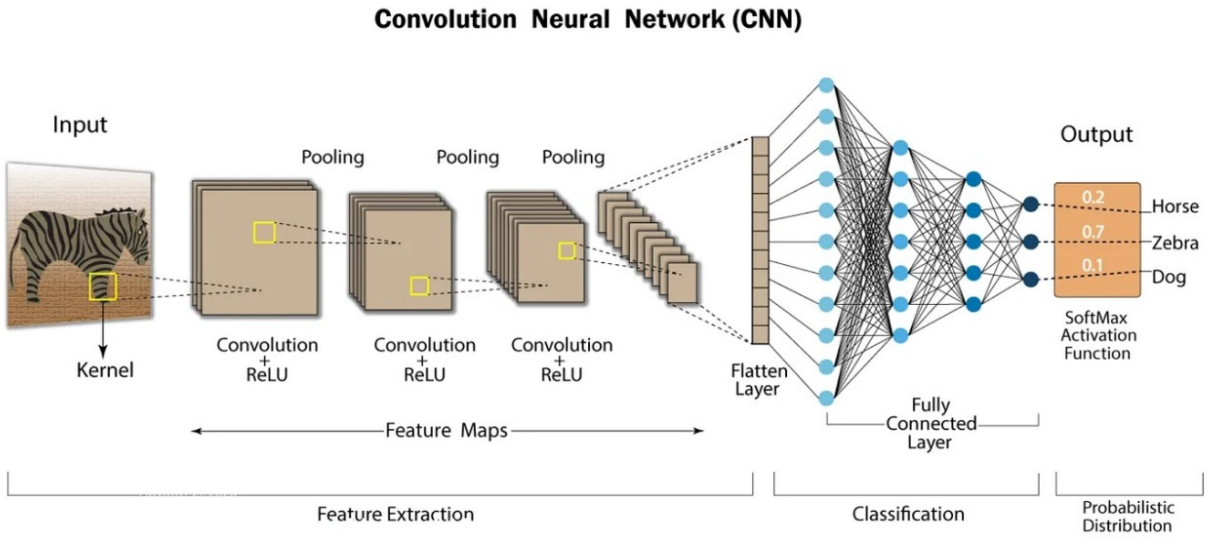
Obrázek 5.1: Struktura umělé neuronové sítě

Ačkoliv ANN jsou úspěšné ve zpracování strukturovaných dat, jejich nevýhodou při práci s obrazovými daty je neschopnost zachytit prostorové vztahy mezi jednotlivými pixely. To vedlo k vývoji CNN, které tuto omezenost překonávají.

5.3.2 Konvoluční neuronová síť

Konvoluční neuronová síť je speciální typ umělé neuronové sítě, která využívá **konvoluční a pooling vrstvy** k efektivnímu zpracování vstupních dat, jako jsou obrázky. Oproti klasickým umělým neuronovým sítím je CNN schopna pracovat s daty velkých rozměrů za použití menšího množství parametrů, díky čemuž je naučení takové sítě správnému fungování mnohem jednodušší. [17, 27, 32]

Kromě konvolučních a pooling vrstev obsahují CNN také **plně propojené vrstvy**, které jsou zodpovědné za finální rozhodnutí modelu, například klasifikaci objektů na obrázcích. Kombinace těchto vrstev umožňuje síti efektivně extrahovat rysy z dat a následně je použít k přesnému rozpoznání a klasifikaci objektů. [17, 27]



Obrázek 5.2: Struktura konvoluční neuronové sítě

5.3.3 Konvoluční a pooling vrstva

Data mohou procházet konvoluční a pooling vrstvou opakovaně v několika vrstvách. S každým dalším opakováním se síť zaměřuje na stále složitější a abstraktnější rysy obrazu, čímž roste její schopnost rozpoznávat konkrétní objekty. [22, 17, 27]

Konvoluční vrstva

Konvoluční vrstva provádí operaci konvoluce, což je matematická operace, jejímž cílem je převést obraz na číselné hodnoty, které může neuronová síť dále analyzovat. [22, 17, 27]

Konvoluce zahrnuje použití filtrů (nebo jader), které jsou menší než celý vstupní obrázek a postupně se posouvají po obraze. Každý filtr detekuje specifické rysy, jako jsou okraje nebo textury, tím, že vynásobí hodnoty pixelů váhami. Součet těchto hodnot je následně zpracován aktivační funkcí. Výstupem této operace je aktivační mapa, která zvýrazňuje detekované rysy. První vrstva obvykle detekuje základní prvky, jako jsou okraje, zatímco hlubší vrstvy se zaměřují na složitější vzory a objekty. Po extrakci těchto rysů jsou výsledky předávány pooling vrstvě. [22, 17, 27]

Pooling vrstva

Pooling vrstva se využívá k redukci rozměrů aktivačních map a tím snižuje výpočetní náročnost sítě. Tento proces zachovává klíčové informace a zároveň zjednodušuje data pro

další zpracování. Nejběžnějšími operacemi v pooling vrstvách jsou max pooling, kdy je vybírána maximální hodnota v určité oblasti, a average pooling, kdy se počítá průměr hodnot v oblasti. [12, 18]

Aktivační funkce

Aktivační vrstva se používá v konvoluční vrstvě, kde upravuje výstupní informace. Jejím úkolem je eliminovat negativní hodnoty z aktivační mapy tím, že je nastaví na nulu, aniž by ovlivnila samotnou konvoluční vrstvu. Tím se snižuje linearita výstupních informací a síť se stává schopná modelovat složitější vzory. V současnosti je nejčastěji používanou funkcí **ReLU (Rectified Linear Unit)**. ReLU nastavuje všechny negativní hodnoty na nulu a zrychluje trénování sítě, i když neřeší chování pro negativní vstupy. [18]

5.3.4 Plně propojené vrstvy

Plně propojené vrstvy (Fully Connected Layers) spojují všechny neurony s předchozími vrstvami a slouží k integraci extrahovaných rysů. Umožňují klasifikaci a rozhodování na základě kombinace informací z celého obrazu. Výstupy této vrstvy určují finální výsledek, například rozpoznání objektu. [22, 17]

5.3.5 Konvoluční neuronové sítě a rozpoznávání obličeje

Konvoluční neuronové sítě se v oblasti rozpoznávání obličeje ukázaly jako vysoce efektivní nástroj díky své schopnosti extrahovat rysy z obrazových dat a následně je použít pro klasifikaci. Pro úspěšnou detekci obličeje je klíčové, jak CNN zachytí různé rysy obličeje, jako jsou oči, nos, ústa a jejich vzájemné vztahy. Tyto informace jsou následně zpracovávány v několika vrstvách, přičemž první vrstvy zachytávají základní rysy (např. okraje), zatímco hlubší vrstvy kombinují složitější informace pro komplexní rozpoznání obličeje.

V dnešní době se pro rozpoznávání obličeje téměř vždy používají algoritmy na bázi CNN. Nejčastěji se používají modely jako VGG-Face, ResNet a Inception. Tyto modely jsou navrženy tak, aby efektivně identifikovaly obličeje v různých podmínkách a dosahovaly vysoké přesnosti při rozpoznávání.

5.4 Hodnocení výkonu algoritmu

Hodnocení výkonu algoritmu pro rozpoznávání obličeje se používá k ověření jeho přesnosti a spolehlivosti. K vyhodnocení výkonu modelu se používají různé metriky jako **přesnost, F1 skóre, či ROC křivka**. Na základě těchto výsledků se poté optimalizují **hyperparametry** modelu, které slouží k nastavení klíčových parametrů ovlivňujících trénování a výkon modelu, jako je rychlost učení nebo počet vrstev. [4]

5.4.1 Metriky pro hodnocení výkonu

Jednou z nejpoužívanějších metrik pro hodnocení výkonnosti modelu je **přesnost (Accuracy)**, která měří podíl správně klasifikovaných vzorků na celkovém počtu vzorků. Tato metrika je jednoduchá a lze snadno pochopit a interpretovat, avšak je potřeba, aby byl počet vzorků v jednotlivých datech vyvážený, aby nebyla tato metrika zavádějící. [4]

F1 skóre představuje **harmonický průměr přesnosti (Precision) a úplnosti (Recall)**. Tato metrika je užitečná zejména v situacích, kdy jsou třídy dat nerovnoměrně zastoupeny, protože kombinuje schopnost modelu správně klasifikovat pozitivní příklady i minimalizovat falešné poplachy. [4]

Dalším nástrojem pro hodnocení je **ROC křivka (Receiver Operating Characteristic)**, která ilustruje vztah mezi pravými pozitivními a falešnými pozitivními případy při různých prahových hodnotách. Klíčovou hodnotou je **AUC (Area Under the Curve)**, která udává schopnost modelu rozlišovat mezi pozitivními a negativními případy. Vyšší hodnota AUC znamená lepší výkon modelu. [4]

5.4.2 Optimalizace hyperparametrů

Hyperparametry jsou parametry, které musí být nastaveny před zahájením trénování modelu, a jejich správná volba může zlepšit přesnost a efektivitu modelu. Mezi běžně laděné hyperparametry patří míra učení, velikost dávky, počet epoch, počet vrstev a neuronů nebo koeficienty regularizace. [6]

Optimalizace hyperparametrů zahrnuje hledání nejlepší kombinace těchto hodnot s cílem dosáhnout co nejlepšího výkonu modelu. Nejčastější metody zahrnují **vyhledávání v mřížce**, kde jsou testovány všechny možné kombinace předdefinovaných hodnot, a náhodné vyhledávání, které vybírá náhodné kombinace parametrů. Pokročilejší metody, jako je **baye-**

sovská optimalizace, využívají pravděpodobnostní modely pro efektivní výběr nejlepších hyperparametrů. [6]

Vhodně nastavené hyperparametry jsou nezbytné pro dosažení vysoké přesnosti a dobré schopnosti modelu generalizovat na nová data. [6]

6 Problematika

Technologie rozpoznávání obličeje je dnes již běžnou součástí života a její přítomnost nelze přehlédnout. Avšak s rozšiřující se a zdokonalující se technologií rozpoznávání obličeje roste i potřeba uvědomit si rizika a nebezpečí spojená právě s touto technologií, která mohou mít zásadní dopad na společnost i jednotlivce. A proto je potřeba si říci něco o ochraně soukromí a možné zaujatosti algoritmů spojených s technologií rozpoznávání obličeje.

6.1 Ochrana soukromí

Jeden z nejčastěji diskutovaných problémů spojených s rozpoznáváním obličeje je ochrana soukromí. V dnešní době, kdy jsou kamery běžně umístěny na veřejných místech, je sběr snímků obličejů velmi snadný. Pokud by se do databáze těchto snímků dostala osoba se zlými úmysly, mohla by údaje zneužít k krádeži identity, loupežím nebo obtěžování. Biometrické údaje, jako jsou obličejové rysy, jsou na rozdíl od hesel nebo čísel kreditních karet nezměnitelné, což činí jejich zneužití obzvláště nebezpečným a může vést k trvalé ztrátě soukromí. [14]

Obavy o soukromí vedly k návrhu legislativy, která má regulovat používání této technologie a chránit osobní údaje. Například v roce 2021 vydal Evropský výbor pro ochranu údajů společné stanovisko k zákonu o umělé inteligenci, v němž požaduje zákaz dálkového biometrického sledování na veřejných místech. Na rizika spojená s možnou krádeží identity upozorňují i organizace na ochranu lidských práv, které varují před nebezpečím používání technologie rozpoznávání obličejů ve veřejném prostoru. [14]

V současnosti je rozpoznávání obličeje součástí široké diskuse o tom, jak najít rovnováhu mezi výhodami, které technologie přináší, a ochranou základních práv. To zahrnuje zajištění transparentnosti v oblasti sběru a uchovávání dat, regulaci veřejného používání technologie a její správné nasazení v souladu s právními normami. [14]

6.2 Diskriminace a zaujatost

Technologie rozpoznávání obličeje je sice v dnešní době téměř dokonalá, avšak ani tak není imunní vůči předsudkům. Systémy mají tendenci snáze rozpoznávat bílé muže než ženy nebo osoby jiných etnických skupin. V roce 2018 došlo k 35% chybám v identifikaci barevných žen, zatímco u bílých mužů to bylo jen 1%. Tyto chyby mohou vést k falešným obvině-ním, jak tomu bylo v případě Nijeera Parkse, který byl nesprávně zatčen na základě chyby v identifikaci. [14]

Zaujatost v technologii může také způsobit diskriminaci a nespravedlivé zacházení, napří-klad v Číně byly zaujaté algoritmy záměrně zneužity, kdy software Huawei sloužil k cílené identifikaci příslušníků ujgurské menšiny. [14]

Část II

Implementace programu pro rozpoznávání obličeje

7 Cíl práce a použité technologie

Praktická část mé práce se zaměřuje na vývoj programu pro rozpoznávání obličeje pomocí strojového učení. Hlavním cílem bylo vytvořit vlastní funkční model, který dokáže správně identifikovat obličeje na základě zadaných dat. Tento proces zahrnoval návrh algoritmu pro rozpoznávání obličeje, trénování algoritmu, jeho testování, vytvoření a správu databází a implementaci uživatelského rozhraní.

Chtěl jsem plně využít možnosti, které rozpoznávání obličeje nabízí, a proto jsem se rozhodl vytvořit aplikaci, která umožňuje klasifikaci osob jak na základě zadaného obrazu, tak i v reálném čase pomocí webkamery. Aplikace obsahuje uživatelskou lištu, která umožňuje uživateli vybrat, zda chce rozpoznat osoby na načteném obrazu, z webkamery nebo přidat novou osobu do databáze známých osob.

Program je vytvořen v jazyce Python a využívá několik důležitých knihoven, které zajišťují jeho funkčnost. Pro návrh a trénování modelu jsem použil TensorFlow, jednu z nejpoužívanějších knihoven pro práci s umělou inteligencí, a NumPy pro efektivní práci s numerickými výpočty. Pro úpravu a předzpracování obrazových dat byla použita knihovna OpenCV, která poskytuje širokou škálu nástrojů pro zpracování obrazu. Knihovna os byla využita pro práci se soubory a adresáři.

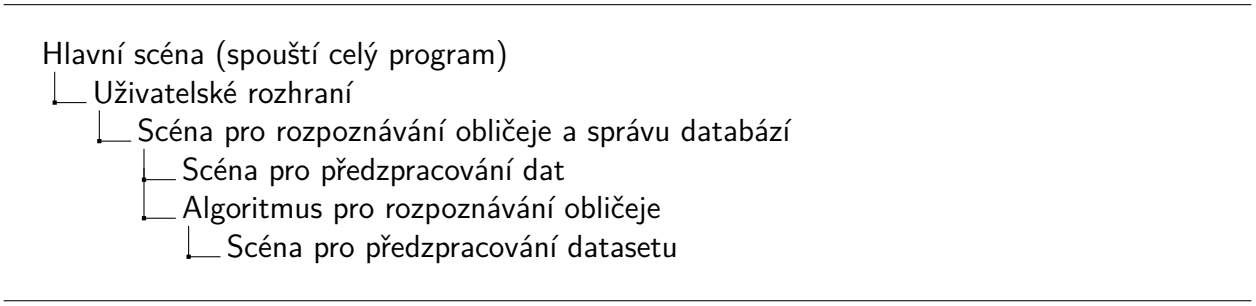
Dále program využívá knihovny FAISS a SQLite3, které slouží k vytvoření a práci s databází. Databáze využívám k ukládání osob, které je program poté schopen rozpoznávat. Konkrétně databáze SQLite slouží pro ukládání metadat o osobách a index FAISS slouží k ukládání embeddingů (vektorová reprezentace obličeje) osob.

Pro detekci obličeje jsem zvolil klasifikátor DNN, který využívá moderní techniky hlubokého učení k dosažení přesných a spolehlivých výsledků.

Dalším důležitým aspektem projektu bylo vytvoření jednoduchého uživatelského rozhraní, které umožňuje snadnou interakci s programem a dodává aplikaci moderní vzhled. Pro tento úkol jsem využil knihovnu PyQt, která poskytuje nástroje pro tvorbu grafického rozhraní.

8 Struktura programu

Výsledný projekt je strukturován do několika navzájem propojených souborů, aby byla zachována modularita a přehlednost kódu. Každý soubor má svůj konkrétní účel a reprezentuje určitou funkční část aplikace. Například jeden soubor obsahuje implementaci modelu pro rozpoznávání obličeje, další se zaměřuje na zpracování obrazových dat pomocí OpenCV a třetí na správu uživatelského rozhraní s knihovnou PyQt.



Obrázek 8.1: Podrobný strom scén programu

9 Předzpracování dat

Pro předzpracování dat jsem vytvořil modul *data_preprocessing*, tento modul slouží k přípravě obrazových dat pro následné zpracování modelem rozpoznávání obličejů. Proces předzpracování zahrnuje detekci obličejů, ořezání obličejových oblastí, redukci rozlišení a normalizaci hodnot pixelů. Každý krok je realizován samostatnou metodou v rámci třídy `DataPreprocessing`, která využívá knihovny OpenCV a NumPy.

9.1 Inicializace třídy `DataPreprocessing`

Třída `DataPreprocessing` je navržena tak, aby zpracovávala vstupní obrazová data a usnadnila práci s modelem pro detekci obličejů. Konstruktor této třídy přijímá vstupní obrazová data a zároveň inicializuje cesty k modelu a jeho konfiguraci. Model je realizován pomocí knihovny DNN (Deep Neural Network) dostupné v OpenCV.

```
1 class DataPreprocessing:
2     def __init__(self, data):
3         self.data = data
4 
5         self.model_path =
          ↪  r"Program\\DNN\\res10_300x300_ssd_iter_140000.caffemodel"
6         self.config_path = r"Program\\DNN\\deploy.prototxt"
```

9.2 Detekce obličejů

Prvním krokem předzpracování je detekce obličejů na vstupním obrazu. Tato úloha je realizována metodou `detect_faces`, která využívá předtrénovaný model CNN (konvoluční neuronové sítě) a funkci `blobFromImage` k přípravě obrazových dat. Výstupem metody je

seznam souřadnic obdélníků ohraničujících detekované obličeje, přičemž jsou zohledněny pouze obličeje s jistotou (confidence) vyšší než 0.5.

```
def detect_faces(self):
    net = cv2.dnn.readNetFromCaffe(self.config_path, self.model_path)
    blob = cv2.dnn.blobFromImage(self.data, scalefactor=1.0, size=(300, 300),
    ↪  mean=(104.0, 177.0, 123.0))
    net.setInput(blob)
    detections = net.forward()

    faces = []
    for i in range(detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > 0.5:y
            box = detections[0, 0, i, 3:7] * numpy.array([w, h, w, h])
            faces.append(box.astype("int"))
    return faces
```

9.3 Ořezávání obličejových oblastí

Na základě souřadnic detekovaných obličejů se pomocí metody `crop_faces` extrahují jednotlivé obličejové oblasti z původního obrazu. Metoda upravuje obdélníkové oblasti tak, aby měly čtvercový formát, což je důležité pro konzistenci vstupních dat modelu. Výsledkem je seznam výřezů obsahujících jednotlivé obličeje.

```
def crop_faces(self, faces):
    cropped_faces = []
    for (x1, y1, x2, y2) in faces:
        width = x2 - x1
        height = y2 - y1
        max_size = max(width, height)

        new_x1 = x1 - (max_size - width) // 2
```

```
        new_y1 = y1 - (max_size - height) // 2
        new_x2 = new_x1 + max_size
        new_y2 = new_y1 + max_size

            if new_x1 < 0:
                new_x2 += - new_x1
                new_x1 = 0

            if new_y1 < 0:
                new_y2 += - new_y1
                new_y1 = 0

            if new_x2 > new_x1 and new_y2 > new_y1:
                cropped_face = self.data[new_y1:new_y2, new_x1:new_x2]
                cropped_faces.append(cropped_face)

    return cropped_faces
```

9.4 Změna velikosti a normalizace

Dalším krokem je standardizace velikosti a hodnot pixelů jednotlivých obličejů. Metoda `resizing_and_normalizing` převádí každý výřez na rozměr 128×128 pixelů a normalizuje hodnoty pixelů do intervalu [0,1], což zlepšuje efektivitu trénování a predikce modelu RO.

```
def resizing_and_normalizing(self, faces):
    preprocessed_faces = []
    for face in faces:
        face_resized = cv2.resize(face, (128, 128))
        face_normalized = face_resized / 255.0
        preprocessed_faces.append(face_normalized)

    return numpy.array(preprocessed_faces)
```

9.5 Kompletní předzpracování

Všechny kroky předzpracování jsou propojeny v metodě `preprocess_faces`. Tato metoda postupně volá metody pro detekci, ořezávání, změnu velikosti a normalizaci, čímž připraví obrazová data pro model RO.

```
def preprocess_faces(self):
    detected_faces = self.detect_faces()
    cropped_faces = self.crop_faces(detected_faces)
    preprocessed_faces = self.resizing_and_normalizing(cropped_faces)
    return preprocessed_faces
```

9.6 Předzpracování datasetu

Pro předzpracování dat jsem využil dataset VGGFace2, který obsahuje více než 3 miliony obrázků od 9 131 unikátních osob. Tento dataset jsem zvolil zejména díky jeho rozsáhlosti a rozmanitosti. Každá osoba je zastoupena přibližně 300 snímky, pořízenými za různých podmínek, což umožňuje modelu dosáhnout lepší generalizace na nových datech.

K procesu předzpracování dat jsem vytvořil vlastní modul `dataset_preparation.py`, který zajišťuje postupné načítání snímků z datasetu. Tyto snímky jsou následně zpracovány pomocí modulu `data_preprocessing.py`. Předzpracovaná data jsou pak ukládána do nové složky, kde jsou jednotlivé obrázky roztříděny podle odpovídajících tříd, reprezentujících unikátní osoby. Tento upravený dataset je připraven k trénování vlastního modelu.

10 Vytvoření vlastního modelu a jeho trénování

Srdcem celého projektu je model, který vytváří embeddingy – číselné vektorové reprezentace obličeje zachycující jeho charakteristické rysy. Díky těmto embeddingům je možné efektivně porovnávat obličeje a analyzovat jejich vzájemnou podobnost. Model je navržen tak, aby generoval tyto embeddingy, které mohou být následně použity k rozlišení různých osob na základě jejich tváří. O vytvoření a přetrénování tohoto modelu se stará modul `model.py`, který využívá konvoluční neuronovou síť. Model je trénován s využitím ztrátové funkce triplet loss, která zajišťuje, že embeddingy pro různé osoby budou co nejvíce odlišné, zatímco embeddingy stejné osoby budou co nejvíce podobné.

10.1 Architektura modelu

Pro implementaci modelu jsem využil knihovny TensorFlow, NumPy a os. Architektura modelu se skládá ze tří konvolučních vrstev, každá se 3x3 filtry a aktivační funkcí ReLU. Za každou konvoluční vrstvou následuje pooling vrstva, která slouží k redukci rozměru vstupu a tím snižuje výpočetní náročnost. Po třech konvolučních vrstvách následuje plně propojená vrstva, která přemění výstupy z předchozích vrstev na embeddingovou reprezentaci obličeje s definovanou velikostí. Po této vrstvě je aplikována normalizace embeddingu pomocí L2 normy, což usnadňuje jejich porovnávání pomocí kosinové podobnosti, která následně určuje podobnost mezi obličeji.

```
1 class L2Normalization(tf.keras.layers.Layer):
2     def call(self, inputs):
3         return tf.math.l2_normalize(inputs, axis=1)
4
```

```
def build_model(input_shape, embedding_size):
    inputs = tf.keras.Input(shape=input_shape)

    x = tf.keras.layers.Conv2D(64, (3, 3), activation="relu",
    ↪  padding="same")(inputs)
    x = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(x)

    x = tf.keras.layers.Conv2D(128, (3, 3), activation="relu",
    ↪  padding="same")(x)
    x = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(x)

    x = tf.keras.layers.Conv2D(256, (3, 3), activation="relu",
    ↪  padding="same")(x)
    x = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(x)

    x = tf.keras.layers.Flatten()(x)
    x = tf.keras.layers.Dense(embedding_size, activation=None)(x)
    outputs = L2Normalization()(x)

    return tf.keras.Model(inputs, outputs)
```

10.2 Ztrátová funkce triplet loss

Triplet loss je ztrátová funkce, která pomáhá modelu lépe rozlišovat mezi různými osobami. Minimalizuje vzdálenost mezi embeddingy (reprezentacemi) stejné osoby a zároveň maximalizuje vzdálenost mezi embeddingy různých osob. Ztráta se počítá jako rozdíl mezi těmito vzdálenostmi, přičemž je udržován minimální odstup mezi pozitivními a negativními páry, definovaný hyperparametrem margin. Tímto způsobem je model nucen, aby se embeddingy stejné osoby co nejvíce podobaly a embeddingy různých osob co nejvíce lišily.

```
def triplet_loss(y_true, y_pred, margin=1.5):
    anchor, positive, negative = tf.split(y_pred, num_or_size_splits=3, axis=0)
    pos_dist = tf.reduce_sum(tf.square(anchor - positive), axis=1)
```

```
    neg_dist = tf.reduce_sum(tf.square(anchor - negative), axis=1)
    basic_loss = pos_dist - neg_dist + margin
    return tf.reduce_mean(tf.maximum(basic_loss, 0.0))
```

10.3 Generování trénovacích dat

Generátor dat vytváří tripletové sady (anchor, pozitivní a negativní příklad). Pro každou sadu se náhodně vybírají dvojice obrázků: jeden obrázek slouží jako anchor (výchozí obrázek), druhý jako pozitivní příklad (obrázek té samé osoby), a třetí obrázek je negativní příklad (obrázek jiného subjektu). Tímto způsobem generátor vytváří trénovací data, která jsou vhodná pro optimalizaci triplet loss funkce.

```
def triplet_generator(file_paths, labels, batch_size):
    label_to_files = defaultdict(list)
    for file_path, label in zip(file_paths, labels):
        label_to_files[label].append(file_path)

    while True:
        batch_images, batch_labels = [], []
        all_labels = list(label_to_files.keys())
        np.random.shuffle(all_labels)

        for label in all_labels:
            pos_files = label_to_files[label]
            if len(pos_files) < 2:
                continue

            anchor, positive = np.random.choice(pos_files, size=2,
            ↪  replace=False)
            neg_label = np.random.choice([l for l in all_labels if l != label])
            negative = np.random.choice(label_to_files[neg_label])

            for img_path in [anchor, positive, negative]:
```

```
                img = preprocess_image(img_path).numpy()
                batch_images.append(img)
                batch_labels.append(label)

            if len(batch_images) >= batch_size:
                yield (np.array(batch_images[:batch_size], dtype=np.float32),
                       np.array(batch_labels[:batch_size], dtype=np.int32))
                batch_images, batch_labels = [], []
```

10.4 Trénování modelu

Model je trénován pomocí optimalizátoru Adam, který je často používán právě v trénování neuronových sítí díky své efektivitě při úpravě rychlosti učení během trénování.

Pro trénování modelu je použíta funkce model.fit, která postupně učí model rozpoznávat obličeje. Model se trénuje na základě obrázků, které mu jsou poskytnuta prostřednictvím generátoru dat. Každý obrázek model porovnává a upravuje své parametry, aby se zlepšil v rozpoznávání podobností mezi obličeji.

Během trénování model také pravidelně testuje, jak si vede na neznámých datech, aby bylo jasné, jak dobře se naučil tvořit embeddingy. Celý proces probíhá v několika cyklech (epochách), přičemž každý cyklus model projde trénovacími daty. Po každém cyklu je model ukládán, aby bylo možné se k němu vrátit.

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
 ↪  loss=triplet_loss)

history = model.fit(
    train_gen,
    steps_per_epoch=train_steps,
    validation_data=val_gen,
    validation_steps=val_steps,
    epochs=5,
    verbose=1,
```

```
    callbacks=[checkpoint_callback]
)
```

10.5 Uložení modelu

Po dokončení trénování je model uložen ve formátu HDF5 pro další využití.

```
model.save("my_face_recognition_model.h5")
```

11 Funkce rozpoznávání a databáze

Po vytvoření a přetrénování vlastního modelu rozpoznávání obličeje je dalším krokem jeho integrace do programu a následné použití pro generování embeddingů. Této problematice se věnuje následující modul `recoengine.py`, který se zaměřuje také na správu databází obsahujících informace o osobách a jejich embeddingy, a na porovnávání nově načtených osob s daty uloženými v těchto databázích.

11.1 Načtení modelu a generování embeddingů

Pro práci s modelem jsem vytvořil třídu `ModelHandler`, která načítá a používá vlastní model RO.

11.1.1 Načtení modelu

Nejprve je potřeba model správně načíst. Model se načítá pomocí metody `load_model`, která kontroluje, zda již model nebyl načten dříve. Pokud model načten nebyl, tak ho načte. Jelikož model obsahuje vlastní komponenty, jako je ztrátová funkce `triplet_loss` a třída `L2Normalization`, která normalizuje embeddingy na jednotkovou velikost, je potřeba tyto komponenty zaregistrovat pomocí parametru `custom_objects`. Tím se zajistí, že TensorFlow správně rozpozná a použije tyto implementace, které nejsou součástí standardní knihovny. Totožné části kódu se nacházejí v architektuře vlastního modelu RO.

```
class L2Normalization(tf.keras.layers.Layer):
    def call(self, inputs, **kwargs):
        return tf.math.l2_normalize(inputs, axis=1)

class ModelHandler:
    def __init__(self, model_path=r"Program\\my_face_recognition_model.h5"):
```

```
        self.model_path = model_path
        self.model = None

    def load_model(self):
        if self.model is None:
            self.model = tf.keras.models.load_model(
                self.model_path,
                custom_objects={
                "L2Normalization": L2Normalization,
                "triplet_loss": self.triplet_loss
                }
            )
        return self.model

    @staticmethod
    def triplet_loss(y_true, y_pred, margin=1.5):
        anchor, positive, negative = tf.split(y_pred, num_or_size_splits=3,
        ↪  axis=0)
        pos_dist = tf.reduce_sum(tf.square(anchor - positive), axis=1)
        neg_dist = tf.reduce_sum(tf.square(anchor - negative), axis=1)
        basic_loss = pos_dist - neg_dist + margin
        return tf.reduce_mean(tf.maximum(basic_loss, 0.0))
```

11.1.2 Generování embeddingů

Pro generování embeddingů slouží metoda `generate_embedding`, která načte model a poté model vytvoří embedding na základě zpracovaného vstupního obrazu.

```
def generate_embedding(self, image):
    model = self.load_model()
    embedding = model.predict(image)
    return embedding[0]
```

11.2 Správa databází

Pro ukládání a správu vytvořených embeddingů slouží třída `DatabaseHandler`. Tato třída poskytuje veškerou funkcionalitu pro správu SQLite databáze a FAISS indexu. SQLite databáze obsahuje metadata o osobách, zatímco index ukládá embeddingy pro rychlé porovnání.

11.2.1 Vytvoření databáze a FAISS indexu

Při inicializaci třídy se automaticky vytváří SQLite databáze a FAISS index, pokud již neexistují. SQLite tabulka je navržena tak, aby obsahovala tabulku, do které se ukládá jméno, příjmení a unikátní ID, které se přiřazuje automaticky každé osobě.

```
def create_database(self):
    conn = sqlite3.connect(r"Program\\database\\face_metadata.db")
    c = conn.cursor()
    c.execute("""
        CREATE TABLE IF NOT EXISTS people (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT,
            surname TEXT
        )
    """)
    conn.commit()
    conn.close()
```

```
def load_or_create_faiss_index(self):
    if os.path.exists(self.faiss_index_path):
        index = faiss.read_index(self.faiss_index_path)
        return index
    return faiss.IndexFlatIP(self.embedding_dim)
```

11.2.2 Přidání osoby do databáze

Pro přidání osoby do databáze slouží metoda `add_person_to_database`. Pokud osoba ještě neexistuje v databázi, tak metoda provede již vytvořené předzpracování obrázku, vygenerování embeddingů a uložení metadat do SQLite databáze a embeddingů do FAISS indexu.

```
def add_person_to_database(self, name, surname, images,
    model_handler=ModelHander()):
    conn = sqlite3.connect(self.metadata_path)
    c = conn.cursor()

    c.execute("SELECT id FROM people WHERE name = ? AND surname = ?", (name,
        surname))
    result = c.fetchone()
    if result:
        print(f"Osoba {name} {surname} již existuje v databázi se záznamem ID:
            {result[0]}")
        conn.close()
        return

    preprocessed_faces = []
    for img in images:
        preprocessor = DataPreprocessing(img)
        preprocessed_faces.extend(preprocessor.preprocess_faces())

    embeddings = []
    for face in preprocessed_faces:
        if len(face.shape) == 3:
            face = np.expand_dims(face, axis=0)
        embedding = model_handler.generate_embedding(face)
        embeddings.append(embedding)

    average_embedding = np.mean(embeddings, axis=0).astype("float32")
    self.faiss_index.add(np.array([average_embedding]))

```

```
27      c.execute("INSERT INTO people (name, surname) VALUES (?, ?)", (name,
        ↪  surname))
28      conn.commit()
29      conn.close()
30
31      faiss.write_index(self.faiss_index, self.faiss_index_path)
```

11.3 Porovnávání osob

Pro porovnávání a identifikaci osob slouží třída `Matcher`, která generuje embeddingy za účelem jejich porovnání s daty uloženými v FAISS indexu. Následně hledá nejpodobnější záznam v databázi a na základě identifikovaného indexu embeddingu získává jméno osoby ze SQLite databáze. Tento proces je umožněn díky propojení FAISS indexu a SQLite databáze, kde indexy v FAISS odpovídají záznamům v tabulce SQLite databáze s minimálním posunem.

11.3.1 Vyhledávání ve FAISS indexu

Porovnávání nových embeddingů s daty uloženými v FAISS indexu je realizováno metodou `find_closest_in_faiss_index`, která vyhledává nejpodobnější embeddingy pomocí kosinové podobnosti a vrací unikátní index embeddingu.

```
1  def find_closest_in_faiss_index(self, query_embeddings, top_k=1):
2      query_embeddings = np.array(query_embeddings).astype("float32")
3      distances, indices = self.faiss_index.search(query_embeddings, top_k)
4      return distances, indices
```

11.3.2 Získání jména na základě indexu

Pro převod indexu embeddingu na jméno osoby slouží metoda `get_name_by_index`, která provádí dotaz na SQLite databázi a vrací odpovídající jméno a příjmení.

```python
def get_name_by_index(self, index):
    conn = sqlite3.connect(self.metadata_path)
    c = conn.cursor()
    corrected_index = int(index) + 1
    c.execute("SELECT name, surname FROM people WHERE id = ?",
    ↪  (corrected_index,))
    result = c.fetchone()
    conn.close()
    print(result)
    if result:
        return f"{result[0]} {result[1]}"
    return "Neznámý"
```

11.3.3 Identifikace osob

K identifikaci osob na základě vstupního obrazu slouží metoda `identify_people`, která kombinuje generování embeddingů, jejich porovnání s FAISS indexem a následné získání jména osoby z databáze. Pokud je podobnost embeddingů nižší než definovaná prahová hodnota, je osoba označena jako ”Neznámý”.

```python
def identify_people(self, image):
    embeddings = self.preprocess_and_embed(image)

    if not embeddings:
        return []

    distances, indices = self.find_closest_in_faiss_index(embeddings, top_k=1)

    identified_names = []
    for i, dist in enumerate(distances):
        if dist[0] > 0.995:
            name = self.get_name_by_index(indices[i][0])
            identified_names.append(name)
```

```
14          else:
15              identified_names.append("Neznámý")
16      return identified_names
```

Kromě identifikace osob modul obsahuje funkci `draw_faces`, která umožňuje vizualizaci detekovaných obličejů na vstupním obrázku spolu s identifikovanými jmény. Tato funkce na základě polohy detekovaných obličejů vykreslí obdélníky a nad ně přidá text s odpovídajícím jménem osoby. Pokud je obličej osoby neznámý, je označen jako ”Neznámý”.

12 Uživatelské rozhraní

Aby byla práce s aplikací zaměřenou na rozpoznávání obličeje co nejjednodušší, obsahuje intuitivní grafické uživatelské rozhraní (GUI), které umožňuje snadnou interakci s jejími klíčovými funkcemi. Rozhraní bylo navrženo s důrazem na jednoduchost a přehlednost, aby bylo snadno použitelné i pro uživatele bez hlubších technických znalostí.

Pro implementaci uživatelského rozhraní byla využita knihovna PyQt5, která nabízí široké možnosti práce s grafickými prvky. Další klíčovou knihovnou je OpenCV, která slouží k zpracování obrazových dat, například při načítání a zobrazování fotografií nebo živého videa z kamery.

Hlavní funkce, které uživatelské rozhraní nabízí, zahrnují:

- **Rozpoznávání obličejů na fotografiích**: Uživatel může nahrát fotografii a spustit analýzu, která identifikuje osoby na obrázku na základě databáze známých osob.
- **Rozpoznávání obličejů v reálném čase**: Aplikace umožňuje zapnout kameru a provádět analýzu obličejů přímo na živém obraze.
- **Správu databáze známých osob**: Uživatel může přidávat nové osoby do databáze pomocí formuláře, kde zadá jejich jméno, příjmení a nahraje jednu nebo více fotografií.
- **Zobrazení notifikací**: GUI zobrazuje upozornění informující o průběhu operací a výsledcích akcí, čímž zlepšuje uživatelský komfort a přehlednost.

Tyto funkce jsou přístupné prostřednictvím boční navigační lišty, která umožňuje snadné přepínání mezi jednotlivými částmi aplikace.

Kromě těchto klíčových funkcí aplikace nabízí několik doplňkových vlastností. GUI zobrazuje notifikace o průběhu operací a výsledcích akcí, čímž uživateli poskytuje jasnou zpětnou vazbu. Aplikace také obsahuje úvodní stránku, která poskytuje přehled o jejím účelu, hlavních funkcích a použitých technologiích. V neposlední řadě je uživatelské rozhraní navrženo

tak, aby se dynamicky přizpůsobovalo velikosti okna aplikace - prvky, jako jsou tlačítka, texty nebo obrázky, mění své rozměry a zajišťují přehlednost při jakémkoli rozlišení.

Závěr

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Bibliografie

1. ADJABI, Insaf; OUAHABI, Abdeldjalil; BENZAOUI, Amir; TALEB-AHMED, Abdelmalik. Past, Present, and Future of Face Recognition: A Review. *Electronics*. 2020, roč. 9, č. 8. Dostupné také z: https://www.mdpi.com/2079-9292/9/8/1188.
2. ANTONIADIS, Panagiotis. *How Do Eigenfaces Work?* 2024. Dostupné také z: https://www.baeldung.com/cs/cv-eigenfaces.
3. AUTORŮ, kolekriv. *JEDNODUŠE: Umělá inteligence*. Universum, 2023.
4. BLOG, Deepchecks Community. *Understanding F1 Score, Accuracy, ROC-AUC, and PR-AUC Metrics for Models*. 2024. Dostupné také z: https://www.deepchecks.com/f1-score-accuracy-roc-auc-and-pr-auc-metrics-for-models/?utm_source=chatgpt.com.
5. *CASIA-WebFace*. [B.r.]. Dostupné také z: https://www.kaggle.com/datasets/debarghamitraroy/casia-webface.
6. *Co znamená ladění hyperparametrů?* 2024. Dostupné také z: https://cs.eitca.org/artificial-intelligence/eitc-ai-gcml-google-cloud-machine-learning/introduction/what-is-machine-learning/what-does-hyperparameter-tuning-mean/.
7. CONTIBUTORS, NIST. *Face Recognition Technology (FERET)*. 2011. Dostupné také z: https://www.nist.gov/programs-projects/face-recognition-technology-feret.
8. CONTRIBUTORS, NEC. *A brief history of Facial Recognition*. 2022. Dostupné také z: https://www.nec.co.nz/market-leadership/publications-media/a-brief-history-of-facial-recognition/.
9. FRANÇOIS, CHOLLET. *Deep learning v jazyku Python: knihovny Keras, Tensorflow. Knihovna programátora*. Grada, 2019.

10. GARGARO, David. *The pros and cons of facial recognition technology*. 2024. Dostupné také z: https://www.itpro.com/security/privacy/356882/the-pros-and-cons-of-facial-recognition-technology.

11. HENDL, Jan. *Big data - Věda o datech, základy a aplikace*. GRADA, 2021.

12. KIŠŠ, Martin. *KONVOLUČNÍ NEURONOVÉ SÍTĚ PRO BEZPEČNOSTNÍ APLIKACE*. 2016. Dostupné také z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=132324. Bakalářská práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, FAKULTA INFORMAČNÍCH TECHNOLOGIÍ.

13. *Labeled Faces in the Wild*. [B.r.]. Dostupné také z: https://vis-www.cs.umass.edu/lfw/.

14. LIBERTIES.EU. *7 Biggest Privacy Concerns Around Facial Recognition Technology*. 2022. Dostupné také z: https://www.liberties.eu/en/stories/facial-recognition-privacy-concerns/44518.

15. LYTVYNENKO, Oleksandr. *Machine Learning Algorithms for Face Recognition*. 2022. Dostupné také z: https://codeit.us/blog/machine-learning-face-recognition#what-is-face-recognition.

16. *MS-Celeb-1M (MS1M)*. [B.r.]. Dostupné také z: https://exposing.ai/msceleb/.

17. NELSON, Daniel. *Co jsou to CNN (konvoluční neuronové sítě)?* 2020. Dostupné také z: https://www.unite.ai/cs/what-are-convolutional-neural-networks/.

18. PETROVIČOVÁ, Klára. *Aplikace konvolučních neuronových sítí*. 2019. Dostupné také z: https://nlp.fi.muni.cz/uui/referaty2019/petrovicova_klara/referat.pdf.

19. QINJUN, Li; CUI TIANWEI, Zhao Yan; YUYING, Wu. Facial Recognition Technology: A Comprehensive Overview. *Academic Journal of Computing & Information Science*. 2023.

20. QIONG CAO, LI SHEN, WEIDI XIE, OMKAR PARKHI, ANDREW ZISSERMAN. *VGGFace2 Dataset*. [B.r.].

21. RUDOLF, Pecinovský. *Python - Kompletní příručka jazyka pro verzi 3.11*. Grada, 2022.

22. SERHII, Bondarenko. *Jak funguje VGG16 – neuronová síť pro extrakci obrazových prvků*. [B.r.]. Dostupné také z: https://robotdreams.cz/blog/317-jak-funguje-vgg16-neuronova-sit-pro-extrakci-obrazovych-prvku.

23. SOUKUPOVÁ, Jana. *Možnosti využití technologie rozpoznávání obličejů v kontextu ochrany osobních údajů v EU*. Praha, 2022. Rigorózní práce. Univerzita Karlova, Právnická fakulta, Katedra evropského práva.

24. SUPPORT, Apple. *Informace o vyspělé technologii Face ID*. 2023. Dostupné také z: https://support.apple.com/cs-cz/102381.

25. *Technologie rozpoznávání obličeje podle fotky: Budoucnost, nebo hrozba?* 2024. Dostupné také z: https://prekon.cz/technologie-rozpoznavani-obliceje-podle-fotky-budoucnost-nebo-hrozba/.

26. *The Cambridge Handbook of Facial Recognition in the Modern State*. Cambridge University Press, 2024. Cambridge Law Handbooks.

27. TOXIN. *Konvoluční neuronové sítě pomáhají detekovat a blokovat nevhodný obsah sociálních sítí*. 2021. Dostupné také z: https://www.toxin.cz/blog/blog/konvolu%C4%8Dn%C3%AD-neuronov%C3%A9-s%C3%ADt%C4%9B-pom%C3%A1haj%C3%AD-detekovat-a-blokovat-nevhodn%C3%BD-obsah-soci%C3%A1ln%C3%ADch-s%C3%ADt%C3%AD.

28. VALLA, Tomáš. *Průvodce labyrintem algoritmů*. CZ.NIC, 2022.

29. WIKIPEDIA CONTRIBUTORS. *DeepFace — Wikipedia, The Free Encyclopedia*. 2024. Dostupné také z: https://en.wikipedia.org/w/index.php?title=DeepFace&oldid=1240075590. [Online; accessed 29-October-2024].

30. WIKIPEDIA CONTRIBUTORS. *Facial recognition system — Wikipedia, The Free Encyclopedia*. 2024. Dostupné také z: https://en.wikipedia.org/w/index.php?title=Facial_recognition_system&oldid=1250429409. [Online; accessed 28-October-2024].

31. WIKIPEDIA CONTRIBUTORS. *FERET (facial recognition technology) — Wikipedia, The Free Encyclopedia*. 2024. Dostupné také z: https://en.wikipedia.org/w/index.php?title=FERET_(facial_recognition_technology)&oldid=1232099360.

32. ZACHA, Jiří. *Konvoluční neuronové sítě pro klasifikaci objektů z LiDARových dat*. 2019. Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra kybernetiky.

Zkratky

AI umělá inteligence. 10

ANN umělá neuronová síť. 19, 20

atd. a tak dále. 10, 15

CNN konvoluční neuronová síť. iv, v, 8, 11, 18–20, 22

FERET Facial Recognition Technology. iv, 7

ML strojové učení. 10

např. například. 22

PCA Principal component analysis. iv, 6

Seznam obrázků

4.1 Detekované obličeje z obrázků z datasetu WGGFace2 14
4.2 Předzpracované obličeje z obrázků z datasetu WGGFace2 15
4.3 Augmentace obličeje z obrázku z datasetu WGGFace2 15

5.1 Struktura umělé neuronové sítě 20
5.2 Struktura konvoluční neuronové sítě 21

8.1 Podrobný strom scén programu 29

Seznam tabulek

Přílohy

A Fotky z pokusů

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

B Příloha další