VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

IDENTIFIKACE OSOB VE VIDEOZÁZNAMU Z KVADROKOPTÉRY

IDENTIFICATION OF PERSONS IN THE VIDEO FROM QUADCOPTER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **TOMÁŠ MOJŽIŠ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. TOMÁŠ GOLDMANN**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav inteligentních systémů (UITS) Akademický rok 2019/2020

Zadání bakalářské práce

Student: **Mojžiš Tomáš**
Program: Informační technologie
Název: **Identifikace osob ve videozáznamu z kvadrokoptéry**
Identification of Persons in the Video from Quadcopter
Kategorie: Umělá inteligence

Zadání:

1. Seznamte se s problematikou identifikace osob na základě snímků obličeje. Zjistěte, jaká řešení se používají pro identifikaci osob ve videozáznamech z CCTV kamer a z kamer dronů.
2. Prostudujte vybraná řešení pro identifikaci osob z kvadrokoptéry a zjistěte jejich limity (např. minimální rozlišení obličeje). Sumarizujte vhodné metody a algoritmy používané pro identifikaci osob na základě snímku obličeje.
3. Navrhněte algoritmus, který ve videozáznamu bude provádět identifikaci osob na základě snímku obličeje. Identifikace se bude provádět vůči databázi, která bude obsahovat fotografii obličeje dané osoby a její ID.
4. Navržené řešení implementujte v libovolném programovacím jazyce a vytvořte jednoduché uživatelské rozhraní. Nasnímaná data z dronu se budou zpracovávat offline.
5. Proveďte experimenty na několika videích a zaměřte se na vyhodnocení úspěšnosti identifikace osob.

Literatura:

- VIOLA, Paul a Michael JONES. Robust Real-Time Face Detection. *International Journal of Computer Vision* [online]. Boston: Kluwer Academic Publishers, 2004, 57(2), 137-154.
- DAMJANOVSKI, Vlado. CCTV, 3rd Edition. 3. Butterworth-Heinemann, 2013. ISBN 9780124045576.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Goldmann Tomáš, Ing.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 14. května 2020
Datum schválení: 31. října 2019

Abstrakt

Cieľom tejto práce je vytvoriť aplikáciu, ktorá bude schopná rozpoznávať ľudí podľa tváre na základe vytvorenej databázy zo záznamov z drona. Databáza pozostáva z fotografií osôb, ktoré sa majú na videu identifikovať. Výstupom aplikácie je video, v ktorom sú hľadané osoby označené ich menom. Pri riešení práce bolo porovnaných niekoľko existujúcich metód, založených na neurónových sieťach, ktoré riešia detekciu tváre a rozpoznávanie tvárí. Vo finálnom riešení bol použitý detektor MTCNN spolu s extraktorom vektorov príznakov na základe ArcFace. Vytvorená multiplatformová aplikácia umožňuje rozpoznať osoby v záznamoch z drona aj pri šírke tváre menšej ako 20 pixelov. Celková funkčnosť aplikácie bola otestovaná na vlastnej dátovej sade pozostávajúcej zo záznamov z drona.

Abstract

The aim of this thesis is to make an application capable of recognizing people's faces based on a user-created database in drone footage. The database is made of pictures of people that should be identified in the footage. The output of this application is a video where the demanded people are labeled with their names. Some face detection and recognition state of the art solutions based on neural networks are compared in this work. The final solution consists of the MTCNN detector and a face embedding extractor based on ArcFace. The created multiplatform application allows to recognize people in drone footage even with face width of less than 20 pixels. The final solution was tested on a private dataset comprised of drone footage.

Kľúčové slová

umelá inteligencia, počítačové videnie, rozpoznávanie tváre, detekcia tváre, dron, kamera, cctv, cmos, ccd, konvolučná neurónová sieť, CNN, MTCNN, ArcFace, OpenCV, Python, PyQt, biometria

Keywords

artificial intelligence, computer vision, face recognition, face detection, drone, camera, cctv, cmos, ccd, convolutional neural network, CNN, MTCNN, ArcFace, OpenCV, Python, PyQt, biometrics

Citácia

MOJŽIŠ, Tomáš. *Identifikace osob ve videozáznamu z kvadrokoptéry.* Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Goldmann

Identifikace osob ve videozáznamu z kvadrokoptéry

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Tomáša Goldmanna. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Tomáš Mojžiš
28. mája 2020

Poďakovanie

Chcel by som sa poďakovať pánovi Ing. Tomášovi Goldmannovi za jeho odborné rady pri tvorbe tejto práce. Tiež ďakujem rodine a priateľom za podporu pri štúdiu a písaní tejto práce. Ďakujem im aj za ochotu a súhlas nachádzať sa na testovacích záberoch z drona použitých v tejto práci.

Obsah

Kapitola 1

Úvod

Identifikácia človeka na základe tváre je pre ľudský mozog zdanlivo jednoduchá úloha. Počítač však obrazové dáta vidí len ako skupinu číselných hodnôt. S vývojom hardvéru sa zvyšuje výpočtový výkon počítačov a vyvíjajú sa nové algoritmy, ktoré umožňujú túto zložitú úlohu splniť aj na voľne dostupných zariadeniach ako je notebook alebo telefón. Ak sú podmienky snímania tváre kontrolované, tak v niektorých prípadoch dosahujú počítače vyššiu presnosť ako samotný človek.

Pri záberoch z dronov sú však podmienky zložitejšie. Tváre sú často malé a snímané pod vyšším uhlom, avšak oproti klasickým statickým kamerám je snímanie ľudí z drona omnoho jednoduchšie. Aj rýchlym preletom ponad dav ľudí je dron schopný získať dostatočné množstvo dát pre identifikáciu vybraných jedincov v dave.

Cieľom tejto práce je preštudovať moderné riešenia umožňujúce detekciu a rozpoznávanie osôb a porovnať ich presnosť pre účel rozpoznávania osôb zo záberov z drona. Následne tieto riešenia použiť pri vytváraní aplikácie s jednoduchým užívateľským rozhraním, ktorá bude schopná získané zábery spracovať, nájsť a označiť v nich konkrétnu, používateľom definovanú množinu ľudí.

Kapitola 2 sa venuje konkrétne oblasti rozpoznávania osôb podľa tvárí. V prvej časti sa venuje základom biometrie a biometrických systémov a postupne prejde k rozpoznávaniu tvárí z biometrického hľadiska. V nasledujúcej časti sú predstavené zdroje obrazových dát, ich vlastnosti a popis kamerových bezpečnostných systémov CCTV. Ďalšia časť sa venuje dronom, kde popisuje ich využitie a predstavuje niekoľko konkrétnych modelov dronov. Posledná časť tejto kapitoly sa venuje niekoľkým moderným riešeniam, založeným na neurónových sieťach, plniacich účel detekcie alebo rozpoznávania osôb podľa tvárí.

V kapitole 3 je predstavený úvod do počítačového videnia, sú tu zhrnuté niektoré algoritmy úzko súvisiace s úlohou rozpoznávania tvárí v obrazových dátach. Posledná časť tejto kapitoly sa venuje neurónovým sieťam – od základnej štruktúry neurónu až po konkrétne vrstvy a súvislosťami medzi nimi. Sú tu popísané aj konvolučné neurónové siete, ktoré sú kľúčové pri spracovaní obrazu.

V kapitole 4 sú predstavené požiadavky na výslednú aplikáciu, je tu porovnaných niekoľko existujúcich riešení pre detekciu, rozpoznávanie a sledovanie tvárí vo videu. Následne je tu popísaný výber konkrétnych algoritmov do výslednej aplikácie, popis databázy tvárí a proces jej vytvárania, návrh a implementácia výslednej aplikácie.

Posledná kapitola 5 obsahuje experiment, kde bol algoritmus z aplikácie odskúšaný na menšej súkromnej dátovej sade záznamov z drona. Zameriava sa na presnosť rozpoznávania v závislosti od rozlíšenia tváre získanej z video záznamu.

Kapitola 2

Rozpoznávanie osôb podľa tváre

Rozpoznávanie konkrétnych ľudí na základe ich tváre je pre človeka zdanlivo jednoduchá úloha. Človek si s touto úlohou poradí aj napriek rôznym svetelným podmienkam alebo rôznym doplnkom, ktoré zakryjú časť tváre. Pre počítač je táto úloha omnoho náročnejšia. Vďaka nárastu výpočtového výkonu a stále sofistikovanejším algoritmom je možné poradiť si s rozpoznaním tváre už aj na obyčajných mobilných zariadeniach. Táto kapitola obsahuje informácie o biometrii, vyhodnocovaní výsledkov biometrických systémov, procese rozpoznávania osôb podľa tváre a v závere sú popísané zdroje obrazových dát pre tento účel.

2.1 Biometria

Podľa [26] je biometria veda o určovaní identity jednotlivca na základe fyzických, chemických alebo povahových vlastností tohto človeka. Význam biometrie v modernej spoločnosti bol posilnený potrebou systémov, ktoré sa spoliehajú na vysokú presnosť určenia identity. Predpokladom je, že jednotlivca je možné rozoznať podľa jeho vlastností. Medzi fyzické vlastnosti patria, napríklad odtlačky prstov, štruktúra dúhovky alebo tvár človeka. Medzi povahové vlastnosti môžme zaradiť napríklad hlas alebo chôdzu.

Hlavnou úlohou systémov pre overovanie identity je určiť alebo overiť identitu jednotlivca. Táto úloha môže byť dôležitá pre viacero účelov, avšak vo väčšine využití je jej primárny zámer zabrániť nepovoleným osobám v prístupe ku chráneným prostriedkom. Tradičné metódy určovania identity človeka sú založené na vedomosti (napr. heslo) alebo na základe bezpečnostného tokenu (napr. ID karta). Tieto možnosti identifikácie však môžu byť ľahko odcudzené, zabudnuté, alebo stratené. Biometrická verifikácia ponúka prirodzené a spoľahlivé riešenie k automatickým možnostiam rozpoznania indivíduí na základe ich biologických charakteristík.

Biometrický systém

Informácie prevzaté z [26]. Biometrický systém je v podstate systém pre rozpoznávanie vzorov. Získava biometrické dáta od indivídua, extrahuje si z dát potrebnú sadu príznakov, porovná sadu s ostatnými v databáze a vykoná akciu na základe výsledku porovnania. Podľa tohto vzoru sa dá na biometrický systém pozerať ako na systém so štyrmi modulami: senzor, extraktor príznakov, modul pre porovnávanie a databáza príznakov.

1. **Senzor:** Pre získanie biometrických dát je nutné zvoliť vhodný senzor. Príliš citlivý senzor môže spôsobiť vysokú mieru zamietnutí aj pri osobe, ktorá by mala mať prístup.

Pri senzore, ktorý má slabú citlivosť by to bolo naopak - prístup by dostali aj nepovolené osoby. Väčšina biometrických dát sa získava z obrázkov (okrem napr. hlasu, alebo zápachu) a kvalita dát je ovplyvnená aj vlastnosťami kamery.

2. **Extraktor príznakov:** Najskôr je kontrolovaná kvalita dát získaných zo senzorov. Dáta zvyčajne prejdú algoritmom na skvalitnenie signálu. Ak je však kvalita stále nízka, používateľ môže byť vyzvaný, aby znova poskytol svoje biometrické dáta. Dáta sú ďalej spracované a sú z nich vybrané dané charakteristiky/príznaky.

3. **Modul pre porovnávanie:** Získané príznaky sú porovnané s už uloženými príznakmi v databáze. Pre porovnania je určené skóre zhody, na základe ktorého sa vyhodnotí celkový výsledok.

4. **Databáza:** Je úložisko biometrických informácii. Pri napĺňaní databázy sú extrahované príznaky uložené spolu s informáciami o danej osobe, napr. meno, adresa, heslo, atď. Proces napĺňania databázy môže a nemusí byť realizovaný pod odborným dozorom. Dáta môžu byť získané z jednej, ale aj z viacerých biometrických vzoriek a následne podľa potreby ďalej spracované.

Verifikácia versus identifikácia

Podľa [28] biometrické systémy pracujú v dvoch rôznych módoch – *verifikácia a identifikácia*.

Verifikácia je synonymom pre autentifikáciu. Používateľ zadá svoj prihlasovací údaj (napr. prihlasovacie meno, PIN, použije čipovú kartu apod.) a tým tvrdí, že je konkrétny používateľ systému. Následne poskytne svoju biometrickú charakteristiku a biometrický systém overí, či je používateľ naozaj tým, kým tvrdí že je. Poskytnutá biometrická charakteristika je porovnaná len s jednou konkrétnou v databáze (tzv. *one-to-one match*). Ak sú biometrické charakteristiky dostatočne zhodné, je užívateľ oprávnený vstúpiť do systému. Zhoda sa dá popísať ako *skóre zhody* alebo ako *vzdialenosť* dvoch charakteristík. Pri *skóre zhody* sa porovnáva podobnosť dvoch charakteristík. Ak je skóre vyššie ako prahová hodnota, používateľ je oprávnený. Pri *vzdialenosti* sa porovnajú biometrické charakteristiky a ak ich vzdialenosť je menšia ako prahová hodnota, tak je užívateľ oprávnený.

Identifikácia môže byť rozdelená na *pozitívnu* a *negatívnu*. Pri *pozitívnej* identifikácii používateľ poskytne svoje biometrické údaje a systém len na základe týchto údajov vyhodnotí či je používateľ oprávnený používať systém na základe databázy používateľov systému. Účelom *negatívnej* identifikácie je zamedziť, aby jeden používateľ mal v systéme viac identít (napr. falošné doklady).

Pri oboch typoch identifikácie sú vstupné biometrické charakteristiky porovnané so všetkými charakteristikami v databáze. Výstupom biometrického systému je v tomto prípade buď používateľ s najväčšou zhodou alebo to, že pre zadané biometrické charakteristiky neexistuje v databáze žiadny používateľ.

Vyhodnocovanie porovnávania biometrických charakteristík

Informácie prevzaté z [26]. V systémoch, kde je pre identifikáciu používané heslo, je potrebné aby sa zadané heslo presne zhodovalo s heslom uloženým v databáze. V biometrických systémoch sa málokedy stane, že poskytnutá biometrická charakteristika sa presne zhoduje

s charakteristikou uloženou v databáze. Dôvodom sú nepresné senzory (napr. zle oskenovaný odtlačok prsta kvôli chybe senzora), zmeny v biometrických charakteristikách používateľa (napr. respirátor zmení hlas danej osoby pri rozpoznávaní hlasu, starnutie alebo ochlpenie na tvári pri rozpoznávaní tvárí), zmeny osvetlenia (najmä pri rozpoznávaní tvárí) alebo chybná interakcia používateľa so senzorom (napr. čiastočne načítaný odtlačok prsta, zakrytá dúhovka apod.). Popravde pri presnej zhode biometrických dát sa skôr dá počítať s pokusom o útok pomocou duplikácie biometrických dát z databázy.

Metriky vyhodnocovania biometrických systémov

Informácie v tejto kapitole sú prebraté z [9]. Pri porovnávaní dvoch biometrických charakteristík môže na základe skóre zhody dôjsť k nasledujúcim štyrom prípadom:

- Používateľ A je prijatý ako používateľ A – **správne prijatie** (angl. *True Accept*)
- Používateľ A je prijatý ako používateľ B – **chybné prijatie** (angl. *False Accept*)
- Používateľ A nie je prijatý ako používateľ A – **chybné odmietnutie** (angl. *False Reject*)
- Používateľ B nie je prijatý ako používateľ A – **správne odmientnutie** (angl. *True Reject*)

z vyššie uvedených prípadov odvodzujeme **chybové miery** biometrických systémov.

Miera chybného prijatia (angl. *False Accept Rate*, **FAR**) je pravdepodobnosť, že biometrický systém vyhodnotí dve biometrické charakteristiky od rôznych osôb ako zhodné. Tým môže povoliť neoprávnenej osobe prístup do systému. Pravdepodobnosť sa dá vypočítať ako:

$$FAR = \frac{\text{počet chybných prijatí}}{\text{celkový počet porovnaní rôznych charakteristík}} \tag{2.1}$$

Miera chybného odmietnutia (angl. *False Reject Rate*, **FRR**) je pravdepodobnosť, že biometrický systém vyhodnotí dve biometrické charakteristiky od tej istej osoby ako rôzne. To zabráni oprávnenej osobe prístup do systému. Pravdepodobnosť sa dá vypočítať ako:

$$FRR = \frac{\text{počet chybných odmietnutí osoby A}}{\text{celkový počet porovnaní charakteristík osoby A}} \tag{2.2}$$

Miera neschopnosti nasnímať (angl. *Failure To Aquire*, **FTA**) udáva mieru kedy senzor nedokázal nasnímať biometrické dáta, aj keď boli prítomné. Táto miera slúži hlavne k hodnoteniu kvality senzorov pre dané biometrické dáta.

Miera neschopnosti zaregistrovať (angl. *Failure To Enroll*, **FTE**) určuje podiel užívateľov, ktorých sa systém nevie naučiť rozpoznať ku všetkým užívateľov systému. Súvisí to s kvalitou biometrickej charakteristiky, kde charakteristiky s nedostatočnou kvalitou systém nie je schopný zaregistrovať. FTE predstavuje údaj, ktorý ohodnotí schopnosť algoritmu pracovať aj s menej kvalitnými biometrickými charakteristikami.

Miera chybnej zhody (angl. *False Match Rate*, **FMR**) udáva podiel chybne prijatých osôb k celkovému počtu. Oproti FAR nie sú do celkového súčtu brané pokusy, ktoré boli neúspešné z dôvodu chybného snímania alebo neschopnosti zaregistrovať (FTA a FTE). FMR sa počíta nasledovne:

$$\mathrm{FMR(T)} = \int_{T}^{1} p_i(s|H_i)ds, \tag{2.3}$$

kde T je prah rozhodovania, H_g je výrok, že porovnávané dáta sú zhodné, p_i je hustota pravdepodobnosti, že výrok v zátvorke je pravdivý a s je skóre zhody.

Miera chybnej nezhody (angl. *False Non-Match Rate*, **FNMR**) je podiel chybne neprijatých osôb k celkovému počtu. Oproti FRR nie sú do celkového súčtu brané pokusy, ktoré boli neúspešné z dôvodu chybného snímania alebo neschopnosti zaregistrovať (FTA a FTE). FNMR je definovaná vzorcom:

$$\mathrm{FNMR(T)} = \int_{0}^{T} p_g(s|H_g)ds, \tag{2.4}$$

kde T je prah rozhodovania, H_g je výrok, že porovnávané dáta sú rozdielne, p_g je hustota pravdepodobnosti, že výrok v zátvorke je pravdivý a s je skóre zhody.

Krivka ROC (*Receiver Operating Characteristic*) popisuje vlastnosti daného systému. Vzhľadom na to, že FMR a FNMR menia svoje hodnoty opačnými smermi v závislosti na hodnote prahu, je potrebné systém hodnotiť pomocou krivky ROC. Tieto krivky sú v dnešnej dobe štandardom pre popis vlastností týchto systémov. Jej alternatívou je DET (*Detection Error Trade-off*), ktorá sa líši len reprezentáciou nanášania hodnôt na graf. ROC krivka znázorňuje hodnoty FMR vzhľadom k FNMR pre rôzne hodnoty prahu. Čím je systém lepší, tým viac krivka inklinuje k ľavému hornému rohu (žiadne chybné prijatie ani chybné odmietnutie). Vďaka tomuto je možné ROC krivku vyhodnotiť pomocou obsahu plochy pod krivkou (tzv. *AUC – Area Under Curve*). Príklad niekoľkých kriviek ROC je na obrázku 2.1.

Faktory biometrických charakteristík pre identifikáciu osôb

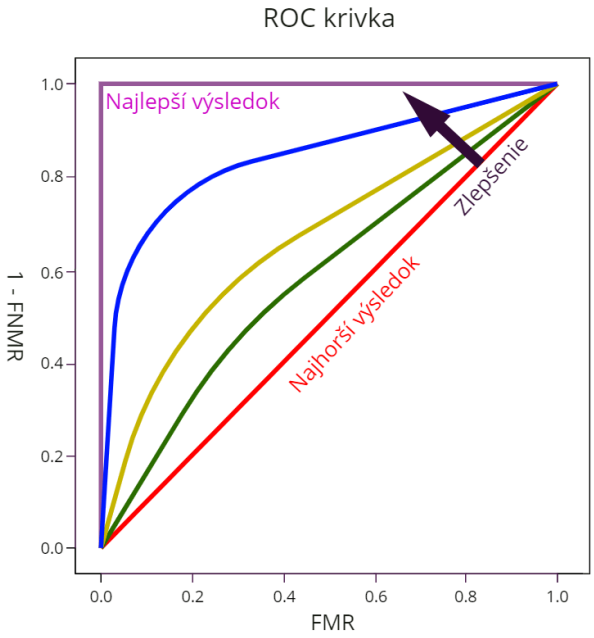
Výber správnej biometrickej charakteristiky záleží na type aplikácie. Každá charakteristika má svoje výhody aj nevýhody. V [28] Jain a kol. určili sedem faktorov, ktoré určujú vhodnosť konkrétnych fyzických alebo povahových charakteristík na základe danej aplikácie.

1. **Univerzalita:** Každý užívateľ používajúci aplikáciu by mal mať danú charakteristiku. Tento faktor ovplyvňuje mieru FTE (*failure to enroll*).

2. **Unikátnosť:** Daná charakteristika by mala byť dostatočne odlišná pre každého používateľa aplikácie. Inak bude miera FAR (*false match rate*) vysoká.

3. **Trvalosť:** Biometrická charakteristika používateľa by sa nemala vo veľkej miere meniť s ohľadom na porovnávací algoritmus. Ak sa charakteristika mení príliš rýchlo bude miera FRR (*false non-match rate*) vysoká.

4. **Merateľnosť:** Charakteristika by mala byť ľahko odoberateľná a digitalizovaná, aby používateľovi proces odoberania nespôsoboval nepríjemnosti. Ďalej by odobrané dáta



Obr. 2.1: Príklad ROC krivky. Čierna šípka znázorňuje akým smerom sa musí krivka posúvať, aby to znamenalo zlepšenie systému. Obrázok prevzatý z [8].

mali byť jednoducho spracovateľné, aby bolo možné extrahovať reprezentatívnu množinu príznakov. Tento faktor ovplyvňuje miery FTE (*failure to enroll*) a FTA (*failure to acquire*) a presnosť rozpoznávania.

5. **Výkon:** Požiadavky na výpočtový výkon pre získanie danej charakteristiky musia spĺňať obmedzenia aplikácie.

6. **Prijateľnosť:** Používatelia aplikácie by mali byť ochotní poskytnúť svoje biometrické dáta danému systému.

7. **Bezpečnosť:** Myslené ako miera jednoduchosti vydávania sa za iného používateľa systému imitáciou jeho biometrickej vlastnosti. Napríklad falošné prsty, alebo napodobňovanie správania.

Rozpoznávanie tvárí

Ľudská tvár má oproti iným biometrickým charakteristikám mnoho výhod. Od používateľa nevyžaduje žiadnu špeciálnu interakciu (ako napr. priloženie prstu pri snímaní odtlačkov prstov), nie je invazívna a jej získanie je bezkontaktné, čo znižuje riziko prenosu nečistôt a s tým spojené zdravotné riziká. Snímanie tváre je v dnešnej dobe veľmi jednoduché pomocou bežne dostupných kamier a fotoaparátov. Tie sú popísane v sekcii 2.2.

História rozpoznávania tvárí

Za výtvorom prvého automatického systému na rozpoznávanie tvárí v roku 1973 stojí Takeo Kanade [30]. Po pár rokoch v roku 1988 aplikovali Sirovich a Kirby PCA (*Principle Component Analysis*) [31] na problém rozpoznávania tvárí. Ich výskumy [50] ukázali, že je

potrebných menej ako sto hodnôt na zakódovanie správne zarovnanej a normalizovanej fotografie tváre. V roku 1991 prišli Turk a Pentland s metódou *Eigenfaces* [56]. Táto metóda umožnila pomerne spoľahlivé automatické rozpoznávanie tvárí v reálnom čase. Aj napriek tomu, že bol tento prístup trochu obmedzený environmentálnymi faktormi, vytvoril významný záujem v zdokonaľovaní systémov pre rozpoznávanie tvárí.

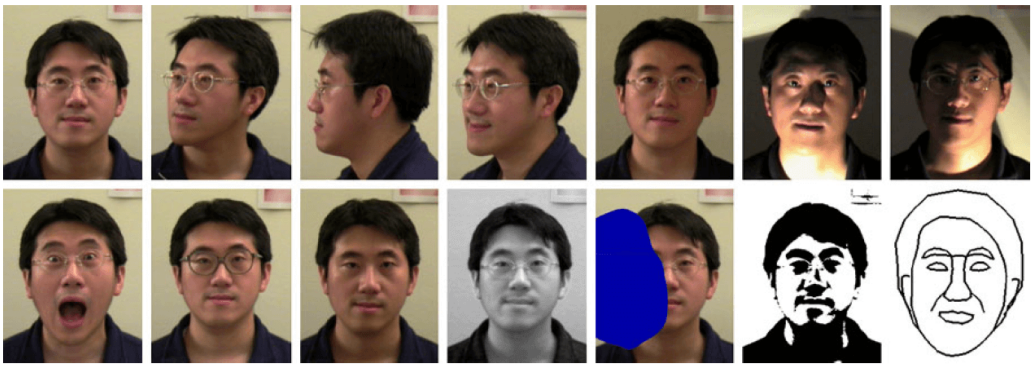
V dnešnej dobe je rozpoznávanie tvárí na významne vyššej úrovni. Podľa [52] už v roku 2011 za umelých podmienok, kde sa dá kontrolovať osvetlenie, póza a mimika, prekonali automatické systémy pre rozpoznávanie tvári aj samotného človeka, hlavne v prípade kedy databáza obsahovala veľké množstvo identít. Avšak za bežných okolností je stále veľké množstvo problémov znižujúcich presnosť automatických systémov.

Faktory ovplyvňujúce rozpoznávanie tvárí

Oproti iným biometrickým charakteristikám má ľudská vysokú vnútro-triednu variabilitu. Hlavnými faktormi sú aj tieto:

- Osvetlenie scény
- Mimika tváre
- Pozícia tváre voči kamere
- Ochlpenie tváre a vlasy
- Rôzne módne doplnky (okuliare, šatka apod.)
- Starnutie
- Prekrytie časti tváre cudzím objektom
- Farba a jas obrazu

Niektoré z týchto faktorov sa dajú minimalizovať pomocou rôznych algoritmov predspracovania obrazu popísaných v sekcii 3.2.



Obr. 2.2: Znázornenie vnútro-triednej variability tváre. Obrázok prevzatý z [20].

Využitie rozpoznávania tvárí

Systém pre rozpoznávanie osôb podľa tvárí ako biometrický systém funguje v dvoch módoch verifikácia a identifikácia popísaných v časti 2.1. V oboch módoch môžme rozpoznávanie osôb podľa tvárí nájsť v dnešnej dobe naozaj všade. Príkladom aplikácie je často napr. bezpečnosť (kontrola vstupu do budov, kontrola na letiskách, kontrola dochádzky, CCTV (viď. sekcia CCTV v časti 2.2), prístup do rôznych zariadení ako osobné počítače, mobilné telefóny apod.), zábavný priemysel (rôzne hry ovládané mimikou alebo pohybmi tváre) alebo na sociálnych sieťach a rôznych aplikáciách pre správu fotografii, ktoré vedia fotografie rozdeliť podľa osôb nachádzajúcich sa na fotografiách. V implementačnej časti tejto práce sa pracuje so systémom rozpoznávania tvárí v móde identifikácie.

Proces rozpoznávania tváre

Rozpoznávanie tvárí je problém rozpoznávania vzorov, kde tvár je reprezentovaná ako trojrozmerný objekt, ktorý je ovplyvňovaný rôznymi faktormi (viď. Faktory ovplyvňujúce rozpoznávanie tvárí v časti 2.1) a je potrebné ho rozoznať na základe poskytnutých fotografií danej osoby. Väčšina systémov pracuje s dvojrozmernými snímkami tváre, avšak sú aplikácie, kde je potrebné zaručiť vyššiu úroveň bezpečnosti a je tak potrebné použiť trojrozmerné modely tvárí. Podľa [52] je systém pre rozpoznávanie osôb podľa tvárí možné rozdeliť na štyri časti:

1. **Nájdenie tváre** – Tvár je potrebné najskôr na snímke nájsť. Ak sa jedná o video, je možné tiež využiť algoritmy na sledovanie objektov, ktoré tvár po jej detekcii vedia vo videu sledovať. Nájdenie tváre poskytne hrubý odhad polohy tváre a jej veľkosti. Bližšie informácie o tvári poskytnú unikátne body nájdené na tvári (tzv. *landmarks*,väčšinou dva body pre oči, jeden pre nos a dva pre ústa), ktoré sa dajú použiť v nasledujúcich krokoch spracovania tváre.

2. **Normalizácia tváre** – Nájdenú tvár je pre vyššiu presnosť rozpoznávacích algoritmov potrebné normalizovať. Je potrebné minimalizovať efekty osvetlenia a upraviť polohu tváre ak je napríklad natočená (viď. sekcia 3.2).

3. **Extrakcia príznakov** – z normalizovanej tváre sa extrahujú príznaky potrebné na rozpoznávanie tvárí a použijú sa pri hľadaní podobných tvárí.

4. **Hľadanie podobných tvárí** – Extrahované príznaky sa porovnávajú s príznakmi uloženými v databáze. Pri verifikácii (1:1) je výstup systému *áno* alebo *nie*, pri identifikácii (1:N) je výstupom jedna alebo viacero podobných tvárí s informáciou o miere zhody s danou tvárou. V prípade, že skóre zhody je nižšie (alebo vzdialenosť tvári je vyššia) ako zadaný prah je tvár vyhlásená za neznámu.

Presnosť týchto systémov záleží na kvalite extrahovaných príznakov, čo záleží na správnej lokalizácii a normalizácii tváre.

Face Embedding

V roku 2015 prišiel tím firmy Google s novým riešením pre rozpoznávanie osôb podľa tváre. Jeho názov je FaceNet [47]. Učí sa mapovať tvár zo snímky priamo do uzavretého

Euklidovského priestoru, kde vzdialenosť dvoch vektorov príznakov (tzv. *embeddings*) extrahovaných zo snímok tvárí priamo odpovedá odlišnosti týchto tvárí. Vďaka tomu sú úlohy ako identifikácia, verifikácia a hľadanie podobných tvárí jednoducho implementovateľné. Predchádzajúce riešenia používajúce hlboké neurónové siete používajú klasifikačnú vrstvu (angl. *classification layer*) trénovanú na množine známych tvárí, ktorá je následne spojená s vrstvou, ktorá má oveľa menej umelých neurónov ako predchádzajúca vrstva (angl. *bottleneck layer*) a tá má za úlohu zovšeobecniť rozpoznávanie tvárí aj mimo trénovacích dát. Sieť FaceNet je trénovaná, aby jej výstupom bol priamo 128-dimenzionálny vektor príznakov. Je trénovaná pomocou funkcie *triplet loss*. Funckia *triplet loss*, architektúra a výsledky na datasetoch tejto siete sú popísané v časti 3.3.

2.2 Kamery

Aby sme boli schopní efektívne nájsť a identifikovať ľudí z fotografií alebo video záznamu, je potrebné, aby bol obraz v dostačujúcej kvalite. V dnešnej dobe to nie je problém. Pri vysokých rozlíšeniach sa ľudská tvár dá rozoznať, aj keď je pomerne vzdialená od objektívu. Na druhej strane však s vyšším rozlíšením narastajú požiadavky na výpočtový výkon pri spracovaní obrazu.

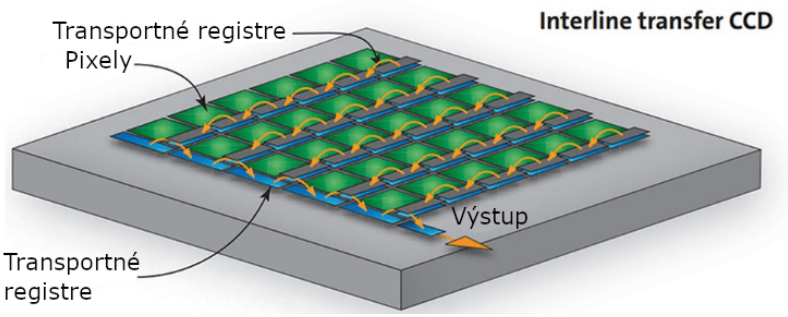
Kamery sa dajú rozdeliť podľa mnohých kritérií. z hľadiska spracovania obrazu sa delia na analógové a digitálne. Ďalej sa delia podľa samotného typu snímacieho čipu, z ktorých sú najviac používané CMOS a CCD. Každá kamera ma potom ďalšie vlastnosti, napr. rozlíšenie, svetelná citlivosť, IR prisvetlenie apod.

Typy snímacích čipov

V dnešnej dobe sú najviac používané snímače typu CCD (angl. *Charge-Coupled Device*) a CMOS (angl. *Complementary Metal Oxide Semiconductor*). Snímací čip je polovodičový prvok, ktorý má tvar mriežky, kde jednotlivé bunky sú nazývané pixely. Fungujú na princípe fotovoltaiky - dopadom svetla na polovodič dochádza k uvoľňovaniu a hromadeniu elektrónov. Snímače typu CMOS a CCD majú rozdielnu architektúru, vďaka čomu majú aj rozdielne vlastnosti.

CCD snímače

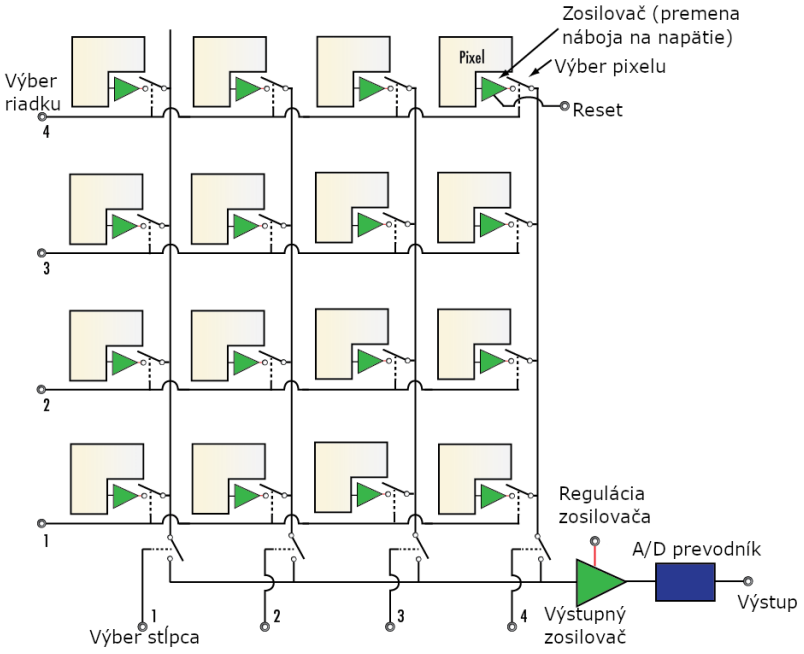
Informácie v odseku pochádzajú z [1]. Pri CCD snímačoch obraz prechádza šošovkou a dopadá na dvojrozmerné pole pixelov. Každý pixel naakumuluje elektrický náboj priamo úmerný intenzite svetla na danom mieste. Po expozícii riadiaci obvod pri full frame CCD senzore posúva náboj nahromadený v každom pixely svojmu susedovi, kde posledný pixel posunie svoj náboj do zosilňovača, ktorý premení náboj na napätie. Pri ILT CCD senzore (*interline transfer CCD*) sa náboj posúva pomocou transportných registrov, ktoré fungujú ako posuvné registre a posúvajú náboj smerom k výstupu bez zmeny jeho hodnoty. Opakovaním tohto procesu sa získa sekvencia napätí, ktorá odpovedá snímanému obrazu. Pri digitálnych kamerách sa táto sekvencia prevedie na digitálne hodnoty pomocou A/D prevodníka a je uložená do pamäte. Pri analógových kamerách je táto sekvencia prevedená na spojitý analógový signál, ktorý je ďalej spracovaný inými obvodmi (napr. je nahrávaný).



Obr. 2.3: Ilustrácia funkcie ILT CCD senzoru. Obrázok prevzatý z [22].

CMOS snímače

Informácie v odseku pochádzajú z [57]. V snímačoch typu CMOS sa za šošovkou taktiež nachádza dvojrozmerné pole pixelov. Každý z týchto pixelov však dokáže naakumulovať náboj a zároveň ho pomocou zosilňovača premeniť na napätie. Pomocou tranzistora sa potom vyberie celý riadok a postupne sa napäťové hodnoty premenia pomocou A/D prevodníku na digitálne, ktoré sú ďalej spracovávané alebo uložené do pamäte.



Obr. 2.4: Zjednodušená schéma CMOS senzoru. Obrázok prevzatý z [42].

Porovnanie CMOS a CCD snímačov

Informácie pochádzajú z [11] a [34]. Aj napriek tomu, že tieto senzory majú toho veľa spoločného, majú často rozdielne vlastnosti. Nedá sa jednoznačne určiť, ktorý je lepší. Záleží na tom, kde ho chceme využiť. CCD senzory zaručujú, že každá hodnota náboja je meraná rovnakým zosilňovačom. To zaručuje vysoko kvalitné a konzistentné dáta s malým šumom. Taktiež vďaka architektúre majú CCD senzory väčšiu plochu pixelu, pretože sa na ňom

nenachádzajú žiadne ďalšie prvky narozdiel od pixelov na CMOS senzore. To znamená že majú vyššiu citlivosť na svetlo a väčšiu hĺbku.

CMOS snímače vďaka svojej architektúre umožňujú rýchlejšie spracovanie obrazu. Často majú A/D prevodníky pre každý stĺpec pixelov. Tieto senzory však neboli navrhnuté preto, aby boli rýchlejšie, ale aby boli lacnejšie. CMOS senzory nepotrebujú veľa externých súčiastok ako napr. hodiny alebo komplexné súčiastky pre čítanie zo senzora a prevod. Všetká elektronika je zabudovaná už priamo na senzore. Vďaka tomuto majú aj nižšiu spotrebu ako CCD senzory. CMOS senzory sú odolné voči tzv. *blooming* (obrázok 2.5), ktorý sa prejavuje, keď je nejaký pixel vystavený veľa svetlu a jeho náboj sa prenesie aj na vedľajšie pixely. Čítanie dát po riadkoch v CMOS senzoroch spôsobuje skreslenie (obrázok 2.5), kedy sa obraz zmení od prečítania predchádzajúcich riadkov. Tento problém sa dá vyriešiť, avšak chce to už súčiastky navyše, čo spôsobí vyššiu cenu.

Obr. 2.5: Vľavo *blooming* vo fotografii vytvorenej CCD senzorom. vpravo skreslenie spôsobené tzv. *rolling shutter* na CMOS senzore. Obrázky prevzaté z [57].

Vlastnosti kamier

Rozlíšenie kamery je jedna z najdôležitejších vlastností. Pri analógových kamerách je rozlíšenie udávané v počte horizontálnych čiar (angl. tzv. *TV Lines*), čo je odvodené podľa televízneho priemyslu. Pri digitálnych kamerách rozlíšenie určuje počet pixelov obrazu. Väčšinou sa udáva v tvare $x \times y$, kde x je počet horizontálnych pixelov a y je počet vertikálnych pixelov. Po vynásobení týchto dvoch čísel dostaneme celkový počet pixelov v obraze, typicky udávaný v jednotke megapixel. Všeobecne platí, že obraz s väčším rozlíšením je kvalitnejší, ale zaberá viac priestoru v pamäti a jeho ďalšie spracovanie je výpočtovo náročnejšie. Pri rozpoznávaní osôb vyššie rozlíšenie pomáha rozpoznať aj malé, od kamery vzdialenejšie tváre, avšak to môže zároveň spomaliť dobu spracovania. Podľa [41] je pri bezpečnostných kamerách potrebných aspoň 70–100 pixelov na pokrytie 1 metra scény na účely pozorovania. Pri úlohách, kde sú potrebné kvalitnejšie obrázky, ako napríklad rozpoznávanie tvárí, by sa množstvo pixelov malo pohybovať okolo 500 pixelov na meter.

Počet snímok za sekundu (ďalej FPS, *Frames per second*) udáva počet snímok, ktoré dokáže kamera zostrojiť za sekundu. Podľa [32] už od 15 FPS sa záznam začína javiť ako spojitý. Bežné kamery v dnešnej dobe poskytujú video s 25-60 FPS. K dispozícii sú

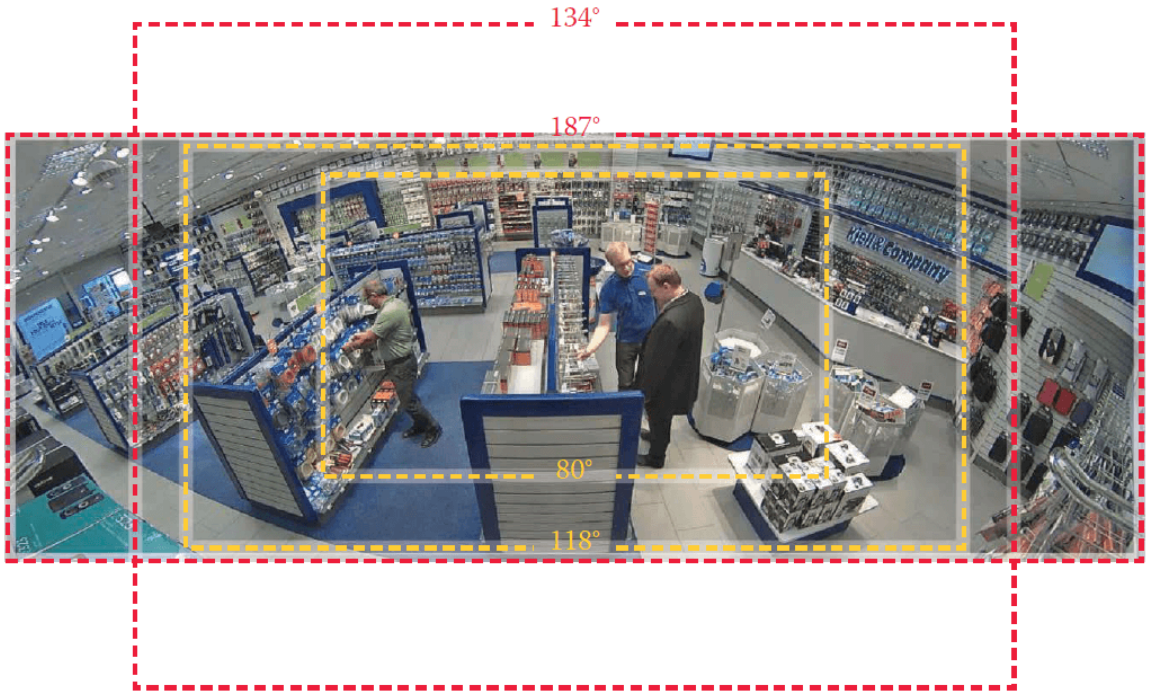
Rozlíšenie	**Počet TV riadkov**	**Šírka**	**Výška**	**Rozlíšenie v počte pixelov**
CIF	400 TVL	352	240	0,086 MPx
VGA	600 TVL	640	480	0,307 MPx
D1 (PAL)	720 TVL	720	576	0,345 MPx
960H (PAL)	960 TVL	960	576	0,460 MPx
HD		1280	720	1 MPx
FullHD		1920	1080	2 MPx
QHD		2560	1440	4 MPx
4K		3840	2160	8 MPx

Tabuľka 2.1: Typické rozlíšenia digitálnych a analógových bezpečnostných kamier. Analógové rozlíšenia sú udané aj v počte TV riadkov.

aj kamery schopné oveľa vyšších FPS, tie však väčšinou slúžia pre potreby spomaleného videa.

Informácie v tomto odseku pochádzajú z [32] a [41]. Svetlo hrá dôležitú úlohu v kvalite obrázku. Všeobecne platí, čím viac svetla, tým lepší obraz. **Svetelná citlivosť** kamery udáva hranicu intenzity svetla, pri ktorej je kamera schopná snímať obraz. Základnou jednotkou je *lux*. Jeden lux znamená, že na plochu $1m^2$ dopadá svetelný tok o veľkosti 1 *lúmen*. Napríklad pri slnečnom letnom dni odpovedá intenzita svetla zhruba 100000 lux, učebne majú intenzitu okolo 250 lux a hviezdna obloha má intenzitu len okolo 0,00005 lux. Pri kamerách, ktoré snímajú farebný obraz, je potrebné, aby mala scéna viac svetla, inak by farby boli skreslené. Podľa [41] niektoré bezpečnostné kamery pri určitej intenzite svetla prepnú do nočného režimu, kedy sa v kamere vypne IR (*infrared*) filter a prepne sa do čiernobieleho režimu pre zníženie šumu a zvýšenie kvality obrazu. Pre zvýšenie citlivosti kamier sa používa napríklad technológia OCL (*on-chip lens*). Ide o techniku, kedy sa pred každý pixel na snímacom čipe umiestni samostatná malá šošovka. Tá zvýši koncentráciu svetla dopadajúcu na pixel a tým sa zvýši samotná citlivosť. Ak je to možné, pre osvetlenie scény sa dá použiť umelé osvetlenie. Najčastejšie sa využívajú volfrámové, volfrámovo-halogénové, halogénové, sodíkové, ortuťové a iné. V niektorých situáciách, kde by svetlo z týchto lámp mohlo vyvolávať pozornosť alebo niekoho otravovať, je vhodnejšie použiť IR svetlo. Niektoré IR osvetľovače produkujú svetlo s vlnovou dĺžkou okolo 850 nm, čo sa pri úplne tmavej miestnosti môže zdať ľudskému oku ako slabý červený odtieň. Sú aj IR osvetľovače, ktoré produkujú svetlo s vlnovou dĺžkou okolo 950 nm, toto svetlo je už pre ľudské oko neviditeľné. Avšak čím vyššia vlnová dĺžka, tým svetlo dosiahne menšie vzdialenosti. V dnešnej dobe sa často používajú LED diódy ako zdroj IR svetla. Bezpečnostné kamery majú často po obvode zapojené IR LED.

Informácie v odseku pochádzajú z [41]. Podľa toho aká šošovka sa nachádza v kamere, môže mať kamera **uhol zorného poľa** až 360°. Pri použití takéhoto širokého uhla zorného poľa je v obraze viac informácií, ale sú skreslené tzv. efektom *rybieho oka*. Toto skreslenie sa dá softvérovo spracovať a odstrániť. Ďalšou nevýhodou je, že je využitá len časť senzoru, čo znižuje rozlíšenie a tým pádom aj kvalitu obrázku. Kamera s uhlom zorného poľa 360° vytvorí snímok v tvare kruhu.



Obr. 2.6: Rôzne uhly zorného poľa dokážu zachytávať rôzne množstvo informácií. Obrázok prevzatý z [41].

CCTV

Informácie v odseku prevzaté z [48]. CCTV (*closed-circuit television*) je TV systém, v ktorom nie je signál určený pre zdieľanie do sveta, ale je monitorovaný, hlavne pre účely dozoru a bezpečnosti. Závisí od strategického umiestnenia kamier a sledovania obrazu týchto kamier. Časť názvu *closed-circuit* alebo „uzavretý okruh“ vyplýva z kabeláže medzi kamerami a zariadeniami určenými na sledovanie záznamu, ktorá je privátna. Staršie CCTV systémy používali malé čierno-biele monitory s nízkym rozlíšením, ktoré nemali žiadne interaktívne schopnosti. Moderné CCTV systémy často poskytujú farebné monitory s vysokým rozlíšením a môžu mať schopnosti ako priblíženie záznamu alebo sledovanie rôznych objektov v zázname. Bežné účely CCTV sú napríklad sledovanie priestoru v zabezpečených areáloch, sledovanie väzňov alebo potencionálne nebezpečných pacientov v nemocniciach, sledovanie dopravnej situácie, sledovanie na miestach, kde človeku hrozí nebezpečie, bezpečnosť budov a miest alebo získavanie záznamov na miestach ako banky, letiska, kasína a pod.

Informácie v odseku prevzaté z [4]. Hlavným rozdielom analógovej CCTV a digitálnej (IP) CCTV je v metóde akou je video doručované. Analógové kamery nahrávajú obraz a posielajú signál pomocou koaxiálnych káblov do DVR (*Digital Video Recorder*). DVR skonvertuje video z analógového signálu do digitálnej podoby, súbor skomprimuje a uloží ho na disk. K DVR musia byť pripojené monitory pre sledovanie záznamov alebo je DVR pripojená k sieti a k záznamom je možný prístup v rámci tejto siete.

Digitálne bezpečnostné kamery založené na IP CCTV systémoch obraz nahrávajú digitálne. Dokážu prijímať a posielať dáta cez sieť, nepotrebujú DVR. K ukladaniu záznamov sa používa tzv. NVR (*Network Video Recorder*) softvér, ktorý beží na počítači v tejto sieti.

Výhody analógovej CCTV sú hlavne v cene a jednoduchosti. Cena analógových kamier je často nižšia ako cena digitálnych. Avšak analógová kamera musí byť zapojená do elektrickej siete aj do DVR. To dáva výhodu digitálnym CCTV systémom, kde sú kamery zapojené pomocou PoE (*Power over Ethernet*) len jedným káblom pre napájanie kamery aj pre získavanie dát z kamery. Kvalita obrazových dát je vyššia pri digitálnych kamerách, čo však prináša vyššie požiadavky na uloženie týchto dát. Digitálne kamery taktiež majú schopnosť bezdrôtového prenosu, čo môže v niektorých prípadoch zjednodušiť celý systém. Digitálny signál sa dá narozdiel od analógového šifrovať, čo zvyšuje bezpečnosť celého systému.

Podľa [41] boli pred 10 rokmi v CCTV kamerách senzory typu CMOS používané len zriedkavo. V dnešnej dobe však dominujú na trhu CCTV kamier vďaka ich zlepšeniu v rámci rozlíšenia a zníženia ceny. Pokroky v tejto technológii sa udiali vďaka dopytu v spotrebnom priemysle.

Informácie v tomto odseku prevzaté z [41]. Mnohé digitálne CCTV kamery sa v dnešnej dobe používajú aj na rôzne typy analýz. Prvou z nich bolo sledovanie pohybu na videách z dôvodu šetrenia diskového priestoru. Ukladali sa len záznamy, v ktorých sa nachádzal pohybujúci sa objekt. Je to základom mnohých pokročilých použití počítačového videnia. V dnešnej dobe sa stretávame napríklad aj s detekciou zakrytia kamery (či už úmyselného alebo neúmyselného), automatické sledovanie nepovoleného vstupu do priestorov, detekcia požiarov, rozpoznávanie a počítanie objektov (ľudí, vozidiel, atď.), rozpoznávanie poznávacích značiek vozidiel, rôzne analýzy pohybu ľudí v priestoroch (napr. obchodoch) alebo rozpoznávanie tvárí. Preto treba na základe použitia kamery zohľadniť jej parametre. Ak je sledovaná scéna tmavá, použiť kameru s IR prisvetlením, ak má byť kamera použitá na rozpoznávanie tvárí, mala by mať dostatočné rozlíšenie. Podľa organizácie Swedish National Laboratory of Forensic Science (SKL) je potrebné, aby tvár človeka mala okolo 80 pixelov na obrázku pre jej rozpoznanie. Štandard publikovaný výborom CELENEC[1] vraví, že 130 - 330 pixelov na m^2 je dostačujúci. Pri priemernej veľkosti ľudskej tváre (zhruba 16 cm) je to zhruba 21 - 52 pixelov pokrývajúcich ľudskú tvár.

Obr. 2.7: Efekt zvýšenia rozlíšenia pre účely rozpoznávania ľudskej tváre. (a) 80 pixelov, (b) 40 pixelov, (c) 20 pixelov pokrývajúcich ľudskú tvár. Obrázok prevzatý z [41].

[1] https://www.cenelec.eu/

2.3 Dron

Bezpilotné lietadlo (UAV – *Unmanned Aerial Vehicle*), tiež všeobecne známe ako dron, je lietajúci objekt, ktorý nemá posádku. Podľa [59] drony patria pod bezpilotné systémy (UAS – *Unmanned Aircraft System*), ktoré sa skladajú zo samotného drona, ovládača na zemi a komunikácie medzi nimi. Drony môžu byť ovládané s rôznou mierou autonómie – trasa môže byť celá predprogramovaná, dron môže byť ovládaný človekom pomocou diaľkového ovládača alebo môže smerovať k manuálne zadanému cieľu (prípadne ho nasledovať, ak je cieľom pohybujúci sa objekt, napr. človek) a vyhýbať sa rôznym prekážkam pomocou kamery a senzorov.

Využitie dronov

Drony sa využívajú vo veľkom množstve odborov. Medzi hlavné dôvody patrí ich cena v porovnaní s lietadlami alebo vrtuľníkmi, ich veľkosť alebo napríklad to, že nepotrebujú posádku. Podľa [59] sa drony využívali primárne v armáde, ale dnes nájdeme využitie pre drony napríklad v komerčnej, vedeckej, rekreačnej alebo hospodárskej oblasti.

V armádnej sfére môžme nájsť rôzne prieskumné drony, ktoré mapujú nepriateľské teritórium alebo drony, ktoré sú vybavené zbraňami. Zbrane nemusia byť smrteľné – môžu to byť rôzne plyny alebo zvukové kanóny, ktoré majú za účel rozdeliť dav búriacich sa ľudí. Drony sa tiež dajú využiť na doručovanie liekov, jedla alebo iných zásielok. Aj napriek tomu, že drony sú bezpochyby schopné doručovať tovar, tento účel je často zamietnutý štátnymi legislatívami. Drony sú využívané aj napríklad na monitorovanie rôznych infraštruktúr ako sú napríklad cesty alebo ropovody. V poľnohospodárskom priemysle si našli využitie na sledovanie dobytku, úrody, množstva vody alebo na postrek úrody. Využívajú sa tiež na produkovanie záberov pre rôzne filmy alebo hudobné klipy. Vďaka snímkam z vtáčej perspektívy alebo rôznym senzorom (napr. *LiDAR*) je vďaka dronom možné vytvárať rôzne 3D modely krajiny alebo budov. Hasiči využívajú drony na monitorovanie požiaru alebo aj samotné hasenie. V rámci policajného dozoru sa dajú drony využiť pre pátranie po nezvestných osobách alebo po zločincoch. Môžu byť použité na monitorovanie zabezpečených objektov ako sú väznice alebo rôzne vojenské objekty. V roku 2012 boli drony prvý krát využité ako umenie na účel podobný ohňostrojom – svetelnú šou.

Technické parametre vybraných dronov

Hlavným faktorom pri výbere drona je jeho účel. Ponuka dronov je veľká a stále rastie. Od veľkých a drahých armádnych dronov až po pomerne lacné drony pre súkromné využitie. Na trhu sa dajú nájsť aj drony bez kamier, napríklad pretekárske drony, drony pre začiatočníkov alebo je možnosť pripevniť na drony bez kamier rôzne akčné kamery (napr. GoPro[2]). Avšak väčšina dronov určených na súkromné alebo komerčné využitie má kameru. Pohyby drona počas letu často nie sú plynulé, aj kvôli poveternostným podmienkam, preto majú často drony kameru pripevnenú na stabilizátore, ktorý zároveň umožňuje kamerou otáčať. Kamera je v týchto prípadoch určená aj pre riadenie dronu.

DJI Mavic 2 Pro

DJI je jeden z najväčších hráčov v oblasti dronov pre širokú verejnosť. Ich produkty zahŕňajú niekoľko typov modelov dronov pre rôzne cieľové skupiny. DJI Mavic 2 Pro je podľa

[2] https://www.gopro.com/

viacerých zdrojov[3] [4] jeden z najlepších dronov pre bežného užívateľa, aký sa dá kúpiť. Váži 907 gramov, keď je zložený má rozmery 214 × 91 × 84 mm a po rozložení sú rozmery 322 × 242 × 84 mm. Dosahuje maximálnu rýchlosť letu 72 km/h a výdrž batérie je maximálne 31 minút. Dolet drona je 18 km za priaznivých podmienok. Dosah ovládača je 8 km. Dron sa dokáže pomocou senzorov vyhnúť prekážkam vo všetkých šiestich smeroch (dopredu, dozadu, doľava, doprava, hore, dole). V kamere je použitý CMOS senzor s rozlíšením 20 MPx. Dokáže nahrávať video v rozlíšení 4K pri 30 snímkach za sekundu. Uhol zorného poľa kamery je 77°. Dron patrí do vyššej cenovej kategórie. Jeho cena 1499 €. Cena a parametre sú z oficiálnych stránok výrobcu [7].

Obr. 2.8: Dron DJI Mavic 2 Pro v rozloženom a zloženom stave. Obrázok prevzatý z [7].

GoPro Karma drone

Spoločnosť GoPro je známa hlavne výrobou akčných kamier. Dron Karma sám o sebe žiadnu kameru nemá. Dá sa k nemu pripevniť kamera minimálne GoPro Hero 4 a novšia. GoPro prišlo na trh s dronom, kde hlavným cieľom bola jednoduchosť ovládania. Dron váži 1006 gramov, keď je zložený má rozmery 365 × 224 × 90 mm a po rozložení 303 × 411 × 117 mm. Jeho maximálna rýchlosť je 56 km/h. Vo vzduchu vydrží 20 minút. Dron má dolet 3 km a dosah ovládača je 1 km. Ovládač má zabudovaný displej. Dron nemá žiadne senzory slúžiace na vyhýbanie sa prekážkam. Výhodou drona je jeho zakomponovanie do prostredia výrobkov GoPro, kedy sa z drona dá odobrať stabilizátor a kamera, ktoré sa dajú pomocou Karma Grip použiť ako klasický stabilizátor pre kameru do ruky. Kamera GoPro 7 Hero Black má rozlíšenie 12 MPx. Dokáže nahrávať video s rozlíšením 4K pri 60 snímkach za sekundu. Uhol zorného poľa tejto kamery je pri pomere strán 16:9 a režime „WIDE“ až 118°, dá sa však zmenšiť na 86° pri režime „LINEAR“ [5]. Dron Karma sa začal predávať v roku 2015, avšak produkcia sa zastavila kvôli niekoľkým problémom s ovládaním drona a aj kvôli silnej konkurencii v oblasti dronov. Spoločnosť GoPro tak v roku 2017 ukončila predaj týchto dronov. Samotný dron so stabilizátorom stál 419,99 € a kamera GoPro 7 Hero Black stojí 229,99 €. Technické parametre drona sú z [3], technické parametre kamery sú a cena drona a kamery sú z oficiálnych stránok výrobcu [12].

[3] https://www.techradar.com/news/best-drones
[4] https://www.wired.com/gallery/best-drones/
[5] https://community.gopro.com/t5/en/HERO7-Field-of-View-FOV-Information/ta-p/390215

Obr. 2.9: Dron Gopro Karma v rozloženom stave aj s kamerou. Obrázok prevzatý z [3].

Parrot Anafi

Spoločnosť Parrot patrí tiež k zvučnejším menám na trhu s dronmi, hlavne tými menšími. Dron Parrot Anafi váži len 300 g, jeho rozmery sú 244 × 67 × 65 mm keď je zložený, rozložený má rozmery 175 × 240 × 65 mm. Dosahuje maximálnu rýchlosť 55 km/h a vo vzduchu vydrží maximálne 25 minút. Dosah ovládača je 4 km. Na ovládanie je potrebné do ovládača pripojiť mobilný telefón. Dron dokáže nasledovať objekty na videu, avšak nemá žiadne senzory, ktoré by zabránili kolíziám s prekážkami. V kamere je zakomponovaný CMOS senzor s rozlíšením 21 MPx. Dokáže nahrávať video v rozlíšení 4K pri 30 snímkach za sekundu. Uhol zorného poľa kamery je 69°. Cenou sa dron radí k lacnejším. Jeho cena sa pohybuje od 660 € pre základný model [6]. Tento dron má viac verzií, jedna z nich napríklad obsahuje termokameru miesto normálnej kamery. Technické parametre drona sú zo stránok výrobcu [43].

Obr. 2.10: Dron Parrot Anafi v rozloženom a zloženom stave. Obrázok prevzatý z [43].

[6] https://www.amazon.com/Parrot-Foldable-Quadcopter-Autonomous-vertical/dp/B07D5R2JKL

Kapitola 3

Počítačové videnie a algoritmy pre rozpoznávanie osôb podľa tváre

Počítačové videnie (angl. *computer vision*) umožňuje počítaču získať schopnosť analyzovať digitálnu fotografiu, či video a získať z nich informácie na vysokej úrovni podobne ako ľudský mozog. Všeobecne sa zaoberá detekciou objektov a ich následným rozpoznávaním a klasifikovaním do predom definovaných tried. Najskôr je potrebné získať digitálne dáta (fotografia alebo video), dáta je následne potrebné predspracovať jednoduchými úpravami (záleží na konkrétnom algoritme, niektoré nepotrebujú predspracovanie vôbec). Ďalším krokom je segmentácia obrazu na časti obrazu, kde by sa mohli nachádzať hľadané objekty. Posledným krokom je klasifikácia výrezov oblastí s potencionálnymi objektami, kde sa rozhodne, či sa jedná o hľadaný objekt alebo nie, resp. o aký objekt sa jedná.

Využitie počítačového videnia je rozsiahle. Od rozpoznávania jednotlivých znakov, vytlačených na písacích strojoch alebo napísaných rukou, cez rozpoznávanie ľudí podľa tvárí, až po autonómne vozidlá alebo roboty, ktoré sú schopné na základe vstupných obrazových dát reagovať na momentálnu situáciu.

V tejto kapitole je vysvetlený princíp neurónových sietí a základné vrstvy konvolučných neurónových sietí, sú tu zmienené niektoré algoritmy slúžiace na predspracovanie obrazu a iné pomocné algoritmy využiteľné pri rozpoznávaní osôb podľa tváre. Na kapitoly konci sú popísané *state of the art* riešenia pre detekciu tvárí a rozpoznávanie osôb na základe tváre.

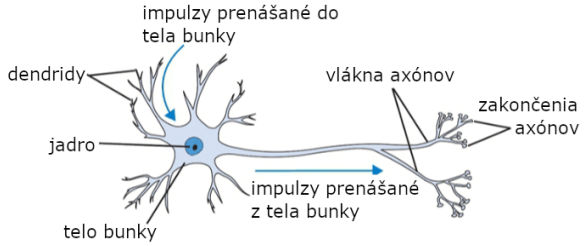
3.1 Neurónové siete

Informácie v tomto odseku sú prevzaté z [27]. Dlhý proces evolúcie človeka, dal ľudskému mozgu mnoho vhodných vlastností, ktoré sa nenachádzajú vo von Neumannovskom, ani v moderných paralelných počítačoch. Sú to vlastnosti ako masívny paralelizmus, distribuovaná reprezentácia a výpočty, schopnosť učiť sa, prispôsobivosť, chápanie v kontexte alebo nízka spotreba. Vďaka neurónovým sieťam je možné aby zariadenia niektoré z týchto vlastností získali.

Neurónové siete sú inšpirované biologickou stavbou nervového systému. Sú to paralelné výpočtové systémy skladajúce sa z množstva jednoduchých poprepájaných procesorov - umelých neurónov. Vďaka tomu, že tieto umelé neuróny sú zložené predovšetkým z trénovateľných parametrov, nadobúdajú schopnosť učiť sa. Vďaka tejto schopnosti sú použiteľné v oblastiach rozpoznávania reči, počítačového videnia, rôznych druhov klasifikácie a ďalších.

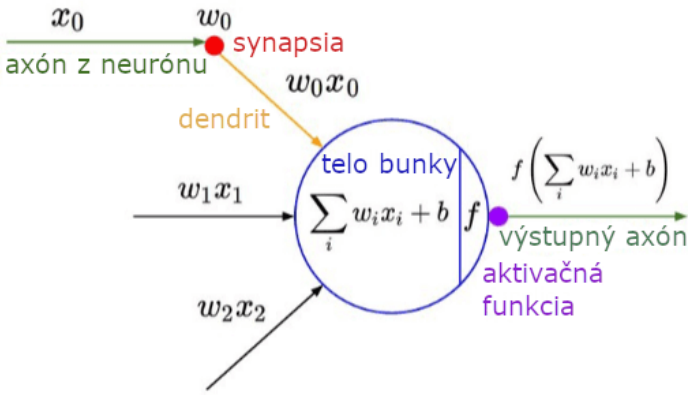
Neurón

Informácie v odstavci pochádzajú z [27]. Neurón je špeciálna biologická bunka. Jej štruktúra je popísaná na obrázku 3.1. Neurón prijíma signály od ostatných neurónov cez *dendridy* (prijímače) a vysiela signály generované jeho telom pomocou *axónu* (vysielač). *Axón* sa delí na vlákna. Na konci týchto vlákien sa nachádzajú *synapsie*. *Synapsia* je základná jednotka medzi neurónmi. Je spojená s *dendridom* druhého neurónu. Počet neurónov v mozgovej kôre človeka je asi 10^{11}, pričom jeden neurón je spojený s 10^3 až 10^4 inými neurónmi. Impulzy medzi neurónmi prebiehajú na frekvencii od pár do niekoľko stoviek hertz. Aj napriek tejto pomerne nízkej frekvencii (v porovnaní s dnešnými procesormi) je človek schopný urobiť komplexné rozhodnutia za pár stoviek milisekúnd. Z toho vyplýva, že v ľudskom mozgu prebiehajú paralelné programy, ktoré majú zhruba 100 krokov.



Obr. 3.1: Štruktúra biologického neurónu. Obrázok prevzatý z [10].

Na základe biologických neurónov vznikol v roku 1943 jeden z najpoužívanejších matematických modelov neurónu [37]. V roku 1957 bol tento model zobecnený na tzv. *perceptron* [46].



Obr. 3.2: Umelý model neurónu, ktorý znázorňuje jeho podobnosť s biologickým neurónom. Obrázok prevzatý z [10].

Umelý neurón má n vstupov a jeden výstup. Dá sa popísať nasledujúcim vzťahom[1]:

$$y = f(b + \sum_{i=1}^{n} w_i x_i), \tag{3.1}$$

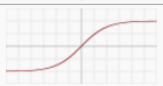
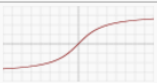
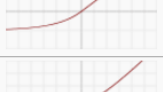
[1] https://medium.com/coinmonks/the-mathematics-of-neural-network-60a112dd3e05

kde y je výstup neurónu, f je aktivačná funkcia neurónu, w_i je odpovedajúca váha pre daný vstup, x_i je príslušná vstupná hodnota do neurónu a b je prahová hodnota aktivačnej funkcie.

Každý vstup x_i je vynásobený príslušnou váhou w_i, potom sú tieto hodnoty sčítané a je k ním pripočítaná prahová hodnota aktivačnej funkcie. Hodnota sa vloží do aktivačnej funkcie a výsledok tejto funkcie je výsledok celého neurónu.

Aktivačná funkcia

Aktivačná alebo prechodová funkcia realizuje zobrazenie $\mathbb{R} \to \mathbb{R}$ a určuje výstup neurónu. Funkcie môžu mať lineárny aj nelineárny priebeh. Niektoré z používaných aktivačných funkcií sú zobrazené na obrázku 3.3.

Názov	Graf	Rovnica	Derivácia
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Obr. 3.3: Používané aktivačné funkcie. Obrázok prevzatý z [49].

V poslednej dobe sa stáva obľúbenou aktivačnou funkciou funkcia s názvom *Rectified Linear Unit* (ReLU). Je definovaná predpisom:

$$f(x) = max(0, x), \tag{3.2}$$

kde $max(0, x)$ znamená maximálna hodnota z 0 a x.

V práci [5], bolo ukázané, že funkcia ReLU zrýchľuje proces trénovania neurónových sietí s konvolučnými vrstvami.

Učenie neurónových sietí

Odstavec bol prevzatý z [55]. Cieľom učenia je nastaviť váhy tak, aby vytvárali správnu odozvu výstupného signálu na daný vstupný signál. Po dokončení procesu učenia neurónovej siete sa na ňu dá pozerať ako na čiernu skrinku, ktorá je vhodná k nasadeniu v zvolených aplikačných rovinách.

Inicializácia váh

Informácie v tento podkapitole pochádzajú z [61]. Inicializáciou váh sa rozumie nastavenie váh neurónov v neurónovej sieti pred začiatkom samotného učenia siete. Keby boli hodnoty váh inicializované na hodnotu 0, skryté vrstvy by sa stali symetrickými a vo výsledku by vznikol lineárny model. Preto je potrebné hodnoty váh nainicializovať náhodným spôsobom. Môže však dôjsť k problému známemu ako *vanishing / exploding gradient* [44]. Z tohto dôvodu sa pre rôzne aktivačné funkcie používajú rôzne inicializačné techniky.

He inicializácia sa používa pri aktivačnej funkcii ReLU. Vygeneruje sa náhodná hodnota pomocou štandardizovaného normálneho rozdelenia[2] a následne sa vynásobí výrazom $\sqrt{\frac{2}{n_{in}}}$, kde n_{in} je počet vstupných neurónov.
Xavier inicializácia sa používa s aktivačnou funkciou *tanh*. Tiež sa vygeneruje náhodná hodnota pomocou štandardizovaného normálneho rozdelenia a vynásobí sa výrazom $\sqrt{\frac{1}{n_{in}+n_{out}}}$, kde n_{in} je počet vstupných neurónov a n_{out} počet výstupných jednotiek vrstvy.

Učenie bez učiteľa

Podľa [18] nemá pri učení bez učiteľa algoritmus k dispozícii žiadne kritérium správnosti hľadanej transformácie vstupných dát. Pracuje na princípe zhlukovania, kde vo vstupnej množine hľadá sebe podobné prvky. Počet hľadaných zhlukov môže byť vopred daný. Do procesu učenia nevstupuje žiadny učiteľ, ktorý by posúdil správnosť. Celé učenie je založené na informáciách obsiahnutých vo vstupných dátach.

Učenie s učiteľom

V prípade učenia s učiteľom má algoritmus dostupnú spočítateľnú množinu dvojíc prvkov, ktoré predstavujú vstup a odpovedajúci korektný výstup. Parametre siete sú ovplyvňované kombináciou tréningového vektoru a chybou. Sieti je predložený vstup a tá na základe aktuálnych hodnôt váhových koeficientov vypočíta výsledok. Výsledok je porovnaný s očakávaným korektným výsledkom a je spočítaná chyba - ako veľmi sa aktuálny výstup siete líši od toho očakávaného. Na základe chyby sú upravované koeficienty váh. Tento proces sa opakuje až kým sa nedosiahne predom definovanú minimálnu veľkosť chyby. Po dosiahnutí minimálnej chyby je sieť považovaná za naučenú. Tento spôsob učenia sa často používa pri rozpoznávaní objektov v obraze, kde sa sieť učí na anotovaných dátach.

Chybové funkcie

Pri učení neurónových sietí je potrebné zistiť ako sa výstup neurónovej siete líši od očakávaného výsledku. Pre získanie chyby sa používajú chybové funkcie, známe aj ako *loss*

[2] https://en.wikipedia.org/wiki/Normal_distribution#Standard_normal_distribution

funkcie. Na základe tejto chyby sú potom upravované váhy jednotlivých neurónov. Chyba by sa mala pri procese učenia postupne znižovať, až pokiaľ nedosiahne určené minimum. Medzi známe chybové funkcie patrí napríklad *Mean squared error* (MSE). Je definovaná nasledujúcou rovnicou:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - f_i)^2, \tag{3.3}$$

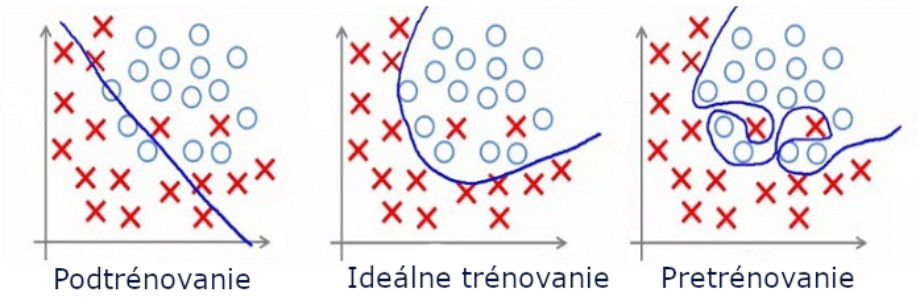
kde x_i je očakávaný výstupný vektor, f_i je výstupný vektor neurónovej siete, n je veľkosť týchto vektorov.

Pretrénovanie a Podtrénovanie

K **pretrénovaniu** [58] siete dochádza pri učení. Model sa môže dostať do stavu kedy dochádza k zníženiu hodnôt chybovej funkcie na trénovacích dátach, avšak na nových dátach sa chybovosť zvyšuje viac, ako tomu bolo pri nižšej iterácii trénovania.

Pre vyhnutie sa pretrénovaniu sa dáta (dataset) rozdelí na 2 časti. Trénovaciu a validačnú. Trénovacia časť obsahuje dáta, na ktorých sa model učí. Väčšinou sa viac dát nachádza práve v trénovacej časti. Validačná časť slúži na sledovanie pretrénovania. Po každej iterácii (epoche) sa spočíta chyba z validačných dát. Ak chyba na validačných dátach rastie a chyba na trénovacích dátach klesá, model sa dostáva do stavu pretrénovania. Je to z dôvodu obmedzenej sady trénovacích dát. Model sa začne zameriavať na konkrétne detaily, ktoré sú až príliš špecifické. Pre predídeniu pretrénovania sa môžu použiť aj tzv. dropout vrstvy spomenuté ďalej v tejto kapitole.

Podtrénovanie [58] je stav kedy chybová funkcia modelu je jednoducho vysoká. Model nedokáže klasifikovať ani validačné, ani trénovacie dáta. K podtrénovaniu môže dôjsť kvôli viacerým dôvodom. Jeden z nich môže byť napríklad, že model nemal dostatok času na učenie alebo nemal dostatok dát. Model tiež nemusí byť dobre navrhnutý pre danú úlohu. V tomto prípade by mohlo byť riešením úprava modelu - preskladanie vrstiev, pridanie nových alebo zmena počtu neurónov.



Obr. 3.4: Podtrénovanie (vľavo) a pretrénovanie (vpravo). V strede je zobrazený ideálny výsledok trénovania. Obrázok prevzatý z [45].

Backpropagation

Informácie v tejto podkapitole pochádzajú z [33]. Algoritmus spätného šírenia chyby je jeden z najčastejšie používaných adaptačných algoritmov pre viacvrstvové dopredné neurónové

siete. Je používaný pri učení s učiteľom. Slúži k počítaniu parciálnych derivácii chybovej funkcie vzhľadom ku váham neurónov.

V rámci algoritmu sa vstupný signál šíri smerom dopredu, cez vstupné vrstvy, skryté vrstvy, až k výstupnej. Tu je spočítaná celková chyba podľa chybovej funkcie. Potom nasleduje spočítanie parciálnych derivácii (gradientov) tejto chyby pre jednotlive neuróny. Na základe týchto gradientov sa pomocou optimalizačného algoritmu vypočítajú nové váhy koeficientov daného neurónu. Tento proces sa opakuje, až kým nie je chybovosť siete prijateľná.

K výpočtu gradientov sa používa reťazové pravidlo (angl. *chain rule*):

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}, \tag{3.4}$$

ktoré hovorí o tom ako je malá zmena v x transformovaná na malú zmenu v y a tá je transformovaná na malú zmenu v z. Na základe reťazového pravidla sa zistí aký majú na seba vplyv váhy a prahové hodnoty aktivačných funkcií jednotlivých neurónov. Potom sú váhy a prahové hodnoty aktivačných funkcií (*bias*) upravované tak, aby sa znížila chybová funkcia. Tento proces je aplikovaný rekurzívne na každú vrstvu siete v smere od výstupnej po vstupnú.

Konvolučné neurónové siete

Umelé neuróny sú zoskupované do väčších celkov, ktoré sa nazývajú vrstvy. Neurónová sieť môže obsahovať viac vrstiev. Výstupy z jednej vrstvy môžu byť vstupom pre ďalšie vrstvy. Spôsob prepojenia neurónov určuje topológiu siete. Viacvrstvové siete majú jednu vstupnú vrstvu (angl. *input layer*), niekoľko skrytých vrstiev (angl. *hidden layer*) a jednu výstupnú vrstvu (angl. *output layer*).

Podľa [15] sú konvolučné neurónové siete viacvrstvové neurónové siete navrhnuté pre rozpoznávanie dvojdimenzionálnych útvarov. Vďaka svojim vlastnostiam nie je odozva siete závislá na posune, rotácii, zmene mierky a ďalších transformáciách. Využívajú extrakciu lokálnych príznakov z recepčných polí neurónov, zdieľanie váh a priestorové prevzorkovanie. Postupne sa z obrázku získajú základné príznaky ako sú napr. hrany alebo rohy, ktoré sú potom ďalej kombinované na zložitejšie príznaky. Toto je zabezpečené postupným striedaním *konvolučných* a *pooling* vrstiev, ktoré sú vysvetlené nižšie.

Plne prepojená vrstva

Informácie sú prevzaté z [38]. Plne prepojená vrstva (angl. *fully connected/dense layer*) sa skladá z určitého množstva umelých neurónov. Každý neurón je napojený na každý vstup tejto vrstvy. Vzhľadom na vysoký počet spojení sa zvyšujú aj výpočtové nároky. Preto sa táto vrstva používa až v posledných vrstvách neurónovej siete, kde má úlohu klasifikácie objektov.

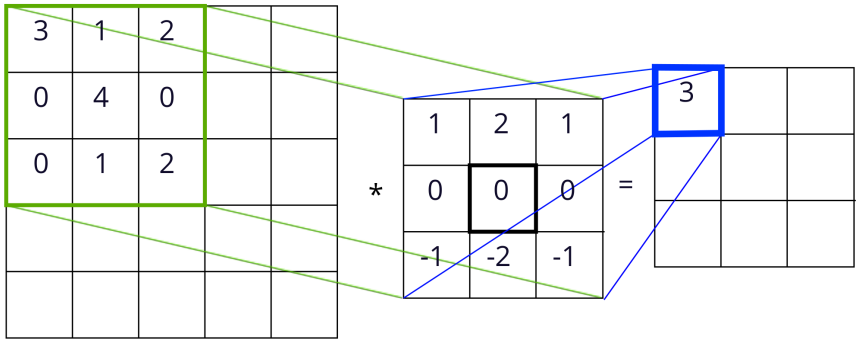
Konvolučná vrstva

Informácie pochádzajú z [15]. Neuróny sú v konvolučných vrstvách prepojené s malým okolím (recepčné pole). Neuróny sú umiestnené v rovinách, kde zdieľajú istú množinu váh. Výstupom tejto roviny je príznaková mapa. Vďaka zdieľaniu váh sa znižuje výpočtová aj priestorová náročnosť pri učení.

Názov tejto vrstvy vychádza z matematickej operácie ktorá sa v nich uskutočňuje - konvolúcia. V konvolučnej vrstve sa aplikuje filter (angl. *kernel*) na vstupný obraz. Obraz je reprezentovaný maticou pixlov. Je teda potrebné použiť diskrétnu 2D konvolúciu.

$$f(x,y) * h(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} f(x-i, y-j) \cdot h(i,j) \tag{3.5}$$

Funkcia f značí vstupnú maticu (obrázok) a funkcia h značí konvolučné jadro (filter). Konvolučné jadro sa postupne posúva po pixloch vstupného obrazu. Vždy vypočíta hodnotu len pre pixel, na ktorom sa práve nachádza. Postupnou aplikáciou konvolúcie získame príznakovú mapu. Každý typ filtra generuje inú príznakovú mapu.



Obr. 3.5: Grafické znázornenie konvolúcie vstupnej matice o veľkosti 5 × 5 a konvolučného jadra o veľkosti 3 × 3.

Výstupná matica má menšie rozlíšenie ako vstupná z dôvodu, že na výpočet jedného bodu potrebujeme na vstupe maticu, ktorá má rovnakú veľkosť ako filter. Dá sa však nastaviť tzv. *padding*, spôsob rozšírenia pôvodného obrazu tak, aby mal výstup z tejto vrstvy rovnaké rozmery ako vstup.

Myšlienkou je získavanie lokálnych charakteristík obrázku. Vďaka zdieľaniu váh medzi neurónmi je dosiahnutá robustnosť voči transformáciám alebo deformáciám vstupných objektov v obrázku.

V obraze sú farby reprezentované hodnotami 0 - 255. Pri konvolúcii sa však môže stať, že výsledok operácie bude záporný. Preto je vhodné použiť aktivačnú funkciu, ktorá tieto záporné hodnoty odstráni. Vhodná funkcia je napríklad spomínaná ReLU.

Pooling vrstva

Pooling vrsta (*subsampling vrstva*) je používaná na zmenšenie priestorového rozlíšenia - podvzorkovanie. Zmenší sa tým výpočtová náročnosť pre ďalšie vrstvy. Vrstva zhlukuje okolité pixely do jedného. Používa sa niekoľko variánt tejto vrstvy. Napríklad max-pooling, ktorý z okna ponechá len pixel s najväčšou hodnotou alebo average-pooling, ktorý ponechá priemer všetkých hodnôt v okne. Často sa používa pred konvolučnou vrstvou.

Dropout vrstva

Informácie sú prevzaté z [51]. Úlohou tejto vrstvy je predísť alebo aspoň znížiť náchylnosť siete na pretrénovanie. Realizuje sa to tak, že sa vyberú náhodné neuróny (aj ich väzby)

a ich výsledky sa zahodia. K tomuto procesu dochádza len pri trénovaní siete. Nastavuje sa tu len parameter miera zahadzovania.

3.2 Lokalizácia a zarovnanie tváre

Aby sme boli schopní efektívne nájsť a identifikovať ľudí či iné objekty z fotografií alebo video záznamu v rozumnom čase, je potrebné zmenšiť plochu, ktorú má systém prehľadávať, hlavne pri záznamoch z dronov, ktoré majú často rozlíšenie Full-HD alebo aj 4K. To by bolo možné jednoduchým znížením rozlíšenia. Pri záberoch z drona sa môže veľkosť tváre pohybovať od jednotiek pixelov až po tisícky pixelov. Zmenšením rozlíšenia by sme stratili možnosť detegovať malé tváre. Ďalej môžu byť tváre v rôznych pózach, čo môže zhoršiť presnosť výsledkov. Preto je nutné využiť rôzne pomocné algoritmy, ktoré v čo najlepšej miere zvýšia presnosť alebo rýchlosť získavania výsledkov.

Posuvné okno

Posuvné okno (angl. *sliding window*) je jeden z najjednoduchších spôsobov ako segmentovať vstupný obrázok. Na začiatku sa zvolí fixná veľkosť okna, ktoré sa bude postupne posúvať po obrázku. Každý jeden výrez sa vloží do klasifikátora a ten rozhodne, či sa jedná o objekt, ktorý hľadáme alebo nie. Môže nastať situácia, kedy je objekt väčší ako samotné posuvné okno. Preto sa využívajú tzv. **obrázkové pyramídy** (angl. *image pyramids*), ktoré zmenia rozlíšenie vstupného obrázku niekoľkokrát a posuvné okno prejde cez každý z týchto obrázkov. Tým je zaručené nájdenie objektov rôznych veľkostí a na rôznych pozíciách. Ďalšou komplikáciou, ktorá môže nastať je, že jeden objekt je nájdený viacerými posuvnými oknami. Pre správny výsledok je nutné, každý objekt mal práve jeden výrez, v ktorom sa nachádza. Preto je nutné použiť algoritmus **NMS** (angl. *non-maximum suppression*). Tento algoritmus nájde skupiny výrezov, kde je miera prekrytia vyššia ako zvolená prahová hodnota a vyberie jeden výrez, ktorý má najvyššie skóre získané klasifikátorom.

Zarovnanie tvárí

Výrez, na ktorom sa nachádza tvár je pre vyššiu presnosť často potrebné správne zarovnať. Tvár môže byť rôzne otočená alebo naklonená. Pre samotné zarovnanie tváre nám stačí poznať pozíciu kľúčových bodov na tvári. Zarovnanie tvárí sa dá rozdeliť na 2D zarovnanie a 3D zarovnanie.

Pri **2D zarovnaní** je potrebná šablóna, ktorá udáva fixnú veľkosť a fixné pozície vybraných kľúčových bodov v cieľovom obrázku. Na základe vstupných kľúčových bodov a kľúčových bodov zo šablóny sú vypočítané transformácie, ktoré vstupný obrázok prevedú tak, aby sa kľúčové body nachádzali na rovnakom mieste ako v šablóne. Využiť sa dajú napríklad afinné (angl. *affine*) alebo podobnostné (angl. *similarity*) transformácie. Afinné transformácie však môžu obrázok zdeformovať, ak je tvár na obrázku priveľmi natočená. Podobnostné transformácie však zachovávajú tvar transformovaných objektov, takže k deformácii nedôjde.

Pri **3D zarovnaní** sa dá použiť podobný spôsob. Najskôr sa tvár frontalizuje pomocou 2D transformácie, potom sa znova nájde niekoľko kľúčových bodov, ktoré sa umiestnia do 3D priestoru na základe jednej [54] alebo viacerých [36] 3D šablón.

Avšak napríklad pri obrázku tváre z profilu nemáme informácie o častiach tváre, ktoré na obrázku nie sú. Pre doplnenie tejto informácie sa dá využiť **syntéza tvárí** pomocou

siete GAN (angl. *Generative adversarial network*). Tieto siete sú schopné generovať dáta podobné tým, na ktorých boli trénované. Pozostávajú z diskriminátora a generátora. Generátor má za úlohu generovať dáta a úlohou diskriminátora je zistiť, či sú dáta pravé alebo vygenerované. Týmto spôsobom je možne na základe snímky z profilu získať obrázok celej tváre a následne ho ďalej spracovať.

Sledovanie objektov

Sledovanie objektov je pre počítačové videnie zložitá úloha, ktorá môže byť riešená viacerými spôsobmi. Pre sledovanie objektov je najskôr nutné použiť detektor, ktorý deteguje objekty a inicializuje samotné sledovanie. Sledovanie funguje na základe informácie o polohe a rýchlosti (na základe predchádzajúcich snímok) objektu a o tom ako vyzerá. Z týchto údajov je možné vypočítať pozíciu objektu v ďalšom obrázku. Sledovanie objektov je rýchlejšie ako ich detekcia a môže pomôcť v prípade, kde detektor zlyhá.

MedianFlow tracker bol predstavený v roku 2010 [29].Sleduje objekty v oboch smeroch (predchádzajúce aj nasledujúce snímky) a meria odchýlku týchto dvoch trajektórií. Minimalizáciou tejto chyby určí polohu objektu na ďalšej snímke. Pri sledovaní veľmi rýchlych objektov jeho presnosť klesá. Jeho výhodou je však jeho rýchlosť.

Goturn tracker je postavený na neurónovej sieti. Sieť sa neučí z aktuálnych snímok videa, ale bola natrénovaná pre daný účel pomocou datasetov. Vstupom sú dve snímky, prvá, na ktorej je ohraničený objekt aj s jeho okolím a druhá, na ktorej je potrebné objekt nájsť. Výstupom siete sú súradnice objektu na druhej snímke. Oba obrázky prejdú sústavou konvolučných vrstiev, výsledky sa spoja sa do jedného vektora, ktorý je vstupom pre tri plne prepojené vrstvy. Posledná plne prepojená vrstva je napojená na výstupnú vrstvu, ktorej výstupom sú spomínané súradnice. Bol predstavený v roku 2016 [17] a dosahoval rýchlosť viac ako 100 FPS s pomocou akcelerácie na grafickej karte. Jeho nevýhodou je však, že často prioritne sleduje objekty, ktoré boli v trénovacích datasetoch, takže pri sledovaní môže ľahko začať sledovať iný objekt.

3.3 Algoritmy pre detekciu tvárí a rozpoznávanie osôb podľa tváre

Podľa správy inštitútu NIST (*National Institute of Standards and Technology*) [13] sa za 5 rokov medzi rokmi 2013 a 2018 masívne zvýšila presnosť systémov pre rozpoznávanie tvárí. Väčšina algoritmov z roku 2018 hravo prekonáva algoritmy z roku 2013. Test z roku 2018 *NIST* zistil, že 0,2 % hľadaní v databáze o veľkosti 1,6 miliónov fotiek dopadli neúspešne. V porovnaní s rokom 2013, kedy táto miera neúspešnosti bola okolo 4 %. To činí zhruba 20 násobné zlepšenie. Tento úspech sa pripisuje skôr revolúcii ako evolúcii. Staré algoritmy sa nahradili algoritmami založenými na (hlbokých) konvolučných neurónových sieťach.

FaceNet

FaceNet [47] je systém pre rozpoznávanie tvárí predstavený v roku 2015 spoločnosťou Google. Výstupom siete je vektor o veľkosti 128 bytov, ktorý sa skladá z príznakov tváre. Je to hlboká konvolučná neurónová sieť, trénovaná pomocou funkcie *triplet loss*. Táto funkcia potrebuje na trénovanie trojice obrázkov: *anchor* - referenčná osoba, *positive* - iný obrázok referenčnej osoby, *negative* - iná osoba. Po extrahovaní príznakov z obrázkov sa vypočíta

Euklidovská vzdialenosť medzi *anchor* a *positive* a medzi *anchor* a *negative*. Hlavným cieľom pri trénovaní siete je minimalizovať vzdialenosť medzi *anchor* a *positive* a maximalizovať vzdialenosť medzi *anchor* a *negative*. Pri trénovaní tejto siete boli zvolené trojice obrázkov tak, aby vzdialenosť medzi *positive* a *negative* bola menšia ako zvolená nízka hranica, pretože zo všetkých vytvorených trojíc obrázkov bolo mnoho Euklidovsky vzdialených. Architektúra siete je zobrazená na obrázku 3.6. V rámci siete sú použité *inception* bloky, predstavené v [53], ktoré sú zložené z konvolučných filtrov viacerých veľkostí, kde je výsledok všetkých konvolúcii na konci spojený do jedného výstupu. Model bol trénovaný na privátnych dátových sadách o veľkosti 100-200 miliónov obrázkov.

type	**output size**	**depth**	#1×1	#3×3 **reduce**	#3×3	#5×5 **reduce**	#5×5	**pool proj (p)**	**params**	**FLOPS**
conv1 (7×7×3, 2)	112×112×64	1							9K	119M
max pool + norm	56×56×64	0						m 3×3, 2		
inception (2)	56×56×192	2		64	192				115K	360M
norm + max pool	28×28×192	0						m 3×3, 2		
inception (3a)	28×28×256	2	64	96	128	16	32	m, 32p	164K	128M
inception (3b)	28×28×320	2	64	96	128	32	64	L_2, 64p	228K	179M
inception (3c)	14×14×640	2	0	128	256,2	32	64,2	m 3×3,2	398K	108M
inception (4a)	14×14×640	2	256	96	192	32	64	L_2, 128p	545K	107M
inception (4b)	14×14×640	2	224	112	224	32	64	L_2, 128p	595K	117M
inception (4c)	14×14×640	2	192	128	256	32	64	L_2, 128p	654K	128M
inception (4d)	14×14×640	2	160	144	288	32	64	L_2, 128p	722K	142M
inception (4e)	7×7×1024	2	0	160	256,2	64	128,2	m 3×3,2	717K	56M
inception (5a)	7×7×1024	2	384	192	384	48	128	L_2, 128p	1.6M	78M
inception (5b)	7×7×1024	2	384	192	384	48	128	m, 128p	1.6M	78M
avg pool	1×1×1024	0								
fully conn	1×1×128	1							131K	0.1M
L2 normalization	1×1×128	0								
total									7.5M	1.6B

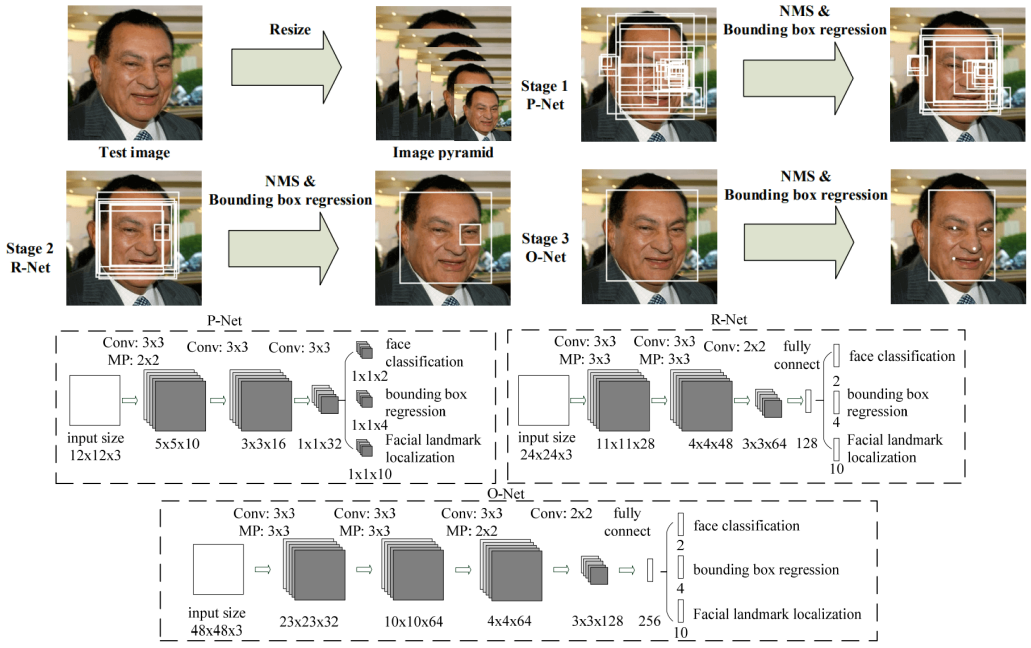
Obr. 3.6: Architektúra siete FaceNet, prevzaté z [47].

Na dátovej sade LFW [21] sieť dosahuje úspešnosť 99,63 % a na dátovej sade Youtube Faces DB [60] je úspešnosť 95,12 %.

Detektor MTCNN

MTCNN alebo *Multi-task Cascaded Convolutional Network* je kaskádová konvolučná neurónová sieť určená pre detekciu tváre na snímke. Popri detekcii tváre, sieť deteguje aj päť kľúčových bodov na tvári, ktoré sa neskôr dajú použiť na účely zarovnania tváre. Riešenie predstavili v dokumente z roku 2016 [63]. Sieť využíva metódu *Multi-task learning*, ktorá umožňuje súčasné riešenie viacerých úloh s pomocou využívania spoločných častí výpočtu. Veľkosť vstupného obrázka sa pred vstupom niekoľkokrát zmenší. Vznikne tzv. obrázková pyramída (popísaná v časti 3.2). Táto pyramída je vstupom pre neurónovú sieť. Kaskádová neurónová sieť MTCNN sa skladá z troch menších sietí. Prvá sieť s názvom P-Net (*Proposal Network*) je najmenšia a obraz spracováva najrýchlejšie. Jej úlohou je nájsť niekoľko kandidátnych oblastí, kde je šanca výskytu tváre. Oblasti, ktoré majú vysokú mieru prekrytia sa spoja do jednej pomocou algoritmu NMS (popísaný v časti 3.2). Všetky tieto oblasti sú následne vstupom pre druhú, väčšiu sieť s názvom R-Net (*Refine Network*). Tá má za úlohu vyradiť čo najviac oblastí, kde sa tvár nenachádza. Znova sa zredukuje aj počet prekrývajúcich sa oblastí pomocou NMS. Tretia, najzložitejšia sieť s názvom O-Net má za úlohu preskúmať zvyšné oblasti. V tejto fáze sú oblasti preskúmané najdôkladnejšie a výstupom sú už ohraničenia tvárí a pätice kľúčových bodov (oči, nos, kútiky úst) pre každú tvár. Aj napriek tomu, že detektor MTCNN vznikol v roku 2016, je dnes stále používaný

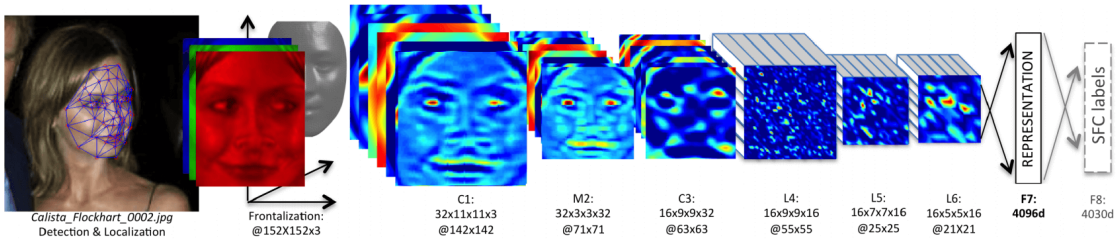
vďaka svojmu pomeru presnosti a rýchlosti. Princíp a architektúra MTCNN je popísaná na obrázku 3.7.



Obr. 3.7: Princíp MTCNN a architektúra jednotlivých sietí. Obrázok prevzatý z [63].

DeepFace

DeepFace [54] je hlboká konvolučná sieť na rozpoznávanie tvárí, predstavená spoločnosťou Facebook v roku 2014. Sieť je ukončená dvojicou plne prepojených vrstiev, pred ktorými sa nachádzajú tri lokálne prepojené vrstvy (viz. obrázok 3.8). Tieto vrstvy fungujú ako klasické konvolučné, avšak ich konvolučné jadrá nezdieľajú váhy, vďaka čomu sa odlišné časti obrázka spracúvajú odlišným spôsobom. Tieto vrstvy žiadajú aby bol obrázok dobre zarovnaný. Modul pre zarovnávanie v sieti DeepFace prevedie detegovanú tvár na jednoduchý 3D model, vďaka ktorému potom tvár zarovná na požadovanú pozíciu. Sieť bola trénovaná na súkromnej dátovej sade, v ktorej sa nachádzalo viac ako 4 milióny obrázkov, kde bolo cez 4000 rôznych identít.



Obr. 3.8: Architektúra siete DeepFace, prevzaté z [54].

Na dátovej sade LFW [21] sieť dosahuje úspešnosť 97,3 % a na dátovej sade Youtube Faces DB [60] dosahuje úspešnosť 91,4 %.

OpenFace

OpenFace [2] je verejne dostupná konvolučná neurónová sieť. Jej architektúra, popísaná na obrázku 3.9, je inšpirovaná sieťami od spoločností Google a Facebook. Na trénovanie používa funkciu *triplet loss* popísanú vyššie. Je však trénovaná na niekoľko násobne menšej dátovej sade ako FaceNet. Dátová sada vznikla spojením sád CASIA-WebFace [62] a FaceScrub [40]. Na zarovnávanie tvárí používa OpenFace afinnú transformáciu, kde neurónová sieť z knižnice *dlib* deteguje 68 kľúčových bodov na tvári, na základe ktorých sa obrázok transformuje tak, aby oči a spodná pera boli vždy na rovnakom mieste pre každý obrázok.

type	output size	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj
conv1 ($7 \times 7 \times 3, 2$)	$48 \times 48 \times 64$						
max pool + norm	$24 \times 24 \times 64$						m 3×3, 2
inception (2)	$24 \times 24 \times 192$		64	192			
norm + max pool	$12 \times 12 \times 192$						m 3×3, 2
inception (3a)	$12 \times 12 \times 256$	64	96	128	16	32	m, 32p
inception (3b)	$12 \times 12 \times 320$	64	96	128	32	64	ℓ_2, 64p
inception (3c)	$6 \times 6 \times 640$		128	256,2	32	64,2	m 3×3, 2
inception (4a)	$6 \times 6 \times 640$	256	96	192	32	64	ℓ_2, 128p
inception (4e)	$3 \times 3 \times 1024$		160	256,2	64	128,2	m 3×3, 2
inception (5a)	$3 \times 3 \times 736$	256	96	384			ℓ_2, 96p
inception (5b)	$3 \times 3 \times 736$	256	96	384			m, 96p
avg pool	736						
linear	128						
ℓ_2 normalization	128						

Obr. 3.9: Architektúra siete `nn4.small2` z OpenFace, prevzaté z [2].

Sieť dosahuje úspešnosť 92,92 % na dátovej sade LFW [21].

ArcFace

Vývojom neurónových sietí sa ukázalo, že viac sofistikované chybové funkcie vedia urýchliť priebeh trénovania a aj presnosť neurónových sietí. V práci [6] predstavili novú chybovú funkciu pre učenie neurónových sietí, ktorých úlohou je rozpoznávanie tvárí. Funkcia je inšpirovaná často používanou klasifikačnou chybovou funkciou *soft-max loss*, ktorá má nasledujúci predpis:

$$S_L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}, \tag{3.6}$$

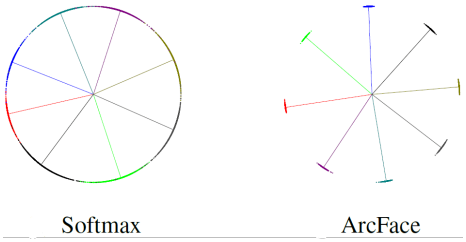
kde $x_i \in \mathbb{R}^d$ je príznakový vektor i-tej vzorky patriaci do y_i-tej triedy. d je rozmer vektorov príznakov, v tejto práci je rovný 512. $W_j \in \mathbb{R}^d$ označuje j-tý stĺpec váhy $W \in \mathbb{R}^{d \times n}$ a $b_j \in \mathbb{R}^n$ je prahová hodnota aktivačnej funkcie. Veľkosť *batch* (počet vzoriek, cez ktoré sieť prejde predtým ako si aktualizuje svoje interné parametre) je N a n.

Funkcia *soft-max loss* však dostatočne neoptimalizuje výstupné vektory príznakov, aby tie, ktoré sú v rámci jednej triedy (identity) boli podobnejšie a tie, ktoré sú z rozličných tried, boli rozličnejšie. To smeruje k poklesu presnosti ak je vysoká vnútro-triedna variabilita (napr. iná póza, vekový rozdiel atď.) a taktiež pri testovaní na veľkom množstve dát.

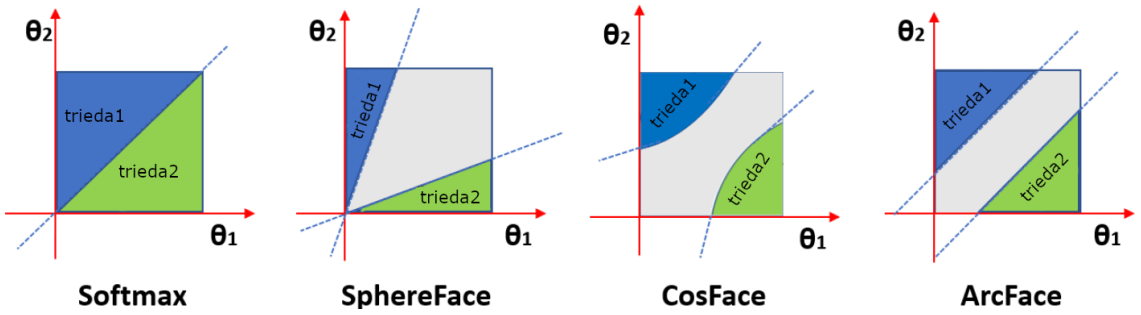
Na základe podobných prác, ktoré začali tieto problémy riešiť pomocou uhlovej (kosínusovej) vzdialenosti, v práci ArcFace upravili rovnicu vyššie určením b_j rovné nule, zmenou $W_j^T x_i$ na $\|W_j\|\|x_i\|cos\theta_j$, kde θ_j je uhol medzi váhou W_j a príznakom x_i. Individuálnu váhu zmenili ako $\|W_j\| = 1$ pomocou L_2 normalizácie. Aj príznak $\|x_i\|$ upravili pomocou L_2 normalizácie a zmenili jeho veľkosť na s. Tento krok normalizácie príznaku a váhy robí predpoveď len na základe uhla medzi príznakom a váhou. Naučené vektory príznakov sú rozmiestnené na hyperguli (*hypersphere*) s polomerom s okolo stredu každého príznaku. K tomuto v práci pridali *additive angular margin penalty* medzi príznakom x_i a váhou W_{y_i} aby naraz zmenšili vnútro-triednu a zväčšili medzi-triednu vzdialenosť. Výsledný predpis funkcie vyzerá nasledovne:

$$A_L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}, \tag{3.7}$$

kde m je spomínaná hodnota *additive angular margin penalty* a ostatné parametre sú popísané v odstavci nad rovnicou.



Obr. 3.10: Ilustračný príklad pri trénovaní s chybovými funkciami Softmax a ArcFace. Príklad pozostáva z 8 identít s 2D príznakmi. V prípade ArcFace vzhľadom na normalizáciu príznakov, sú všetky príznaky tváre umiestnené do priestoru s malým polomerom. Osem identít je tu viditeľných na prvý pohľad aj bez toho, aby boli farebne rozlíšené. Obrázok prevzatý z [6].



Obr. 3.11: Porovnanie *margin* podobných chybových funkcii pri binárnej klasifikácii. Obrázok prevzatý z [6] a preložený.

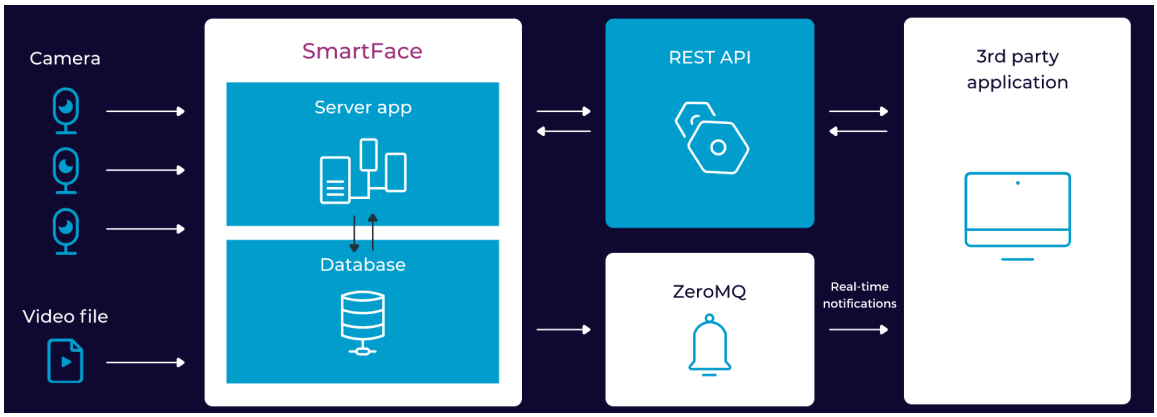
V práci využili architektúru *ResNet100* [16], kde upravili prvú konvolúciu, aby bol vstup pre sieť o veľkosti 112×112 pixelov. Tiež upravili spôsob finálneho výpočtu príznakového vektora. Za poslednou konvolučnou vrstvou je spravená *batch normalizácia*, dropout vrstva, plne prepojená vrstva a znova *batch normalizácia* [25], čo umožňuje výstup vo forme 512-D face embedding.

Na dátovej sade [21] sieť dosahuje úspešnosť 99,83 % a na dátovej sade Youtube Faces DB [60] dosahuje úspešnosť 98,02 %.

Innovatrics

Innovatrics je slovenská spoločnosť, ktorá patrí medzi svetovú špičku v oblasti biometrických technológií, najmä identifikáciu a verifikáciu osôb na základe rozpoznávania tváre, rohovky a odtlačkov prstov. Podľa benchmarkov z novembra 2019 inštitútu NIST[3] bol algoritmus firmy Innovatrics najrýchlejší pri rozpoznávaní osoby z galérie o veľkosti 12 miliónov osôb. Trvalo len 13 milisekúnd rozpoznať osobu v tejto databáze. Celkovo sa v presnosti umiestnil na 5. mieste z viac ako 200 rôznych súťažiacich. Výsledky sú prebraté zo stránky firmy Innovatrics [23].

Informácie o produkte SmartFace prevzaté zo stránky firmy Innovatrics [24]. SmartFace je komerčný produkt firmy Innovatrics slúžiaci na rozpoznávanie tvárí. Využíva sa na dozor a bezpečnosť alebo kontrolu vstupu do budov. Medzi jeho výhody patrí jednoduchá inštalácia, škálovateľnosť, vysoká presnosť aj pri real-time spracovaní. Pre fungovanie potrebuje aplikáciu na serveri, databázu pre tváre a videá a IP kamery, ktoré podporujú protokol RTSP[4]. Princíp je popísaný na obrázku 3.12.



Obr. 3.12: Princíp produktu SmartFace, prevzaté z [24].

[3] https://www.nist.gov/
[4] https://tools.ietf.org/html/rfc2326

Kapitola 4

Návrh riešenia a implementácia

Systémy, ktoré dokážu rozpoznávať tváre sú v dnešnej dobe veľmi rozšírené. Aj vďaka ich širokej možnosti použitia je o ne veľký záujem. Z tohto dôvodu existuje mnoho riešení pre problémy detekcie tváre a rozpoznávanie osôb podľa tváre. Riešenia predstavené v posledných rokoch dosahujú v niektorých prípadoch lepšie výsledky ako samotný ľudský mozog. Preto som sa v tejto práci rozhodol použiť už existujúce riešenia. Prvým cieľom bola implementácia experimentálnych nástrojov, na základe ktorých bude možné existujúce riešenia porovnať a vybrať do finálnej aplikácie.

4.1 Požiadavky na aplikáciu

Po analýze záznamov z dronov a na základe výskumu v predchádzajúcej časti práce boli pre aplikáciu špecifikované nasledujúce požiadavky. Pri záberoch z dronov sú zvyčajne ľudia vzdialení od kamery. Preto bolo vhodné, aby implementované riešenie dokázalo rozpoznávať jednotlivé osoby z rozumnej vzdialenosti s čo najpresnejšími výsledkami. Aplikácia mala byť schopná rozpoznať ľudí v databáze aj z malého množstva snímok záznamu (napr. pri rýchlom prelete drona ponad skupinu ľudí). Pri spracovaní záznamov offline je podstatná presnosť spracovania. Rýchlosť tu nehrá podstatnú rolu. Spracovanie by malo prebiehať na výkonnejších počítačoch s možnosťou akcelerácie výpočtu pomocou dedikovanej grafickej karty.

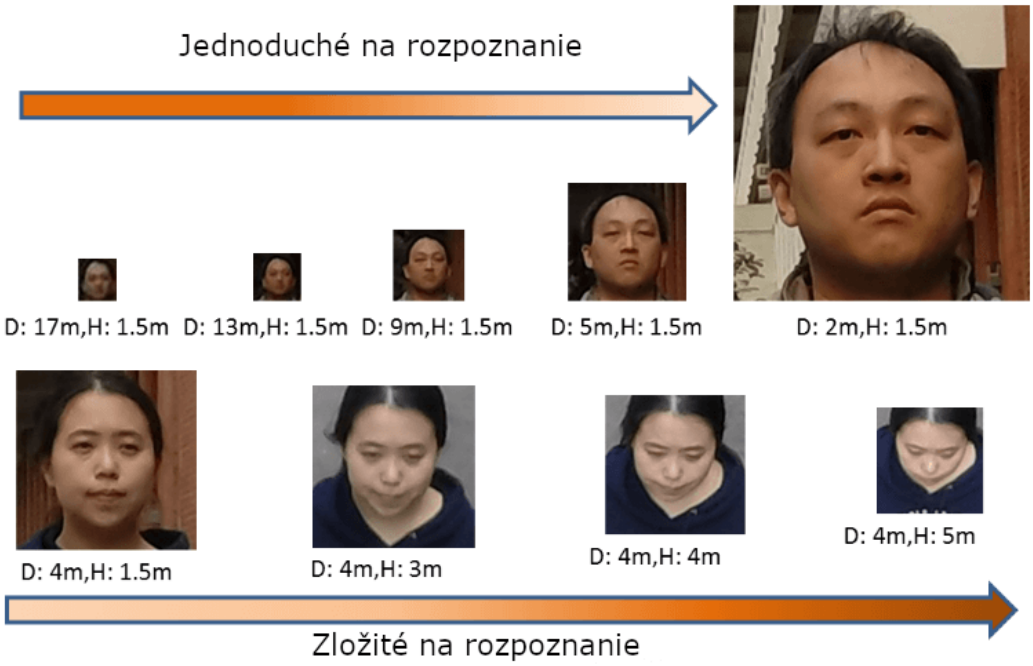
4.2 Testovacie dáta

Pre splnenie vyššie spomínaných požiadaviek bolo nutné nájsť dátovú sadu, ktorá by splnila potrebné parametre. Aj keď existuje veľké množstvo dátových sád pre účel rozpoznávania tvárí, veľmi malé množstvo z nich je odlišných od dát, ktoré sa dajú získať z drona. Podarilo sa však nájsť aj jeden vyhovujúci dataset, ktorý však pozostáva len z fotografií. Preto bolo nutné získať aj zábery pozostávajúce z videa. V rámci práce boli použité aj moje súkromné zábery z drona, avšak zábery obsahujú len obmedzené množstvo identít.

DroneFace dataset

DroneFace [19] je dataset určený pre testovanie riešení pre rozpoznávanie tváre na záberoch imitujúcich zábery z drona. Pozostáva zo 620 fotografií s rozlíšením 3680×2760 px a ultra širokým uhlom zorného poľa 170° fotených pomocou kamery GoPro Hero3+ Silver Edition. Zábery sú snímané v štyroch rôznych výškach (1 a pol, 3, 4, 5 metrov) a 31 rôznych

vzdialenostiach (2 - 17 metrov s veľkosťou kroku pol metra). Na fotografiách je 11 rôznych dobrovoľníkov vo veku 23 - 36 rokov. Obsahuje aj frontálne fotografie tvárí dobrovoľníkov pre vytvorenie databázy tvárí. Názvy jednotlivých fotografií poskytujú informáciu o osobách, nachádzajúcich sa na danej fotografii, výške a vzdialenosti snímania.



Obr. 4.1: Ukážka vystrihnutých tvárí z datasetu DroneFace. D značí vzdialenosť v metroch a H značí výšku v metroch. Obrázok prevzatý z [19].

4.3 Porovnanie algoritmov pre detekciu a rozpoznávanie tváre

Pre implementáciu skriptu na porovnanie existujúcich riešení som zvolil programovací jazyk *Python* [1], hlavne kvôli jednoduchosti inštalovania potrebných knižníc a algoritmov pre počítačové videnie pomocou nástroja *pip*. Pre spracovanie obrazu bola použitá knižnica *OpenCV* [2] a ako framework pre prácu s neurónovými sieťami *Tensorflow* [3] kvôli jeho jednoduchej inštalácii a možnosti akcelerácie výpočtov pomocou dedikovanej grafickej karty s podporou architektúry *CUDA* [4].

Porovnanie algoritmov pre detekciu tváre

Pri výbere algoritmu na detekciu tvárí som preskočil staršie algoritmy a siahol hneď po algoritmoch využívajúcich neurónové siete. Tieto algoritmy dosahujú väčšiu presnosť a menšie množstvo falošných detekcií. Skúsil som viacero algoritmov, avšak niektoré z nich sa mi nepodarilo otestovať z dôvodov ich vyššej náročnosti na hardvérové zdroje. Rozhodol som sa popísať len algoritmy, ktoré som bol schopný otestovať.

[1] https://www.python.org/
[2] https://opencv.org/
[3] https://www.tensorflow.org/
[4] https://docs.nvidia.com/cuda/

Vzhľadom na to, že na spracovanie obrazových dát som využil knižnicu OpenCV, prvý testovaný detektor bol práve z tejto knižnice. Konkrétne to bol **detektor využívajúci hlboké neurónové siete** založený na Single-Shot-Multibox detektore [35] s architektúrou *ResNet-10*. Ako vstup som detektoru určil fotografie z datasetu DroneFace. Pri pôvodných nastaveniach sa fotografie pred vstupom do neurónovej siete interne zmenšili a preto detekcia malých tvárí nebola možná. Ďalším pokusom bolo pridanie posuvného okna, čo presnosť vylepšilo, avšak algoritmus sa niekoľkonásobne spomalil. Po niekoľkých úpravách, ktoré zahrňovali aj zmenu interného zmenšenia vstupu, sa výsledky o niečo zlepšili, avšak algoritmus nebol schopný detegovať tváre, ktoré boli vzdialené od kamery viac ako 6 metrov pri rozlíšení 4K.

Ako ďalší som skúsil detektor MTCNN (popísaný v časti 3.3), konkrétne jeho open source implementáciu [5] využívajúcu jazyk *Python* a knižnicu *Keras*, ktorá je súčasťou *Tensorflow*. Túto implementáciu som si vybral z dôvodu jej jednoduchej inštalácie pomocou nástroja *pip*. Algoritmus bol schopný detegovať tváre v datasete DroneFace aj z maximálnej vzdialenosti 17 metrov. Jeho rýchlosť bola porovnateľná s detektorom z knižnice *OpenCV*. Algoritmus má nastaviteľné parametre, avšak najlepšie výsledky boli dosiahnuté s odporúčanými parametrami. Oproti detektoru z knižnice *OpenCV* je detektor MTCNN schopný určiť aj päť kľúčových bodov na tvári, ktoré je neskôr možné použiť na prípadné zarovnanie tváre.

Obr. 4.2: Porovnanie maximálnej vzdialenosti pri úspešnej detekcii tvárí na datesete DroneFace. Vľavo je hlboká neurónová sieť z knižnice *OpenCV* (6 metrov), vpravo je detektor MTCNN (17 metrov).

Porovnanie algoritmov pre rozpoznávanie tváre

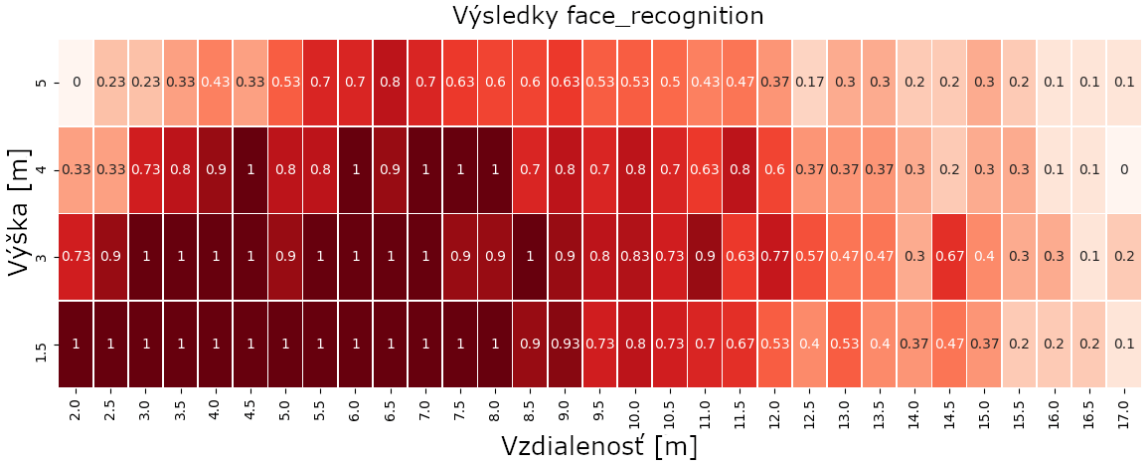
Aj pri výbere algoritmu pre rozpoznávanie tvárí som vybral algoritmy založené na neurónových sieťach. V dnešnej dobe veľa z nich funguje tak, že z orezaných fotografií tvárí dokážu extrahovať vektory príznakov, ktorých vzájomná vzdialenosť priamo odpovedá odlišnosti daných tvárí. Z tohto dôvodu bolo najskôr potrebné navrhnúť aj štruktúru databázy tvárí, ktorá slúži na zistenie identity získaných tvárí zo záznamov.

[5] https://github.com/ipazc/mtcnn

V práci [39] bolo ukázané, že pri rozpoznávaní tvárí s malým rozlíšením za použitia moderných extraktorov vektorov príznakov, je vhodné vytvárať referenčné vektory príznakov z fotografií s nižším rozlíšením. Vzhľadom na to, že táto práca je zameraná na určovanie identity ľudí podľa tváre zo záberov z drona, kde môže byť rozlíšenie tváre premenlivé, som sa rozhodol využiť toto zistenie. Z každej referenčnej fotografie určenej pre databázu sa vystrihne tvár, jej rozlíšenie sa upraví na tri rôzne veľkosti, z ktorých je následne extrahovaný vektor príznakov pre každé rozlíšenie zvlášť. Pre každú vstupnú fotografiu teda vzniknú tri rôzne vektory príznakov a sú uložené tak, aby bolo možné ďalej sledovať ich vplyv na rozpoznávanie osôb. Taktiež je pre jednu osobu možné vložiť viacero referenčných fotografií tváre.

V testovacom skripte som postupne testoval tri rôzne algoritmy pre extrakciu vektorov príznakov. Na vyhodnotenie algoritmov som využil dataset DroneFace.

Prvým z nich bol modul pre jazyk *Python* s názvom `face_recognition` [6]. Využíva algoritmy knižnice *dlib* [7]. V tomto module je aj algoritmus pre detekciu tvárí, avšak v práci som použil len extraktor príznakových vektorov z tvárí. Po vložení výrezu tváre algoritmy z knižnice *dlib* určia 68 kľúčových bodov na tvári, na základe ktorých sa prevedie transformácia obrazu, kde sa tvár správne otočí, zarovná ju na stred a zväčší jej veľkosť na 150×150 pixelov. Následne je z takto zarovnanej tváre získaný 128 dimenzionálny vektor príznakov popisujúci danú tvár. Výsledky algoritmu za použitia detektoru MTCNN pri datasete DroneFace sú zobrazené na obrázku 4.3. Po sčítaní týchto hodnôt je možné určiť celkové skóre. Pri tomto algoritme celkové skóre činilo **75,9** z celkových 124 bodov.

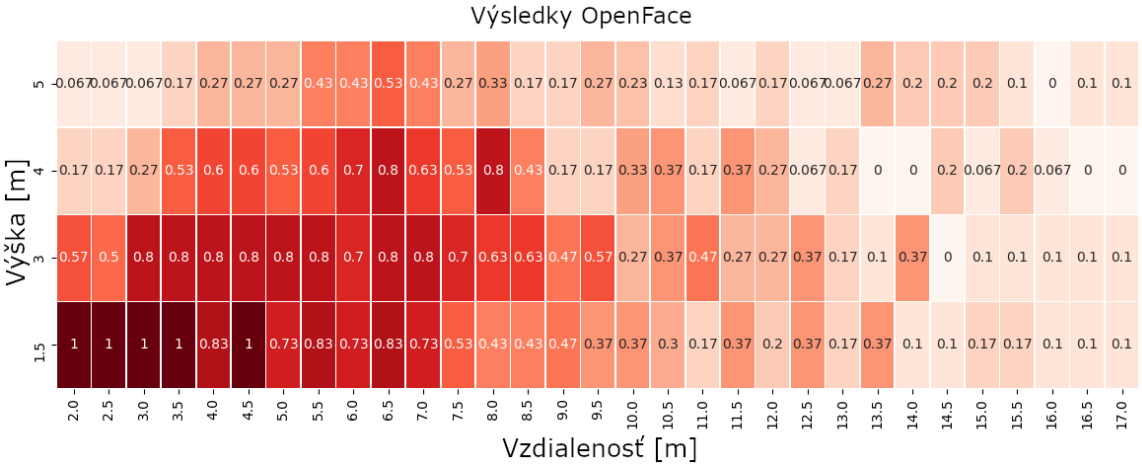


Obr. 4.3: Výsledky algoritmu z modulu `face_recognition`. Na osi X je vzdialenosť snímania (2 - 17 metrov s krokom o veľkosti 0,5 metra), na osi Y je výška snímania v metroch. Jednotlivé hodnoty znamenajú presnosť rozpoznávania osôb (hodnota 1 znamená, že všetky osoby boli rozpoznané správne). Celkové skóre je **75,9** z 124.

Ďalším testovaným algoritmom bol `OpenFace` (popísaný v časti 3.3). Konkrétne jeho *Tensorflow* implementácia od M. Krasser [8]. Využíva architektúru neurónovej siete nazývanú `nn4.small2` [9] (viď. 3.9) v projekte OpenFace. Architektúra modelu bola v tejto im-

[6] https://pypi.org/project/face-recognition/
[7] http://dlib.net
[8] https://github.com/krasserm/face-recognition
[9] https://cmusatyalab.github.io/openface/models-and-accuracies/#model-definitions

plementácii prevzatá z repozitára [10], ktorý konvertoval pôvodnú implementáciu OpenFace vo frameworku *Torch* [11] do frameworku *Keras*. Neurónová sieť vyžaduje na vstupe orezaný a zarovnaný obrázok tváre s rozlíšením 96 × 96 pixelov. Pre zarovnanie využíva tiež knižnicu *dlib* a jej zarovnanie tváre na základe 68 kľúčových bodov. Tvár je reprezentovaná 128 dimenzionálnym vektorom príznakov. Výsledky na datasete DroneFace je možné vidieť na obrázku 4.4. Pri sčítaní hodnôt je celkové skóre **45,6** z celkových 124 bodov.



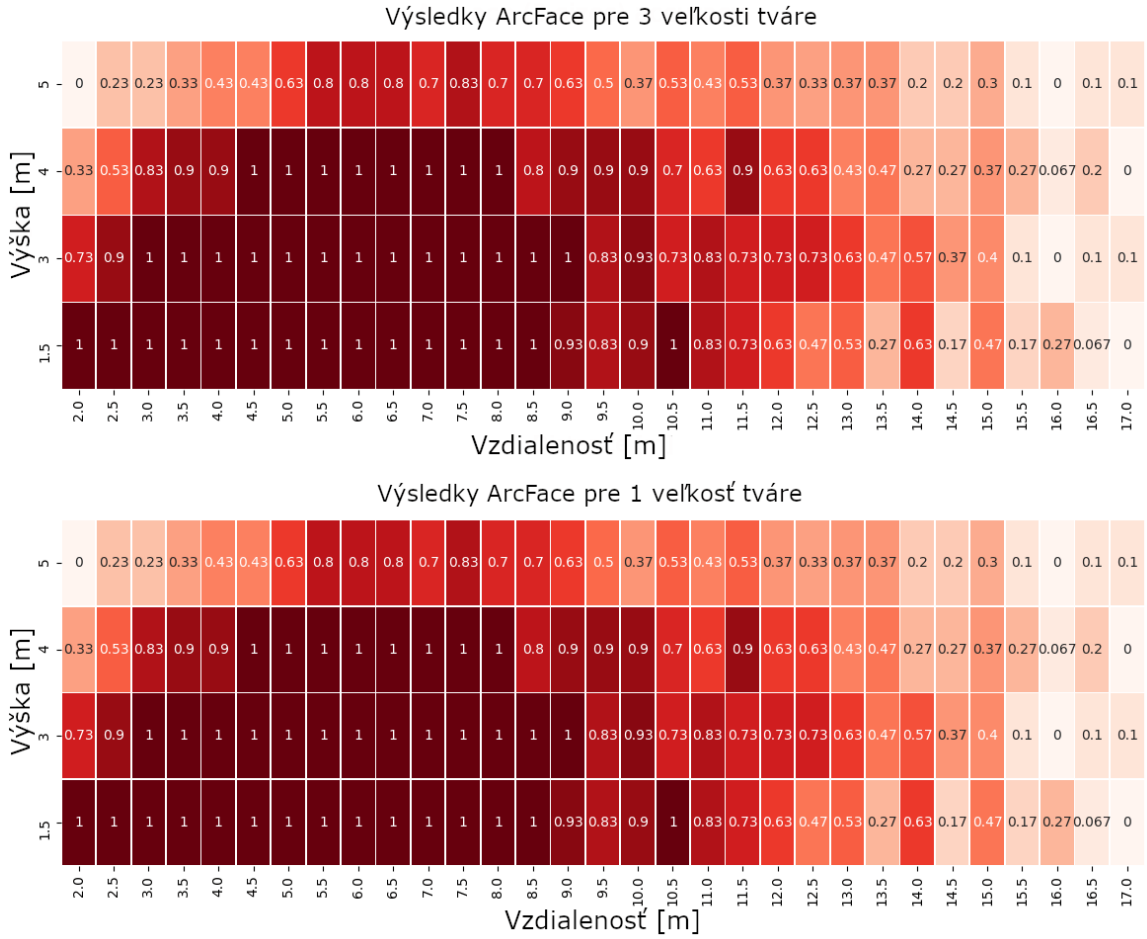
Obr. 4.4: Výsledky algoritmu `OpenFace`. Popis osí je rovnaký ako na predchádzajúcom obrázku 4.3. Celkové skóre **45,6** z 124.

Posledným testovaným algoritmom bol `ArcFace` (popísaný v časti 3.3). Aj pri tomto algoritme som hľadal jeho implementáciu vo frameworku *Tensorflow*. Použitá implementácia [12] je od autora Kuan-Yu Huang. Implementácia využíva architektúru neurónovej siete *ResNet50* aj so zmenami popísanými v časti 3.3. Neurónová sieť bola v tomto repozitári trénovaná na datasete *MS-Celeb-1M* [14]. Obrázky tvárí sú pred vstupom do siete zarovnané s použitím piatich kľúčových bodov na tvári získaných z detekcie tváre pomocou detektora MTCNN. Následne je ich veľkosť zmenená na 112 × 112 pixelov. Pri vyhodnocovaní tohto algoritmu ma zaujímalo, aký vplyv má na výsledky zmena veľkosti tvárí na tri rôzne veľkosti pri vytváraní databázy, popísaná v úvode tejto časti. Po vyhodnotení je celkové skóre s použitím troch veľkostí tvárí rovné **79,7** z celkových 124 bodov. Pri použití len jednej pôvodnej fotografie je skóre rovné **79,36** z celkových 124 bodov. Celkové vyhodnotenie je zobrazené na obrázku 4.5. Vplyv spomínanej zmeny veľkostí tvárí je na celkový výsledok v tomto prípade minimálny. Avšak vzdialenosti získaných tvárí od referenčných obrázkov sa pri väčšine fotografií zmenšili zhruba o jednu desatinu a viac.

Porovnanie algoritmov pre sledovanie tvárí

Pri implementácii testovacieho skriptu som sa z dôvodu väčšej rýchlosti spracovania rozhodol, že detekcia a rozpoznávanie tvárí nebude prebiehať v každej snímke videa. Preto bolo vhodné použiť algoritmus určený pre sledovanie objektov, aby sa poloha tvárí aktualizovala aj v snímkach, kde neprebieha detekcia. V prvej snímke videa sa detegujú tváre, určí sa ich identita a inicializuje sa sledovací algoritmus. Na nasledujúcich X (X je nastaviteľný

[10] https://github.com/iwantooxxoox/Keras-OpenFace
[11] http://torch.ch/
[12] https://github.com/peteryuX/arcface-tf2



Obr. 4.5: Výsledky algoritmu `ArcFace`. Popis osí je rovnaký ako na predchádzajúcom obrázku 4.3. Vrchný obrázok je pri použití troch veľkostí tvárí pri vytváraní databáze so skóre **79,7** z 124. Spodný obrázok je pri použití len jednej veľkosti tváre pri vytváraní databáze so skóre **79,36** z 124.

parameter) snímkach sú polohy tvárí aktualizované len sledovacím algoritmom, pričom ich identita ostáva rovnaká, ako bola zistená pri prvej snímke. Po X snímkach sa znova využije algoritmus pre detekciu a rozpoznávanie. Celý tento proces sa opakuje počas celého spracovania videa. Sledovacie algoritmy som testoval na sade domácich záberov z drona. Celkovo som otestoval päť rôznych algoritmov [13] implementovaných v knižnici *OpenCV*.

Prvým z nich bol algoritmus **CSRT Tracker**. Algoritmus zvládal sledovať aj malé tváre o veľkosti okolo 15 pixelov. Za testovacích podmienok (štyri sledované tváre, Full HD video, započítaný aj čas pre detekciu a rozpoznávanie tvárí) bola rýchlosť tohto algoritmu zhruba 6 snímok za sekundu. Do tejto rýchlosti je započítaný aj čas na spracovanie snímky, kde prebehla detekcia a rozpoznávanie.

Ďalším testovaným bol algoritmus **BOOSTING Tracker**. Tento algoritmus však nehlásil chybu sledovania aj keď očividne začal sledovať iný objekt. Ak sledovaná tvár zmizla

[13] https://docs.opencv.org/3.4/d9/df8/group___tracking.html

z obrazu, algoritmus stále aktualizoval jej polohu na okraji snímky. Rýchlosť tohto algoritmu za rovnakých podmienok bola asi 8 snímok za sekundu.

Tretím algoritmom bol **MedianFlow Tracker** popísaný v časti 3.2. Tento algoritmus nemal problémy ani so sledovaním malých tvárí, ani s tvárami, ktoré zmizli z obrazu. Jeho nevýhodou však je, že pri rýchlejších pohyboch stráca presnosť. Jeho rýchlosť je pri testovacích podmienkach okolo 19 snímok za sekundu.

Ďalším testovaným algoritmom bol **GOTURN Tracker** popísaný v časti 3.2. Jeho presnosť bola vysoká, nemal problém so sledovaním malých objektov. Jeho nevýhodami bola však zložitejšia inštalácia a menšia rýchlosť pri nevyužití akcelerácie na dedikovanej grafickej karte. Pri testovacích podmienkach bola jeho rýchlosť bez akcelerácie na dedikovanej grafickej karte 4 snímky za sekundu.

Predposledným testovaným algoritmom bol **MOSSE Tracker**. Tento algoritmus nedokázal sledovať malé objekty takmer za žiadnych podmienok. Jeho rýchlosť pri testovacích podmienkach však bola okolo 50 snímok za sekundu.

Posledným algoritmom bol **MIL Tracker**. Algoritmus mal problém s malými objektami, kedy sa štvorec vykreslený okolo tváre pri sledovaní zväčšoval (aj v prípade keď bola veľkosť tváre rovnaká) a niekedy sa nestíhal premiestniť zároveň s tvárou, a tak bol na niektorých snímkach vykreslený o pár pixelov vedľa. Rýchlosť algoritmu za testovacích podmienok bola 3 snímky za sekundu.

4.4 Návrh a implementácia aplikácie

Prvým krokom pre implementáciu finálnej aplikácie bol výber algoritmov pre detekciu, rozpoznávanie a sledovanie tvárí z algoritmov testovaných v testovacom skripte. Ďalším krokom bol výber nástroja pre tvorbu užívateľského rozhrania, vytvorenie návrhu užívateľského rozhrania a jeho implementácia. Aplikácia bola implementovaná na operačnom systéme Ubuntu 19.10. Jej funkčnosť bola však úspešne odskúšaná aj na operačnom systéme Windows 10. Na oboch operačných systémoch bola nainštalovaná knižnica `tensorflow-gpu` a knižnice potrebné pre akceleráciu výpočtov na dedikovanej grafickej karte s podporou architektúry *CUDA*. Použitá bola grafická karta NVIDIA GeForce GTX 1060 3GB.

Algoritmy použité v aplikácii

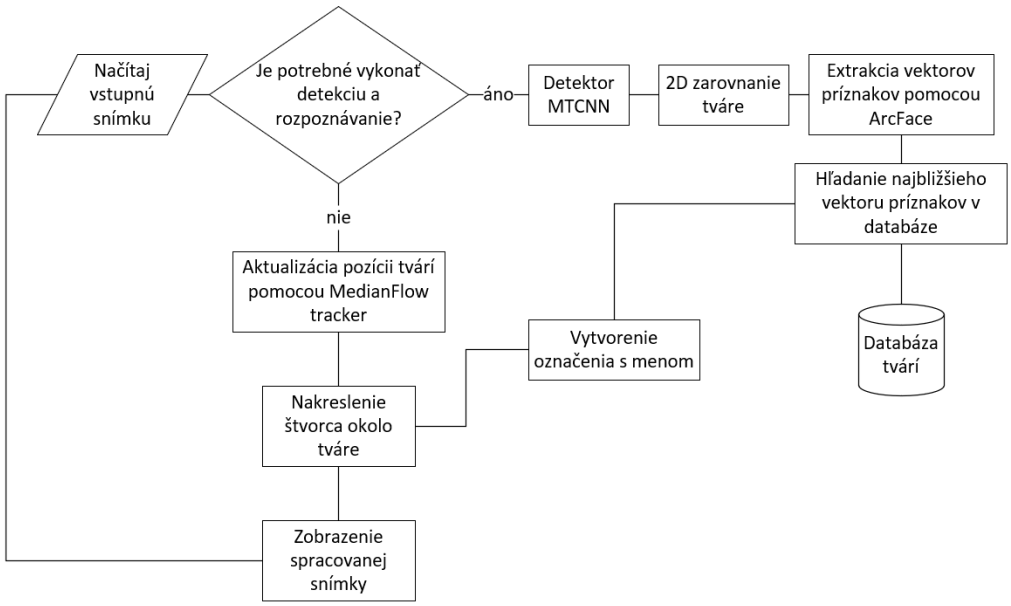
Pri výbere algoritmov do aplikácie som pracoval s kandidátmi spomínanými v časti 4.3. Pre detekciu tvárí som zvolil **detektor MTCNN**. Bol schopný detegovať tváre vzdialené aj 17 metrov od kamery v datasete DroneFace. Detektor ma však problémy s detekciou tvárí, ktoré sú snímané z veľkého uhla, sú otočené alebo nesmerujú do kamery. Pri záberoch z drona sú však tváre vo väčšine prípadov otočené správne. Pri snímaní z veľkého uhla a pri tvárach, ktoré nie sú otočené do kamery je tento problém zanedbateľný, keďže z týchto tvárí by nebolo možné extrahovať dostatočné množstvo tvárových príznakov na identifikáciu človeka. Päť kľúčových bodov na tvári získaných týmto detektorom som využil na zarovnanie tváre pomocou **2D zarovnania** popísaného v časti 3.2. Šablónu pre päť kľúčových bodov na tvári som získal z workshopu *Deep face recognition using Tensorflow*[14].

[14] https://github.com/Alireza-Akhavan/deep-face-recognition

Ako algoritmus pre extrakciu vektorov príznakov z tvárí som vybral **ArcFace** a jeho implementáciu popísanú v časti 4.3. Tento algoritmus dosahoval najvyššie skóre podľa vytvorenej metriky. Použil som **tri veľkosti tvárí pri vytváraní databáze**. Získaná tvár sa zarovná a následne sa jej rozlíšenie zmení na 40×40, 80×80 a 112×112 pixelov. Tieto veľkosti som nazval S (*small*), M (*medium*) a L (*large*). Z každého z vytvorených obrázkov sa extrahujú vektory príznakov a sú uložené do databázy. Veľkosti týchto tvárí som určil experimentálne a dbal som na to, aby nepokazili presnosť rozpoznávania.

Pri algoritme na sledovanie objektov nebolo rozhodovanie tak jednoznačné. Nakoniec som zvolil algoritmus **MedianFlow Tracker**, ktorý dosahoval vyššiu rýchlosť ako väčšina testovaných algoritmov. Pri testovaných záberoch sa nestalo, že by stratil sledované objekty aj pri pohybe alebo otáčaní drona.

Výsledná schéma hlavného procesu aplikácie je zobrazená na obrázku 4.6



Obr. 4.6: Schéma zobrazujúca hlavný proces aplikácie.

Požiadavky na grafické užívateľské rozhranie

V rámci zadania práce neboli určené žiadne konkrétne požiadavky na užívateľské rozhranie. Aby som mohol pokračovať v návrhu užívateľského rozhrania, bolo potrebné vytvoriť si konkrétne požiadavky. V aplikácii by malo byť možné vidieť proces spracovávania videozáznamu s možnosťou pozastavenia obrazu. Tiež by malo byť možné zobraziť si informácie o aktuálnych tvárach na videu. Výsledné video by malo byť možné uložiť do súboru. Vo výslednom videu by mali byť tváre ohraničené obdĺžnikom (angl. *bounding box*) a mala by byť určená ich identita. Taktiež by malo byť možné vizualizovať si aktuálnu databázu tvárí, s ktorou aplikácia pracuje. Databáza tvárí by sa mala dať vytvoriť nová, alebo načítať už existujúca zo súboru.

Grafické užívateľské prostredie

Užívateľské rozhranie som sa rozhodol implementovať vo frameworku *PyQt* verzia 5[15]. Je založený na frameworku *Qt*[16]. Je to nástroj na vytváranie multiplatformových aplikácií. Súčasťou je aj nástroj s užívateľským rozhraním pre tvorbu užívateľských rozhraní *Qt Designer*. Pomocou tohto nástroja som si vytvoril jednoduchý návrh užívateľského rozhrania.

Užívateľské rozhranie som v návrhu rozdelil na dve hlavné okná. Prvé okno bude určené pre databázu tvárí. Tu bude možné zvoliť natrénovanie novej databázy alebo otvoriť už existujúcu databázu. V druhom okne prebieha spracovanie videozáznamu. Je tu niekoľko tlačidiel, ktorými je možné načítať video, spustiť, zastaviť, pozastaviť spracovanie, zobraziť stav spracovávania a otvoriť nastavenia spracovania, kde je možné zmeniť parametre pred začiatkom spracovávania. Pre štýly v užívateľskom rozhraní som využil balíček inštalovateľný pomocou nástroja *pip* s názvom `qdarkstyle`[17], ktorý aplikácii pridá jednotný moderný tmavý vzhľad.

Databáza tvárí

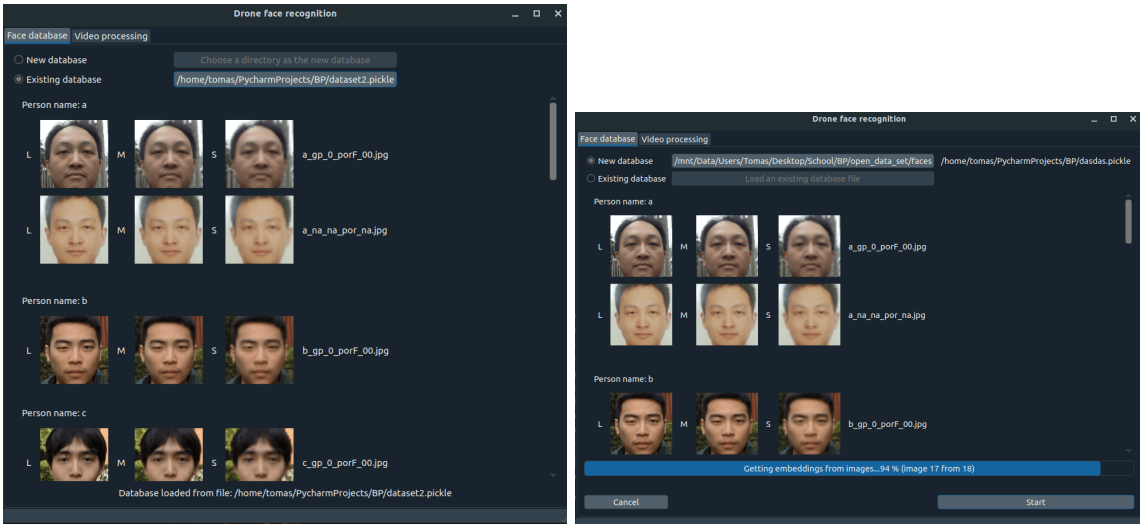
Ďalej bolo potrebné navrhnúť štruktúru databázy tvárí, ktorá by umožňovala databázu vizualizovať v aplikácii a uložiť ju do súboru. Pri vytváraní novej databázy je potrebné určiť cestu k priečinku s tvárami. Hlavný priečinok musí mať štruktúru, v ktorej každá osoba databázy bude mať samostatný priečinok, ktorého názov je meno danej osoby. V každom priečinku konkrétnej osoby môže byť až 100 fotografií tejto osoby. Databáza je dátového typu `dict`, kde je kľúčom názov priečinku osoby a hodnotou je zoznam položiek typu `FaceDatabaseItem`. Položka `FaceDatabaseItem` obsahuje meno osoby, názov zdrojovej fotografie, výrez tváre vo formáte `numpy` poľa[18] a extrahovaný vektor príznakov z tváre. Vyhľadávanie v databáze prebieha na základe euklidovskej vzdialenosti vektorov príznakov, porovnaním získaného vektoru so všetkými vektormi v databáze. Bolo testované aj vyhľadávanie pomocou KNN klasifikátora[19], ale pri malej databáze tvárí (menej ako 50 identít) tento klasifikátor presnosť znižoval.

Databáza je serializovaná pomocou nástroja `pickle`[20] a uložená do súboru, z ktorého je možné jej opätovné načítanie do aplikácie. Pri vytváraní novej databázy je tento proces predaný novému vláknu. V užívateľskom rozhraní sa postupne pridávajú spracované osoby. Stav spracúvania je zobrazený v stavovom pruhu (angl. *progress bar*). Zobrazenie existujúcej databázy v aplikácii a vytváranie novej je zobrazené na obrázku 4.7.

Vizualizácia spracovávania videozáznamu

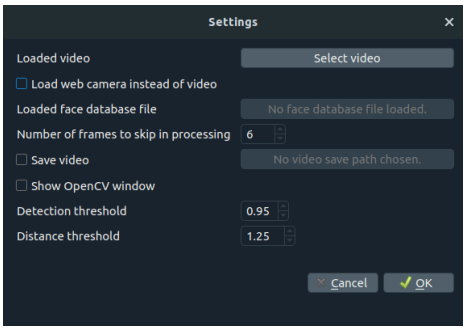
Vizualizácia spracovávania je hlavným prvkom aplikácie. Samotná vizualizácia prebieha na prvku `QLabel`. Tento prvok z frameworku `PyQt` je schopný zobraziť text alebo tzv. `pixmap`, ktorú je možné získať konverziou snímok získaných a upravených pomocou *OpenCV*. Po spustení aplikácie sa v samostatnom vlákne načítajú váhy neurónovej siete ArcFace. Po načítaní je možné odštartovať spracovávanie videozáznamu ak je zvolený zdroj. Zdrojom môže byť video zo súboru alebo webová kamera. Pred spracovaním je možné otvoriť nastavenia (viz.

[15] https://wiki.python.org/moin/PyQt
[16] https://www.qt.io/
[17] https://pypi.org/project/QDarkStyle/
[18] https://docs.scipy.org/doc/numpy/reference/generated/numpy.array.html
[19] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[20] https://docs.python.org/3/library/pickle.html



Obr. 4.7: Časť finálnej aplikácie, ktorá zobrazuje existujúcu databázu načítanú zo súboru (vľavo) a časť aplikácie ktorá vytvára novú databázu (vpravo).

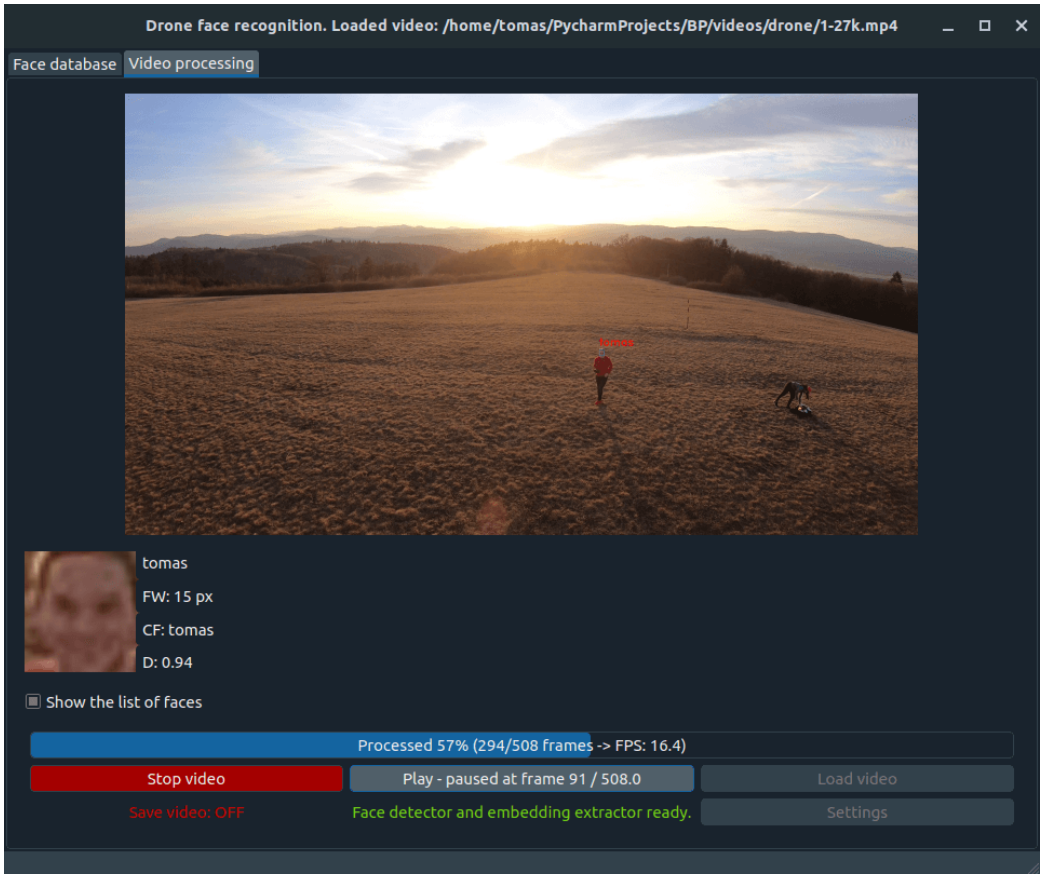
obrázok 4.8), kde je možné zvoliť video zo súboru alebo webovú kameru, je tu zobrazený aktuálny súbor s databázou tvárí, je možné nastaviť počet snímok medzi snímkami, na ktorých prebieha detekcia a rozpoznávanie, je možné zvoliť cestu kam sa má výsledné video uložiť, je možné zapnúť zobrazenie okna knižnice *OpenCV* popri zobrazení v aplikácii, je tu možné nastaviť parametre pre prah detekcie a prah vzdialenosti tvárí pre rozpoznávanie.



Obr. 4.8: Nastavenie spracovania vo finálnej aplikácii.

Po spustení spracovania sa na základe rýchlosti zdrojového videa (FPS) nastaví `QTimer`, ktorý volá funkciu pre vykreslenie snímky v pravidelných intervaloch. Ak sú nejaké snímky spracované, zobrazia sa. Pri spracovaní je v stavovom pruhu (angl. *progress bar*) zobrazený stav spracovania. Spracovanie prebieha v samostatnom vlákne. Vizualizáciu spracovania je možné pozastaviť, pričom samotné spracovanie prebieha v pozadí ďalej. Snímky spracované v pozadí sa ukladajú do zoznamu. Po zrušení pauzy sa spracované snímky zo zoznamu prehrajú rýchlosťou vstupného videa. Pod videom je možné zobraziť informácie o všetkých detegovaných tvárach na aktuálnej snímke. Je tu zobrazený zarovnaný výrez tváre, priradené meno, šírka tváre v pixloch, najbližšia podobná tvár (priradené meno môže byť `*` v prípade, že vzdialenosť k najbližšej tvári v databáze je väčšia ako prah vzdialeností tvárí) a vzdialenosť od najbližšej tváre. Pri pozastavení vizualizácie a prejdení myšou ponad tieto informácie sa zobrazí podrobnejší náhľad (angl. *tooltip*), kde sú zobrazené aj informácie

o najbližšej tvári – z ktorej fotografie je jej vektor príznakov a o akú veľkosť sa jedná (veľkosti S, M, L popísané v časti 4.4). Okrem pozastavenia vizualizácie je možné tlačidlom ukončiť proces spracovania. Ak je nastavená cesta k výstupnému videu, sú uložené snímky z videa, ktoré sa stihli spracovať. Vzhľad tejto časti užívateľského prostredia je na obrázku 4.9.



Obr. 4.9: Vizualizácia procesu spracovania vo finálnej aplikácii.

Aplikácia využíva akceleráciu výpočtov na dedikovanej grafickej karte s podporou architektúry *CUDA*. Rýchlosť spracovania bola závislá aj od počtu osôb nachádzajúcich sa na zázname a rozlíšení tohto záznamu. Pri detekcii a rozpoznaní tvárí na každej siedmej snímke (1. snímka - detekcia a rozpoznávanie, 6 snímok len sledovanie pomocou sledovacieho algoritmu) sa rýchlosť spracovania na grafickej karte NVIDIA GeForce GTX 1060 3GB pohybovala nasledovne:

- rozlíšenie 1920×1080 pixelov (Full HD): 6 - 8 FPS,
- rozlíšenie 2704×1520 pixelov (2,7K): 4 - 6 FPS,
- rozlíšenie 3840×2160 pixelov (4K): 2 - 3 FPS.

Kapitola 5

Experimenty

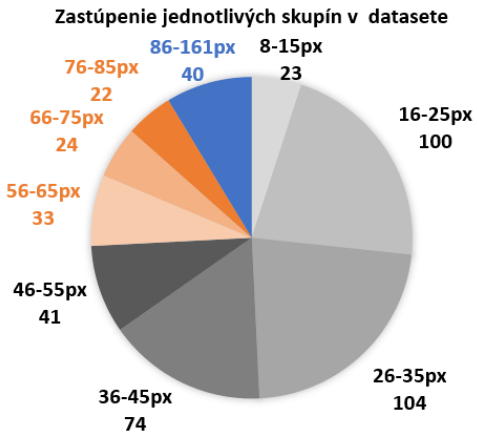
V rámci experimentov som využil moje súkromné záznamy z drona. Zo záznamov som vybral časti, kde sa nachádzajú ľudia v rozumnej vzdialenosti. Záznamy sú vytvorené pomocou drona GoPro Karma drone s kamerou GoPro 7 Hero Black (popísané v časti 2.3), za použitia najvyššieho uhla zorného poľa 118°, rozlíšenia 1920×1080 (Full HD), 2704×1520 (2,7K) a 3840×2160 (4K) a rýchlosti snímania 60 snímok za sekundu.

5.1 Testovacie dáta

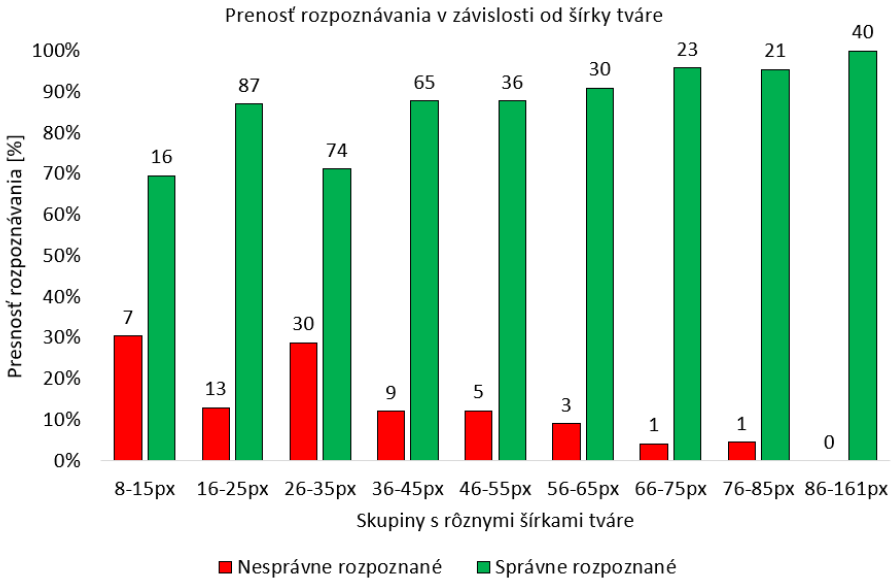
Vytvorená databáza obsahovala **10 identít**. Z toho 7 sa nachádzalo na záznamoch. V záznamoch sa však nachádzali aj traja ľudia, ktorí neboli súčasťou databázy. Pri spracovávaní záznamov som zvolil parameter prah vzdialenosti (angl. *distance threshold*) rovný $1,25$ a prah detekcie (angl. *detection threshold*) rovný $0,95$. Tieto prahy boli zvolené na základe predchádzajúcich experimentov popri implementácii. V aplikácii som spracoval celkovo sedem videí zo štyroch rôznych miest. Celková dĺžka videí bola 3 minúty a 12 sekúnd. Z videí som každú sekundu odobral všetky tváre, ktoré bol detektor schopný nájsť. Extrahoval som aj informácie o tvárach, ktoré som následne spracoval. Celkový počet **obrázkov** bol **192**. Nachádzalo sa na nich celkovo **461 tvárí**. Následne som ručne skontroloval, či boli tváre správne identifikované. Celkovo bolo **392** tvárí rozpoznaných správne a **62** nesprávne. Sledoval som aj závislosť šírky tváre v pixloch na presnosti rozpoznávania. Pre jednoduchšie spracovanie som si tváre rozdelil do skupín podľa rozlíšenia. Celkové zastúpenie tvárí v skupinách je možné vidieť na obrázku 5.1. V datasete majú početnú prevahu tváre s nižším rozlíšením.

5.2 Vplyv rozlíšenia na úspešnosť rozpoznávania

Ďalej som skúmal presnosť rozpoznávania v jednotlivých skupinách tvárí. Výsledok je na obrázku 5.2. Na grafe je viditeľné, že so zväčšujúcim rozlíšením sa podiel správne rozpoznaných tvárí podľa očakávania zvyšuje. Až na jednu výnimku pri skupine tvárí so šírkou 26 - 35 pixelov, kedy oproti predchádzajúcemu menšiemu rozlíšeniu presnosť klesla. To je pravdepodobne spôsobené nedostatkom dát.



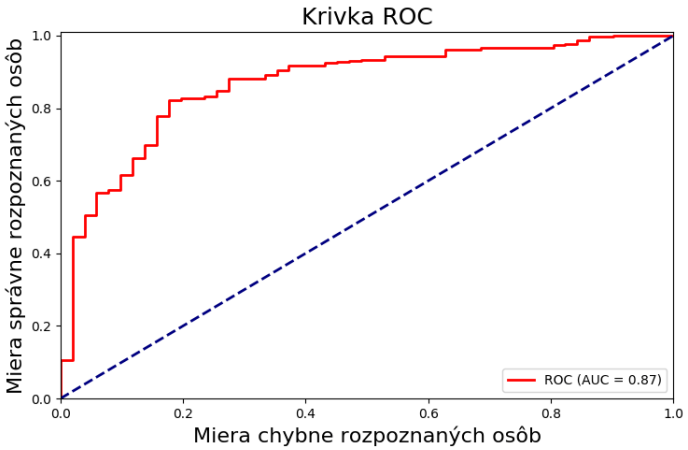
Obr. 5.1: Graf zastúpenia počtu tvárí v skupinách podľa šírky tváre. Vedľa výseku je rozsah šírky tváre v pixloch a počet tvárí v danom rozsahu.



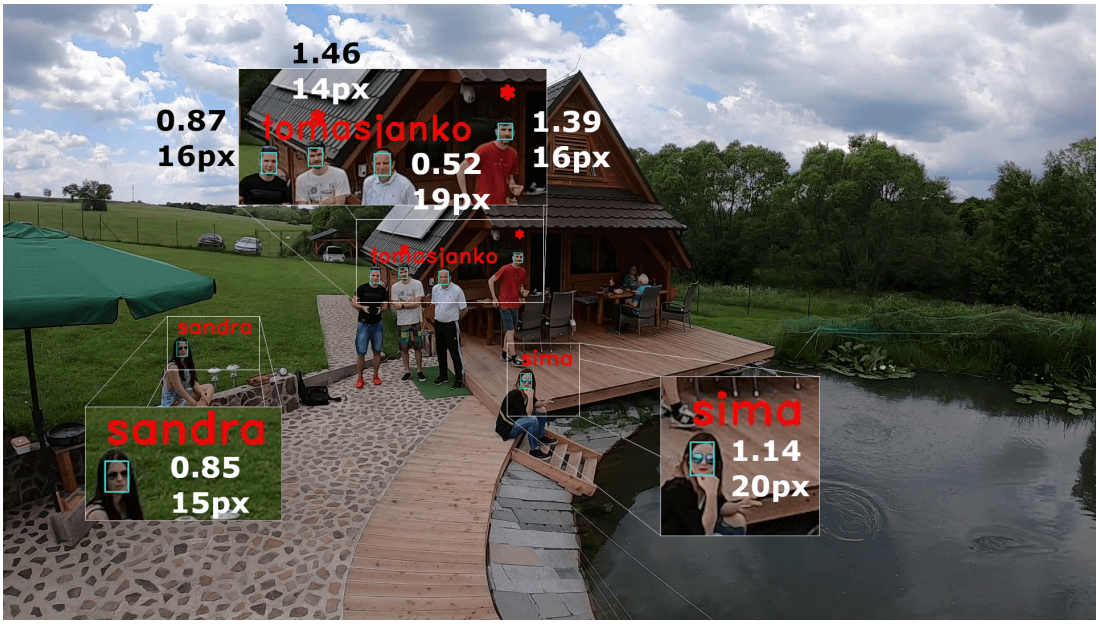
Obr. 5.2: Graf znázorňujúci presnosť rozpoznávania pre rôzne šírky tváre v pixloch. Prvý stĺpec v dvojici pre každú skupinu (červený) znázorňuje nesprávne rozpoznanú tvár, druhý stĺpec (zelený) znázorňuje správne rozpoznanú tvár. Výška stĺpcov odpovedá percentuálnemu podielu v danej skupine. Nad každým stĺpcom je celkový počet správne alebo nesprávne rozpoznaných tvárí v danej skupine.

Detektor bol schopný detegovať aj niekoľko tvárí so šírkou menšou ako 10 pixlov. Rozpoznávanie je však pri takýchto veľkostiach nepresné, keďže na tvári je malé množstvo informácii. Pri použití kamery so širokým uhlom zorného poľa táto veľkosť tváre odpovedá vzdialenosti okolo 16 metrov pri rozlíšení 4K. Pri spracovaní videa v rámci experimentu som okrem ohraničujúceho obdĺžnika okolo tváre a mena osoby nad ním nechal zobraziť na výsledných snímkach aj šírku tváre v pixloch a vzdialenosť k najbližšej tvári. Obrázky 5.4, 5.5 a 5.6 som upravil, aby v nich bol jednoduchšie viditeľný výsledok. Osoby, ktoré neboli rozpoznané, majú namiesto mena priradený znak „* “.

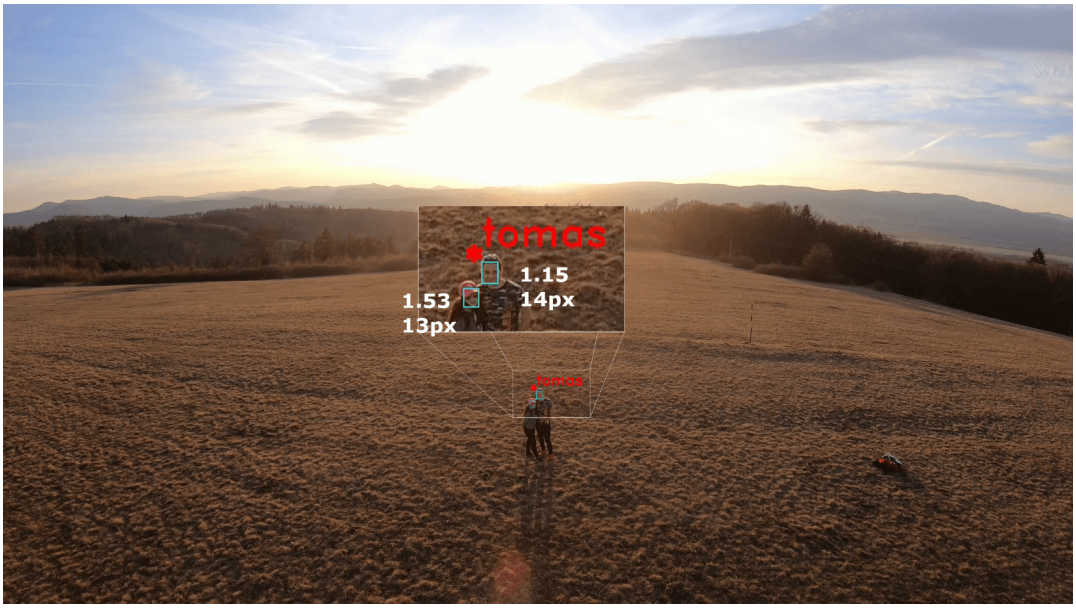
Vlastnosti implementovaného systému pre rozpoznávanie tvárí som vyhodnotil pomocou krivky ROC (viď. obrázok 5.3). Hodnoty sú získané z výsledkov v tomto experimente, kde boli vyradené prípady, kedy sa na obrázku nachádzala neznáma osoba (tj. osoba, ktorej tvár nebola zaregistrovaná v databáze tvárí). Krivka je len orientačná, keďže sa jedná o dátovú sadu s malým množstvom dát. Plocha pod krivkou (AUC) je rovná 0,87.



Obr. 5.3: Krivka ROC získaná z výsledkov rozpoznávania na vlastných testovacích dátach. Plocha pod krivkou (AUC) je rovná 0,87.



Obr. 5.4: Video snímané v rozlíšení Full HD. Dokopy sa na videu nachádza 7 osôb, z toho 2 osoby nie sú v databáze. Ich vzdialenosť od najbližšej tváre v databáze je 1,39 a 1,46, čo je vzdialenosť vyššia ako zvolený prah. Šírka tvárí sa pohybuje v rozmedzí 14 - 20 pixelov a známe tváre sú rozpoznané správne. Vzdialenosť drona od osôb je asi 9 metrov a výška snímania je zhruba 4 metre.



Obr. 5.5: Video snímané v rozlíšení 2,7K. Na zázname sa nachádzajú 2 osoby, ktoré sú aj v databáze tvárí. Jedna osoba nie je rozpoznaná, pretože vzdialenosť k najbližšej tvári v databáze je 1,53. To je spôsobené tým, že táto osoba má na zázname slnečné okuliare a čiapku. Druhá osoba je rozpoznaná správne. Šírka tvárí je 13 a 14 pixelov. Vzdialenosť drona od osôb je asi 12 metrov a výška snímania je asi 8 metrov.

Obr. 5.6: Video snímané v rozlíšení 4K. Na zázname sa nachádza 5 ľudí, pričom jedna z nich sa nenachádza v databáze. Jedna osoba nachádzajúca sa v databáze nebola v snímke rozpoznaná, pretože je jej vzdialenosť od najbližšej tváre v databáze rovná 1,32, čo je pravdepodobne spôsobené kombináciou malej šírky tváre a zlého osvetlenia. Neznáma osoba má túto vzdialenosť rovnú 1,42. Ostatné osoby sú rozpoznané správne. Veľkosť tvárí sa pohybuje v rozmedzí 21 - 32 pixelov. Vzdialenosť drona je v tomto prípade 8 metrov a výška snímania je 4 metre.

Kapitola 6

Záver

Cieľom tejto práce bolo vytvoriť aplikáciu, ktorá je schopná rozpoznať ľudí podľa tvárí z databáze z video záznamov z drona. Bolo nutné preštudovať si problematiku zaoberajúcu sa neurónovými sieťami, spracovaním obrazu a biometriou. Po analýze dát z dronov a preskúmaní existujúcich možností detekcie a rozpoznávania osôb boli zistené limity existujúcich riešení. Niekoľké existujúce riešenia boli porovnané na dátach z dátovej sady DroneFace, ktorá napodobňuje zábery z drona a na dátach z vlastnej dátovej sady. Porovnanie bolo vykonané niekoľkými jednoduchšími skriptami.

Tieto skripty poslúžili ako predloha pre hlavnú funkciu aplikácie. V tejto aplikácii bol použitý detektor MTCNN pre detekciu tvárí, extraktor príznakových vektorov z tvárí na základe ArcFace a algoritmus pre sledovanie tvárí MedianFlow tracker. Výsledná aplikácia bola na zázname s Full HD rozlíšením schopná rozpoznať tvár, ktorá mala šírku menej ako 10 pixlov s rýchlosťou spracovania okolo 8 snímok za sekundu. Aplikácia je viacvláknová a multiplatformová. Je schopná vytvoriť databázu z tvárí určených používateľom a uložiť túto databázu do súboru, ktorý je neskôr možné znovu načítať do aplikácie vizualizovať ho a použiť ho pri spracovaní videa v aplikácii. Výsledný algoritmus bol otestovaný na menšej dátovej sade, kde bol schopný správne rozpoznať 392 zo 461 detegovaných tvárí, ktoré mali šírku od 8 do 161 pixlov.

V budúcnosti je možné aplikáciu rozšíriť vo viacerých smeroch. Bolo by vhodné pridať funkcionalitu, kedy by sa vytvorená databáza tvárí dala upravovať priamo v aplikácii. V rámci algoritmu pre rozpoznávanie tvárí by pri väčšej databáze tvárí bolo vhodné vytvoriť klasifikátor. Do aplikácie je v budúcnosti možné pridať funkcionalitu, ktorá by dokázala načítať aj jednotlivé fotografie určené pre spracovanie. V aplikácii bol použitý algoritmus pre detekciu tvárí starý 5 rokov. Aj napriek tomu, že algoritmus dosahuje dobré výsledky, by bolo vhodné použiť novší, kvalitnejší algoritmus. To sa mne nepodarilo z dôvodov vyšších hardvérových nárokov modernejších algoritmov. S rýchlosťou vývoja neurónových sietí nebude v budúcnosti problém prekonať presnosť detekcie a rozpoznávania tejto aplikácie. Samotné porovnanie presnosti algoritmov prebehlo na pomerne malej a slabo oanotovanej dátovej sade. Výsledky testovania by mohli byť presnejšie, keby existovalo viacero dátových sád zaoberajúcich sa problematikou rozpoznávania osôb na záberoch z drona.

Literatúra

[1] Amelio, G. F. CHARGE-COUPLED DEVICES. *Scientific American*. Scientific American, a division of Nature America, Inc. 1974, zv. 230, č. 2, s. 22–31. ISSN 00368733, 19467087.

[2] Amos, B., Ludwiczuk, B. a Satyanarayanan, M. *OpenFace: A general-purpose face recognition library with mobile applications*. CMU-CS-16-118, CMU School of Computer Science, 2016.

[3] Brown, J. *GoPro Karma: Features, Reviews, Specifications, Competitors* [online]. My Drone Lab [cit. 2020-04-21]. Dostupné z: https://www.mydronelab.com/reviews/gopro-karma.html.

[4] Communication, C. 1st. *ANALOG VS. DIGITAL IP SECURITY CAMERAS & CCTV SYSTEMS* [online]. 2017 [cit. 2019-12-02]. Dostupné z: https://www.c1c.net/blog/analog-vs-digital-security-cameras-cctv/.

[5] Dahl, G. E., Sainath, T. N. a Hinton, G. E. Improving deep neural networks for LVCSR using rectified linear units and dropout. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, s. 8609–8613. ISBN 978-1-4799-0356-6.

[6] Deng, J., Guo, J., Xue, N. a Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach Convention and Entertainment Center, Long Beach, California, USA: IEEE, June 2019, s. 4690–4699. ISBN 978-1-7281-3293-8.

[7] DJI. *Mavic 2 Specifications* [online]. DJI [cit. 2020-04-21]. Dostupné z: https://www.dji.com/sk/mavic-2/info#specs.

[8] Draelos, R. *Measuring Performance: AUC (AUROC)* [online]. 2019 [cit. 2020-04-13]. Dostupné z: https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/.

[9] Drahanský, M., Orság, F., Doležel, M. et al. *Biometrie*. 1. vyd. Computer Press, s.r.o, 2011. 294 s. ISBN 978-80-254-8979-6.

[10] Fumo, D. *A Gentle Introduction To Neural Networks Series — Part 1* [online]. August 2017 [cit. 2019-12-24]. Dostupné z: https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc.

[11] George., D. *CCD versus CMOS: Which is Better?* [online]. Diffraction Limited, 9. apríla 2018 [cit. 2020-02-10]. Dostupné z: https://diffractionlimited.com/ccd-versus-cmos-better/.

[12] GoPro. *HERO7 Black Tech Specs* [online]. GoPro [cit. 2020-04-21]. Dostupné z: https://gopro.com/en/gr/shop/hero7-black/tech-specs?pid=CHDHX-701-master.

[13] Grother, P. J., Ngan, M. L. a Hanaoka, K. K. *Ongoing face recognition vendor test (frvt) part 2: Identification.* 2018.

[14] Guo, Y., Zhang, L., Hu, Y., He, X. a Gao, J. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In: *Computer Vision – ECCV 2016.* Cham: Springer International Publishing, 2016, s. 87–102. ISBN 978-3-319-46487-9.

[15] Haykin, S. *Neural networks: a comprehensive foundation.* 1. vyd. USA: Prentice Hall PTR, 1994. 768 s. ISBN 978-0-02-352761-6.

[16] He, K., Zhang, X., Ren, S. a Sun, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* Las Vegas, NV, USA: IEEE, June 2016, s. 770–778. ISBN 978-1-4673-8851-1.

[17] Held, D., Thrun, S. a Savarese, S. Learning to track at 100 fps with deep regression networks. In: *European Conference on Computer Vision.* Amsterdam, The Netherlands: Springer, 2016, sv. 9905, s. 749–765. ISBN 978-3-319-46447-3.

[18] Holčík, J., Komenda, M. a kol. *Matematická biologie: e-learningová učebnice* [online]. 1. vyd. Brno: Masarykova univerzita, 2015 [cit. 2019-12-18]. Dostupné z: https://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence.

[19] Hsu, H.-J. a Chen, K.-T. DroneFace: An Open Dataset for Drone Research. In: *Proceedings of ACM MMSys 2017 (Dataset Track).* Taipei, Taiwan: Association for Computing Machinery, Jun 2017, s. 187–192. ISBN 978-1-4503-5002-0.

[20] Hsu, R.-L. *Face Detection and Modeling for Recognition.* 2002. Dizertačná práca. Citeseer.

[21] Huang, G. B., Ramesh, M., Berg, T. a Learned Miller, E. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.* 07-49. University of Massachusetts, Amherst, October 2007.

[22] Imaging, S. *CCD sensor types* [online]. [cit. 2020-02-10]. Dostupné z: https://www.stemmer-imaging.com/en-gb/knowledge-base/ccd/.

[23] Innovatrics, s. *Innovatrics: Top Performer in Every NIST FRVT Category* [online]. [cit. 2020-01-07]. Dostupné z: https://www.innovatrics.com/awards/nist-frvt-top-performer-categories/.

[24] Innovatrics, s. *SmartFace* [online]. [cit. 2020-01-07]. Dostupné z: https://www.innovatrics.com/face-recognition-solutions/.

[25] Ioffe, S. a Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ArXiv preprint arXiv:1502.03167.* 2015.

[26] Jain, A. K., Flynn, P. a Ross, A. A. *Handbook of biometrics*. 1. vyd. Springer, 2008. ISBN 978-0-387-71040-2.

[27] Jain, A. K., Mao, J. a Mohiuddin, K. M. Artificial neural networks: A tutorial. *Computer*. IEEE. 1996, zv. 29, č. 3, s. 31–44. ISSN 1558-0814.

[28] Jain, A. K., Ross, A. A. a Nandakumar, K. *Introduction to Biometrics*. 1. vyd. Springer, 2011. ISBN 978-0-387-77325-4.

[29] Kalal, Z., Mikolajczyk, K. a Matas, J. Forward-backward error: Automatic detection of tracking failures. In: *2010 20th International Conference on Pattern Recognition*. Istanbul, Turkey: IEEE, 2010, s. 2756–2759. ISBN 9781424475414.

[30] Kanade, T. Picture processing system by computer complex and recognition of human faces. Kyoto University. 1974.

[31] Kirby, M. a Sirovich, L. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern analysis and Machine intelligence*. IEEE. 1990, zv. 12, č. 1, s. 103–108. ISSN 0162-8828.

[32] Kruegle, H. *CCTV Surveillance: Video practices and technology*. 2. vyd. Elsevier, 2007. ISBN 978-0-7506-7768-4.

[33] LeCun, Y., Bengio, Y. a Hinton, G. Q. *Nature*. Nature Publishing Group. 2015, zv. 521, č. 7553, s. 436–444.

[34] Litwiller, D. Ccd vs. cmos. *Photonics spectra*. 2001, zv. 35, č. 1, s. 154–158.

[35] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E. et al. SSD: Single Shot MultiBox Detector. *CoRR*. 2015, abs/1512.02325.

[36] Masi, I., Tran, A. T., Hassner, T., Leksut, J. T. a Medioni, G. Do we really need to collect millions of faces for effective face recognition? In: Springer. *European Conference on Computer Vision*. 2016, s. 579–596.

[37] McCulloch, W. S. a Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*. Springer. 1943, zv. 5, č. 4, s. 115–133.

[38] Mehrotra, K., Mohan, C. K. a Ranka, S. *Elements of artificial neural networks*. 2. vyd. MIT press, 2000. ISBN 0-262-13328-8.

[39] Mynepalli, S. C. *Recognizing Tiny Faces*. Pittsburgh, Pennsylvania 15213, 2019. Diplomová práca. Carnegie Mellon University.

[40] Ng, H.-W. a Winkler, S. A data-driven approach to cleaning large face datasets. In: *2014 IEEE International Conference on Image Processing (ICIP)*. 2014, s. 343–347.

[41] Nilsson, F. et al. *Intelligent Network Video: Understanding Modern Video Surveillance Systems*. 2. vyd. CRC Press, 2016. ISBN 978-1-4665-5521-1.

[42] Optics, E. *Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications* [online]. [cit. 2020-02-10]. Dostupné z: https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-camera-sensors-for-machine-vision-applications/.

[43] Parrot. *Drone Camera 4k HDR ANAFI | Parrot Official* [online]. Parrot [cit. 2020-04-21]. Dostupné z: https://www.parrot.com/global/drones/anafi.

[44] Pascanu, R., Mikolov, T. a Bengio, Y. Understanding the exploding gradient problem. *CoRR, abs/1211.5063*. 2012, zv. 2.

[45] Patel, A. *Chapter-7 Under-fitting, over-fitting and its solution* [online]. Medium, 24. júla 2018 [cit. 2020-05-14]. Dostupné z: https://medium.com/ml-research-lab/under-fitting-over-fitting-and-its-solution-dc6191e34250.

[46] Rosenblatt, F. Perceptron simulation experiments. *Proceedings of the IRE*. IEEE. 1960, zv. 48, č. 3, s. 301–309.

[47] Schroff, F., Kalenichenko, D. a Philbin, J. Facenet: A unified embedding for face recognition and clustering. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, s. 815–823.

[48] Security, C. *CCTV Explained* [online]. 2016 [cit. 2019-12-02]. Dostupné z: https://www.complete-security.co.uk/what-is-cctv.

[49] SHARMA, S. *Activation Functions in Neural Networks* [online]. September 2017 [cit. 2019-12-24]. Dostupné z: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[50] Sirovich, L. a Kirby, M. Low-dimensional procedure for the characterization of human faces. *Josa a*. Optical Society of America. 1987, zv. 4, č. 3, s. 519–524.

[51] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. a Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*. JMLR. org. 2014, zv. 15, č. 1, s. 1929–1958.

[52] Stan Z. Li, S. Z. L. A. K. J. e. *Handbook of Face Recognition*. 2. vyd. Springer-Verlag London, 2011. ISBN 978-0-85729-931-4.

[53] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. et al. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, s. 1–9.

[54] Taigman, Y., Yang, M., Ranzato, M. a Wolf, L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In:. September 2014, s. 1701–1708. DOI: 10.1109/CVPR.2014.220.

[55] Trenz, O., Fejfar, J., Popelka, O., Kolomazník, J. a Štencl, M. *ELearningová opora k předmětu Umělá inteligence 1* [online]. Brno: [b.n.], 2010 [cit. 2019-12-18]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/index.pl?opora=2068.

[56] Turk, M. a Pentland, A. Eigenfaces for recognition. *Journal of cognitive neuroscience*. MIT Press. 1991, zv. 3, č. 1, s. 71–86.

[57] Wikipedia. *Active-pixel sensor* [online]. Apríl 2020 [cit. 2020-02-10]. Dostupné z: https://en.wikipedia.org/wiki/Active-pixel_sensor.

[58] Wikipedia. *Overfitting* [online]. Apríl 2020 [cit. 2018-12-18]. Dostupné z: https://en.wikipedia.org/wiki/Overfitting.

[59] Wikipedia. *Unmanned aerial vehicle* [online]. Máj 2020 [cit. 2020-04-21]. Dostupné z: https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle.

[60] Wolf, L., Hassner, T. a Maoz, I. *Face recognition in unconstrained videos with matched background similarity*. IEEE, 2011.

[61] Yadav, S. *Weight Initialization Techniques in Neural Networks* [online]. November 2018 [cit. 2019-12-25]. Dostupné z: https://towardsdatascience.com/weight-initialization-techniques-in-neural-networks-26c649eb3b78.

[62] Yi, D., Lei, Z., Liao, S. a Li, S. Z. Learning face representation from scratch. *ArXiv preprint arXiv:1411.7923*. 2014.

[63] Zhang, K., Zhang, Z., Li, Z. a Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*. Oct 2016, zv. 23, č. 10, s. 1499–1503. DOI: 10.1109/LSP.2016.2603342. ISSN 1070-9908.

Príloha A

Obsah priloženého pamäťového média

Na priloženom pamäťovom sa nachádza:

- adresár `/app` – zdrojové kódy k aplikácii, vyhodnocovacím skriptom a nástroju spustiteľnému z príkazovej riadky. Adresár má nasledovnú štruktúru:
  - súbor `/app/README.md` – súbor s návodom pre spustenie všetkých skriptov vrátane spustenia aplikácie, sú tu odkazy na spracované videá pomocou aplikácie a nachádza sa tu aj bližší popis jednotlivých súborov v adresári `app`,
  - adresár `/app/models` – obsahuje váhy pre model ArcFace,
  - adresár `/app/openface_req` – obsahuje váhy potrebné pre model OpenFace,
  - adresár `/app/qt` – obsahuje súbory vytvorené pomocou nástroja QtDesigner definujúce užívateľské rozhranie.
- adresár `/data` – testovacie videá a fotografie pre vytvorenie databáze pre aplikáciu. Adresár má nasledovnú štruktúru:
  - adresár `/data/faces` – adresárová štruktúra pre vytvorenie databáze tvárí z vlastnej dátovej sady,
  - adresár `/data/faces_drone_face` – adresárová štruktúra pre vytvorenie databáze tvárí z dátovej sady DroneFace,
  - adresár `/data/videos` – záznamy pochádzajúce z vlastnej dátovej sady.
- adresár `/tex` – zdrojové texty a súbory technickej správy v jazyku LaTeX,
- súbor `demo.mp4` – krátke demonštračné video k aplikácii,
- súbor `text.pdf` – technická správa,
- súbor `text_print.pdf` – tlačiteľný formát technickej správy (odkazy sú čiernobiele).