



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

KAMERA PRO ROZPOZNÁVÁNÍ OSOB PODLE OBLIČEJE
BIOMETRICKÉ SYSTÉMY

Bc. Ján Jakub Kubík

24. novembra 2022

Obsah

1	Úvod	2
2	Teória	3
2.1	Rozoznávanie tváre	3
2.1.1	Detekcia	3
2.1.2	Extrakcia rysov	4
2.1.3	Porovnávanie rysov	4
3	Použité modely	5
3.1	FaceNet	5
3.2	ArcFace	6
3.3	SFace	6
4	Návrh a implementácia	7
4.1	Použitý software	7
4.2	Návrh a Implementácia	7
5	Testovanie	9
5.1	Použitý hardware	9
5.2	Nastavenie testov	9
5.3	Testy	11
5.4	Zhrnutie testov	14
6	Záver	15
	Literatúra	16
A	Súbory k projektu	17

1 Úvod

Rozoznávanie tvári prešlo veľkým vývojom od pôvodne štatistických metód, ktoré sú v dnešnej dobe takmer úplne nahradené neurónovými sieťami. V poslednom desaťročí sa stáva rozoznávanie tvári čím ďalej viac populárne. Poskytuje diskretný nerušivý a bezkontaktný spôsob detekcie, identifikácie a verifikácie osôb bez ich vedomia. Dnes je táto technológia úplne bežná na miestach ako sú letiská a iné frekventované miesta.

Hlavným cieľom projektu je použiť už predtrénované modely neurónových sietí na rýchle rozoznávanie osôb vo videu z USB kamery, ktorá je pripojená k embedded počítaču. A následne tieto modely vyhodnotiť so zameraním sa na rýchlosť.

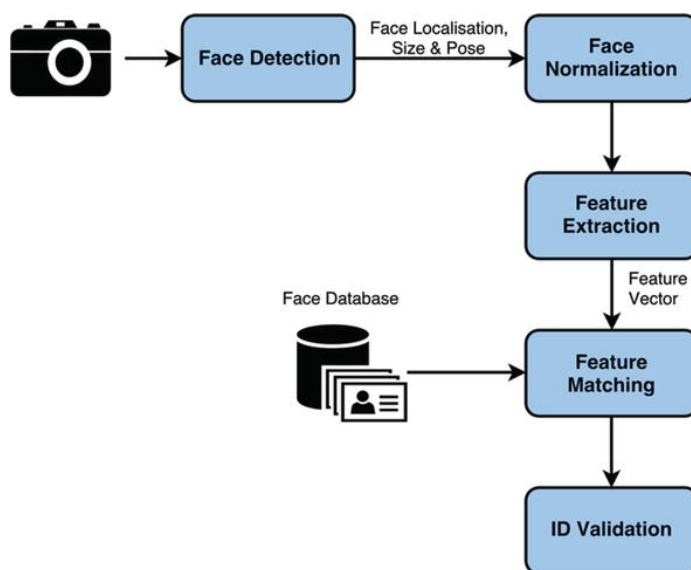
Kapitola 2 obsahuje stručný popis problematiky rozoznávania tvári. V Kapitole 3 je vysvetlený výber modelov neurónových sietí spolu s ich funkcionalitou. Kapitola 4 popisuje použitý hardware a software spolu s návrhom a implementáciou výsledného riešenia. Kapitola 5 sa zaoberá podrobným testovaním.

2 Teória

Kapitola sa zaoberá dôležitou teóriou pre rozoznávanie tvári, ktoré sa skladá z viacerých krokov. Postupne sú tieto kroky ďalej v tejto kapitole vysvetlené.

2.1 Rozoznávanie tváre

Obrázok 2.1 znázorňuje všeobecný diagram jednotlivých aktivít systémov pre rozoznávanie tváre. Na vstupe je obrázok alebo video. V ktorom sa najskôr musia rozoznať tváre, ďalej sa vykonávajú rôzne orezávania, rotácie a iné úpravy obrázku pre lepšiu extrakciu rysov z tváre. Tento krok nie je nutný ale je vhodný. Ďalej sa pomocou neurónovej siete z obrázku extrahuje vektor rysov, ktorý je následne porovnávaný s databázou známych tvári.



Obr. 2.1: Diagram všeobecného postupu pre rozoznávanie tvári [4]

2.1.1 Detekcia

Správna detekcia tváre je fundamentálny krok, bez ktorého žiadne rozoznávanie tváre nemôže fungovať. Detekcia je samostatná oblasť ktorá je kľúčová aj v iných oblastiach ako len rozoznávanie tvári. Je kľúčová pre samo-zaostrovanie na mobilných zariadeniach, rozoznávanie rôznych vyjadrení tváre, značkovanie na sociálnych sieťach a mnoho ďalších. Rozšírené

metódy na detekciu tváre založené na neurónových sieťach sú:

- Single Shot Multibox Detector (SSD) [3]
- Max-Margin Object Detection (MMOD) [2]
- Multi-task Cascaded Convolutional Networks (MTCNN) [6]

Cieľom tejto práce je porovnávať modely na rozoznávanie osôb. Z toho dôvodu som sa na základe článku ¹ rozhodol použiť metódu SSD. V danom článku vychádza SSD z vyššie zmienených metód ako najrýchlejšia a je dosť presná.

2.1.2 Extrakcia rysov

Táto fáza je považovaná za najdôležitejšiu z fáz v systémoch na rozoznávanie tvári. Pretože kvalita celého systému je priamo závislá na kvalite extrahovaných rysov. Extrahované rysy sú reprezentované ako viac-dimenzionálne vektory. Modely neurónových sietí na extrakciu rysov:

- FaceNet
- ArcFace
- SFace

Výberom a popisom týchto modelov sa zaoberá ďalšia kapitola.

2.1.3 Porovnávanie rysov

Po samotnej extrakcii rysov prichádza na rad porovnávanie extrahovaných rysov s už uloženými rysmi, ktoré patria konkrétnym osobám uloženým v databáze. Metódy na porovnávanie sú:

- kosínová vzdialenosť
- euklidovská vzdialenosť

¹<https://sefiks.com/2020/08/25/deep-face-detection-with-opencv-in-python/>

3 Použité modely

V tejto kapitole je vysvetlený výber použitých modelov. Ich opis a opis použitých dátových súd na základe ktorých boli jednotlivé modely predtrénované a testované.

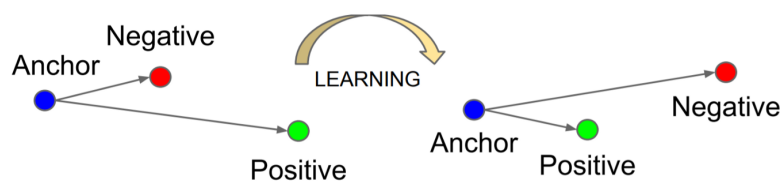
Pre testovanie som si vybral modely na základe vyhodnotenia presnosti jednotlivých modelov v mnou používanom frameworku ¹. Sú tam porovnávané presnosti jednotlivých modelov na dátovej sade Labelled Faces in the Wild (LFW). Vybrané predtrénované modely sú FaceNet (99.2%), ArcFace (99.41%) a SFace (99.60%). LFW je benchmarková dátová sada pozostávajúca z 13 233 obrázkov pre 5749 rôznych osôb ².

Všetky použité modely boli autorom frameworku predtrénované na dátovej sade VGG-Face ³. VGG-Face je dátová sada obsahujúca celebrity v rôznych prostrediach. Obsahuje 2600 000 obrázkov s viac ako 2000 rôznymi osobami ⁴.

3.1 FaceNet

FaceNet je jedna z prvých hlbokých neurónových sietí, ktoré dosiahli lepšie výsledky v rozpoznávaní obrázkov ako človek. Princíp fungovania tejto neurónovej siete je založený na tom, že z detekovaných a normalizovaných obrázkov tvári vytvára takzvané embedding vektory, kde vzdialenosti medzi týmito vektormi odpovedajú miere podobnosti medzi skúmanými obrázkami.

Na trénovanie je použitá takzvaná triplet loss funkcia, ktorá sa snaží minimalizovať vzdialenosť medzi vektormi jednotlivých obrázkov z 1 rovnakej triedy a zároveň maximalizovať vzdialenosť tejto triedy on iných.



Obr. 3.1: Triplet loss [5]

Na obrázku 3.1 je znázornená triplet loss funkcia. Trénovacia sada dát je rozdelená na trojice,

¹<https://github.com/serengil/deepface>

²<http://vis-www.cs.umass.edu/lfw/>

³https://github.com/serengil/deepface_models/

⁴https://www.robots.ox.ac.uk/vgg/data/vgg_face/

kde je v každej trojice vzorový obrázok (anchor) spolu s obrázkom rovnakej osoby z rovnakej triedy (positive) a obrázok z inej triedy (negative).

FaceNet tripple loss ma problém dĺžky tréovania na veľkých dátových sadách. Je to zapríčinené veľkým množstvom tripletov na tréovanie [5].

3.2 ArcFace

ArcFace funguje tiež na rovnakom princípe ako FaceNet (vytváranie emebdding vektorov) ale tréovacia funkcia je iná. Je to additive angular margin stratová funkcia využívajúca uhlovú distribúciu vektorov. Použitím tejto tréovacej funkcie sa podarilo znížiť celkovú dĺžku tréovania. Bližšie informácie o ArcFace sú [1].

3.3 SFace

SFace funguje tiež na rovnakom princípe ako vyššie zmienené modely (vytváranie emebdding vektorov) ale tréovacia funkcia je iná. Je to sigmoid-constrained hypersphere stratová funkcia. Podrobnejšie informácie o SFace sú v [7].

4 Návrh a implementácia

V tejto kapitole sú vysvetlené použité technológie, návrh a implementácia.

4.1 Použitý software

Pre projekt som použil programovací jazyk Python vo verzii 3.8 s nasledujúcim softwarom:

- **Deepface** ¹ je odľahčený framework pre rozoznávanie tvári a analýzu rôznych výrazov tváre. Ja som z neho využil predtrénované modely na rozoznávanie a detekciu tvári. Tieto modely sú viac vysvetlené v kapitole 3.
- **OpenCV** ² je knižnica rôznych funkcií zameraná hlavne na počítačové videnie v reálnom čase.
- **Tensorflow 2** ³ je komplexný framework určený k strojovému učeniu.
- **NumPy** ⁴ a **Pandas** ⁵. Tieto dve knižnice slúžia na zjednodušenie rôznych výpočtov z programátorského hľadiska spolu so zjednodušením manipulácie s rôznymi dátami.

4.2 Návrh a Implementácia

Návrh má štruktúru vychádzajúcu z diagramu všeobecného postupu pre rozoznávanie tvári 2.1. Tento diagram je pre jeden konkrétny obrázok. Keďže môj program má na vstupe live video z kamery, tak pozostáva z nekonečného cyklu, kde sa v tele cyklu zachytí 1 konkrétny obrázok, vykoná sa pre neho celá pipeline zmieneného diagramu a tak dokola pre ďalšie obrázky.

Implementácia pozostáva zo 4tyroch modulov:

- **main.py** – hlavný vstupný bod programu, ktorý slúži na spúšťanie samotného live rozoznávanie tvári pomocou funkcie `live_video_face_recognition`

¹<https://github.com/serengil/deepface>

²<https://pypi.org/project/opencv-python/>

³<https://pypi.org/project/tensorflow/>

⁴<https://pypi.org/project/numpy/>

⁵<https://pypi.org/project/pandas/>

- **settings.py** – slúži na nastavovanie dôležitých konštánt ako je cesta k DB, názov modelu na detekciu tvári, názov použitého modelu na extrakciu rysov, metriku na meranie vzdialeností medzi vektormi a časovú konštantu, ktorá sa používa na počítanie FPS.
- **utils.py** – v sebe zahŕňa pomocné funkcie. Predovšetkým na počítanie FPS, a tvorbu embedding vektorov z obrázkov uložených vo DB na rozoznávanie.
- **face_recognition.py** – obsahuje implementáciu funkcie `live_video_face_recognition`. Táto funkcia zahŕňa celkovú logiku pre rozoznávanie.

5 Testovanie

V tejto kapitole je vysvetlený použitý hardware pri testovaní, problémy pri testovaní a samotné testovanie.

5.1 Použitý hardware

Na projekt som dostal zapožičaný hardware od vedúceho projektu Ing. Tomáša Goldmanna. Konkrétne:

- Logitech webkamera
- embedded zariadenie Up core s procesorom intel Atom a 2 GB RAM

Software pre rozoznávanie tvári som vyvíjal na mojom osobnom PC. Testovanie som chcel pôvodne robiť podľa zadania na UP core s pripojenou Logitech webkamerou. Ale po inštalácii SW na UP core som dostával chybu kôli nepodporovaným AVX jednotkám na UP core. Spôsoboval to Tensorflow, ktorý používa predkompilovaný tensorflow pre AVX jednotky. Pokúšal som sa to rôznymi návodmi opraviť. Ale nič nepomohlo a preto som sa rozhodol nakoniec robiť testovanie na mojom osobnom PC MacBook Pro 13-inch, 2019 i7 16GB RAM. Zapožičaná USB kamera taktiež nefungovala, tak som musel použiť vstavanú web kameru v mackbooku. Tieto zmeny som konzultoval s vedúcim projektu. Odkaz ¹ na video z Logitech webkamery.

5.2 Nastavenie testov

Rozhodol som sa, že otestujem akými rýchlosťami som schopný v rôznych testovacích scenároch spracovávať jednotlivé videá následovne:

- Čisto len obrázok po obrázku bez toho aby som niečo v programe robil.
- Pridaná detekcia tvári.
- Samotné rozoznávanie tvári pre každý model zvlášť.

¹<https://drive.google.com/file/d/1ELJKGa3AQPqmIzagghAg9-udxBRgiXZE/view>

Rýchlosť som sa rozhodol testovať pomocou FPS, ktoré je zobrazované v jednotlivých videách. Na záver je vypísané priemerné FPS pre daný testovací scenár a priemerná normalizovaná Euklidovská vzdialenosť porovnaných vektorov tvári s uloženými vektormi v DB.

Testovacie scenáre boli nasledovné:


- Scenár s 1 osobou.
- Scenár s viacerými osobami.
- Scenár s 1 osobou s pokrývkou hlavy.
- Scenár s 1 osobou s rôznymi rotáciami tváre.
- Scenár s 1 osobou v tmavom prostredí.

Pre správnosť otestovania som sa rozhodol pomocou webkamery na mojom notebooku nahráť všetky zmienené testovacie scenáre a použiť ich ako vstupné videá pre jednotlivé modely, nie ako live videá. Dôvodom je to, že všetky tri modely som chcel testovať na totožných scenároch. Pri live videu by ten pohyb bol vždy aspoň trochu iný.

Testovacie videá spolu s výslednými videami kde sú rozoznávané jednotlivé osoby sú na Google Disku ².

V používanom frameworku na modely autor tvrdí, že Euclidean L2 je stabilnejšie ako klasická euklidovská a kosínová vzdialenosť. Preto som sa ju rozhodol použiť vo všetkých testoch. Euklidean L2 je normalizovaná Euklidovská vzdialenosť. Normalizovaná Euklidovská vzdialenosť udáva druhú mocninu vzdialenosti medzi dvoma vektormi, ktoré boli normalizované. Pre jednotlivé modely sú hranice vzdialenosti rôzne. Pre Facenet to je 0.8. pre Sface to je 1.055836701022614. Pre Arcface to je 1.13. Ak je Euklidovská vzdialenosť vektoru rozoznanej tváre v analyzovanom obrázku v porovnaní s najlepším vektorom uloženým v DB menšia alebo rovnaká ako hranica pre daný model, tak sú tváre týchto dvoch osôb považované za totožné. Za najlepší vektor v DB je považovaný ten, ktorý je najbližšie k vzorovému vektoru.

Všetky testy bežali na CPU, ako je zrejmé z obrázku 5.1.

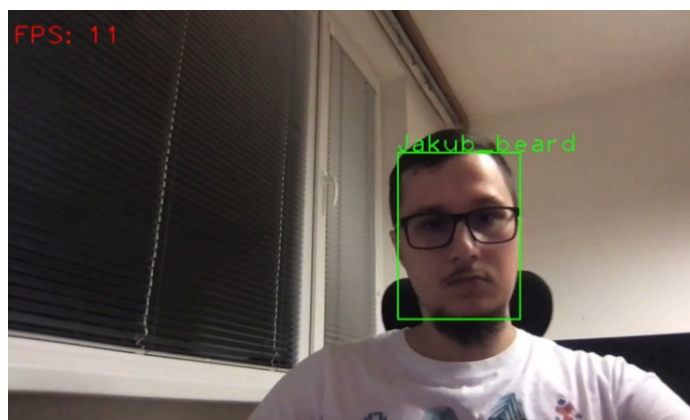
Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	% GPU	GPU Time
 Python	268.1	3:56.93	14	110	1.0	2.23

Obr. 5.1: Vyťaženie CPU a GPU pri behu testov.

²https://drive.google.com/drive/folders/1WJT8B76npY7avFYU8smD_9CMTDsZaNY4?usp=sharing

5.3 Testy

1 osoba



Obr. 5.2: Ilustračný obrázok z testovania na jednej osobe

- Priemerné FPS: 46
- Priemerné PFS s detekciou tváre: 24
- Facenet: priemerné PFS s detekciou tváre 5.5, priemerné euklidovské vzdialenosti 1.12, hraničné vzdialenosti 0.8.
- Arcface: priemerné PFS s detekciou tváre 4.21, priemerné euklidovské vzdialenosti 1.034, hraničné vzdialenosti 1.13.
- Sface: priemerné PFS s detekciou tváre 9.17, priemerné euklidovské vzdialenosti 1.037, hraničné vzdialenosti 1.056.

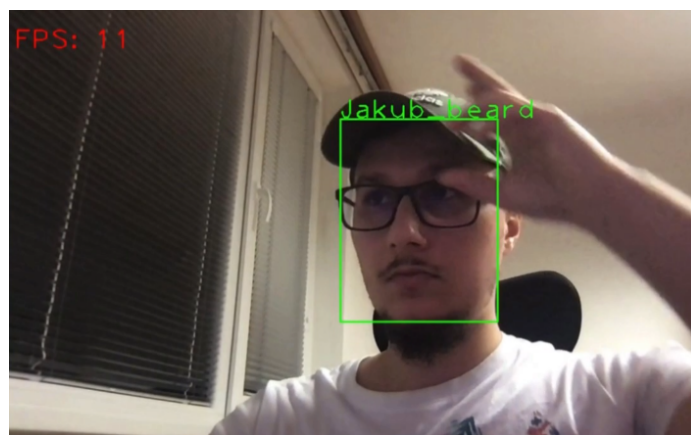
Viac osôb



Obr. 5.3: Ilustračný obrázok z testovania viacerých osôb

- Priemerné FPS: 47
- Priemerné PFS s detekciou tváre: 20
- Facenet: priemerné PFS s detekciou tváre 4.30, priemerné euklidovské vzdialenosti 1.114, hraničné vzdialenosti 0.8.
- Arcface: priemerné PFS s detekciou tváre 3.54, priemerné euklidovské vzdialenosti 1.085, hraničné vzdialenosti 1.13.
- Sface: priemerné PFS s detekciou tváre 7.58, priemerné euklidovské vzdialenosti 1.154, hraničné vzdialenosti 1.056.

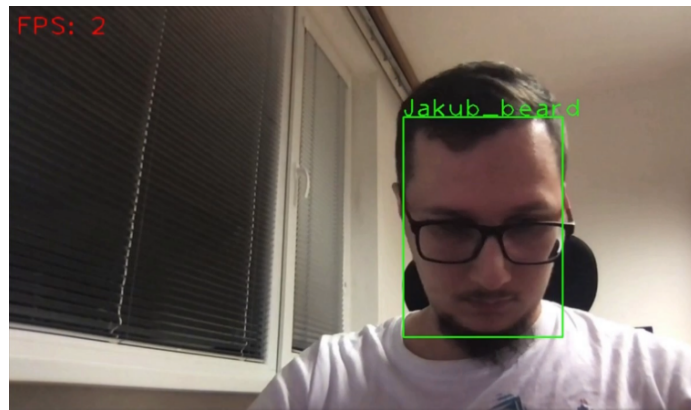
Pokrývka hlavy



Obr. 5.4: Ilustračný obrázok z testovania pokrývky hlavy

- Priemerné FPS: 44
- Priemerné PFS s detekciou tváre: 25
- Facenet: priemerné PFS s detekciou tváre 5.13, priemerné euklidovské vzdialenosti 1.179, hraničné vzdialenosti 0.8.
- Arcface: priemerné PFS s detekciou tváre 4.568, priemerné euklidovské vzdialenosti 1.079, hraničné vzdialenosti 1.13.
- Sface: priemerné PFS s detekciou tváre 9.055, priemerné euklidovské vzdialenosti 1.1036, hraničné vzdialenosti 1.056.

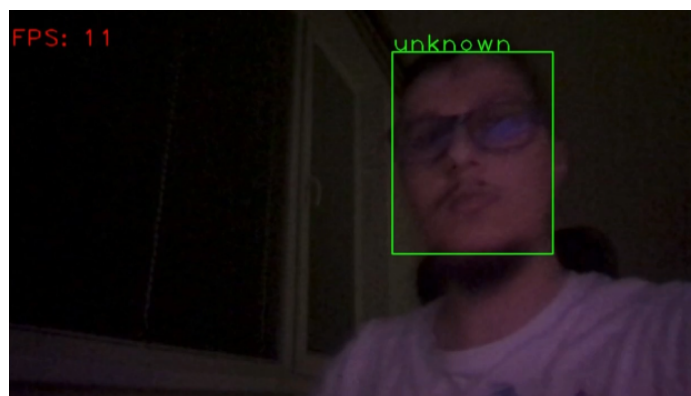
Rotácie tváre



Obr. 5.5: Ilustračný obrázok z testovania rotácie

- Priemerné FPS: 46
- Priemerné PFS s detekciou tváre: 23
- Facenet: priemerné PFS s detekciou tváre 4.986, priemerné euklidovské vzdialenosti 1.152, hraničné vzdialenosti 0.8.
- Arcface: priemerné PFS s detekciou tváre 4.183, priemerné euklidovské vzdialenosti 1.099, hraničné vzdialenosti 1.13.
- Sface: priemerné PFS s detekciou tváre 7.936, priemerné euklidovské vzdialenosti 1.105, hraničné vzdialenosti 1.056.

Tmavé prostredie



Obr. 5.6: Ilustračný obrázok z testovania v šeru

- Priemerné FPS: 46
- Priemerné PFS s detekciou tváre: 21

- Facenet: priemerné PFS s detekciou tváre 5.498, priemerné euklidovské vzdialenosti 1.058, hraničné vzdialenosti 0.8.
- Arcface: priemerné PFS s detekciou tváre 4.541, priemerné euklidovské vzdialenosti 1.082, hraničné vzdialenosti 1.13.
- Sface: priemerné PFS s detekciou tváre 9.336, priemerné euklidovské vzdialenosti 1.142, hraničné vzdialenosti 1.056.

5.4 Zhrnutie testov

Najhorší model zo všetkých čo sa týka rýchlosti aj presnosti bol určite Facenet. Veľmi podobnú rýchlosť mal aj Arcface model, ale tento model bol zas najpresnejší. Najväčšiu rýchlosť ma Sface model a bol stredne presný. Ale stále to bolo celkom rozumne použiteľné.

Facnet modelu sa nepodarilo takmer žiadnú tvár rozoznať. Možno to je absenciou predspracovania jednotlivých obrázkov. No to som nepoužil z dôvodu, že by to ešte viac spomalilo samotné už tak dosť pomalé rozoznávanie tvári.

Najhoršie dopadli všetky modely v tme a s veľkým prekryvom tváre. Rozoznanie tvári viacerých osôb dopadla na moje prekvapenie celkom dobre pre Sface a Arcface.

6 Záver

Podarilo sa mi vytvoriť program na live rozoznávanie osôb použitím už predtrénovaných modelov na rozoznávanie tvári. Tieto modely som následne otestoval pomocou rôznych testovacích scenárov. Testovanie kôli technickým problémom neprebiehало na zapožičanom embedded zariadení a externej kamere ale na mojom osobnom PC.

Najrýchlejší bol model Sface. Tento model bol stredne presný. Za ním sa umiestnil Arcface, ktorý bol síce najpresnejší, ale pomalý. Rovnako pomalý ako ArcFace bol tiež Facenet, ktorý bol veľmi nepresný.

Bolo by vhodné ešte pridať predspracovávanie obrázkov pred samotnou detekciou a rozoznávaním tvári. To by presnosti jednotlivých modelov mohlo vylepšiť ale na druhej strane aj spomaliť. Bolo by vhodné to otestovať.

Literatúra

- [1] DENG, J., GUO, J. a ZAFEIRIOU, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *CoRR*. 2018, abs/1801.07698. Dostupné z: <http://arxiv.org/abs/1801.07698>.
- [2] KING, D. E. Max-Margin Object Detection. *CoRR*. 2015, abs/1502.00046. Dostupné z: <http://arxiv.org/abs/1502.00046>.
- [3] LIU, W., ANGUELOV, D., ERHAN, D., SZEGEDY, C., REED, S. E. et al. SSD: Single Shot MultiBox Detector. In: LEIBE, B., MATAS, J., SEBE, N. a WELLING, M., ed. *ECCV (1)*. Springer, 2016, s. 21–37. Lecture Notes in Computer Science, sv. 9905. Dostupné z: <http://dblp.uni-trier.de/db/conf/eccv/eccv2016-1.html#LiuAESRFB16>. ISBN 978-3-319-46447-3.
- [4] RICARDO RIBEIRO, D. L. a NEVES, A. *Access Control in the Wild Using Face Verification* [online]. [cit. 2022-10-27]. Dostupné z: <https://www.intechopen.com/chapters/62461>.
- [5] SCHROFF, F., KALENICHENKO, D. a PHILBIN, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. *CoRR*. 2015, abs/1503.03832. Dostupné z: <http://arxiv.org/abs/1503.03832>.
- [6] ZHANG, K., ZHANG, Z., LI, Z. a QIAO, Y. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *CoRR*. 2016, abs/1604.02878. Dostupné z: <http://arxiv.org/abs/1604.02878>.
- [7] ZHONG, Y., DENG, W., HU, J., ZHAO, D., LI, X. et al. SFace: Sigmoid-Constrained Hypersphere Loss for Robust Face Recognition. *IEEE Transactions on Image Processing*. 2021, roč. 30, s. 2587–2598.

A Súbory k projektu

Odkaz na github repozitár ¹.

Dodatok obsahujem zoznam súborov k projektu.

- database/
- README.md
- face_recognition.py
- main.py
- requirements.txt
- settings.py
- utils.py

¹<https://github.com/Jakub-Kubik/BIO>