VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

APLIKACE PRO ANALÝZU ODOLNOSTI ALGORITMŮ PRO ROZPOZNÁVÁNÍ PODLE OBLIČEJE VŮČI DEEP-FAKE SNÍMKŮM

AN APPLICATION FOR ANALYZING THE RESILIENCE OF FACIAL RECOGNITION ALGORITHMS AGAINST A DEEPFAKE IMAGE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **ADAM KUČÍK**
AUTHOR

VEDOUCÍ PRÁCE **Ing. TOMÁŠ GOLDMANN, Ph.D.**
SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

162134

Ústav:	Ústav inteligentních systémů (UITS)
Student:	**Kučík Adam**
Program:	Informační technologie
Název:	**Aplikace pro analýzu odolnosti algoritmů pro rozpoznávání podle obličeje vůči deepfake snímkům**
Kategorie:	Umělá inteligence
Akademický rok:	2023/24

Zadání:

1. Seznamte se s dostupnými metodami pro generování deepfake snímků obličeje.
2. Seznamte se s algoritmy pro rozpoznávání osob podle snímku obličeje.
3. S využitím existujících řešení navrhněte aplikaci, která bude generovat deepfake snímky a provádět rozpoznávání podle obličeje vůči referečním snímkům.
4. Navržené řešení implementujte v programovacím jazyce Python.
5. Proveďte experimenty pro určení odolnosti dostupných algoritmů pro rozpoznávání osob podle obličeje vůči deepfake snímkům.

Literatura:

- WU, Jian, et al. A forensic method for deepfake image based on face recognition. In: *Proceedings of the 2020 4th High Performance Computing and Cluster Technologies Conference & 2020 3rd International Conference on Big Data and Artificial Intelligence*. 2020. p. 104-108.
- AHMED, Saadaldeen Rashid, et al. Analysis survey on deepfake detection and recognition with convolutional neural networks. In: *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 2022. p. 1-7.

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

Vedoucí práce:	**Goldmann Tomáš, Ing., Ph.D.**
Vedoucí ústavu:	Hanáček Petr, doc. Dr. Ing.
Datum zadání:	1.11.2023
Termín pro odevzdání:	9.5.2024
Datum schválení:	10.7.2024

Abstrakt

Práca sa primárne zaoberá analýzou algoritmov pre rozpoznanie tváre voči deepfake snímkom. Časť riešenia práce je založená na konvolučných neurónových sieťach, ktoré sú schopné určiť, či zadaný snímok je alebo nie je deepfake. Práca skúma siamskú neurónovú sieť a jej vhodnosť pre detekciu deepfake. Jej výsledky porovnávame s modernými algoritmami pre rozpoznanie tváre ako napríklad ArcFace, DeepFace, MagFace a FaceNet. Na základe výsledkov z týchto algoritmov pre rozpoznanie tváre vyhodnocujeme dôveryhodnosť vstupných snímkov voči deepfake technológii. Taktiež porovnávame rôzne konfigurácie neurónových sietí. Zaoberáme sa aj rôznymi programami pre tvorbu deepfake snímkov ako napríklad FaceFusion alebo SimSwap. Programová časť práce je implementovaná v jazyku Python.

Abstract

This thesis primarily focuses on the analysis of algorithms for facial recognition against deepfake images. Part of the solution is based on convolutional neural networks, which are capable of determining whether a given image is a deepfake or not. The thesis examines the Siamese neural network and its suitability for deepfake detection. Its results are compared with state-of-the-art facial recognition algorithms such as ArcFace, DeepFace, MagFace, and FaceNet. Based on the results of these face recognition algorithms, we evaluate the credibility of the input images against deepfake technology. Different configurations of neural networks are also compared. Additionally, we explore different programs for creating deepfake images, such as FaceFusion and SimSwap. The programming part of the thesis is implemented in Python.

Kľúčové slová

Deepfake, Neurónové siete, Konvolučné neurónové siete, Algoritmy rozpoznanie tváre, Deepfake detekcia, Vgg sieť, Siamská neurónová sieť, DeepFace, ArcFace, FaceNet, MagFace

Keywords

Deepfake, Neural network, Convolutional neural network, Face recognition algorithms, Deepfake detecion, Vgg net, Siamese neural network, DeepFace, ArcFace, FaceNet, MagFace

Citácia

KUČÍK, Adam. *Aplikace pro analýzu odolnosti algoritmů pro rozpoznávání podle obličeje vůči deepfake snímkům.* Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. TOMÁŠ GOLDMANN, Ph.D.

Aplikace pro analýzu odolnosti algoritmů pro rozpoznávání podle obličeje vůči deepfake snímkům

Prehlásenie

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Tomáša Goldmanna, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Adam Kučík
19. júla 2024

Poďakovanie

Chcel by som poďakovať pánovi Ing. Tomášovi Goldmannovi, Ph.D za odbornú pomoc pri vedení práce a za cenné rady poskytnuté na konzultáciách pri tvorbe tejto práce. Taktiež sa chcem poďakovať rodičom za podporu počas štúdia. Ďalej sa chcem poďakovať Metacentru za poskytnutie výpočtového zdroja v rámci projektu e-INFRA CZ (ID:90254), ktorý podporilo Ministerstvo školstva, mládeže a telovýchovy Českej republiky.

Obsah

Kapitola 1

Úvod

Žijeme v dobe najväčšieho technologického rozmachu. Za poslednú dekádu sme ako ľudstvo dosiahli neuveriteľného pokroku, najmä vo vývoji umelej inteligencie. Tá nám umožňuje simulovať funkcie ľudského mozgu. Pomocou umelej inteligencie vieme vykonať veľa užitočných vecí a môže mať pre ľudstvo nesmierne pozitívny dopad vo forme urýchlenia výskumov a zautomatizovania našich každodenných aktivít, tak taktiež sa dá aj zneužiť. Jednou z čoraz populárnejších technológií sa stáva deepfake. Dnes už sme ťažko schopný rozoznať či prejav politika propagujúci nejakú myšlienku alebo investičnú ponuku je reálny, alebo zmanipulovaný za použitia deepfake algoritmov. Pre túto skutočnosť je potreba vyvíjať prostriedky, ktoré sú schopné zaručiť odolnosť voči nim.

Cieľom práce je vytvoriť aplikáciu, ktorá bude schopná otestovať robustnosť rozpoznávacích algoritmov pre rozpoznanie tváre voči deepfakom a bude schopná určiť či sa jedná o deepfake alebo nie.

V prvej kapitole si popíšeme princíp neurónových sietí. Jedná sa o spoľahlivejšie riešenie ako za použitia algoritmov. Existuje mnoho typov neurónových sietí, my budeme predovšetkým pracovať s konvolučnými sieťami, ktoré na základe trénovania na rozsiahlych dátach dokážu klasifikovať vstupné snímky. Ich sila spočíva v tom, že dokážu nájsť aj malé rozdiely alebo súvislosti na dátach, taktiež dokážu riešiť nelineárne problémy. Medzi nevýhody patrí potreba silného výpočtového zdroja. Aby dávali rozumný výstup potrebujeme mať obšírny dataset, ktorý zahŕňa všetky prípady potrebné pre klasifikáciu. V nasledujúcej kapitole zhrnieme technológiu deepfake. Táto technológia využíva najmä neurónové siete. Používajú sa pri tvorbe deepfake, ale taktiež aj pri ich detekcií. Posledná kapitola v rámci teórie bude o rozpoznávacích algoritmov, kde si bližšie popíšeme metodiky a jednotlivé systémy rozpoznania tváre.

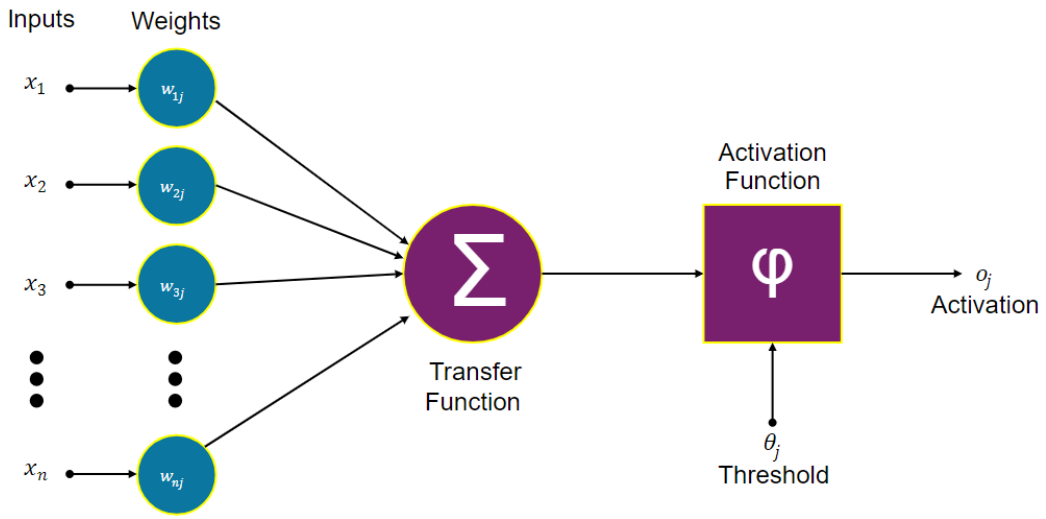
V rámci praktickej časti práce popisujeme návrh aplikácie a výsledky experimentov. Po kapitole Návrh a implementácia najskôr popisujeme tvorbu deepfake snímkov, pre náš dataset, ktorý ďalej využijeme v experimentálnej časti. Využili sme 4 deepfake programy, ktoré sú voľne dostupné na GitHube. Bližšie popisujeme prácu s konvolučnými neurónovými sieťami, kde sa najmä zameriavame na architektúru siamskej neurónovej siete. Nami vytvorené modely porovnávame s komerčným deepfake detektorom BioID a následne porovnávame s výsledkami rozpoznávacích algoritmov voči deepfake a určujeme dôveryhodnosť vstupných snímkov.

Kapitola 2

Neurónové siete

Neurónová sieť je výpočtový model používaný v oblasti umelej inteligencie a sú srdcom *deep learning* algoritmov. Vzorom a chovaním neurónov umiestnených v ľudskom mozgu sa podobajú. Umelé neurónové siete [87] sú zložené z vrstiev uzlov, obsahujúcich vstupnú vrstvu, jednu alebo viacej skrytých vrstiev a výstupnú vrstvu. Každý uzol alebo umelý neurón, je prepojený s ďalšími a má priradenú váhu a prah. Pokiaľ je výstup nejakého individuálneho uzla nad stanovenú hodnotu prahu, tento uzol je aktivovaný a posiela dáta do nasledujúcej vrstvy siete. V opačnom prípade dáta neprichádzajú do nasledujúcej vrstvy. Do aktivačnej funkcie sa teda posiela suma vstupných hodnôt vynásobených s váhami [87].

Neurónové siete spoliehajú na trénovanie dáta k učeniu a zlepšovaniu svojej presnosti v priebehu času. Akonáhle sú tieto učiace sa algoritmy doladené pre presnosť, stávajú sa mocnými nástrojmi v oblasti počítačovej vedy a umelá inteligencia, nám umožňuje klasifikovať a zhlukovať dáta s vysokou rýchlosťou. Najznámejšou neurónovou sieťou je vyhľadávací algoritmus Google [36].



Obr. 2.1: Umelý neurón [85].

Model na obrázku 2.1 je možné popísať rovnicou 2.1:

$$o_j = \varphi\left(\sum_{i=1}^{N}(w_i \cdot x_i) - \theta\right) \tag{2.1}$$

kde x_i sú vstupy neurónu, w_i sú váhy, θ je prah, φ je aktivačná funkcia neurónu a o_j je výstup neurónu.

Veľkosť váh w_i vyjadruje uloženie skúseností do spojení neurónov. Čím je väčšia hodnota, tým je daný vstup dôležitejší. ϑ označuje prahovú hodnotu aktivácie neurónu [87].

Neurónové siete môžu pomocou počítačov vykonávať inteligentné rozhodnutie za pomoci minimálneho zásahu človeka. To preto, lebo dokážu modelovať vzťahy medzi vstupnými a výstupnými dátami, ktoré sú nelineárne a zložité [4].

Medzi ich výhody patrí schopnosť paralelného spracovania dát. Siete dokážu vykonávať viacej úloh naraz. Nelinearita nám umožňuje modelovať reálne vzťahy medzi vstupom a výstupom. Sú odolné voči chybám, keďže porucha alebo strata jednej, či viacerých buniek nezastaví generovanie výstupu. Rozhoduje na základe pozorovania, vedia sa naučiť rozhodnúť na základe pozorovania vstupných trénovacích dát. Nakoniec majú schopnosť zovšeobecňovať dáta, vedia si odvodiť vzťahy na nevidených dátach. To znamená, že sieť dokáže predpovedať a správne klasifikovať výstupy [50].

K nevýhodám môžeme povedať, že je problematické im dôverovať, pretože nevieme dostatočne presne povedať ako sa sieť dostala k výsledku. Môžu byť aj nepresné ak sme nemali dostatočne natrénovaný model alebo obmedzenú veľkosť dataset. Nakoniec je ťažké zistiť ako kategorizujú dáta alebo vytvárajú predikcie, čiže javia sa nám ako Black box [50].

2.1 Typy neurónových sietí

V rámci existencie a vývoja neurónových sietí nám vznikli rôzne architektúry a konfigurácie. Ako to väčšinou v živote býva, každá z nich má svoje pozitívne a negatívne stránky a niektoré sú vhodnejšie na niektoré typy úloh. V nasledujúcom zozname si popíšeme niektoré známe typy neurónových sietí [72]:

- Feed-forward neurónová sieť – Jedná sa o najjednoduchšie neurónové siete. Dáta putujú iba jedným smerom. Jej výhody sú rýchlosť a krátka doba na trénovanie. Väčšina aplikácií v oblasti rozpoznania vizuálneho vstupu používa tento typ sietí.

- Kohonenova samo organizujúca neurónová sieť – Vektory náhodného vstupu sú vstupom do diskrétnej mapy skladajúcej sa z neurónov. Vektory sú tiež nazývané rozmery alebo roviny. Aplikácie zahŕňajú použitie k rozpoznaniu vzorov v dátach, ako je lekárska analýza.

- Rekurzívne neurónové siete – V tomto type sietí si skrytá vrstva ukladá svoj výstup k použitiu pre budúcu predpoveď. Výstup sa stáva súčasťou nového vstupu. Aplikácie zahŕňajú konverziu textu na reč.

- Konvolučné neurónové siete – Jedná sa o typ *Feed-forward* sietí. Tento typ sietí berie vstupné dávky po dávkach(batch). To umožňuje sieti pamätať si obraz po častiach. Aplikácie zahŕňajú spracovanie signálov a obrazov, napríklad rozpoznanie tváre.

V tejto práci budeme výhradne pracovať s konvolučnými neurónovými sieťami.

2.1.1 Aktivačné funkcie

Aktivačné funkcie sú neoddeliteľnou súčasťou neurónových sietí, ktoré im umožňujú naučiť sa zložité vzory v dátach. Transformujú vstupní signál uzlu v neurónovej sieti na výstupný signál, ktorý sa následne predáva do ďalšej vrstvy. Prinášajú nelinearitu, čo umožňuje

schopnosť sieťam naučiť sa zložité mapovanie medzi vstupmi a výstupmi. Správny výber je kľúčový pre trénovanie neurónových sietí, ktoré sa dobre generalizujú a poskytujú presné predikcie.

Najjednoduchším typom je lineárna aktivačná funkcia. Používajú sa vo výstupnej vrstve pre regresívne úlohy, neposkytujú nelinearitu. Jedná sa o najjednoduchšiu funkciu.

Na výstupe sa používajú dva typy buď funkcia *sigmoid* [3] alebo *softmax* [3]. *Sigmoida* vracia hodnoty medzi 0 a 1, užitočná pre binárnu klasifikáciu. *Softmax* je všeobecnejšia sigmoida, používa sa pre viac násobnú klasifikáciu, čiže nebinárne úlohy. Obidve funkcie trpia problémom miznúceho gradientu.

Ako poslednú si spomenieme ReLu funkciu, ktorá sa často používa v neurónových sieťach, konkrétne v skrytých vrstvách. Funkcia vie riešiť problém miznúceho gradientu. Vie prahovať vstupné hodnoty na 0, vracia 0 pre záporné hodnoty a samotný výstup pre kladné hodnoty [3].

2.1.2 Presnosť a strata

Presnosť (*accuracy*) a strata (*loss*) sú dôležité metriky pri strojovom učení.

Presnosť sa počíta ako počet správnych predikcií/počet všetkých predikcií, podrobnejšie v podkapitole 4.5.6. Presnosť je metóda na meranie výkonnosti klasifikačného modelu. Zvyčajne sa vyjadruje v percentách. Presnosť je počet predpovedí, kde sa predpokladaná hodnota rovná skutočnej hodnote. Je binárna pre konkrétnu vzorku. Presnosť sa zobrazuje v grafe a monitoruje sa počas tréningovej fázy [84].

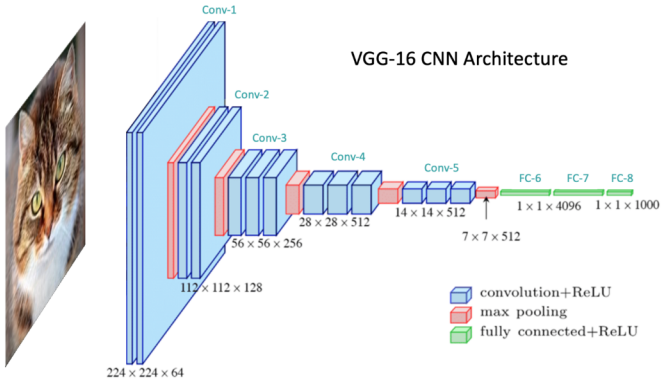
Strata alebo stratová funkcia berie do úvahy pravdepodobnosť alebo neistotu predpovede na základe toho, do akej miery sa predpoveď líši od skutočnej hodnoty. To nám poskytuje presnejší pohľad na výkonnosť modelu [84]. Existuje niekoľko typov stratových funkcií, ktoré sa používajú v závislosti od úlohy. *Cross-entropy loss* sa používa pri klasifikačných úlohách a hodnotí rozdiel medzi predikovanými a skutočnými kategóriami. *Triplet loss* sa používa pri úlohách rozpoznania tvárí. Snažia sa minimalizovať vzdialenosť medzi pozitívnym párom a maximalizovať medzi negatívnym. *Contrastive loss* používa sa pri párových úlohách, ako je rozpoznanie tváre, snaží sa priblížiť podobné páry a oddialiť nepodobné. Podrobnejšie tieto funkcie rozoberáme v podkapitole 2.3.3. Tieto stratové funkcie umožňujú lepšie prispôsobenie modelu na danú úlohu a poskytujú jasnejší obraz o jeho výkonnosti počas tréningu a testovania.

2.2 Konvolučné neurónové siete

Pred konvolučnými neurónovými [35] sieťami boli používané časovo náročné metódy extrakcie rysov k identifikácií objektov v obrazoch. Konvolučné neurónové siete teraz poskytujú jednoduchší prístup ku klasifikácií obrazov a úlohe rozpoznania objektov, využívajúcich princípy lineárnej algebry, konkrétne násobenie matíc, k identifikácií vzorov v obraze. Môžu byť výpočtovo náročné a vyžadovať grafické procesory k trénovaniu modelov.

Konvolučná vrstva je prvou vrstvou konvolučnej siete. Zatiaľ čo konvolučné vrstvy môžu byť nasledované ďalšími konvolučnými vrstvami alebo *pooling* vrstvami, plne prepojená vrstva je finálnou vrstvou. S každou vrstvou konvolučnej siete rastie zložitosť, identifikuje väčšiu časť obrazu [35].

Konvolučné neurónové siete sa odlišujú od ostatných neurónových sietí svojou nadriadenou výkonnosťou pri spracovaní vstupu obrazu alebo zvukových signálov. Majú tri hlavné typy vrstiev:



Obr. 2.2: Konvolučná sieť Vgg-16 [51].

- Konvolučná vrstva
- *Pooling* vrstva
- Plne prepojená vrstva

Na obrázku 2.2 môžeme vidieť ukážku konvolučnej neurónovej siete.

2.2.1 Konvolučná vrstva

Konvolučná vrstva je jadrom stavebného bloku konvolučnej neurónovej siete, a je to miesto, kde prebieha väčšina výpočtov, vo forme konvolúcie. Vyžaduje niekoľko komponent, a to vstupné dáta, filter a príznakovú mapu. Pre každé konvolučné operácie sa aplikuje konvolučná transformácia ReLU (Rectified Linear Unit) na príznakovú mapu, čo do modelu vnáša nelinearitu. Rovnica konvolúcie pre dvoj rozmerné obrázky [32]:

$$y[i,j] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h[m,n] \cdot x[i-m, j-n] \tag{2.2}$$

kde x predstavuje maticu vstupného obrazu, ktorá sa konvoluje s maticou jadra h, aby sa vytvorila nová matica y, ktorá predstavuje výstupný obraz. Indexy i a j patria obrazovým maticiam, zatiaľ čo indexy m a n patria jadru matice.

2.2.2 Pooling vrstva

Pooling vrstvy, sú známe taktiež ako redukcia dimenzie, prevádzajú redukciu dimenzionality, znižujú počet parametrov vstupov. Podobne ako u konvolučnej vrstve, operácie zlievania prechádzajú filtrom cez celý vstup, ale rozdielom je, že tento filter nemá žiadne váhy. Namiesto toho jadro aplikuje agregačné funkcie na hodnoty v reaktívnom poli, naplňuje výstupné pole. Existujú dva hlavné typy zlievania [35]:

- *Max pooling*: Keď filter prechádza vstupom, vyberá pixel s maximálnou hodnotou, ktorý zašle do výstupného poľa. Mimochodom, tento prístup je obvykle používaný častejšie než priemerné zlievanie.
- *Average pooling*: Keď filter prechádza vstupom, vypočíta priemernú hodnotu v reaktívnom poli, ktorú zašle do výstupného poľa.

Aj keď vo vrstve zlievania dochádza k strate mnoho informácií, tak má tiež niekoľko výhod pre konvolučné neurónové siete. Pomáha znižovať zložitosť, zlepšovať efektivitu a obmedzovať riziko nadmernej montáži, *overfitting* [35].

2.2.3 Plne prepojená vrstva

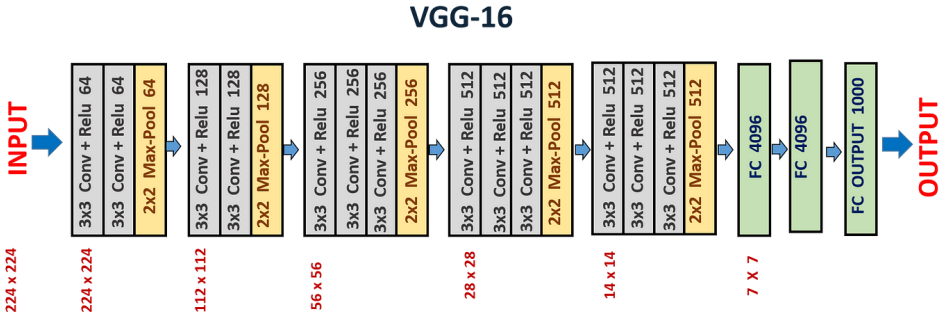
Názov plno prepojenej vrstvy dobre vystihuje jej povahu. Tak ako sme už spomenuli, hodnoty pixelov vstupného obrazu nie sú priamo prepojené s výstupnou vrstvou v čiastočne prepojených vrstvách. Viacmennej, v plno prepojenej vrstve je každý uzol vo výstupnej vrstve priamo pripojený s uzlom v prechádzajúcej vrstve. Používa sa na klasifikáciu. Zatiaľ čo konvolučné a *pooling* vrstvy obvykle používajú funkcie ReLu, plno prepojené vrstvy obvykle využívajú aktivačnú funkciu softmax [61] alebo sigmoid k vhodnej klasifikácií vstupov, produkovanými pravdepodobnosťou od 0 do 1 [35].

2.3 Architektúra neurónových sietí

V tejto kapitole sa zameriame na rôzne architektúry neurónových sietí, ktoré zohrávajú kľúčovú úlohu v oblastiach generovania, detekcie a extrakcie vlastností. GAN (*generative adversarial network*) siete slúžia pre generovanie realistických obrázkov, využívajú sa pri tvorbe deepfake. Siamské [27] sú špecifické tým, že obsahujú dve alebo viac identických sietí a určujú ich podobnosť, využitie pri detekcií deepfake. Pre extrakciu vlastností zo snímkov sú často využívané architektúry ako ResNet a Vgg. Nasledujúce podkapitoly sa podrobnejšie venujú jednotlivým spomenutým architektúram neurónových sietí.

2.3.1 Vgg-Net

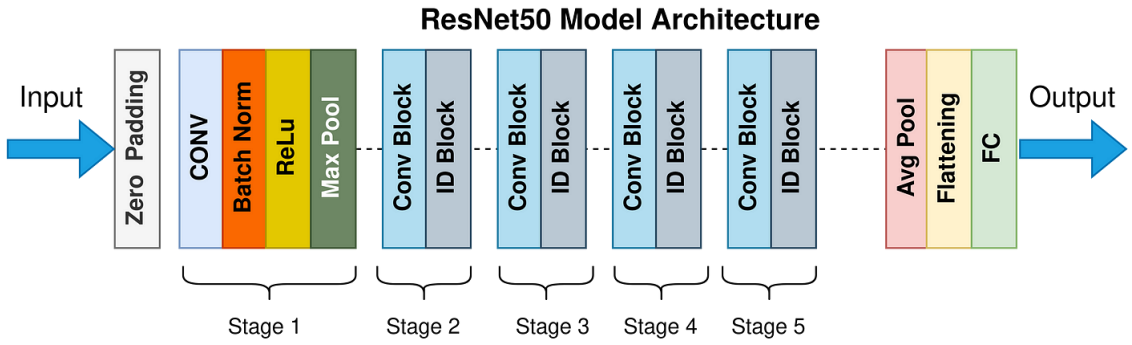
Skratka Vgg znamená Visual Geometry Group [7], jedná sa o architektúru konvolučnej neurónovej siete. Existujú dve verzie Vgg a to Vgg-16 a Vgg-19, kde číslovka 16 a 19 označujú počet konvolučných vrstiev. Na obrázku 2.3 môžeme vidieť architektúru siete Vgg16. Táto neurónová sieť sa používa pre identifikáciu objektov. Jedná sa o populárnu sieť, ktorá vyhrala prvé a druhé miesto v súťaži ImageNet Challenge 2014 [7, 71, 26]. Táto architektúra sa dá spoľahlivo použiť na rozpoznávanie tváre, ak je učená na dátovej sade zloženej z tvári ľudí. Nasledujúci obrázok 2.3 zobrazuje architektúru siete Vgg16.



Obr. 2.3: Architektúra siete Vgg-16 [7].

2.3.2 ResNet

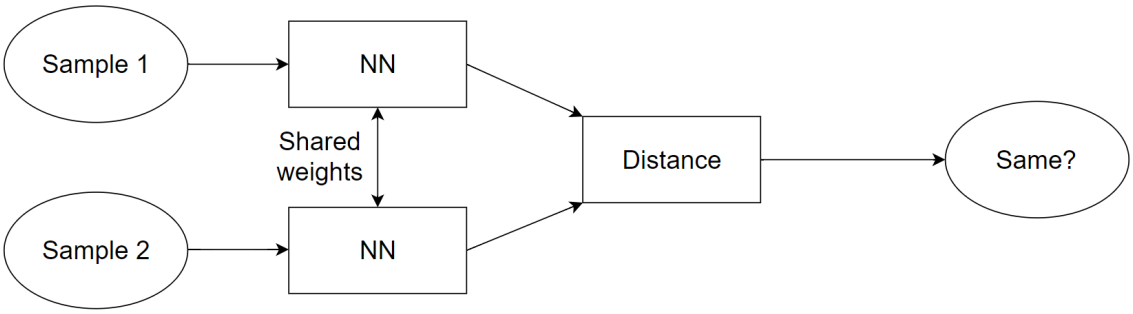
Residual Learning alebo skrátene ResNet architektúra je navrhnutá tak, aby pomohla modelom hlbokého učenia učiť sa rýchlo a efektívne, pomocou zvyškových blokov. Riešia problém degradácie, kedy hlbšia sieť začína fungovať horšie, a taktiež riešia problém miznúceho gradientu, ktorý bráni efektívnemu trénovaniu veľmi hlbokých sietí. Architektúra má rôzne hĺbky s verziami ako 50, 101 alebo 152. Táto sieť našla svoje uplatnenie v aplikáciách hlbokého učenia, najmä počítačového videnia [34]. Nasledujúci obrázok 2.4 zobrazuje architektúru siete ResNet50.



Obr. 2.4: Architektúra siete ResNet50 [55].

2.3.3 Siamská neurónová sieť

Siamská neurónová sieť [27] je trieda architektúr neurónových sietí, ktoré obsahujú dve alebo viacero identických sietí. Identické znamená, že majú rovnakú konfiguráciu s rovnakými parametrami a váhami. Aktualizácia parametrov počas trénovania je zrkadlená naprieč všetkými sieťami a používa sa k nájdeniu podobnosti medzi vstupmi [10]. Nasledujúci obrázok 2.5 znázorňuje štruktúru architektúru siamskej siete.



Obr. 2.5: Architektúra siamskej neurónovej siete. Obrázok vychádza z [80].

Siamská sieť dokáže za použitia zopár obrázkov dávať dobré predikcie. Schopnosť učiť sa z malého množstva dát urobilo siamské siete populárnymi. Táto výhoda je vykompenzovaná dlhším potrebným časom na tréning siete [10].

Siamské siete našli široké spektrum využitia v rôznych aplikáciách rozpoznávania obrazu vďaka ich schopnosti efektívne sa učiť reprezentáciu obrazu a vzájomného porovnávania.

V sfére výskumu rozpoznania ľudskej tváre sú používané k porovnávaniu dvoch obrazov tváre a určenia, či patria k tej istej osobe alebo nie. Môžu byť použité v sledovaní objektov vo video sekvenciách. Používajú sa k overovaniu podpisov alebo k overeniu autentickosti. Kedysi boli využívané aj v biometrických autentizačných systémov [27].

Trénovanie siamských sietí môže byť riadené (*supervised*) alebo bez dohľadu (*unsupervised*) [27]. Pri riadenom učení je sieť trénovaná na označených dátach. Pri trénovaní bez dozoru je sieť trénovaná na neoznačených dátach a musí sa naučiť generovať vlastné označenie na základe vstupných snímkov [27].

Pri trénovaný je vždy potrebné zvoliť správnu stratovú funkciu. Pri siamských sieťach sa najčastejšie používajú buď *contrastive loss* s *triplet loss*, každopádne existuje pár scenárov kde *binary cross-entropy* je použiteľná ako napríklad pri úlohách rozpoznania tváre. Stratová funkcia *binary cross-enthropy* má tvar [81]:

$$L_L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{2.3}$$

kde $\hat{y}_i$ je predpokladaná pravdepodobnosť pre i-tú vzorku, y_i je označenie (*label*) i-tej vzorky. N je celkový počet vzoriek.

Triplet loss [9] používa na vstup 3 obrázky vstupný, pozitívny a negatívny. Funkcia porovnáva vstup s pozitívnym a negatívnym obrázkom. Vzdialenosť medzi vstupom k pozitívnemu sa počas trénovania minimalizuje a vzdialenosť medzi vstupom k negatívnemu maximalizuje.

Contrastive loss [9]. je založená na vzdialenosti. Táto stratová funkcia sa snaží pre dva podobné body dávať na výstup nízku euklidovskú vzdialenosť a pre dva odlišné body veľkú euklidovskú vzdialenosť. Pri tom používa hodnotu *margin*, ktorá udáva minimálnu vzdialenosť, ktorou sa musia nepodobné body držať, takže trestá nepodobné vzorky za to, že sú bližšie než je zadaná hranica. Stratová funkcia *contrastive loss*, využitá v rámci tejto práci, má tvar [53]:

$$\mathcal{L} = (1-l)D^2 + l\{max(0, m-D)\}^2 \tag{2.4}$$

kde l je pravdivé binárne označenie snímky, podľa toho či sa jedná o deepfake ($l = 1$) alebo originál ($l = 0$), $margin > 0$ je hranica pre nepodobné páry a $D = ||f(Ref) - f(Cand)||_2$ je Euklidovská vzdialenosť medzi vektorom referenčného obrázka $f(Ref)$ a snímkom pre porovnanie s referenciou $f(Cand)$ vstupných snímkov modelu. Ref označuje referenčný snímok a $Cand$ kandidátny snímok, čo môže byť potenciálny deepfake. Nepodobné páry prispievajú k stratovej funkcii iba vtedy, ak je ich vzdialenosť v medziach hodnoty *margin*. Táto stratová funkcia podporuje zhodné páry, aby boli blízko seba vo funkčnom priestore, zatiaľčo odlišné páry sa od seba vzďaľujú [53].

2.3.4 GAN architektúra

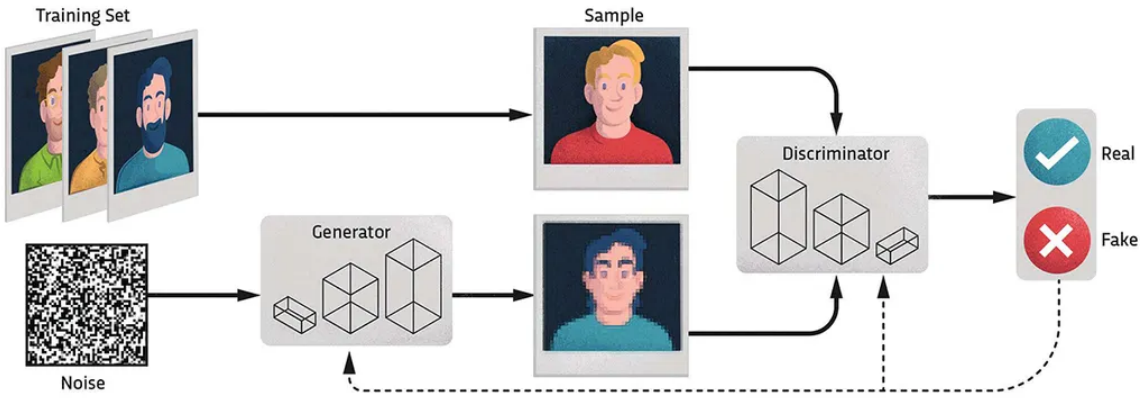
GAN architektúra zahŕňa dve neurónové siete: generátor a diskriminátor, ako je znázornené na Obrázku 2.6. Ak máme súbor skutočných obrázkov x s distribúciou $p_{data}(x)$, cieľom generátora G je vytvoriť obrázky $G(z)$ podobné skutočným obrázkom x s obrázkami z ako šumovým signálom s distribúciou $p_z(z)$. Cieľom diskriminátora G je správne klasifikovať obrázky generované G a skutočné obrázky x. Diskriminátor D je trénovaný na zlepšenie svojej klasifikačnej schopnosti, teda maximalizovať $D(x)$, čo predstavuje pravdepodobnosť, že x je skutočný obrázok, a že jeho výstupy budú klasifikované D ako syntetické obrázky,

teda minimalizovať $1 - D(G(z))$. Ide o minimaxovú hru medzi dvoma hráčmi D a G, ktorú možno opísať nasledujúcou hodnotovou funkciou 2.5:

$$\min_G \max_D V(D,G) = E_{x\ p_{data}(x)}[logD(x)] + E_{z\ p_z(z)}[log(1 - D(G(z)))] \tag{2.5}$$

kde G je generátor, D reprezentuje diskriminátor, $V(D,G)$ je minimax hra medzi generátorom a diskriminátorom, x reprezentuje reálny obrázok s distribúciou $p_{data}(x)$, z je zašumený vstup do generátora, $G(z)$ reprezentuje výstup generátora – tu sa jedná o nepravý snímok, $D(x)$ je výstup diskriminátora – čo obsahuje pravdepodobnosť pravosti vstupného obrázky, $D(G(z))$ reprezentuje výstup diskriminátora pri prijatí falošného obrázka z $G(z)$, $p_{data}(x)$ je pravdepodobnostná distribúcia reálneho snímku, $p_z(z)$ pravdepodobnostná distribúcia falošného snímku, $E_{x\ p_{data}(x)}[logD(x)]$ je priemerná logaritmická pravdepodobnosť D, keď je zadaný skutočný obrázok a nakoniec $E_{z\ p_z(z)}[log(1 - D(G(z)))]$ je priemerná logaritmická pravdepodobnosť D pri vstupe vygenerovaného obrázka, čiže falošného [23, 62].

Po dostatočnom tréningu obidve site zlepšujú svoje schopnosti, teda generátor G je schopný vytvárať obrázky, ktoré sú veľmi podobné skutočným obrázkom, zatiaľ čo diskriminátor D je schopný rozlišovať falošné obrázky od skutočných [57]. Nasledujúci obrázok 2.6 znázorňuje architektúru GAN sietí.



Obr. 2.6: Štruktúra GANu [56].

Kapitola 3

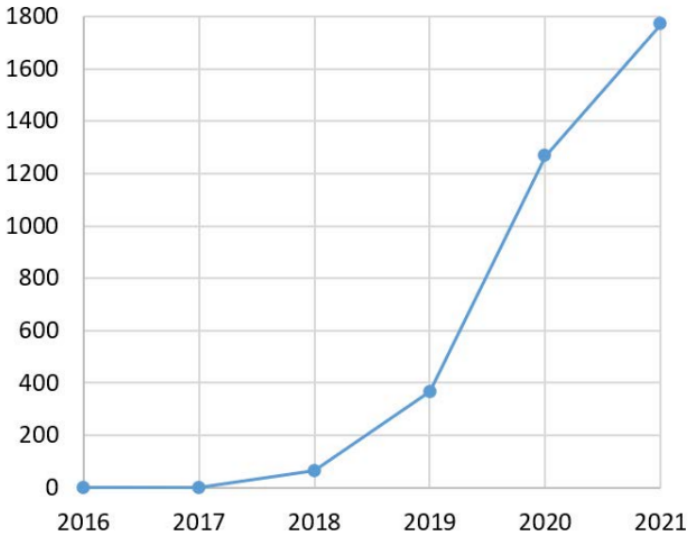
Deepfake tváre

Myšlienka manipulácie fotografií sa v ľudských mysliach začala tvoriť pri vzniku prvých fotografií, niekedy v 19. storočí. Tieto techniky predovšetkým radi využívajú totalitné režimy [86]. Ich prudký nárast však prišiel so vznikom videí a vyvrcholil pri pokrokoch umelej inteligencie. Podobne ako bolo zneužívané manipulovanie fotografií v histórii, tak skoro totožne dopadlo využívanie deepfake technológie. Deepfake môže byť výraznou hrozbou pre spoločnosť vo forme hoaxov v médiách, politických systémov, jednotlivcov, podnikateľov a podobne. Z týchto dôvodov je potreba ich detekcie a regulácie pri zneužívaní [2]. Väčšina deepfake algoritmov je založených buď na princípe konvolučných neurónových sietí alebo GAN sietí. Oba tieto prístupy majú kľúčovú úlohu pri tvorbe syntetických obrázkov, tieto obrázky sú zamerané na vizuálnu manipuláciu tváre osôb.

3.1 História

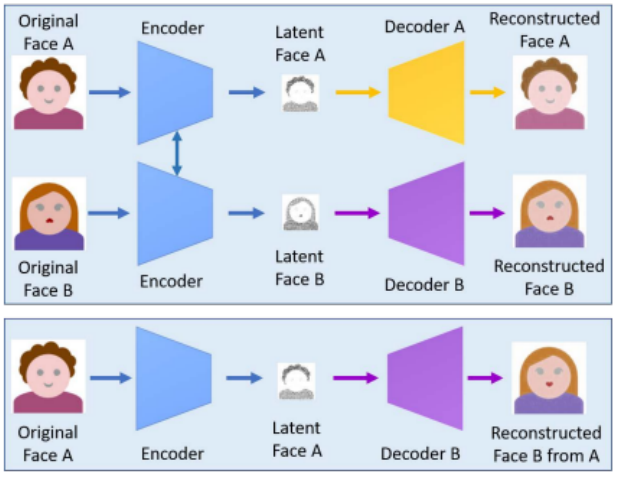
Deepfake označuje špecifický druh synteticky vytvorenej snímky alebo fotografie, ktorý zabezpečuje zmenu tváre osoby alebo osôb za akúkoľvek inú podľa preferencie. Termín deepfake vznikol v roku 2017 na Reddite po užívateľovi, ktorý tento Reddit spravoval [2]. Tento názov je spojenie slov deep learning a fake. Na začiatku boli pre tvorbu deepfake algoritmov použité neurónové siete *Autoencoders* [38]. *Autoencoder* je špeciálny typ neurónovej siete, ktorej cieľom je zhodovať sa so vstupom, ktorý bol poskytnutý. Pri deepfakoch sú trénované dva páry autoenkóderoch súčasne a ich parametri sú súčasne zdieľané. Táto stratégia umožňuje bežnému enkodéru nájsť a naučiť sa podobnosť medzi dvoma sadami tvárí, čo je relatívne jednoduché, pretože tváre obvykle majú podobné rysy, ako sú poloha očí, nosa a úst. Obrázok 3.2 zobrazuje proces vytvárania deepfake, pri ktorom je sada čŕt tváre A spojená s dekóderom B na rekonštrukciu tváre B z pôvodnej tváre A. Na obrázku 3.2 môžeme vidieť názornú ukážku použitia enkóderov a dekóderov pri autoenkóderov. Táto verzia sa neskôr vylepšila minimalizujúc množstvo strát. Toto vylepšenie bolo založené na práci Ian Goodfellowa GAN z roku 2014 [29]. Tento pokrok prispel k popularite deepfake technológie ako môžeme vidieť na obrázku 3.1.

V rokoch 2015 a 2016 sa technológia tvorby deepfake zdokonaľovala a jej prvotný výbuch nastal v roku 2017, keď sa spoločnosti Nvidia podarilo dosiahnuť významný pokrok v kvalite výstupu GANu [67]. V istom roku sa do popredia dostáva už spomínaní reddit server deepfakes, ktorý spopularizoval deepfake technológiu. Tento nárast popularity bol predovšetkým spôsobený propagovaním obsahu pre dospelých, kde tváre protagonistov v jednotlivých scénkach boli nahradené za tváre známych celebrít, predovšetkým ženského



Obr. 3.1: Graf popularity deepfake v priebehu rokov [57].

pohlavia. Tento reddit bol zrušený v roku 2018 z dôvodu vysokého výskytu nemravného obsahu. Od vtedy prebiehali ďalšie vylepšenia, nové regulácie na sociálnych sieťach, ktoré sa rozhodli, že nechcú na svojich stránkach žiadne deepfake snímky [67]. Od tejto chvíle vzniklo veľa *open-source* projektov, ktoré naďalej rozvíjajú túto technológiu.



Obr. 3.2: Zobrazenie učenia autoenkóderu pri výmene tváre [57].

3.2 Problematika deepfake

Deepfake technológia je v dnešnej dobe veľmi ľahko dostupná, čo je jeden z najväčších problémov. Tento fakt podporuje aj skutočnosť rýchlo rastúceho počtu podvodných článkov a videí zachytávajúcich ľudí čo sú pristihnutí pri činnostiach, ktoré nikdy nevykonávali. Ďalší problém vyplýva z ich sofistikovanosti a reálnosti. Už dávno sme prekonali dobu kedy tieto pokusy o podvody boli na prvý okamih jasné, dnes je situácia o dosť horšia a algoritmy

deepfake technológie dokážu vytvoriť snímky, ktoré nie sú ľahko ľudským okom odlíšiteľné od reálnych. Nasledujúce podkapitoly sa budú venovať problematike pri zmene tváre ľudí a ich hrozbou pre známych osobností ale aj menej známych a nakoniec popíšeme potrebu ich detekcie a či vlastne má význam. Na nasledujúcom obrázku 3.3 môžeme vidieť príklad tvorby deepfake.

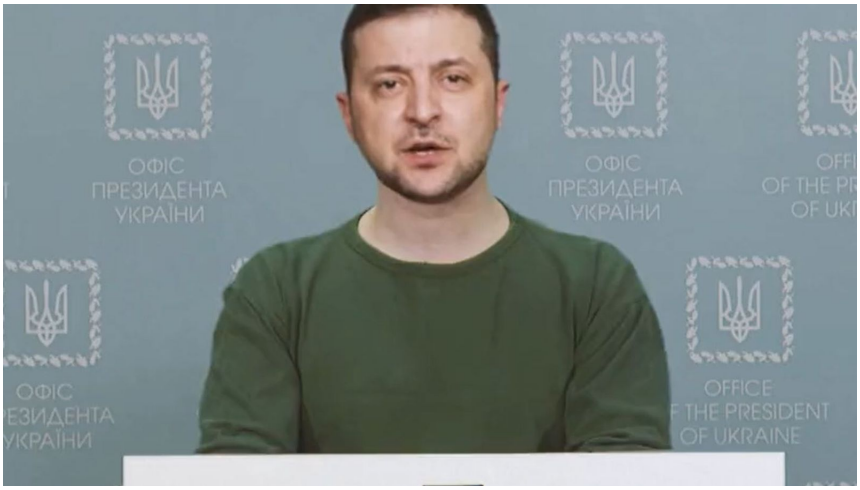
Obr. 3.3: Príklad deepfake [6].

3.3 Zneužívanie deepfake technológie

Ako už bolo spomínané deepfake je veľmi populárny na stránkach pre dospelých. Väčšina týchto snímkov a na základe článku [73] až 96 % je vytvorených bez súhlasu protagonistov. Tento žáner reprezentuje videá kde tvár jednej osoby, obvykle ženskej celebrity, je umiestnený na telo druhej. Na podobnom princípe môžu podvodníci využiť vašu fotku a vytvoriť videá alebo obrázky, za cieľom vymáhať z vás peniaze [66]. Na druhú stránku existujú aj humorné účty na sociálnych sieťach, ktoré za súhlasu celebrít, vytvárajú obsah kde si svoju tvár zmenia na celebritu, a tak zvyšujú popularitu tejto osobnosti [45].

Ďalšie odvetvie zneužívania deepfake technológie je v oblasti politiky. Väčšinou sa jedná o sfalšovaný prejav politika tak, aby odradili podporovateľov danej osoby. Najlepšia ochrana proti týmto podvodom je prezerať oficiálne profily politikov na sociálnych sieťach, ktoré sú verifikované, či naozaj niečo také vykonal. Takéto výtvory majú potenciál ohroziť demokraciu tak ako sa spomína v článku [59] a preto je potrebná rýchla detekcia na limitovanie škôd. Jeden z najväčších pokusov o politickú manipuláciu bol pokus Ruskej federácie o vytvorenie deepfake snímku, kde bol vyobrazený Ukrajinský prezident Zelenskyy [12]. Nejednalo sa o kvalitný snímok a nebol zásadný problém ho detekovať spravodajskými spoločnosťami [73]. Výsledný snímok môžeme vidieť na obrázku 3.4.

Práve internet a mediálne portály sa snažia byť využité na rozposielanie týchto falošných správ. Novinári už od pradávna bojujú s hoaxami a v priebehu rokov vynašli taktiky ako zabrániť šíreniu falošných správ, avšak tieto praktiky sú prikrátke pri deepfakoch. Obrana voči deepfakom môže stáť nemalý kapitál, preto novinári musia ručne overovať či je daná správa reálna alebo vymyslená. Z tohto dôvodu spoločnosti ako Google alebo Meta vytvorili vlastné projekty, s ktorými chcú limitovať počet a rýchlo detekovať deepfake. Do tohto



Obr. 3.4: Zelenskyy deepfake [12].

rozsiahleho výskumu je aj zapojených mnoho univerzít, ktoré sa snažia hľadať lepšie a účinnejšie riešenia detekcie deepfake [5].

3.4 Pozitívne využitie

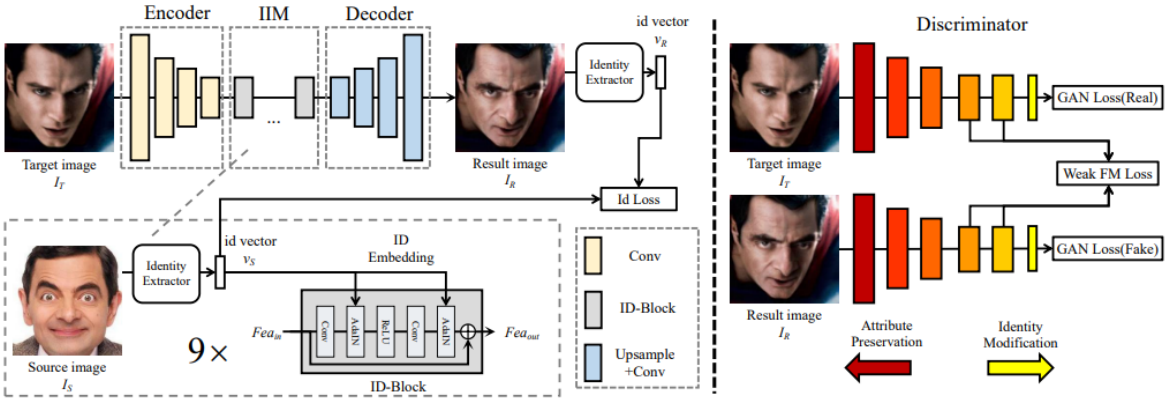
Doteraz boli spomenuté iba negatívne stránky, na ktoré sa žiaľ deepfake technológia prevažne využíva, avšak deepfake má možnosť pozitívne ovplyvniť budúcnosť pri mnohých aplikáciách. V súčasnosti bol deepfake využitý napríklad Davidom Beckhamom, kedy jeho hlas bol menený do rôznych jazykov, aby upozornil čo najväčší dosah ľudí o malárií [60]. Predpokladá sa, že deepfake technológia by mohla zefektívniť a z atraktívni vzdelanie. Vďaka deepfakom namiesto online kurzov, kde iba niekoho počúvame, by mohli vzniknúť interaktívne kurzy kde by sme mali učiteľa, s ktorým by sme mohli interagovať a pýtať sa ho otázky na objasnenie vecí, ktoré nechápeme. Na stávajúcich výukách v školách by napríklad na hodinách dejepisu mohli byť oživené postavy s dejín, s ktorými by mohli študenti komunikovať a názorne ich vidieť pre lepšiu predstavivosť [58]. Na školách medicíny by si žiaci mohli lepšie odsimulovať operovanie a komunikáciu s pacientov. Môžeme vytvárať rýchlejšie a lacnejšie umelecké obrazy, ktoré by boli schopné zrýchliť vývoj softwaru alebo výrobu filmov. Ďalšie odvetvie kde by deepfake vedel výrazne prispieť je virtuálna realita, kde by pomohol naše pomerne realisticky tváre previesť do virtuálnej reality [43].

3.5 Deepfake algoritmy

V nasledujúcej sekcií si bližšie popíšeme dva použité deepfake programy, ktoré sme v práci využili. Tieto dva boli vytvorené pomocou skupiny výskumníkov, čiže sú k nim dostupné aj vedecké články. Ostatné použité programy, boli *open-source* projekty, ktoré neobsahujú žiadne vedecké články.

3.5.1 SimSwap

SimSwap [16] je deepfake program, ktorým cieľom je preniesť identitu zdrojovej tváre na cieľovú tvár pri zachovaní atribútov cieľa, ako je výraz a osvetlenie.



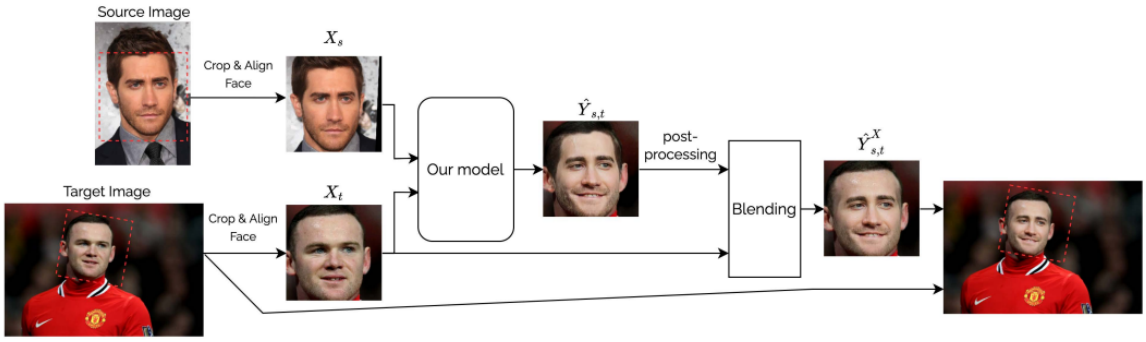
Obr. 3.5: SimSwap pipeline [16].

Implementácie je vytvorená za použitia technológie GAN, konkrétne enkóderu a dekóderu. Medzi nimi je pridaný *Injection Module*(IIM) [16], ktorý zabezpečuje prenos tváre na cieľovú osobu. Enkóder extrahuje tvár osoby F_{ea_T} z cieľového obrázka I_T, injekčný model vyextrahuje tvár požadovanej osoby z I_S do F_{ea_T} a dekóder sa postará o vytvorenie výsledného snímku. Ako stratovú funkciu využívajú *Weak Feature Matching Loss* [16], aby sa implicitne zachovali atribúty cieľa. Architektúra je navrhnutá tak, aby sa dala prispôsobiť ľubovoľným identitám. Model bol natrénovaný na datasete VggFace2[1], ktorý obsahuje širokú škálu tvári v rôznych pozíciách a vekových kategóriách. Využívali snímky ktoré boli väčšie ako 250 × 250 pixelov boli zarovnané a orezané. Autori svoj model popisujú ako revolučný a jeden z najlepších tej doby, ale už nie je najnovší a dnes toto tvrdenie už neplatí [16].

3.5.2 Ghost

Ghost [30] je deepfake program na jednorazové zamieňanie tvárí v obrázkoch a videách, ktorého cieľom je zlepšiť kvalitu a výkon voči moderným architektúram SoTA [76]. Ghost stavia na architektúre *FaceShifter AEI-Net* [48] s vylepšeniami, ako je stratová funkcia pre oči, algoritmus vyhladzovania tvárovej masky, stabilizačná technika a následné spracovanie v vysokom rozlíšený. Stratová funkcia očí zlepšuje výstup zámeny tváre a vytvára realistickejší výstup. Ďalej sa zameriavajú na straty rekonštrukcie tváre, atribútovej a nepriaznivej straty. Model bol natrénovaný taktiež za pomoci datasetu VggFace2, pričom obrázky boli orezané a zarovnané na rozlíšenie 256 × 256 pixelov. V závere autori prezentujú výsledky, ktorými vykazujú lepšie výkony ako spomínané SoTA architektúry [30].

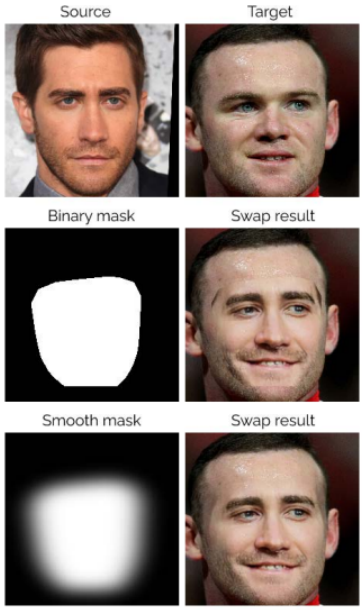
[1] https://paperswithcode.com/dataset/vggface2-1



Obr. 3.6: Ghost pipeline [30].

Ghost pracuje s orezanými tvárami, preto najskôr pred výmenou tváre musíme orezať zdrojový a cieľový snímok. Po prevedený výmeny tváre sa výsledok vloží do pôvodného cieľového obrázka. Hlavný problém nastáva pri zachovaný atribútov z X_t na $\hat{Y}_{s,t}$, pretože nie sú úplne rovnaké. Preto ak by sme vložili $\hat{Y}_{s,t}$ priamo späť do cieľového obrázka, boli by viditelné okraje ako môžeme vidieť na obrázku 3.7. Riešeným je využiť vyhladzovaciu masku. Všeobecný postup riešenia Ghost zobrazený na obrázku 3.6 [30]:

- Detekcia a orezanie tvárí zo zdrojového a cieľového obrázku pomocou detektora tváre. Nech X_s a X_t sú orezané obrázky.
- Aplikácia Ghost modelu na X_s a X_t na získanie výsledku $\hat{Y}_{s,t}$.
- Zmiešanie $\hat{Y}_{s,t}$ a X_t na získanie finálnej tváre $\hat{Y}^X_{s,t}$
- Vloženie výsledku $\hat{Y}^X_{s,t}$ späť do cieľového snímku.



Obr. 3.7: Ghost maska [30].

3.6 Princíp detekcie deepfake

Detekcia deepfake je dôležitý výskum, ktorý pomáha nájsť a zabrániť škodám, ktoré môže vzniknúť z falošných videí a obrázkov. Pri detekovaní deepfake sa experti snažia nájsť stopy, chyby a rozdiely vo videu, ktoré by naznačili upravený súbor. Hľadajú sa divné vzory, rozdiely v osvetlení alebo čokoľvek neprirodzené. Naviac za pomoci rôznych programov môžeme pozorovať rozdiely v žmurkaní alebo pohybu tváre osôb. Ďalší spôsob je zaviesť metódy hlbokého učenia, ktoré dokážu zachytiť drobné detaily ľahko nepovšimnuté ľudskými zmyslami. V nasledujúcej časti si bližšie priblížime techniky odhaľovania deepfake [15].

3.6.1 Analýza tváre

Analýza tváre detekuje deepfake prostredníctvom skúmania rysov tváre a výrazov, identifikujúce nekonzistenciu a nepravidelnosť. Tento proces využíva počítačového videnia a strojového učenia k rozlíšeniu medzi skutočným a zmanipulovanými tvarmi. Znaky tváre ako oči, nos, ústa a obočie sú veľmi dôležité pre detekciu deepfake pri analýze tváre. Algoritmy detekcie tieto znaky tváre analyzujú, aby našli rozdiel naznačujúci zmanipulovaný snímok, deepfake snímky často obsahujú v oblasti tváre rôzne anomálie a nedokonalosti. Techniky detekujú príznaky ako rozmazané hrany, spomínané osvetlenie a nesúlad v textúrach tváre. Strojové učenie zlepšilo analýzu tváre pre detekciu, že analyzuje artefakty a rozlišuje medzi autentickou a zmanipulovanou tvárou. Hlboké učenie môže detekovať manipulácia prostredníctvom analýzy jemných vizuálnych signálov v rozsiahlych dátových sadách s veľkou presnosťou [15].

3.6.2 Forenzná analýza

Forenzná analýza detekuje deepfake skúmaním digitálnych dôkazov a identifikáciu manipulácie. Táto metóda odhaľuje skryté stopy zanechané behom vytvárania alebo úprav deepfake. Forenzná analýza je zásadná pre rozlíšenie autentickosti a detekcii deepfake. Forenzná analýza detekuje obsah deepfake a skúma metadáta. Metadáta ukazujú históriu súboru. Analýza metadát rozlišuje autentické od zmanipulovaných, zatiaľ čo forenzná analýza identifikuje nezrovnalosti v obsahu deepfake. Analýza nezrovnalostí v deepfakoch zahŕňa skúmanie zarovnania, osvetlenia, tieňov a odrazov k detekcii zmanipulovaných rôznych tvárí na telách [15].

3.6.3 Analýza synchronizácie pier

Analýza synchronizácie pier detekuje deepfake, ktoré manipulujú alebo syntetizujú reč. Prostredníctvom skúmania synchronizácie pohybu pier a zvukov môžu vedci identifikovať nesúlad naznačujúci deepfake. Analýza pier je zásadná pre odhalenie deepfake. Jedná z variant je sledovať a študovať pohyb pier vo videu. Počítačové videnie detekuje a sleduje polohu, tvár a pohyb pier. Potom sa porovná s audiom, aby sa vyhodnotila synchronizácia. Analýza skúma vizuálne signály prirodzenej reči. Analýza synchronizácie pier kombinovaná s forenznou a tvárovou alebo audio analýzou zlepšuje celkovú pravdepodobnosť detekovať deepfake [15].

3.6.4 Analýza metadát

Analýza metadát je kľúčová pre identifikáciu deepfake prostredníctvom posudzovania metadát mediálnych súborov. Metadáta poskytujú podrobnosti o pôvode, úpravách a histórii súboru, ponúkajú užitočné informácie o autentickosti a možnej manipulácií. Analýza metadát môže odhaliť deepfake skúmaním časovej značky. Časové značky ukazujú aktivitu súboru. Nekonzistentné časové značky zvyšujú podozrenie na možný deepfake. Dôležité je tiež skúmať informácie o zariadeniach v metadátach. Digitálne zariadenia ukladajú metadáta, ktoré môžu overiť autentickosť. Tvorcovia deepfake môžu naraziť na problémy s replikáciou presných metadát fotoaparátov, čo vedie k nezrovnalostiam. Analýza metadát taktiež môže zahŕňať skúmanie digitálnych podpisov a vodoznakov. Skúmaním vlastníctva deepfake súboru môžeme ukázať či sa jedná o legitímny obsah. Ďalšou technikou detekcie deepfake je použitie rekurzívnych neurónových sietí navrhnutých k analýze časových informáciám vo videách. To napomáha identifikovať manipulované znaky, ako sú žmurkanie alebo problémy so synchronizáciou pier. GANy sú taktiež používané v deepfake detekcií. Sú trénované, aby sme dostávali realistický výstup pomocou generátoru, pričom trénujeme aj diskriminátor, ktorý umožňuje detekovať deepfake [15].

3.7 Detekcia deepfake

K využitiu strojového učenia pri detekcií deepfake treba kvalitné rozsiahle dátové sady. Dátové sady obsahujú autentické, tak aj deepfake súbory, poctivo vybrané, aby poskytovali rôznorodé scenáre pre efektívnu detekciu manipulovaného obsahu. Konvolučné neurónové siete sú populárnou metódou pre analýzu vizuálnych dát, z pravidla pri rozpoznávaný obrazových dát. Ich trénovanie na dátových sadách deepfake im umožňuje rozoznať vzory, ktoré odlišujú skutočný obsah od zmanipulovaného [15]. V nasledujúcej časti si popíšeme 3 algoritmy na detekciu deepfake. V nasledujúcej časti si popíšeme zopár existujúcich riešení detekcie deepfake.

3.7.1 Detekcia za použitia konvolučných sietí

V tomto článku autori riešia všeobecne problém a rozvoj manipulácie tváre za posledné roky. Článok sa zaoberá detekciou manipulácie tváre vo videách za použitia state-of-art metód. Ponúkajú riešenie problematiky pomocou konvolučných neurónových sietí, konkrétne prezentujú využitie siamskej siete a trénovanie za použitia *end-to-end* prístupu. Do trénovania pomocou *end-to-end* prístupu posielajú na vstup tvár a sieť pošle výsledok na výstup. Ako stratovú funkciu využívajú *LogLoss* funkciu, inak prezívanú aj *binary cross-entropy*, v tvare [11]:

$$L_L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(S(\hat{y}_i)) + (1 - y_i)\log(1 - S(\hat{y}_i))] \tag{3.1}$$

kde $\hat{y}_i$ predstavuje výstup i-tej tváre, y_i patrí 0,1 štítku (*label*) tváre. 0 označuje reálne tváre a 1 označuje zmanipulované snímky. N je celkový počet tvárí použitých na trénovanie a $S(x)$ je Sigmoidná funkcia.

Pri trénovaný siamskej siete bola využitá stratová funkcia *triplet loss* v tvare [11]:

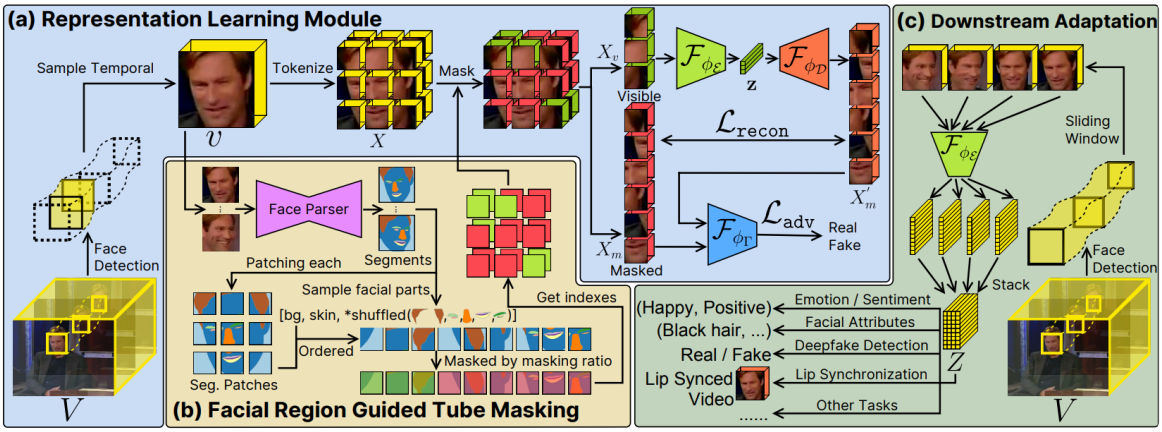
$$L_T = \max(0, \mu + \delta_+ - \delta_-) \tag{3.2}$$

kde $\delta_+ = |f(I_a) - f(I_p)|_2$, $\delta_- = |f(I_a) - f(I_n)|_2$ a μ je striktne pozitívny okraj. V tomto prípade I_a je kotva (*anchor*), reálny snímok, I_p je pozitívny snímok patriaci rovnakej triede ako I_a, ďalší reálny a I_n je negatívny snímok patriaci inej triede ako I_a, zmanipulovaný snímok [11].

V rámci práce využívajú viacero architektúr neurónových sietí ako napríklad EfficientNetB4 alebo XceptionNet. Prezentujú výsledky na dvoch verejne dostupných datasetoch s viac ako 119 000 videami. Podrobnejšie informácie sú vo vedeckom článku [11].

3.7.2 Detekcia za použitia autoenkóderu

V tomto článku autori predstavujú riešenie detekcie zmanipulovaných obrázkov za použitia autoenkóderu s názvom MARLIN. MARLIN sa zameriava na rôzne úlohy analýzy tváre, ako sú rozpoznávanie vlastností tváre, rozpoznávanie výrazov tváre, detekcia deepfake a synchronizácia pier. Na nasledujúcom obrázku 3.8 si bližšie popíšeme jednotlivé fázy MARLINu:



Obr. 3.8: Architektonický prehľad MARLIN [13].

MARLIN pozostáva hlavne z modulu *Representation Learning Module* (a), *Facial Region guided Tube Masking* (b) a Downstream Adaptation (c). (a) Reprezentačný modul: MARLIN sa učí tvár reprezentovať z neoznačených dát. Jedná sa o *self-supervised* učenie, zvýraznené modrou farbou. (b) Vedenie podľa oblasti tváre: Pomocou maskovacej trubice vedenej v oblasti tváre (zvýraznené žltou) získava MARLIN dočasné časopriestorové poznatky, ktoré následne využíva na zvýšenie výkonu. Stratégia maskovania trubice vnáša do procesu znalosti domény. (c) Prispôsobenie prúdu: Na prispôsobenie prúdu sa pri špecifických úlohách tváre využíva lineárne snímanie (LP) a jemné ladenie (FT), ktoré sa starajú o robustnosť, generalizáciu a o presnosť načítaného prvku, zvýraznenie zelenou farbou.

Použité dáta pre trénovanie pochádzajú z verejne dostupných videí na YouTube a datasetov ako LFW, CelebA, Vgg-FACE a ďalších. Pre vyhodnotenie výsledkov pri detekcií deepfaku využili dataset FaceForensic++. Podrobnejšie informácie sú vo vedeckom článku [13].

Kapitola 4

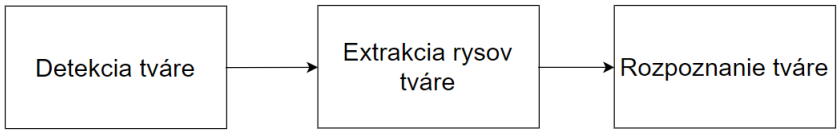
Algoritmy pre rozpoznanie tváre

Oblasť skúmania tváre je jednou z najrozšírenejších oblastí v počítačovom videní. Rozpoznanie tváre je spôsob identifikácie alebo potvrdenia identity jednotlivca pomocou jeho tváre [41]. Systémy rozpoznania tváre slúžia k identifikácií ľudí na fotografiách, videách alebo v reálnom čase. Všetky tieto funkcionality sú vytvorené pomocou algoritmov na rozpoznanie tváre, ktoré sú vyvíjané od roku 1960 a od tej doby ich vzniklo mnoho rôznych variant [52].

V tejto kapitole priblížime pozadie za algoritmami a metódami pre rozpoznanie tváre ich architektúru a problematiku, ktorá musí byť prekonaná pri detekciách tváre osoby zo snímky.

4.1 Architektúra systému

Vstupom do systému rozpoznávania je vždy fotografia alebo video. Výstupom je identita, vektor alebo boolovská hodnota áno/nie, závisí podľa danej úlohy. Existujú prístupy, ktoré definujú systém rozpoznania ako trojstupňový proces. Z tohoto hľadiska by bolo možné vy-



Obr. 4.1: Obecný systém rozpoznania tváre [52].

konať fázu detekcie tváre a extrakcie rysov súčasne. Detekcia tváre je definovaná ako proces extrakcie tvár zo scén. Systém tak pozitívne identifikuje určitú oblasť obrazu ako tvár. Tento prístup má mnoho aplikácií, ako je sledovanie tváre, odhad polohy alebo kompresia.

Existuje niekoľko kategórií pre algoritmy rozpoznania tváre. Prvý je založený na geometrickom prístupe. Táto kategória zahŕňa algoritmy, ktoré analyzujú rysy tváre a ich geometrické vzťahy. Môžu extrahovať parametre ako vzdialenosť medzi očami, nosom a ústami alebo uhly medzi nimi. Ďalšia kategória je na základe holistických modelov, kde sa na tvár dívame ako na celok. Namiesto toho, aby sa zameriavali na jednotlivé rysy, tieto algoritmy skúmajú celkovú štruktúru tváre. V tomto prípade sú používane konvolučné neurónové siete, aby sa naučili reprezentáciu tváre [24, 63].

Ďalším krokom môže byť extrakcia rysov, kde získavame relevantné tvarové rysy z dát. Tieto rysy môžu obsahovať relevantné údaje o tvári ako časti tváre, uhly alebo rozmery ako napríklad rozostup očí, najčastejšie sa snaží ako prvé vyhľadať ľudské oči, pretože sa jedná o jeden z najjednoduchších rysov pri detekcií. Potom sa snaží detekovať charakteristické body tváre, ako sú obočie, ústa, nos [52, 8].

Najčastejšie používaný prístup namiesto extrakcií rysov je *Face embedding*. Táto metodika reprezentuje tvár ako číselné vektory, zachytávajúce detaily tváre. *Face embedding* je numerická reprezentácia tváre v nízkorozmernom priestore, ktorá umožňuje porovnávať a určovať podobnosť medzi tvárami na základe ich vzdialeností v tomto priestore. [79].

4.2 Detekcia tváre

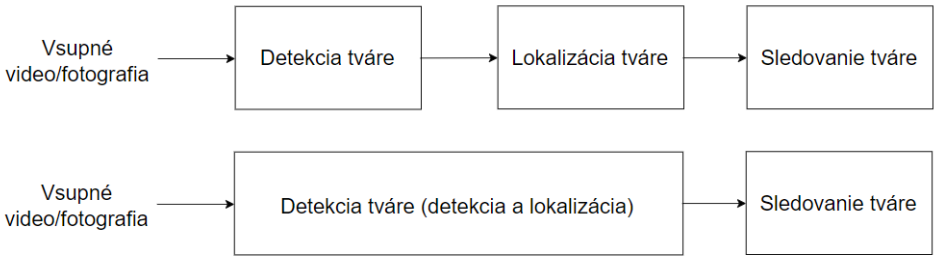
V niektorých prípadoch sa dá detekcia tváre preskočiť. Príkladom môže byť databáza kriminálnikov, kde orgány vyšetrovania ukladajú tváre ľudí s kriminálnym záznamom. Avšak bežné vstupné obrazy systémov počítačového videnia nie sú tak vhodné. Môžu obsahovať mnoho položiek alebo tvárí. V týchto prípadoch je detekcia tváre povinná. Je tiež nevyhnutná, ak chceme vyvinúť automatizovaný systém sledovania tváre. Preto je rozumné predpokladať, že detekcia tváre je súčasťou širšieho problému rozpoznania tváre [52].

Detekcia tváre je ovplyvňovaná niekoľkými výzvami ako napríklad [52, 2]:

- Starnutie – Tvár sa mení s časom, čo môže mať dopad na algoritmy pre rozpoznanie tváre. Využívajú sa spojené auto-enkodéry pre starnutie a odstarnutie a skúma ich vplyv na rôzne vekové skupiny.
- Termálne obrazy – Pre rozpoznanie tváre z termálnych obrazov je hlavné nájsť vhodnú techniku pre extrakciu rysov z tváre. Vo výskumu sa využívajú Gaborove systémy nízkeho rozlíšenia.
- Zakrytie čŕt – Prítomnosť prvkov ako brady, okuliare alebo klobúk zavádza vysokú variabilitu. Tváre môžu byť tiež čiastočne zakryté objektmi alebo inými tvárami.
- Variácia postoja – Ideálny scenár pre detekciu tváre by bol taký, v ktorom by boli zahrnuté len čelné obrazy. Ale, ako je uvedené, to je všeobecne veľmi nepravdepodobné v nekontrolovaných podmienkach. Navyše výkon algoritmov detekcie tváre dramaticky klesá pri veľkých variáciách postoja. Je to závažný výskumný problém. Variácia postoja môže nastať v dôsledku pohybov subjektu alebo uhlu kamery.
- Výraz tváre – Črty tváre sa tiež veľmi menia kvôli rôznym výrazom tváre.
- Podmienky snímania – Rôzne kamery a podmienky okolia môžu ovplyvniť kvalitu obrazu, ovplyvňujúc vzhľad tváre.

4.3 Architektúra štruktúry systému rozpoznania tváre

Detekcia tváre je koncept, ktorý zahŕňa mnoho podproblémov. Niektoré systémy detekujú a lokalizujú súčasne, iné najskôr vykonajú rutinnú detekciu a potom, ak je výsledok pozitívny, sa pokúsia lokalizovať tvár. Potom môže byť potrebných niekoľko sledovacích algoritmov - ukážka obrázka 4.2. Algoritmy detekcie tváre zvyčajne zdieľajú spoločné kroky. Po prvé, sa vykoná redukcia dimenzie dát, aby sa dosiahla prijateľná doba odpovede. Niektoré predspracovania môžu byť vykonané aj na prispôsobenie vstupného obrazu predpokladom



Obr. 4.2: Architektúra detekcie systémov rozpoznania tváre [52].

algoritmu. Potom niektoré algoritmy analyzujú obraz taký, aký je, a iné sa snažia extrahovať určité relevantné oblasti tváre. Ďalšia fáza zvyčajne zahŕňa extrakciu tvárových čŕt alebo meraní. Tieto budú následne vážené, vyhodnotené alebo porovnané, aby sa rozhodlo, či je prítomná tvár a kde sa nachádza. Nakoniec niektoré algoritmy majú rutinu učenia a zahŕňajú nové údaje do svojich modelov. Detekcia tváre je teda problém s dvomi triedami, kde musíme rozhodnúť, či je na obrázku tvár alebo nie. Sledovanie tváre sa rieši najmä vtedy, keď na vstup príde video. V tejto práci budeme pracovať iba so snímkami tváre [52].

4.4 Metódy detekcie tváre

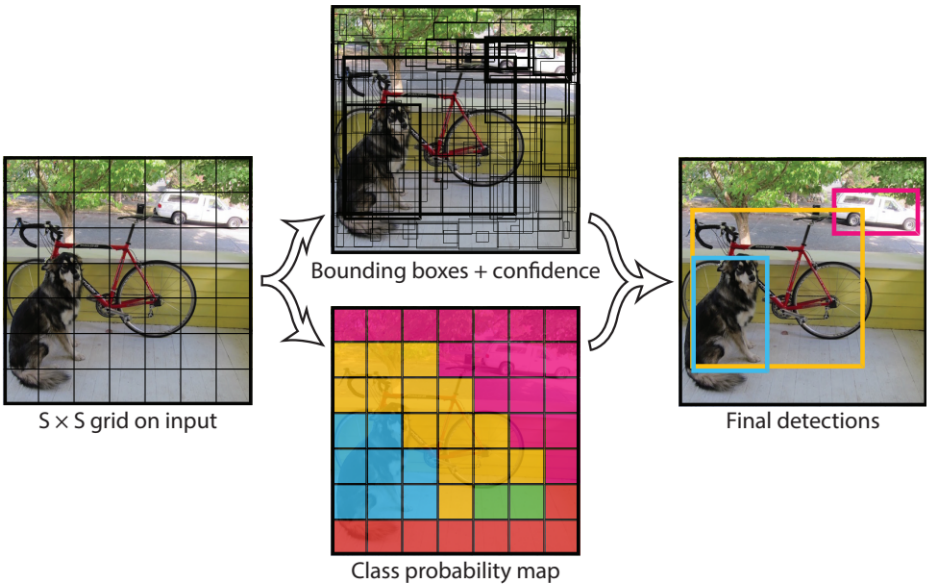
Nie je ľahké poskytnúť všeobecnú metódu detekcie tváre. Neexistuje všeobecne akceptované kritérium pre všetky skupiny. Obvykle sa miešajú a prekrývajú. V tejto časti budú predstavené rôzne metódy detekcie tváre.

- Algoritmus Viola-Jones [77] – Táto metóda je založená na tréningu modelu, ktorý rozumie čo je a čo nie je tvár. Jedná sa o populárny *framework* pre rozpoznávanie tváre v reálnom čase, má problémy s identifikáciou tváre, ktorá je zakrytá alebo natočená do strany.
- Detekcia tváre založená na znalostiach – Táto metóda spolieha na sadu pravidiel vyvinutých ľuďmi podľa našich znalostí. Vieme, že tvár musí obsahovať oči, nos a ústa v určitej vzdialenosti a polohách navzájom. Problém metódy je vytvoriť vhodnú a jasnú sadu pravidiel. Treba nájsť stred medzi všeobecnými a veľmi špecifickými [69].
- Založené na vzhľade – Táto metóda používa statickú analýzu a strojové učenie k nalezeniu relevantných charakteristík obrazu tváre.
- Založené na konvolučných neurónových sietí – Sieť sa učí rozpoznať a klasifikovať tvár osoby [39, 8]. V nasledujúcej časti si popíšeme niektoré takéto riešenia.

4.4.1 YOLO (You Only Look Once)

Jedná sa o inovatívny prístup na detekciu a lokalizáciu objektov, ktorý spracúva obrazy v reálnom čase za použitia konvolučnej neurónovej sieti. Rozdeľuje vstupný obraz na mriežku buniek a predikuje ohraničujúce rámy rôznej veľkosti a tvaru, ktoré slúžia ako referencia predikcie. Toto ohraničenie sa nazýva kotvou, ktoré sú preddefinované ohraničujúce rámy a práve kotvy pomáhajú YOLO generalizovať detekcie pre rôzne typy objektov a mierky.

Ohraničujúce rámy sú predikované pomocou konvolučných sietí *region proposal neural networks* (RPN) [40]. Pomáhajú zvýšiť presnosť detekcie. Práve vďaka svojej flexibilite a efektívnosti sa dostal do širokej škály odvetví aplikácií počítačového videnia [64, 19].



Obr. 4.3: Princíp YOLO modelu [64].

Na obrázku 4.3 môžeme vidieť rozdelenie obrázku do mriežok. Ďalej odhad rámov/kotiev, na ich základe sa určí odhad o aký objekt sa jedná. Na konci môžeme vidieť výsledný obrázok, s určením hraníc jednotlivých objektov.

4.4.2 RetinaFace

RetinaFace je jednostupňový detektor na lokalizáciu tváre a ich príznakov. Na rozdiel od tradičných dvojstupňových detektorov, využíva konvolučnú sieť na predikcie oblastí tváre a kľúčových príznakov ako sú oči, nos a ústa. RetinaFace využíva viacúlohové učenie na zlepšenie robustnosti a presnosti detekcie. Táto architektúra vyniká pri riešení prekrytých častiach tváre, rôznych výrazoch tváre a rôznych environmentálnych podmienkach, čo ju robí ideálnou pre aplikácie v bezpečnosti, interakcii človek-počítač a biometrickej autentizácii [20, 22].

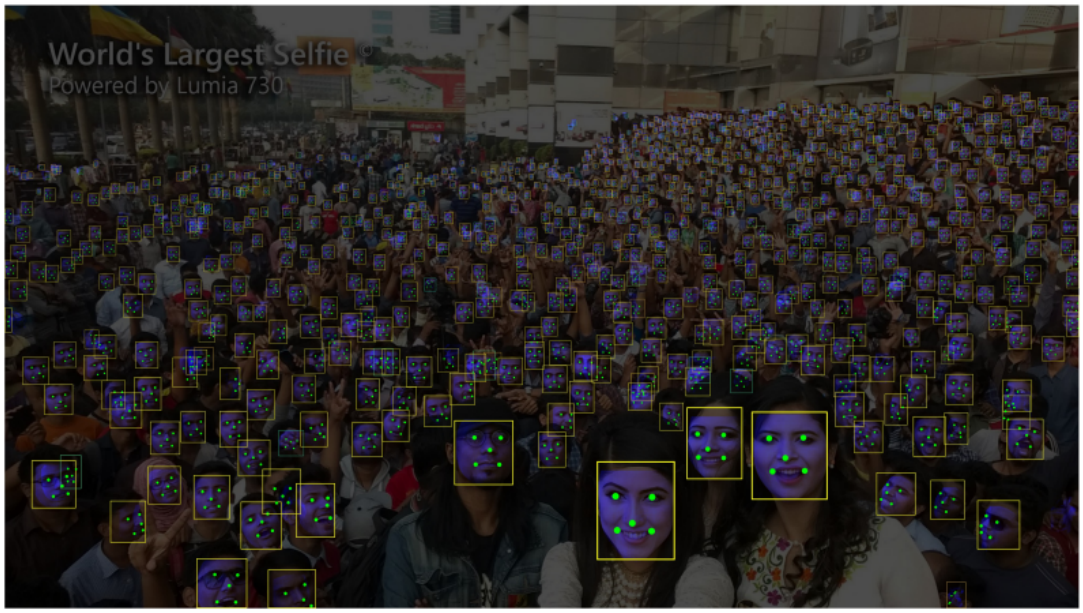
Počas trénovania RetinaFace využíva viacúlohovú stratovú funkciu na súčasné optimalizovanie predikcie ohraničujúcich rámov a príznakov tváre. Stratová funkcia zvyčajne pozostáva z niekoľkých komponentov. Na nasledujúcej rovnici si podrobnejšie popíšeme každú prítomnú komponentu [22]:

$$L = L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{box}(t_i, t_i^*) + \lambda_2 p_i^* L_{pts}(l_i, l_i^*) + \lambda_3 p_i^* L_{pixel} \tag{4.1}$$

Prvá komponenta $L_{cls}(p_i, p_i^*)$ je strata klasifikácie tváre, kde p_i je predpovedaná pravdepodobnosť, že kotva i je tvárou a p_i^* je 1 pre kladnú kotvu a 0 je pre zápornú. Klasifikačná strata L_{cls} je *softmax* strata pre binárne triedy (je alebo nie je tvár). Druhá komponenta regresná strata $L_{box}(t_i, t_i^*)$, kde $t_i = t_x, t_y, t_w, t_{h_i}$ a $t_i^* = t_x^*, t_y^*, t_w^*, t_{h_i}^*$ predstavujú súradnice

predpovedanej mriežky okolo tváre osoby. Tretia komponenta $L_{pts}(l_i, l_i^*)$ predstavuje predpovedaných päť orientačných bodov tváre. Štvrtá komponenta L_{pixel} predstavuje regresnú stratovú funkciu [22].

Celková stratová funkcia je teda váženým súčtom týchto jednotlivých komponentov, pričom sa váhy môžu líšiť v závislosti od dôležitosti pri danej úlohe. Váhy upravujeme pomocou premenných $\lambda_1 - \lambda_3$. Tento prístup zabezpečuje, že RetinaFace sa učí presne lokalizovať tváre a ich príznaky pri zachovaní robustnosti pri rôznych polohách tváre, rôznych vzdialenostiach alebo prekryvov tváre [22].



Obr. 4.4: Ukážka výstupu RetinaFace [20].

Na obrázku 4.4 môžeme vidieť výstup z modelu RetinaFace. Jedná sa o selfie obrázok s najväčším počtom tvárí. RetinaFace dokázal nájsť okolo 900 ľudí z 1151. Maska žltej farby značí spoľahlivý výsledok, zatiaľ čo rámčeky tmavšej farby značia menej spoľahlivý výsledok [20].

4.5 Rozpoznanie tváre

V modernom svete sa čoraz viac spoliehame na technológie schopné identifikovať a rozpoznať objekty a osoby na základe vizuálnych dát. Rozpoznávacie algoritmy zohrávajú kľúčovú úlohu v mnohých aplikáciách, od bezpečnostných systémov a biometrických overení až po autonómne vozidlá a inteligentné zariadenia. Tieto algoritmy umožňujú systémom nielen vidieť, ale aj chápať a interpretovať vizuálny svet okolo nich [37, 88, 14].

V tejto sekcii sa pozrieme na niektoré rozpoznávacie algoritmy používané dnes. Patria sem ArcFace, DeepFace a FaceNet. Každý z týchto algoritmov prináša rôzne prístupy a techniky, ktoré umožňujú identifikáciu a overovanie tvárí. Preskúmame, ako tieto algoritmy fungujú, aké problémy riešia a v akých konkrétnych aplikáciách nachádzajú svoje uplatnenie.

4.5.1 DeepFace

DeepFace je model rozpoznávania tváre vyvinutí výskumníkmi z Facebooku. DeepFace bol trénovaný na označenom datasete o veľkosti viac ako 4 miliónov tvárí, čo bol najväčší súbor údajov o tvári v čase vydania. Prístup je založený na hlbokej neurónovej sieti s deviatimi vrstvami. Model dosahuje presnosť 97,35 % na datasete LFW [82].

DeepFace ako stratovú funkciu využíva *cross-enthropy*, namiesto aktivačnej funkcie *sigmoid* využíva *softmax*, v tvare [75]:

$$L = -log(p_k) \tag{4.2}$$

$$p_k = \frac{exp(o_k)}{\sum_h exp(o_h)} \tag{4.3}$$

kde p_k je výstup zo *softmax* funkcie a o_k je výstup poslednej plne pripojenej vrstvy [75].

4.5.2 FaceNet

FaceNet je systém pre rozpoznávanie a verifikáciu tvárí. Dosahuje to učením mapovania z obrazov tvárí do euklidovského priestoru, kde sa na základe L2 vzdialenosti (podkapitola 4.5.5) určuje podobnosť tvárí. Táto vzdialenosť pre podobné snímky udáva nižšie hodnoty a pre rozdielne vyššie. FaceNet používa *triplet loss*, čo umožňuje efektívne učenie sa rozdielov medzi tvárami. Táto stratová funkcia minimalizuje vzdialenosť medzi kotvou a pozitívnym príkladom (rovnaká identita) a maximalizuje vzdialenosť medzi kotvou (*anchor*) a negatívnym príkladom (rôzna identita). *Triplet loss* je definovaná ako [68]:

$$L = \sum_{i}^{N} \left[\|f(x_a^i) - f(x_p^i)\|_2^2 - \|f(x_a^i) - f(x_n^i)\|_2^2 + \alpha \right]_+ \tag{4.4}$$

kde:

- L: Celková hodnota stratovej funkcie, ktorú minimalizujeme.
- $\sum_i^N$: Suma cez všetky vzorky v trénovacej množine.
- N: Počet vzoriek v trénovacej množine.
- $f(x)$: Funkcia, ktorá premieta obraz x do vektorového priestoru.
- x_a^i: Ukotvený (*anchor*) obraz vo vzorke i.
- x_p^i: Pozitívny obraz vo vzorke i (obraz tej istej osoby ako ukotvený obraz).
- x_n^i: Negatívny obraz vo vzorke i (obraz inej osoby než ukotvený obraz).
- $\|\cdot\|_2$: Euklidovská vzdialenosť (L2 vzdialenosť).
- α: Okraj (*margin*), ktorý oddeľuje pozitívne a negatívne páry.
- $[\cdot]_+$: Funkcia $\max(0, x)$, ktorá zaručuje, že príspevok k strate je nezáporný.

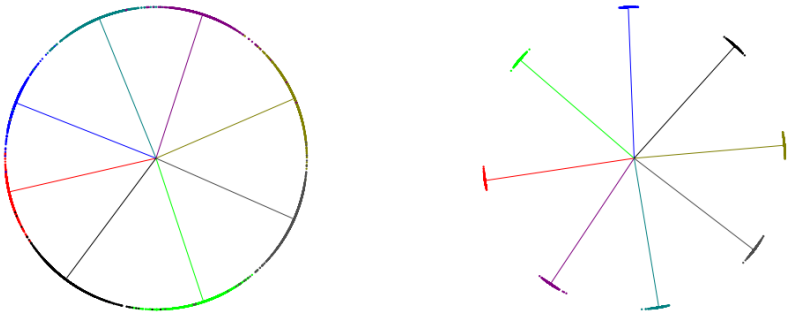
Na súbore dát LFW FaceNet dosiahol rekordnú presnosť 99,63 %. Tento algoritmus sa široko používa v systémoch na overovanie identity a bezpečnostných aplikáciách, kde je potrebná rýchlosť a presnosť pričom [68].

4.5.3 ArcFace

ArcFace je algoritmus pre rozpoznávanie tvárí, ktorý optimalizuje presnosť pomocou *Additive Angular Margin Loss*. Táto stratová funkcia je modifikáciou tradičnej *softmax* stratovej funkcie. Oproti *softmaxu* zahŕňa uhlovú penalizáciu, k výpočtu využíva funkciu arkus kosínus. Celkovo táto modifikácia umožňuje ArcFace lepšie zvládať prácu s nečistými dátami. Stratová funkcia ArcFace je definovaná ako [21]:

$$L = -\log\left(\frac{e^{s\cdot\cos(\theta_{y_i}+m)}}{e^{s\cdot\cos(\theta_{y_i}+m)} + \sum_{j\neq y_i}^{N}(e^{s\cdot\cos\theta_j})}\right) \tag{4.5}$$

kde θ_j predstavuje úhol medzi váhou W_j a vlastnosťou, x_i, y_i je trieda i-tej vzorky, s je škálovací faktor na kontrolu veľkosti vstupných vlastností a m je uhlová penalizácia. Na nasledujúcom obrázku 4.5 môžeme vidieť rozdiel stratových funkcií medzi *Softmax* a ArcFace. Z obrázku 4.5 je vidieť lepšiu jednoznačnosť ArcFace oproti *Softmax*. Práve vďaka tejto vlastnosti predstavuje ArcFace významnú úlohu pri rozpoznávaný tváre [21].



(a) Norm-Softmax (b) ArcFace

Obr. 4.5: Rozdiel stratovej funckie *Softmax* a ArcFace [21].

Táto metóda zabezpečuje, že rozdiely medzi vektormi tvárí rôznych osôb sú maximalizované, čo zvyšuje spoľahlivosť rozpoznávania. ArcFace sa využíva v biometrických overovacích systémoch, kde je vysoká presnosť kľúčová, napríklad pri zabezpečení prístupu do citlivých oblastí [21].

4.5.4 MagFace

MagFace je algoritmus na rozpoznávanie tváre a pre hodnotenie kvality. Vychádza z ArcFace, kde sa snaží vytlačiť nejednoznačné vzorky preč z trénovacej množiny, aby sa zlepšil celkový výstup. K ArcFace stratovej funkcií pridáva uhlový okraj $m(a_i)$ a regularizátor $g(a_i)$. Uhlový okraj $m(a_i)$ sa stará o vyradzovanie menej kvalitných snímkov, pričom regularizátor $g(a_i)$ sa snaží docieliť, aby sa model hlavne učil na kvalitnejších snímkov. Stratová funkcia MagFace vyzerá nasledovne [54]:

$$L = -\log\left(\frac{e^{s\cdot\cos(\theta_{y_i}+m(a_i)}}{e^{s\cdot\cos(\theta_{y_i}+m(a_i))} + \sum_{j\neq y_i}^{N}(e^{s\cdot\cos\theta_j})}\right) + \lambda_g g(a_i) \tag{4.6}$$

kde θ_j predstavuje úhol medzi váhou W_j a vlastnosťou, x_i, y_i je trieda i-tej vzorky, s je škálovací faktor na kontrolu veľkosti vstupných vlastností a m je uhlová penalizácia,

táto časť je totožná s ArcFace. MagFace pridáva $m(a_i)$, čo predstavuje úhlový okraj, $g(a_i)$ regularizátor, ktorý oceňuje kvalitnejšie snímky a λ_g je hyper-parameter pre balancovanie klasifikačnej a regulačnej straty [54].

4.5.5 Porovnanie embedding vektorov

Pri rozpoznávaní tváre sú *embedding* vektory kľúčovým komponentom, ktorý umožňuje porovnávať rôzne tváre. Na porovnanie týchto vektorov sa používajú rôzne metriky vzdialenosti, medzi najčastejšie patrí Euklidovská $L2$ vzdialenosť, kosínusová vzdialenosť a Euklidovská $L2$ normovaná.

Euklidovská $L2$ vzdialenosť má nasledujúcu matematickú definíciu [78]:

$$d(\overrightarrow{v}, \overrightarrow{w}) = \sqrt{\sum_{i}^{n}(v_i - w_i)^2} \tag{4.7}$$

kde x a y sú *embedding* vektory dvoch tvárí. Používa sa na meranie priamej vzdialenosti medzi dvoma bodmi v n-rozmernom priestore. Čím je vzdialenosť vektorov menšia, tým je väčšia pravdepodobnosť, že sú si podobné [78].

Kosínusová vzdialenosť má nasledujúcu matematickú definíciu [78]:

$$cos(\beta) = \frac{\overrightarrow{v} \cdot \overrightarrow{w}}{\|\overrightarrow{v}\| \times \|\overrightarrow{w}\|} \tag{4.8}$$

kde v a w sú *embedding* vektory, $v \cdot w$ je skalárny súčin a $\|v\|$ je $L2$ norma vektora v. Používa sa na meranie uhla medzi dvomi vektormi, čím sa ohodnotí ich orientácia. Vhodná pre viac dimenzionálne priestory [78].

Euklidovská $L2$ normovaná vzdialenosť má nasledujúcu matematickú definíciu [83]:

$$\|\overrightarrow{v}\|_2 = \sqrt{\sum_{i}^{n} v_i^2} \tag{4.9}$$

kde v je *embedding* vektor. Používa sa na normalizáciu vektorov, aby sa zabezpečilo, že všetky vektory majú rovnakú dĺžku. Normalizované vektory môžu byť následne porovnávané Euklidovskou alebo kosínusovou vzdialenosťou [83, 31].

4.5.6 Kľúčové metriky v oblasti rozpoznania tváre

V oblasti rozpoznania tváre, najmä pri detekcii deepfake, je nevýhnutné hodnotiť výkon rôznych algoritmov pomocou štandardných metrík pre binárnu klasifikáciu. Tieto metriky nám pomáhajú pochopiť ako dobre dokáže algoritmus vykonávať danú úlohu, v našom prípade ako dokáže rozlíšiť medzi skutočnými a zmanipulovanými deepfake obrazmi. Kľúčové metriky zahŕňajú *True Positive*, *True Negative* a presnosť [25].

True Positive (TP) metrika predstavuje podiel prípadov, kedy algoritmus správne identifikuje pozitívny prípad. V našich experimentoch rozpoznania tváre ide o prípad, keď algoritmus správne identifikuje o dvoch skutočných obrázkov rovnakej osoby budú vyhodnotené, že patria tej istej osobe [25].

True Negative (TN) metrika predstavuje podiel prípadov, kedy algoritmus správne identifikuje negatívny prípad. V našich experimentoch, práve vtedy keď ku skutočnému obrázku

pôjde súčasne na vstup deepfake snímok, čiže výsledný výstup bude, že vstupné snímky neobsahujú rovnakú osobu [25].

Výsledná presnosť, pre našu úlohu sa následne počíta ako:

$$Accuracy = \frac{TP + TN}{Cases} \tag{4.10}$$

kde *Accuracy* je presnosť a *Cases* značí počet prípadov. Vysoká presnosť naznačuje, že algoritmus je spoľahlivý v rozlišovaní medzi skutočnými obrázkami a deepfake[25].

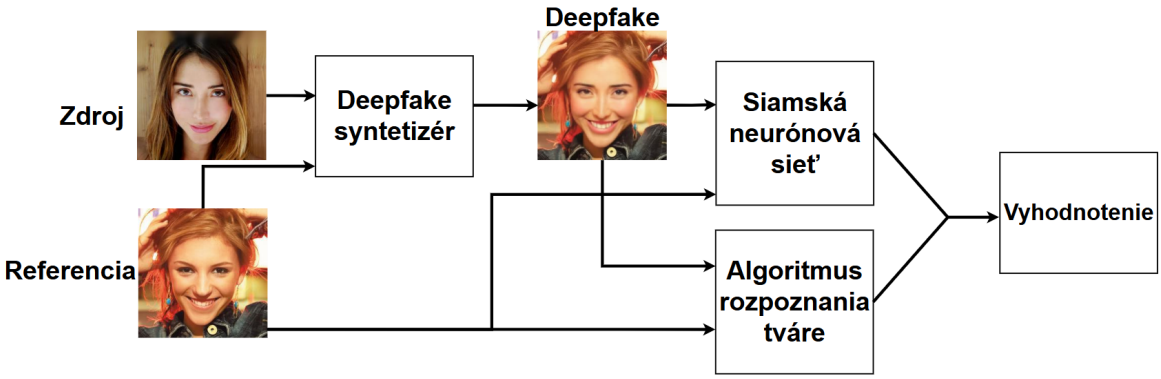
Kapitola 5

Návrh a implementácia

Táto kapitola sa venuje návrhu a implementačnej časti tejto práce. V úvode práce bude popísaný návrh riešenia, ďalej sa pozrieme na nástroje a frameworky pre prácu s rozpoznaním tváre. Na záver je podrobnejší popis natrénovaných model konvolučných neurónových sietí.

5.1 Návrh riešenia

Nasledujúci obrázok 5.1 obsahuje diagram aplikácie, ktorý si bližšie popíšeme. Na vstup aplikácie prichádzajú dva vstupné obrázky. Jeden z ktorého beriem tvár osoby (zdroj) a druhý referenčný na ktorý vložíme tvár zo zdroja. Z deepfake syntetizéru nám vypadne deepfake snímok. Ten spoločne s referenciou putuje do siamskej neurónovej siete a algoritmu rozpoznania tváre. V aplikácií je na výber z ArcFace, DeepFace, FaceNet a MagFace. Po tejto fáze dochádza k vyhodnoteniu výsledkov z oboch modelov.



Obr. 5.1: Diagram aplikácie.

5.2 Nástroje a frameworky

Pre detekciu tváre existujú rôzne nástroje, ktoré obsahujú knižnice pracujúce s rozpoznaním tváre. Tieto nástroje nám môžu byť sprístupnené pomocou frameworku pre zjednodušenie práce s nimi. V tejto časti si popíšeme nástroje detekcie tváre a k nim 2 frameworky, využité pri implementácií tejto práce.

5.2.1 Nástroje detekcie tváre a rozpoznávania

Na detekciu tváre sa používajú rôzne nástroje. V nasledujúcom zozname si predstavíme zopár známych nástrojov detekcie [39, 8]: V tejto podsekcií si predstavíme zopár nástrojov používaných pre detekciu tváre.

- Dlib[1]. Jedná sa o *toolkit* používaný v bezpečnosti a analýze obrazu.
- Megvii Face++[2] je často používaný pre kontrolu prístupu, elektronické obchodníctvo a sociálne média.
- Deepface[3] je *framework* pre Python, ktorý poskytuje analýzu tvárových vlastností ako je vek, pohlavie, rasa a emocionálny stav.
- FaceNet[4] vyvíjaný Googlom používa Python knižnice.
- InsightFace[5] je ďalšia Python knižnica dostupná na GitHube.
- OpenCV[6] je *open-source* Python knižnica pre počítačové videnie
- PIL[7] je knižnica pre spracovanie obrázkov, implementovaná v jazyku Python.

5.2.2 Knižnica TensorFlow a Keras API

TensorFlow [1] je *open-source* knižnica pre strojové učenie vytvorená Googlom. TensorFlow sa používa na tvorenie a trénovanie modelov hlbokého učenia, pretože uľahčuje tvorbu výpočetných grafov a zaisťuje efektívny beh modelov na rôznych druhoch hardwaru. Knižnica Tensorflow v sebe obsahuje populárne vysoko úrovňové aplikačné programové rozhranie (API) Keras [17]. Keras poskytuje prístupné rozhranie pre riešenie problémov strojového učenia. Keras pokrýva každý krok postupu pri strojovom učení od spracovania cez ladenie až po nasadenie. Cieľ jeho vývoja bol umožniť rýchle experimentovanie pri tvorbe sietí.

5.2.3 Frameworky

Prvý framework, ktorý sme v rámci práci využili je Deepface. Umožňuje prepínať medzi niekoľkými *state-of-art* modelmi ako sú FaceNet, ArcFace, a mnohé ďalšie. Tieto modely používa na detekciu, zarovnanie, reprezentáciu a overenie tváre. S pomocou pár riadkov kódu dokáže prispôsobiť rozpoznávanie tváre na úrovni ľudí [70]. Na nasledujúcom obrázku 5.2 môžeme vidieť ukážku výstupu Deepface frameworku.

[1] https://github.com/davisking/dlib
[2] https://en.megvii.com/technologies/face_recognition
[3] https://github.com/serengil/deepface
[4] https://github.com/davidsandberg/facenet
[5] https://github.com/deepinsight/insightface
[6] https://opencv.org/
[7] https://pillow.readthedocs.io/en/stable/

Obr. 5.2: Výstup deepfacu pri overení tvári dvoch snímkov [70].

Na obrázku môžeme vidieť výstup deepface, kde sa snaží zistiť, či tvár na dvoch snímkoch patrí rovnakej osobe. Na výstupe môžeme vidieť, že sa jedná o rovnakú osobu.

Druhý framework bola knižnica FaceAIKit [28], ktorá sa zameriava na detekciu a rozpoznanie tváre. Umožňuje jednoducho integrovať *state-of-the-art* modely rozpoznania tváre. Pre detekciu tváre využíva RetinaFace a na rozpoznanie ArcFace alebo MagFace [28].

5.3 Implementácia hodnotenia vierohodnosti

Pre analýzu algoritmov rozpoznania tváre boli využité dva *frameworky*, FaceAIKit a Deepface.

Prvý si popíšeme Deepface. Deepface ponúka na výber z troch funkcii na výpočet vzdialeností - kosínovskú, euklidovskú a L2 euklidovskú, pričom L2 by mala dosahovať najlepšie výsledky podľa autorov. Deepface obsahuje v sebe veľa modelov pre úlohy rozpoznania tváre, mi sme si vybrali FaceNet, DeepFace a ArcFace. Keď máme vybratý model a funkciu pre výpočet vzdialenosti môžeme zavolať funkciu *verify*, ako môžeme vidieť v nasledujúcom úseku kódu 1:

Algorithm 1 Príklad práce s Deepface

```
result = DeepFace.verify(img1_path = reference,
           img2_path = image,
           model_name = "euclidean_l2",
           distance_metric = "Facenet")
```

V *result* potom dostaneme výsledky vyhodnotenia, či sa referenčná osoba zhoduje s testovanou.

Druhý použitý *framework* FaceAIKit pre výpočet vzdialenosti využíva tiež L2 euklidovskú vzdialenosť. FaceAIKit ponúka model ArcFace a MagFace. Knižnica sa na začiatku načíta a potom sú z nej volané funkcie ako môžeme vidieť na nasledujúcej ukážke 2:

Algorithm 2 Príklad práce s FaceAIKit

```
face_lib = FaceRecognition(recognition="arcface")

results1 = face_lib.face_detection(reference, align='keypoints')
results2 = face_lib.face_detection(image, align='keypoints')

distance = face_lib.verify(results1, results2)
```

Na základe hodnoty L2 sa potom určí či sa jedná o rovnaké osoby alebo nie.

5.4 Návrh modelov

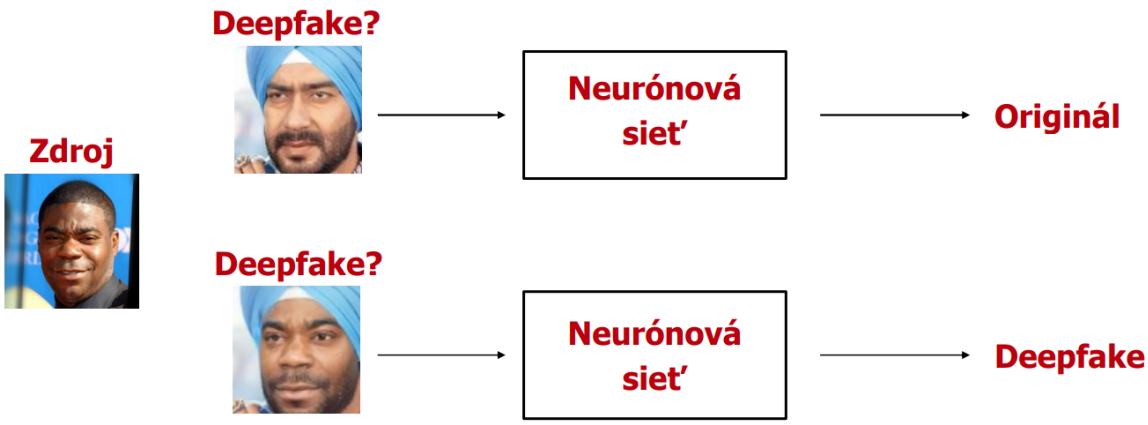
Táto sekcia sa bude venovať návrhu modelu. Podrobnejšie si popíšeme architektúru siamskej siete, ktorá bola vychádzala z vedeckého článku.

Návrh modelov s jedným vstupným snímkom obsahoval jednu sieť, ktorá bola poskytnutá skrz knižnicu TensorFlow ako môžeme vidieť v nasledujúcej ukážke *Algorithm* 3.

Algorithm 3 Príklad so sieťou Vgg19

```
base_model = VGG19(include_top=False, input_shape=(128,128,3))
```

Schéma modelu potom vyzerá nasledovne 5.3:



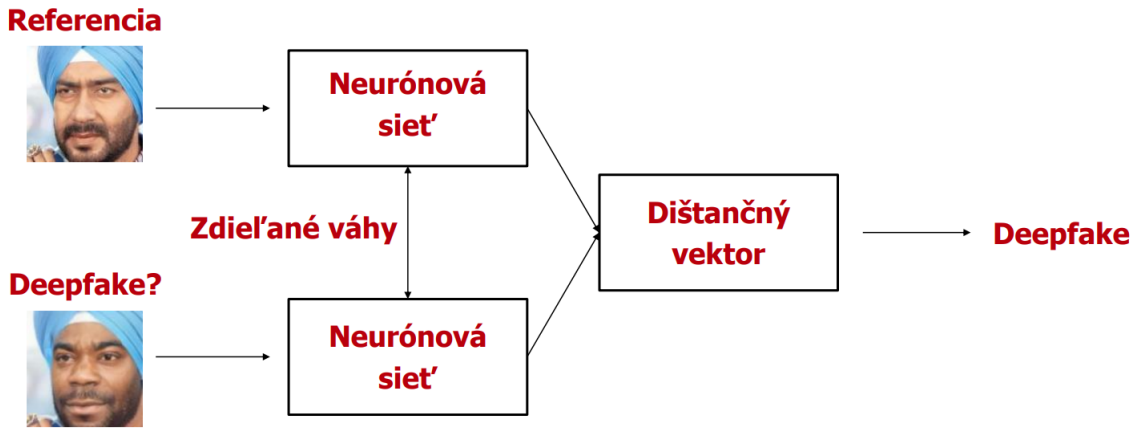
Obr. 5.3: Schéma návrhu modelu s jedným snímkom na vstupe.

Na obrázku 5.3 môžeme vidieť zdrojový obrázok, ktorý bol použitý pri vytvorení deepfaku. Prvý obrázok obsahuje originálny snímok a druhý je deepfake. Snímky idú do neurónovej siete, ktorá následne rozhodne či sa jedná o deepfake alebo nie.

Návrh siamskej siete bol komplexnejší a bol inšpirovaný článkom [46]. Štruktúra súboru bola prevzatá z práce [17], z ktorej sme čerpali poznatky. Narozdiel od článkov sme nedefinovali vlastný model neurónovej siete, ale použili sme architektúru Vgg alebo ResNet. Predstavený model v článku [17] nedosahoval dobré výsledky pre našu problematiku.

Siete boli vložené do zdrojového kódu rovnako ako je naznačené v predchádzajúcej ukážke *Algorithm* 3. Podrobnejšie detaily implementácie sú popísané v nasledujúcej časti.

Schéma modelu siamskej siete vyzerá nasledovne 5.4:



Obr. 5.4: Schéma návrhu modelu siamskej neurónovej siete.

Na obrázku 5.4 môžeme vidieť ako do modelu vstupujú dva vstupné snímky. Prvý z nich je vždy referencia a druhý je obrázok na porovnanie. Oba snímky idú do zvlášť neurónovej siete, avšak tieto dve siete zdieľajú váhy. Každá z nich pošle svoj výstup do dištančného vektora na základe jeho hodnoty sa určí výsledok.

5.5 Implementácia a trénovanie modelu

Výsledná aplikácia je tvorená v programovacom jazyku Python, prevažne za použitia knižnice TensorFlow [1] a Kerasu [17].

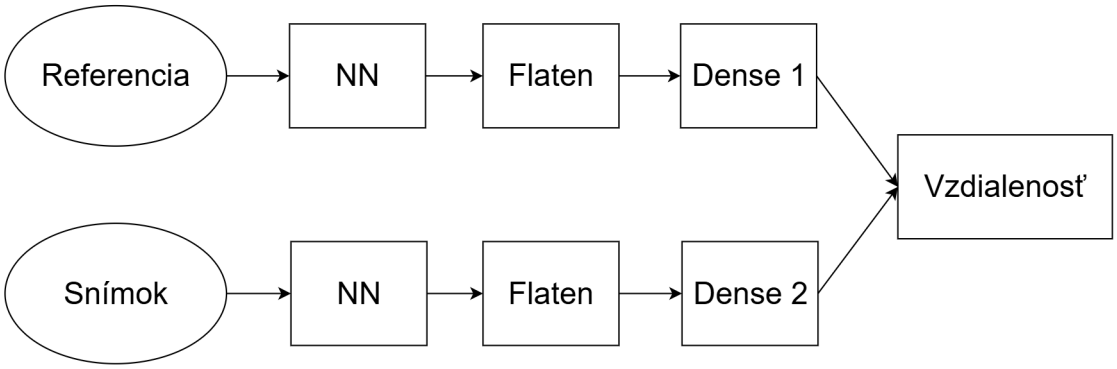
Po vytvorení datasetu sa začali vytvárať modely neurónových sietí. V rámci práce sú demonštrované pokusy na dvoch rôznych modelov neurónových sietí. Prvý model berie na vstup jeden vstupný snímok a ako architektúru používa jednu sieť Vgg. Druhý model berie na vstup dva vstupné obrázky a je vytvorený podľa schémy siamskej neurónovej siete z vedeckého článku [44].

Nami vytvorený dataset je uložený do dvoch priečinkov Deepfake a Original. Deepfake priečinok obsahuje 4900 snímkov a priečinok Original 1400 snímkov. V položke Original sú snímky pomenované ako Origin_počet a v položke deepfake ako DeepX_počet, kde X značí o ktorý deepfake algoritmus sa jedná, a počet je poradové číslo snímky. Deep1 je pre program Roop, Deep2 pre program FaceFusion, Deep3 pre program Ghost a Deep4 pre program SimSwap.

Trénovanie prvého modelu prebiehalo lokálne sa použitia grafickej karty RTX 3050 Ti. Verzia využitej TensorFlow knižnice bola 2.10.0 a CUDA 11.2 pre grafickú akceleráciu. Model obsahuje funkciu z knižnice TensorFlow, ktorá na základe pozície položky v priečinku vygeneruje označenie pre dáta. Deepfake dáta sú označené číslom 0 a originálne dáta sú označené číslom 1. Dáta sú potom normalizované, aby obsahovali hodnoty v rozmedzí (0, 1), čo umožňuje rýchlejšiu konvergenciu modelu, pre lepšie trénovanie neurónovej siete. Následne sú dáta zamiešané a rozdelené do trénovacích, validačných a testovacích dát. Sú rozdelené v pomere 70 % pre trénovacie dáta, 20 % pre validačné dáta a zvyšok je priradený testovacím dátam. Jednotlivé *batche* obsahujú 32 snímkov, čo je východzia hodnota,

ktorá bola pre našu problematiku dostačujúca. Ako architektúra neurónovej siete je použitá primárne, už spomínaná architektúra Vgg-net. V experimentálnej časti sú popísané experimenty s Vgg16 a Vgg19. V rámci porovnania výsledkov boli využité aj siete ResNet, konkrétne ResNet50 a ResNet101. Je možné naimportovať predtrénované váhy z datasetu ImageNet do sietí alebo môžeme si váhy natrénovať samy na základe datasetu. Koncové vrstvy vynechávame, aby sme si ich mohli nadefinovať podľa seba. Ako poslednú vrstvu používame plne prepojenú vrstvu s funkciou sigmoid, ktorá sa hodí pre binárnu klasifikáciu. Optimalizátor je použitý Adam [42] ako stratovú funkciu sme použili *binary cross-entropy*, keďže klasifikujeme dva typy dát.

Druhý model vyžadoval dataset uložiť ako páry dvoch obrázkov, keďže na vstup potrebujeme dva vstupné obrázky. Z toho dôvodu sme sa rozhodli vytvoriť si vlastnú funkciu na načítanie dát a ich následné uloženie pre tréning siete. Dokopy je uložených 11200 párov z nich 4900 je vytvorených spojením originálnych snímkov a deepfake snímkov rovnakej osoby s označením 1, ďalších 4900 je vytvorených spojením originálneho snímku s deepfakom inej osoby s označením 0 a zvyšok je originál a originál rovnakého človeka označení ako 0. Následne ako pri prvom modely sa jednotlivé snímky normalizujú na hodnotu v rozmedzí (0, 1) a potom sa zamieša ich poradie. Rovnako ich pomer je 70 % pre testovacie dáta, 20 % pre validačné dáta a zvyšok sú testovacie dáta. Veľkosť *batchu* je nastavená na 64 snímkov. Architektúra modelu sa skladá z architektúry Vgg-net a ResNet. Na vstup sa berie jeden pár snímkov, jednotlivý snímok z páru ide cez Vgg sieť a následne sa pomocou výsledku z jednotlivej siete počíta dištančný vektor v tvare $D = |dense1 - dense2|$. Na nasledujúcom obrázku 5.5 môžeme vidieť podrobnejšie schéma na ktorom môžeme vidieť ako sa počíta spomínaný dištančný vektor.



Obr. 5.5: Podrobnejšie schéma siamskej siete.

Na obrázku môžeme vidieť ako výstup jednotlivej siete ide do *Flaten* vrstvy, ktorej výstup ide do *Dense* na základe ich výstupu sa vypočíta dištančný vektor. Výsledný dištančný vektor ide do plne prepojenej vrstvy s aktivačnou funkciou sigmoid. Optimalizér je použitý Adam, ktorý inicializujeme s rýchlosťou učenia $I_r = 1e-4$. Tento parameter ovplyvňuje veľkosť kroku počas optimalizácie, určujúc ako veľmi sú upravené parametri modelu v každej iterácií tréningu. Ako stratové funkcie sme využili *contrastive loss* a *binary cross-entropy*.

Kapitola 6

Experimenty a výsledky

V tejto kapitole sa venujeme experimentálnej časti práce. Na začiatku kapitoly bude popis prípravy datasetu, vytvárania deepfakov a zhrnutie použitých programov. Následne bude popis experimentov modelov neurónových sietí s rôznymi architektúrami a porovnanie ich výsledkov. Ďalej bude nasledovať vyhodnotenie experimentov algoritmov rozpoznania tváre a nakoniec vyhodnotenie dosiahnutých výsledkov.

6.1 Príprava datasetu

Dataset je dátový súbor, ktorý je organizovaný a uložený spoločne pre analýzu alebo spracovanie. Datasety sú základným nástrojom v oblasti analýzy dát a strojové učenie [18].

V rámci implementácie bolo ako prvé treba vytvoriť vlastný deepfake dataset za použitia už existujúcich datasetov. Dáta pre náš dataset boli čerpané zo známeho datasetu ,ktorý sa často využíva pre prácu rozpoznania ľudí v obraze, Labeled Faces in the Wild(LFW) [33] a CelebA HQ [49]. Dáta z týchto datasetov boli využité pre evaluáciu algoritmov. Z LFW datasetu bolo vybraných 700 osôb ako cieľové obrázky pre tvorbu deepfake snímkov a k nim bolo náhodne vybraných 14 osôb, ktoré sa použili ako zdroj pre deepfake snímky. Tvár zdrojovej osoby je použitá na mapovanie 50 ľudí. Pri datasete CelebA bolo taktiež použitých 700 osôb aj s rovnakým postupom tvorby deepfakov ako tomu bolo pri LFW. Dokopy boli použité 4 deepfake algoritmy pre väčšiu rozmanitosť deepfake datasetu a zaistenia väčšej robustnosti modelu neurónovej siete. Pre experimenty algoritmov rozpoznania tváre bol zvolený dataset CFP-FP.

6.2 Tvorba deepfake

V nasledujúcej sekcií si popíšeme jednotlivé použité deepfake algoritmy. Pri každom bude uvedený názov, pocity, vhodnosť a nakoniec zobrazíme ich výstupy.

6.2.1 Deepfake program - Roop

Prvý deepfake program, ktorý bol v tejto práci použitý sa nazýva Roop [74]. Jedná sa o *open-source*, ktorý umožňuje vymieňať tváre ľudí na vstupe. Tento program poskytoval grafické užívateľské rozhranie (GUI), z ktorého bolo na prvý pohľad jasné ako sa má program používať. Pri výmene tvárí, keď boli značné rozdiely vo farbe alebo množstve vlasov na hlave, tak na výslednom obrázku vznikali viditeľné nedostatky vo vlasoch osôb, ktoré boli

na prvý pohľad zjavne, že sa nejedná o reálnu osobu, obrázok 6.1. Na obrázku 6.2 môžeme vidieť výstup programu.

Obr. 6.1: Príklad problému s vlasmi – Roop aj FaceFusion. Sprava prvý výsledok z FaceFusion, druhý sprava je z Roop.

Obr. 6.2: Príklad výstupu – Roop: zľava zdrojový obrázok, v strede je cieľový a vpravo je výsledný deepfake.

6.2.2 Deepfake program - FaceFusion

Druhý deepfake program, ktorý bol v tejto práci využití sa nazýva FaceFusion [65]. Aplikácia obsahuje prehľadný návod na inštaláciu, čo ju umožňuje sprístupniť viacerým ľudom. Aplikáciu hodnotím pozitívne, pocitovo sa aj príjemnejšie ovládala a po skončení bolo ihneď vidieť výsledok v GUI aplikácie. Na obrázku 6.3 môžeme vidieť výstup programu.

Obr. 6.3: Príklad výstupu – FaceFusion: zľava zdrojový obrázok, v strede je cieľový a vpravo je výsledný deepfake.

6.2.3 Deepfake program - Ghost

Tretí použitý program pre tvorbu deepfake snímkov v tejto práci bol Ghost [30]. Jednalo sa o konzolovú aplikáciu. Neobsahuje oproti predchádzajúcim aplikáciám GUI a taktiež sa nedá nastaviť ktorú osobu, čo berieme za menšie negatívum. Proces tvorby deepfake snímkov bol vykonávaný rýchlejšie oproti predchádzajúcim aplikáciám. Program ako jediný zo všetkých použitých mal problémy vytvárať deepfake snímky za použitia datasetu CelebA HQ. V experimentálnej časti sme sa preto rozhodli využiť iba výstupy programu pre dataset LFW. Na obrázku 6.4 môžeme vidieť výstup programu.

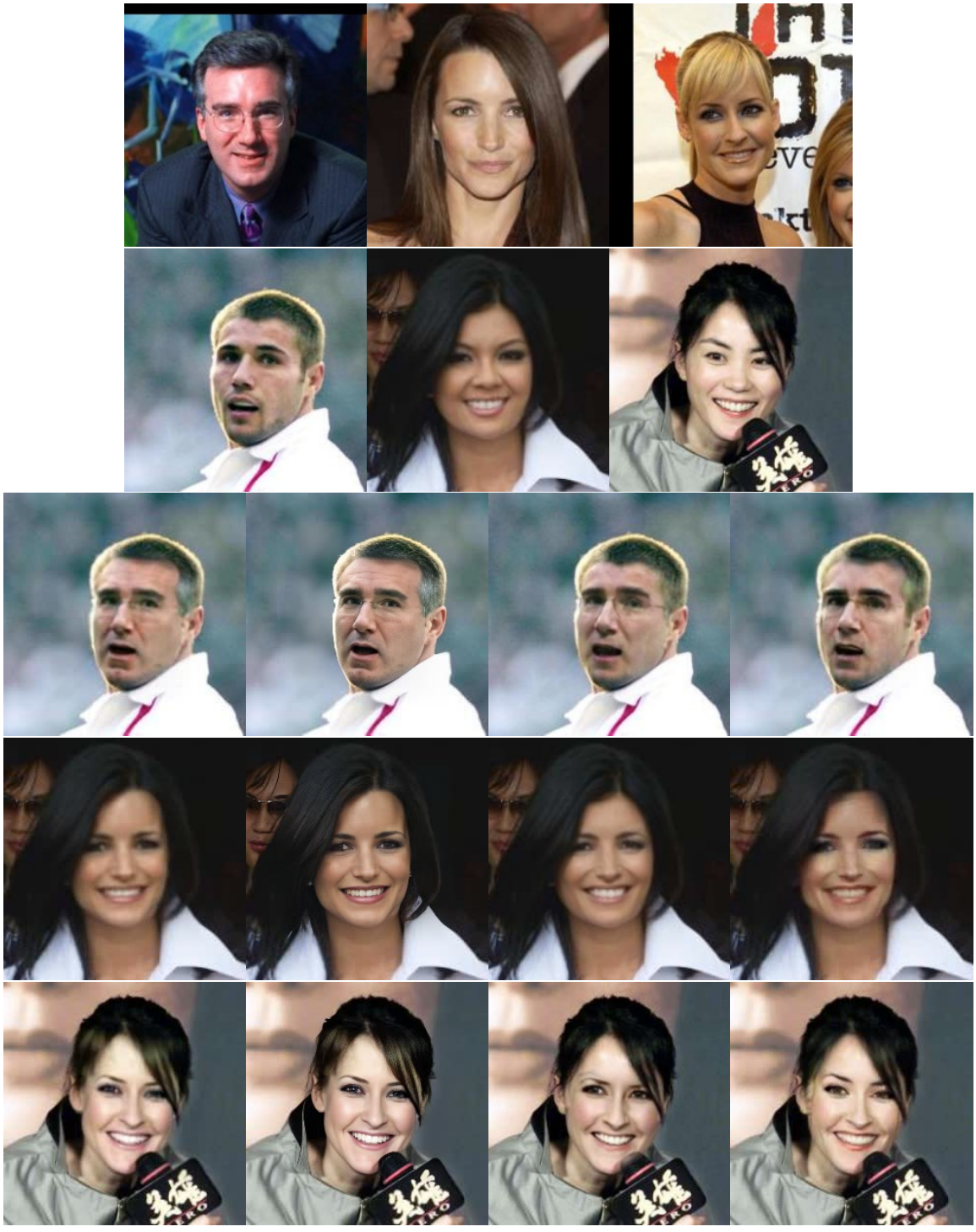
Obr. 6.4: Príklad výstupu – Ghost: zľava zdrojový obrázok, v strede je cieľov a vpravo je výsledný deepfake.

6.2.4 Deepfake program - SimSwap

Posledným použitým programom pre tvorbu deepfake snímkov bol SimSwap [16]. Jedná sa o konzolovú aplikáciu tak ako tomu bolo pri Ghoste. Pri veľkej podobnosti osôb boli vytvorené výsledky dobrej kvality. Aplikáciu bolo náročné nainštalovať, jedná sa o starší projekt, ktorý nie je extra udržiavaný. Celkový dojem radíme medzi horšie, najmä z dôvodu už spomínanej zaťaženej inštalácie programu. Na obrázku 6.5 môžeme vidieť výstup programu.

Obr. 6.5: Príklad výstupu – SimSwap: zľava zdrojový obrázok, v strede je cieľov a vpravo je výsledný deepfake.

Na nasledujúcom obrázku 6.6 môžeme vidieť výstup všetkých deepfake programov, s cieľovými obrázkami aj s originálmi.



Obr. 6.6: Prvý riadok obsahuje snímky s tvárami osôb pre tvorbu deepfake. Druhý riadok obsahuje cieľové snímky. Pod prvým riadkom sú jednotlivé výstupy deepfakov – Zľava originál, nasleduje roop, FaceFusion, Ghost, SimSwap.

6.3 Orezanie datasetu podľa tváre

Po úspešnom vytvorení deepfake snímkov a usporiadaným datasetu sa pred samotným implementovaní modelu neurónovej siete sme najskôr samotný LFW dataset zmenšili z pôvodnej veľkosti 250×250 na 128×128. Dataset CelebA obsahoval snímky veľkosti 256×256, ktoré taktiež boli zmenšené na 128×128. To isté sme vykonali aj pre testovací dataset CFP-FP. Cieľ zmenšenia jednotlivých snímkov bolo, aby sa neurónová sieť zameriavala primárne na črty ľudskej tváre, aby sme dospeli k lepším výsledkom. Orezávanie obrázka bolo vykonané za použitia *open-source* projektu autocrop [47] dostupného na GitHube. Algoritmus fungoval rovnako úspešne pre pôvodné obrázky ako aj pre deepfake snímky. Občas nejaké snímky orezal chybne alebo vôbec. Jednalo sa asi o 5 % datasetu, tieto snímky boli napokon orezané ručne. Ukážka výsledných snímkov na obrázku 6.7.

Obr. 6.7: Ukážka finálneho stavu datasetu pred samotným trénovaním modelu neurónovej siete.

6.4 Model za použitia jednej siete architektúry Vgg-Net

V tejto sekcií si popíšeme výsledky konfigurácií použitia siete Vgg s jedným vstupným snímkom. Tento model dostával postupne na vstup snímky po jednom a na ich základe sa snažil naučiť črty tváre, pre rozoznanie deepfake snímku od skutočného. Experimenty boli prevedené s obomi verziami siete Vgg a s dvomi verziami siete ResNet, konkrétne ResNet50 a ResNet101. Taktiež bolo vyskúšané použitie predtrénovaných váh ImageNet.

Pred samotnou výslednou tabuľkou si popíšeme ako sme dostali výsledné hodnoty v tabuľkách 6.1 a 6.2. Trénovanie modelov bolo nastavené na 50 epóch. Prvý stĺpec tabuľky obsahuje názov využitej architektúry neurónovej siete. Druhý stĺpec obsahuje hodnotu ImageNet alebo None. ImageNet udáva využitie predtrénovaných váh z datasetu ImageNet a None udáva, že neboli využité žiadne predtrénované váhy. Ďalšie stĺpce (Roop, Face, Ghost, Sim) obsahujú názov využitého deepfake programu pre tvorbu deepfake snímkov. Percentuálna úspešnosť udáva v koľkých percentách prípadoch správne klasifikoval vstupný snímok, táto hodnota bola vypočítaná ako aritmetický priemer predchádzajúcich stĺpcov tabuľky. Príklad ak na vstup prišiel deepfake a neurónová sieť určila, že sa o deepfake jedná percentuálna úspešnosť sa zvýšila. Ak výstup z nejakého z deepfake programov model určil ako originálny úspešnosť sa nezvýšila. Podobne je tomu aj pri stĺpci Origin, kde sa snažíme zistiť v koľkých prípadoch neurónová sieť správne určí, že vstupný snímok je originál a nie deepfake. Posledný stĺpec uvádza aritmetický priemer predchádzajúcich hodnôt, aby sme následne boli schopný modely medzi sebou porovnať. Na testovanie bolo využitých 500 snímkov. 100 originálnych snímkov a dokopy 400 deepfakov, pričom 100 snímkov bol výstup z každého deepfake programu. 50 snímok bolo z datasetu LFW a 50 z datasetu CelebA HQ. V nasledujúcej tabuľke 6.1 môžeme vidieť výsledok experimentov so sieťou Vgg:

Model	**Váhy**	**Roop**	**Face**	**Ghost**	**Sim**	**Origin**	**Suma**
Vgg16	ImageNet	100 %	87 %	91 %	88 %	38 %	**80,8** %
Vgg16	None	100 %	75 %	88 %	88 %	41 %	78,4 %
Vgg19	ImageNet	100 %	75 %	88 %	88 %	41 %	78,4 %
Vgg19	None	100 %	65 %	71 %	82 %	57 %	75,0 %

Tabuľka 6.1: Výsledky experimentov so sieťou Vgg.

Z výsledkov môžeme vidieť, že modely mali najväčší problém s klasifikáciou originálnych snímkov. Prvý deepfake algoritmu určili všetky konfigurácie bez problémov, pri ostatných sa výsledky líšili podľa zvolenej konfigurácie. Najúspešnejší model bol za použitia siete Vgg16 s využitím predtrénovaných váh ImageNet, ktorý dosiahol úspešnosť 80,8 %. Na nasledujúcej tabuľke 6.2 môžeme vidieť výsledky experimentov so sieťou ResNet. Sieť ResNet sa pri tejto úlohe neosvedčila, nebola vôbec schopná správne určiť kedy sa jedná o originálny snímok. Mala tendenciu označovať všetky vstupné snímky ako deepfake.

Model	**Váhy**	**Roop**	**Face**	**Ghost**	**Sim**	**Origin**	**Suma**
ResNet50	ImageNet	100 %	92 %	99 %	96 %	0,1 %	77,5 %
ResNet50	None	100 %	93 %	99 %	99 %	0,1 %	78,2 %
ResNet101	ImageNet	100 %	97 %	100 %	99 %	0 %	79,2 %
ResNet101	None	100 %	96 %	100 %	99 %	0,1 %	79,0 %

Tabuľka 6.2: Výsledky experimentov so sieťou ResNet.

Prvá sada experimentov ukázala, že aj za použitia iba jednej konvolučnej neurónovej siete je možné vytvoriť jednoduchý deepfake detektor. V rámci experimentov sme vyskúšali rôzne konfigurácie sietí, kde najlepšie výsledky boli dosiahnuté pomocou siete Vgg16. ResNet architektúra sa pri tejto úlohe neosvedčila, čo môže byť spôsobené veľkou komplexnosťou siete pre našu úlohu. Celkovo konfigurácie dosahovali úspešnosť okolo 78 %. V nasledujúcej kapitole prevedieme rovnaké experimenty, ale budeme využívať architektúru siamskej siete.

6.5 Model za použitia siamskej neurónovej siete

V tejto časti si popíšeme experimenty s rôznymi architektúrami neurónových sietí s cieľom vybrať najlepší. Ten následne použijeme pri vyhodnotení výsledkov algoritmov rozpoznania tváre. Oproti predchádzajúcemu experimentu 6.4 siamské siete berú na vstup minimálne dva vstupné snímky, referenciu a potenciálny deepfake snímok. Trénovanie prebiehalo na externom výpočetnom stroji, ktorý bol poskytnutí Metacentrom[1]. Trénovanie bolo nastavené na 35 epoch všetkých experimentov, po trénovaný modely dosahovali vysoké presnosti. Veľkosť *batchu* bola nastavená na 64. Do modelov vstupovali počas trénovania a validácie jednotlivé páry z celkového počtu 11200. 4900 párov obsahovalo deepfake a referenciu osoby, ktorá bola využitá pre tvorbu deepfake, 4900 párov boli referencia a deepfake inej osoby z nášho datasetu a zvyšok snímkov obsahoval referenciu a rovnakú referenciu, čiže istý obrázok reálnej osoby.

V rámci experimentov boli skúšané dve rôzne stratové funkcie, a to konkrétne *binary cross-entropy* a *contrastive loss*. *Binary cross-entropy* bola použitá z knižnice TensorFlow a *contrastive loss* sme si museli implementovať, na základe rovnice 2.4.

Nasledujúce tabuľky budú obsahovať výsledky rôznych architektúr na trénovacích a validačných dátach. Do modelu vstúpia dva vstupné obrázky. Jeden referenčný, druhý potenciálny deepfake. Model má za úlohu správne klasifikovať druhý obrázok (potencionálny deepfake). Ak je pôvod obrázku Deepfake a model správne určí, že sa o deepfake jedná, tak sa zvyšuje percentuálna hodnota. Ak bol snímok deepfake a model ho určí ako originálny snímok hodnota sa nezvyšuje. Obdobne je tomu je pre originálne snímky, len naopak, kde sa snažíme určiť v koľkých percentách prípadov model správne určí, že sa jedná o originálny snímok. Schéma modelu je možné vidieť na obrázku 5.4.

Prvý stĺpec tabuľky obsahuje názov architektúry neurónovej siete. Druhý stĺpec obsahuje váhu, teda či boli využité predtrénované váhy z datasetu ImageNet alebo nie. Tretí stĺpec obsahuje využitú stratovú funkciu, konkrétne buď *binary cross-entropy* alebo *contrastive loss*. Ďalšie stĺpce obsahujú výsledky vyhodnocovania pre jednotlivé deepfake programy – Roop, Face, Ghost Sim. Predposledný stĺpec obsahuje výsledky pre originálne snímky. Posledný stĺpec obsahuje priemer predchádzajúcich hodnôt pre porovnanie celkovej úspešnosti modelov medzi sebou. Na základe tejto hodnoty sa vyberie najlepší model, ktorý bude ďalej využitý pri vyhodnocovaní experimentov. Dáta pre experimenty boli využité rovnaké ako v predchádzajúcej sekcií 6.4. Nasledujúca tabuľka 6.3 obsahuje výsledky experimentov so sieťou Vgg.

Model	Váhy	Strat	Roop	Face	Ghost	Sim	Origin	Suma
Vgg16	ImageNet	Cross	82 %	62 %	60 %	44 %	72 %	64 %
		Contrast	81 %	67 %	64 %	56 %	70 %	67,6 %
Vgg16	None	Cross	87%	68 %	66 %	57 %	67 %	**69** %
		Contrast	85 %	67 %	64 %	53 %	69 %	67,6 %
Vgg19	ImageNet	Cross	77 %	65 %	52 %	67 %	71 %	66,4 %
		Contrast	62 %	48 %	34 %	49 %	87 %	56 %
Vgg19	None	Cross	80 %	67 %	52 %	68 %	70 %	67,4 %
		Contrast	75 %	61 %	54 %	68 %	73 %	66,2 %

Tabuľka 6.3: Výsledky experimentov siamskej siete so sieťou Vgg.

[1] https://metavo.metacentrum.cz/

Celkové výsledky zaznamenali horšie hodnoty oproti predchádzajúcemu experimentu. Jednotlivé stratové funkcie udávali podobné úspešnosti. Pri určovaný originálnych snímkov sa výstup zlepšil o mnoho oproti predchádzajúcim experimentom. Siamská sieť mala väčší problém správne určovať deepfake snímky. Najlepší model dokázal správne klasifikovať 69 % testovacieho datasetu. Využíval sieť Vgg16 bez predtrénovaných váh so stratovou funkciou *binary cross-entropy*. Pri experimentoch siamskej siete sme sa rozhodli nevyužiť sieť ResNet. Model sa natrénoval, ale nedosahoval dobré úspešnosti obdobne ako v predchádzajúcich experimentoch. Taktiež potreboval väčší výpočetný výkon čo spôsobilo, že sa nedal zakomponovať do výslednej aplikácie. V nasledujúcej časti vezmeme z každej sekcie najlepší model a vykonáme porovnanie s voľne dostupným deepfake detektorom BioID.

6.6 Porovnanie výsledkov s deepfake detektorom BioID

V tejto sekcií, vykonáme porovnanie nášho najlepšieho modelu za použitia jednej siete Vgg16. Cieľom experimentu je porovnať úspešnosť nášho modelu s komerčným deepfake detektorom od firmy BioID[2], ktorý je verejne dostupný.

Pre porovnanie výsledkov bolo náhodne vybraných 20 snímkov z testovacieho datasetu. Boli skúšané všetky kategórie snímkov - Originálne, Roop, Facefusion, Ghost a SimSwap. V nasledujúcej tabuľke 6.4 môžeme vidieť jednotlivé percentuálne úspešnosti správnej klasifikácie testovacieho datasetu.

Model	**Roop**	**FaceFusion**	**Ghost**	**SimSwap**	**Originál**	**Celkovo**
BioID	20 %	50 %	65 %	40 %	60 %	47 %
Vgg16	85 %	90 %	90 %	85 %	40 %	78 %

Tabuľka 6.4: Výsledky porovnania nášho modelu s BioID.

Do BioID detektoru bolo nutné posielať na vstup neorezané snímky, pri orezaných bola vyhodená chyba. Detektor od BioID primal na vstup jeden snímok na základe ktorého sa rozhodoval, či sa jedná o deepfake alebo nie, podobne ako náš Vgg16 model. Najťažšia úloha pre BioID detektor bola rozoznať snímky vygenerované deepfake programami SimSwap a Roop. Túto skutočnosť môžeme vidieť vo výslednej tabuľke 6.4. V rámci porovnania skončil deepfake detektor od BioID horšie ako náš natrénovaný model. Je zvláštne, že BioID mal tendenciu klasifikovať väčšinu snímkov, vytvorených pomocou programu SimSwap a Roop za originál a originálne pomerne často naopak za deepfake. Na výsledkoch môžeme vidieť, že sa nejedná o jednoduchú úlohu správne rozlíšiť deepfake od originálnych snímkov.

6.7 Experimenty algoritmov pre rozpoznanie tváre

V tejto časti budú popísané výsledky experimentov s algoritmami pre rozpoznanie tváre voči deepfakom. V rámci týchto experimentov sme si vytvorili dataset ktorý obsahoval dva snímky rovnakej osoby a jeden deepfake snímok ako môžeme názorne vidieť na nasledujúcom obrázku 6.8:

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

Obr. 6.8: Ukážka datasetu pre experimenty s algoritmami pre rozpoznanie tváre. Zľava referencia (*Real1*), druhý snímok rovnakej osoby ako je na referencií (*Real2*), deepfake predchádzajúceho snímku (*Fake*)

Pri tvorbe testovacieho datasetu sme vybrali dvojicu fotiek pre každú osobu z datasetu CFP-FP. Dokopy sme mali snímky 200 osôb, ku ktorým sme vytvorili jeden deepfake snímok, tak ako je zobrazené na ukážke na obrázku 6.8. Pre tvorbu deepfake bol využitý program FaceFusion, podkapitola 6.2.2. Následne sme pri algoritmov pre rozpoznanie tváre určovali presnosť, či je na oboch snímkoch rovnaká osoba. Ak na vstup prišli obrázky *Real1* (1. obrázok v 6.8) a *Real2* (2. obrázok v 6.8) výstup mal byť, že sa jedná o rovnakú osobu, a teda sa presnosť zvýšila. Ak výstup bol, že na obrázkoch nie sú rovnaké osoby presnosť zostala nezmenená. Podobne tomu bolo keď na vstup prišiel deepfake. Pri deepfake sme robili dva experimenty, kde na vstup išla *Real1* a *Fake* (3. obrázok v 6.8) alebo *Real2* a *Fake*. Princíp bol rovnaký, keď algoritmus určil, že sa nejedná o rovnakú osobu presnosť sa zvýšila, ak tomu bolo naopak tak zostala nezmenená. Pri experimentoch sme skúšali dve rôzne techniky výpočtu dištančného vektora a to kosínusovú a Euklidovskú L2. Prvá tabuľka 6.5 obsahuje výsledky Euklidovskej L2 vzdialenosti a druhá tabuľka 6.6 obsahuje výsledky kosínovskej vzdialenosti. V zvlášť tabuľke 6.7 sme vykonali rovnaký experiment s naším najlepším modelom siamskej siete, aby sme porovnali výsledky s algoritmami pre rozpoznanie tváre. Bližší popis k metrikám v podkapitole 4.5.6. V tabuľke sú taktiež uvedené prahové hodnoty pre jednotlivé algoritmy pre rozpoznanie tváre, na základe ktorých sa určilo či sa jedná o rovnakú osobu alebo nie. Prahové hodnoty sú určené pre najlepšiu presnosť jednotlivých algoritmov.

Model a prah	**True Positive**	**True Negative**		**Presnosť**
	Real1 a Real2	**Real1 a Fake**	**Real2 a Fake**	
FaceNet – 0,8	74,5 %	100 %	100 %	91,5 %
DeepFace – 0,64	64 %	58,5 %	24,5 %	49 %
ArcFace – 1,13	95,5 %	99 %	94 %	96,2 %
MagFace – 1,1	100 %	100 %	100 %	100 %

Tabuľka 6.5: Výsledky experimentov algoritmov pre rozpoznanie tváre za použitia Euklidovskej L2 vzdialenosti.

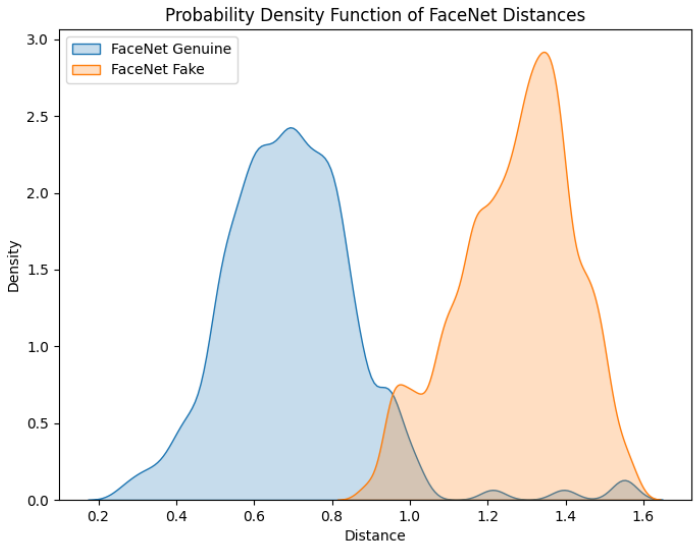
Model a prah	**True Positive**	**True Negative**		**Presnosť**
	Real1 a Real2	**Real1 a Fake**	**Real2 a Fake**	
FaceNet – 0,4	88,5 %	99,5 %	98,5 %	95,5 %
DeepFace – 0,23	72,5 %	46 %	18,5 %	45,6 %
ArcFace – 0,68	96,5 %	94 %	88 %	92,8 %

Tabuľka 6.6: Výsledky experimentov algoritmov pre rozpoznanie tváre za použitia Cos vzdialenosti.

Model	True Positive	True Negative		Presnosť
	Real1 a Real2	Real1 a Fake	Real2 a Fake	
Siamese	89 %	56,5 %	91,5 %	79 %

Tabuľka 6.7: Výsledky experimentov siamskej siete pre porovnanie výsledkov voči algoritmom pre rozpoznanie tváre.

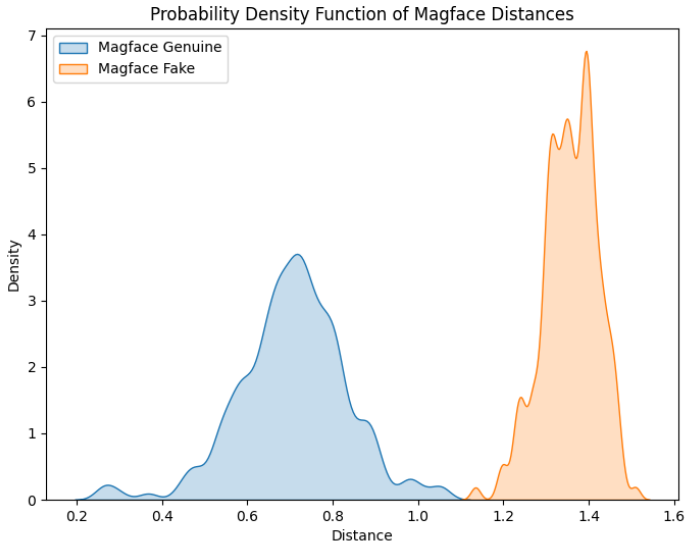
Z výsledkov môžeme vidieť, že väčšina algoritmov dosahovala dobré výsledky. Najhoršie skončil DeepFace a potom náš model siamskej siete. Náš model nepoužíval rôzne metriky na vyhodnocovanie experimentov, preto je uvedený zvlášť. Najlepší výsledok dosiahol MagFace s presnosťou 100 %. Euklidovská vzdialenosť dosahovala všeobecne lepšie výsledky oproti kosínovskej, jedinou výnimkou bol FaceNet. MagFace má experimenty iba pomocou Euklidovskej vzdialenosti, lebo v použitej knižnici kosínus nebol dostupný. Jednotlivé algoritmy rozpoznania tváre sme vyniesli na grafy hustoty pravdepodobnosti pre Euklidovskú L2 vzdialenosť. Môžeme vidieť na nasledujúcich obrázkov grafov. Pre prehľadnosť následujúce grafy budú pre Euklidovskú L2 vzdialenosť. Kosínus dosahoval podobné výsledky, ako môžeme vidieť v predchádzajúcich tabuľkách.



Obr. 6.9: Funkcia hustoty pravdepodobnosti v jednom grafe za použitia L2 vzdialenosti - FaceNet.

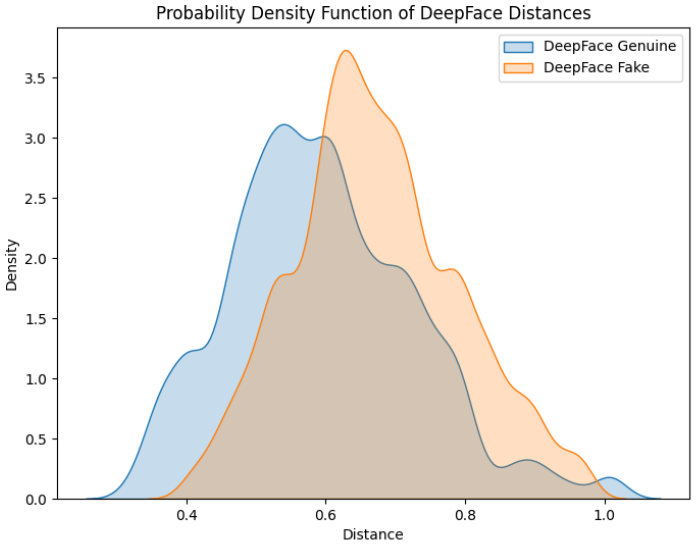
Na obrázku 6.9 môžeme vidieť rozdielne stredové hodnoty pre jednotlivé hodnoty. Modrá krivka znázorňuje hodnoty pre originálne snímky a oranžová pre deepfake. Väčšina snímkov bola zvlášť klasifikovaná, ale môžeme vidieť prekryv medzi hodnotami 0,8 až 1,0, kde snímky boli určované aj ako originálne, ale aj ako deepfake. Niektoré originálne snímky boli určené s veľkou vzdialenosťou, ktorá bola typickejšia pre deepfake snímky ako môžeme vidieť malé vlnky v modrom grafe. Prahová hodnota pre rozlíšenie, či sa jedná o rovnaké alebo rôzne osoby má FaceNet nastavenú na 0,8 pre Euklidovskú L2 vzdialenosť (obrázok 6.9) a pre kosínovskú 0,4.

Na nasledujúcom grafe 6.10 si ukážeme výstup MagFace, ktorý dosiahol najlepší výsledok v experimentoch:



Obr. 6.10: Funkcia hustoty pravdepodobnosti v jednom grafe za použitia L2 vzdialenosti - MagFace.

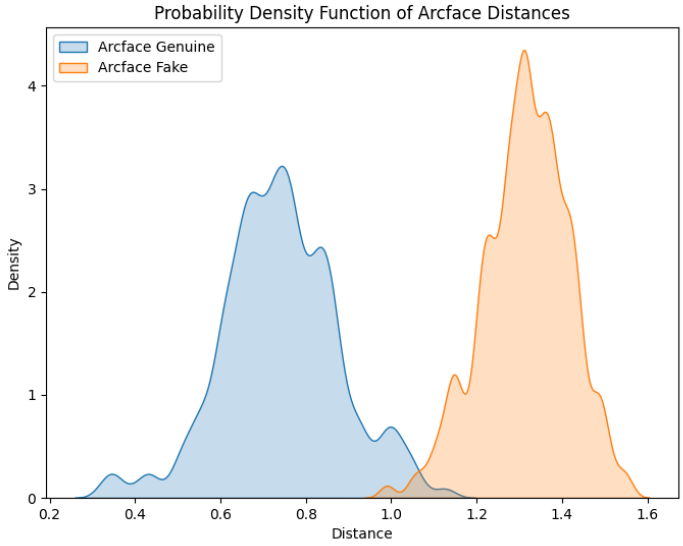
Na grafe 6.10 môžeme vidieť, že výstupy pre originály nemajú prienik hodnôt s výstupmi pre deepfake. Prahová hodnota pre MagFace je 1,1 pre Euklidovskú L2 vzdialenosť. Na nasledujúcom grafe 6.11 si môžeme výstupy porovnať s najhorším výsledok v rámci experimentov, ktorý bol dosiahnutý algoritmom DeepFace.



Obr. 6.11: Funkcia hustoty pravdepodobnosti v jednom grafe za použitia L2 vzdialenosti - DeepFace.

Na grafe 6.11 môžeme vidieť veľký prienik hodnôt. DeepFace má prahovú hodnotu nastavenú na 0,64 pri Euklidovskej L2 vzdialenosti a 0,23 pri kosínusovej. DeepFace nevedel

spoľahlivo určiť, či sa na vstupných snímkoch nachádza rovnaká osoba. Tento fakt je viditeľný na grafe, keďže sa tam nachádza vysoký prienik hodnôt. DeepFace je najstarší z hodnotených algoritmov a od jeho vzniku pribudli mnohé ďalšie algoritmy pre rozpoznanie tváre, ktoré ho prekonávajú v rôznych *benchmarkoch*. Na základe výsledkov sa jeho použitie neosvedčilo. Na poslednom obrázku 6.12 môžeme vidieť graf posledného testovacieho algoritmu ArcFace:



Obr. 6.12: Funkcia hustoty pravdepodobnosti v jednom grafe za použitia L2 vzdialenosti - ArcFace.

Na poslednom obrázku 6.12 môžeme vidieť graf ArcFace. ArcFace má prahovú hodnotu nastavenú na 1,13 pri Euklidovskej L2 vzdialenosti a 0,68 pri kosínusovej. ArcFace dosiahol druhý najlepší výsledok. Na grafe môžeme vidieť, že prienik hodnôt nebol vysoký.

Z experimentov vyplýva odolnosť určitých algoritmov pre rozpoznanie tváre voči deepfake. Najlepšie výsledky boli za použitia MagFace, kde presnosť dosahovala 100%. Najhoršie výsledky boli za použitia DeepFace, kde model väčšinu snímkov vyhodnocoval na základe vzdialenosti ako deepfake. Náš model siamskej neurónovej siete bol schopný určovať či sa jedná o deepfake alebo nie, avšak výsledky boli slabšie oproti algoritmom rozpoznania tváre. Pri originálnych snímkoch na vstupe dával pomerne správne výsledky vo viac ako 91% prípadov. Podobný trend bol aj keď na vstup išla referencia a deepfake referencie. Akonáhle sme na vstup poslali inú referenciu ako bol deepfake snímok vytvorený, tak presnosť rapidne klesla. Poradie od najlepšieho modelu po najhorší na základe výsledkov je následovné: MagFace, ArcFace, FaceNet, Siamská sieť, DeepFace.

6.8 Zhrnutie výsledkov

V experimentálnej časti sme demonštrovali a porovnali jednotlivé konfigurácie vytvorených modelov. Jednotlivé modely boli schopné klasifikovať, či sa jedná o deepfake alebo nie, s rôznymi úspešnosťami. Z jednotlivých experimentov lepšie výsledky dosahovala sieť Vgg16 oproti Vgg19. Model, ktorý bral na vstup jeden vstupný snímok dosahoval úspešnosť okolo 80%. Pri siamskej sieti úspešnosť klesla pod 70%. Po zhrnutí experimentov našich vytvorených modelov neuronových sietí sme vykonali porovnanie výsledkov s komerčným detektorom od firmy BioID. Ten dosahoval horšie výsledky oproti nášmu modelu. Na výsledkoch môžeme vidieť, že nie je ľahké spoľahlivo určiť, či vstupný snímok je alebo nie je deepfake. Následne sme vykonali experimenty s algoritmami pre rozpoznanie tváre, pričom sme ich výsledky porovnávali s našim najlepším modelom siamskej siete. Pri experimentoch sme vyskúšali dva rôzne spôsoby výpočtu dištančného vektora. Z výsledkov sa osvedčilo využitie Euklidovskej L2 vzdialenosti oproti kosínovskej. Algoritmy rozpoznania tváre dosahovali lepšie výsledky ako náš model siamskej siete s výnimkou DeepFace. Najlepší výsledok bol dosiahnutý s MagFace, ktorý na testovacom datasete dosiahol presnosť 100 %. Následoval ArcFace, FaceNet a posledný spomínaný DeepFace. Všetky algoritmy okrem DeepFace boli do určitej miery odolné voči deepfake.

Kapitola 7

Záver

Cieľom práce bolo vytvoriť aplikáciu, ktorá bude kombinovať rozpoznávanie a odolnosť voči deepfake snímkom. V úvode práce bola vysvetlená technológia za neurónovými sieťami. Ďalej bola vysvetlená technológia deepfake s jej rizikami a dôvodmi pre tvorbu nástrojov voči tejto hrozbe. V závere teoretickej časti práce popisujeme algoritmy rozpoznania tváre.

Práca si ako prvé vyžadovala vytvoriť vlastný dataset. Ten bol vytvorený za použitia dvoch datasetov LFW a CelebA HQ pre experimenty s neurónovými sieťami. Pre experimenty algoritmov pre rozpoznanie tváre bol využitý dataset CFP-FP. Vybraná časť snímkov slúžila ako vstup do 4 voľne dostupných deepfake programov. Dataset bol následne upravený, aby obsahoval iba tváre osôb. V kapitole Návrh a implementácia je popísaný návrh aplikácie a do podrobna popísaný proces tvorby modelov neurónových sietí.

V kapitole Experimenty a výsledky popisujeme tvorbu datasetu. Následne skúšame rôzne konfigurácie neurónových sietí za účelom vybrať najlepší pre ďalšie experimenty. Prvý experiment sa sústredil na sieť, ktorá brala na vstupe jeden snímok a na jeho základe rozhodla, či sa jedná o deepfake alebo nie. Druhý experiment pozostával z architektúry siamskej neurónovej sieti, ktorá na vstupe berie dva vstupné snímky. Najlepšie výsledky dosiahla architektúra Vgg16.

Model s jedným vstupným snímkom bol následne porovnaný s komerčným online deepfake detektorom BioID. Náš model dosahoval lepšie výsledky. Ďalej sme porovnali model siamskej siete s výsledkami algoritmov rozpoznania tváre a určovali ich odolnosť. Testované algoritmy rozpoznania tváre boli ArcFace, DeepFace, FaceNet a MagFace. Z testovaných algoritmov Deepface dopadol najhoršie. Druhý najhorší výsledok dosiahol náš model siamskej siete. Ostatné algoritmy rozpoznania tváre boli na tom omnoho lepšie, pričom MagFace dokázal správne rozlíšiť celý testovací dataset, ktorý obsahoval kolekciu obrázkov 200 ľudí, s presnosťou 100 %. Na základe výsledkov sa nám podarilo dokázať odolnosť algoritmov pre rozpoznanie tváre voči deepfake snímkov, s výnimkou DeepFace.

V budúcnu by bolo možné prácu rozšíriť o zabudovaný orezávač fotografií, ktorý by fotku upravil, aby obsahovala iba tvár osoby. Taktiež by bolo možné pridať užívateľské rozhranie alebo vyskúšať iné architektúry neurónových sietí.

Literatúra

[1] Abadi, M., Agarwal, A., Barham, P. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015 [cit. 2024-03-23]. Software available from tensorflow.org. Dostupné z: https://www.tensorflow.org/.

[2] Agarwal, A. a Ratha, N. Chapter 8 - Manipulating faces for identity theft via morphing and deepfake: Digital privacy. In: *Deep Learning*. 2023, s. 223–241 [cit. 2024-03-20]. ISSN 0169-7161. Dostupné z: https://www.sciencedirect.com/science/article/pii/S016971612200058X.

[3] All, M. *Introduction to Activation Functions in Neural Networks* [online]. 2023 [cit. 2024-04-30]. Dostupné z: https://www.datacamp.com/tutorial/introduction-to-activation-functions-in-neural-networks.

[4] amazon, A. *What is a Neural Network?* [online]. 2024 [cit. 2024-03-20]. Dostupné z: https://aws.amazon.com/what-is/neural-network/.

[5] Angel Vizoso, M. V.-A. a López García, X. *Fighting Deepfakes: Media and Internet Giants' Converging and Diverging Strategies Against Hi-Tech Misinformation*. 2021 [cit. 2024-01-25]. ISSN 2183-2439. Dostupné z: https://www.cogitatiopress.com/mediaandcommunication/article/view/3494/1992.

[6] Artguru. *Free Online AI Face Swap*. 2024 [cit. 2024-04-10]. Dostupné z: https://www.artguru.ai/blogs/face-swap/.

[7] Bangar, S. *VGG-Net Architecture Explained*. 2022 [cit. 2024-03-27]. Dostupné z: https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f.

[8] Barney, N. *Face detection* [online]. 2023 [cit. 2024-03-27]. Dostupné z: https://www.techtarget.com/searchenterpriseai/definition/face-detection.

[9] Bekuzarov, M. *Losses explained: Contrastive Loss*. 2020 [cit. 2024-04-1]. Dostupné z: https://medium.com/@maksym.bekuzarov/losses-explained-contrastive-loss-f8f57fe32246.

[10] Benhur, S. *A Friendly Introduction to Siamese Networks* [online]. 2022 [cit. 2024-03-27]. Dostupné z: https://builtin.com/machine-learning/siamese-network.

[11] Bonettini, N., Cannas, E. D., Mandelli, S., Bondi, L., Bestagini, P. et al. *Video Face Manipulation Detection Through Ensemble of CNNs*. 2020 [cit. 2024-05-01].

[12] Burgess, S. *Ukraine war: Deepfake video of Zelenskyy telling Ukrainians to 'lay down arms' debunked* [online]. 2022 [cit. 2024-03-26]. Dostupné z:

https://news.sky.com/story/ukraine-war-deepfake-video-of-zelenskyy-telling-ukrainians-to-lay-down-arms-debunked-12567789.

[13] Cai, Z., Ghosh, S., Stefanov, K., Dhall, A., Cai, J. et al. *MARLIN: Masked Autoencoder for facial video Representation LearnINg*. 2023 [cit. 2024-05-01].

[14] Centre, N. C. S. *Biometric recognition and authentication systems* [online]. 2024 [cit. 2024-06-15]. Dostupné z: https://www.ncsc.gov.uk/collection/biometrics/face.

[15] Chandrasekara, N. *Deepfake Detection*. 2024 [cit. 2024-04-8]. Dostupné z: https://medium.com/@nilushihansika531/deepfake-detection-7c73b0acc5a7.

[16] Chen, R., Chen, X., Ni, B. a Ge, Y. SimSwap: An Efficient Framework For High Fidelity Face Swapping. In: *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, Október 2020 [cit. 2024-05-03]. MM '20. DOI: 10.1145/3394171.3413630. Dostupné z: http://dx.doi.org/10.1145/3394171.3413630.

[17] Chollet, F. et al. *Keras*. 2015 [cit. 2024-03-23]. Dostupné z: https://keras.io.

[18] Databrics.com. *Dataset*. 2024 [cit. 2024-03-21]. Dostupné z: https://www.databricks.com/glossary/what-is-dataset.

[19] DataScientest. *You Only Look Once (YOLO): What is it?* [online]. 2024 [cit. 2024-06-14]. Dostupné z: https://datascientest.com/en/you-only-look-once-yolo-what-is-it.

[20] Deng, J., Guo, J., Ververas, E., Kotsia, I. a Zafeiriou, S. RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, s. 5202–5211 [cit. 2024-06-14]. DOI: 10.1109/CVPR42600.2020.00525.

[21] Deng, J., Guo, J. a Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *CoRR*. 2018, abs/1801.07698, [cit. 2024-06-15]. Dostupné z: http://arxiv.org/abs/1801.07698.

[22] Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I. et al. RetinaFace: Single-stage Dense Face Localisation in the Wild. *CoRR*. 2019, abs/1905.00641, [cit. 2024-06-14]. Dostupné z: http://arxiv.org/abs/1905.00641.

[23] Dhamecha, H. *A Detailed Explanation of GAN with Implementation Using Tensorflow and Keras* [online]. 2021 [cit. 2024-05-01]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/06/a-detailed-explanation-of-gan-with-implementation-using-tensorflow-and-keras/.

[24] Dwivedi, D. *Face Recognition for Beginners* [online]. 2018 [cit. 2024-04-30]. Dostupné z: https://towardsdatascience.com/face-recognition-for-beginners-a7a9bd5eb5c2.

[25] Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters*. 2006, zv. 27, č. 8, s. 861–874, [cit. 2024-07-09]. DOI: https://doi.org/10.1016/j.patrec.2005.10.010. ISSN 0167-8655. ROC Analysis in Pattern Recognition. Dostupné z: https://www.sciencedirect.com/science/article/pii/S016786550500303X.

[26] Fei Fei, L., Deng, J., Russakovsky, O., Berg, A. a Li, K. *ImageNet*. 2020 [cit. 2024-03-28]. Dostupné z: https://www.image-net.org/index.php.

[27] Georgios Nanos, M. A. *How Do Siamese Networks Work in Image Recognition?* [online]. 2024 [cit. 2024-03-27]. Dostupné z: https://www.baeldung.com/cs/siamese-networks.

[28] Goldmann, T. *FaceAIKit* [online]. 2024 [cit. 2024-06-23]. Dostupné z: https://github.com/tgoldmann/face-ai-kit.

[29] Goodfellow, I. J., Pouget Abadie, J., Mirza, M., Xu, B., Warde Farley, D. et al. *Generative Adversarial Networks*. 2014 [cit. 2024-01-25]. Dostupné z: https://arxiv.org/abs/1406.2661.

[30] Groshev, A., Maltseva, A., Chesakov, D., Kuznetsov, A. a Dimitrov, D. GHOST—A New Face Swap Approach for Image and Video Domains. *IEEE Access*. 2022, zv. 10, s. 83452–83462, [cit. 2024-05-03]. DOI: 10.1109/ACCESS.2022.3196668.

[31] Gupta, S. *Is L2-Norm = Euclidean Distance?* [online]. 2022 [cit. 2024-07-02]. Dostupné z: https://medium.com/@guptasaurav/is-l2-norm-euclidean-distance-a9c04be0b3ca.

[32] H.L., S. *2D Convolution in Image Processing* [online]. 2018 [cit. 2024-04-29]. Dostupné z: https://www.allaboutcircuits.com/technical-articles/two-dimensional-convolution-in-image-processing/.

[33] 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, 2007 [cit. 2024-03-21].

[34] I, A. *Architecture: A Deep Dive into Residual Neural Network* [online]. 2023 [cit. 2024-05-01]. Dostupné z: https://medium.com/@ibtedaazeem/understanding-resnet-architecture-a-deep-dive-into-residual-neural-network-2c792e6537a9.

[35] IBM. *Convolutional Neural networks* [online]. 2023 [cit. 2024-01-25]. Dostupné z: https://www.ibm.com/topics/convolutional-neural-networks.

[36] IBM. *Neural networks* [online]. 2023 [cit. 2024-01-25]. Dostupné z: https://www.ibm.com/topics/neural-networks.

[37] Innovatrics. *Facial Recognition Technology* [online]. 2024 [cit. 2024-06-15]. Dostupné z: https://www.innovatrics.com/facial-recognition-technology/.

[38] Ippolito, P. P. *Introduction to Autoencoders: From The Basics to Advanced Applications in PyTorch* [online]. 2023 [cit. 2024-04-29]. Dostupné z: https://www.datacamp.com/tutorial/introduction-to-autoencoders.

[39] KALRA, K. *FACE RECOGNITION* [online]. 2023 [cit. 2024-03-27]. Dostupné z: https://medium.com/@khwabkalra1/face-recognition-e45aff329fba.

[40] Karmarkar, T. *Region Proposal Network (RPN) — Backbone of Faster R-CNN* [online]. 2018 [cit. 2024-07-15]. Dostupné z: https://medium.com/@codeplumber/region-proposal-network-rpn-backbone-of-faster-r-cnn-4a744a38d7f9.

[41] KASPERSKY. *What is Facial Recognition – Definition and Explanation* [online]. 2024 [cit. 2024-01-25]. Dostupné z: https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition.

[42] KERAS. *Adam* [online]. 2024 [cit. 2024-05-03]. Dostupné z: https://keras.io/api/optimizers/adam/.

[43] KNOWLEDGENILE. *Applications of Deepfake Technology: Positives and Dangers* [online]. 2023 [cit. 2024-01-25]. Dostupné z: https://www.knowledgenile.com/blogs/applications-of-deepfake-technology-positives-and-dangers.

[44] KOCH, G., ZEMEL, R. a SALAKHUTDINOV, R. Siamese Neural Networks for One-shot Image Recognition. In:. 2015 [cit. 2024-03-23].

[45] KORSHUNOVA, I., SHI, W., DAMBRE, J. a THEIS, L. *Fast Face-swap Using Convolutional Neural Networks*. 2017 [cit. 2024-01-25]. Dostupné z: https://arxiv.org/pdf/1611.09577v2.pdf.

[46] KUMAR, P. *Siamese Networks Introduction and Implementation*. 2023 [cit. 2024-03-27]. Dostupné z: https://medium.com/@prabhattgs12345789/siamese-neural-network-enhancing-ai-capabilities-with-pairwise-comparisons-4f00e2dd8256.

[47] LEBLANC, F. *Autocrop*. 2022 [cit. 2024-03-22]. Dostupné z: https://github.com/leblancfg/autocrop.

[48] LI, L., BAO, J., YANG, H., CHEN, D. a WEN, F. *FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping*. 2020 [cit. 2024-05-03].

[49] LIU, Z., LUO, P., WANG, X. a TANG, X. Deep Learning Face Attributes in the Wild. In: *Proceedings of International Conference on Computer Vision (ICCV)*. December 2015 [cit. 2024-04-29].

[50] LUTKEVICH, B. *Pros and cons of facial recognition* [online]. 2024 [cit. 2024-03-20]. Dostupné z: https://www.techtarget.com/whatis/feature/Pros-and-cons-of-facial-recognition.

[51] MALLICK, S. *Understanding Convolutional Neural Network (CNN): A Complete Guide* [online]. 2024 [cit. 2024-01-25]. Dostupné z: https://learnopencv.com/understanding-convolutional-neural-networks-cnn/.

[52] MARQUÉS, I. *Face Recognition Algorithms*. 2010 [cit. 2024-01-25]. Dostupné z: https://www.ehu.eus/ccwintco/uploads/d/d2/PFC-IonMarqu%C3%A9s.pdf.

[53] MELEKHOV, I., KANNALA, J. a RAHTU, E. Siamese network features for image matching. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. 2016, s. 378–383 [cit. 2024-04-29]. DOI: 10.1109/ICPR.2016.7899663.

[54] MENG, Q., ZHAO, S., HUANG, Z. a ZHOU, F. MagFace: A Universal Representation for Face Recognition and Quality Assessment. *CoRR*. 2021, abs/2103.06627, [cit. 2024-06-23]. Dostupné z: https://arxiv.org/abs/2103.06627.

[55] MUKHERJEE, S. *The Annotated ResNet-50* [online]. 2022 [cit. 2024-07-15]. Dostupné z: https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758.

[56] N, G. V. *GAN — Generative Adversarial Network*. 2024 [cit. 2024-04-10]. Dostupné z: https://medium.com/@ngreeshmavamsi/gan-generative-adversarial-network-f1dcbefa43ed.

[57] Nguyen, T. T., Nguyen, Q. V. H., Nguyen, D. T., Nguyen, D. T., Huynh The, T. et al. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*. 2022, zv. 223, s. 103525. DOI: https://doi.org/10.1016/j.cviu.2022.103525. ISSN 1077-3142. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1077314222001114.

[58] Patterson, D. *Deepfakes for good? How synthetic media is transforming business* [online]. 2023 [cit. 2024-01-25]. Dostupné z: https://techinformed.com/deepfakes-for-good-how-synthetic-media-is-transforming-business/.

[59] Patterson, D. *How synthetic media can upend reality* [online]. 2023 [cit. 2024-01-25]. Dostupné z: https://techinformed.com/how-synthetic-media-can-upend-reality/.

[60] Payne, L. *Deepfake* [online]. 2024 [cit. 2024-01-25]. Dostupné z: https://www.britannica.com/technology/deepfake.

[61] Philips, H. *A Simple Introduction to Softmax* [online]. 2023 [cit. 2024-04-29]. Dostupné z: https://medium.com/@hunter-j-phillips/a-simple-introduction-to-softmax-287712d69bac.

[62] Pra, M. D. *Generative Adversarial Networks* [online]. 2023 [cit. 2024-05-01]. Dostupné z: https://medium.com/@marcodelpra/generative-adversarial-networks-dba10e1b4424.

[63] RECFACES. *Understanding facial recognition algorithms* [online]. 2024 [cit. 2024-04-30]. Dostupné z: https://recfaces.com/articles/facial-recognition-algorithms.

[64] Redmon, J., Divvala, S. K., Girshick, R. B. a Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *CoRR*. 2015, abs/1506.02640, [cit. 2024-06-14]. Dostupné z: http://arxiv.org/abs/1506.02640.

[65] Ruhs, H. *FaceFusion*. 2024 [cit. 2024-03-22]. Dostupné z: https://github.com/facefusion/facefusion.

[66] Ruiter, A. de. The Distinct Wrong of Deepfakes. *Philosophy & Technology*. 2021, [cit. 2024-01-25]. Dostupné z: https://doi.org/10.1007/s13347-021-00459-2.

[67] Schreiner, M. *Deepfakes: HOW it all began - and where it could lead us* [online]. 2022 [cit. 2024-01-25]. Dostupné z: https://the-decoder.com/history-of-deepfakes/.

[68] Schroff, F., Kalenichenko, D. a Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. *CoRR*. 2015, abs/1503.03832, [cit. 2024-06-15]. Dostupné z: http://arxiv.org/abs/1503.03832.

[69] Sciforce. *Face Detection Explained: State-of-the-Art Methods and Best Tools* [online]. 2021 [cit. 2024-03-27]. Dostupné z: https://medium.com/sciforce/face-detection-explained-state-of-the-art-methods-and-best-tools-f730fca16294.

[70] Serengil, S. I. a Ozpinar, A. LightFace: A Hybrid Deep Face Recognition Framework. In: *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. 2020, s. 1–5 [cit. 2024-06-15]. DOI: 10.1109/ASYU50717.2020.9259802.

[71] Simonyan, K. a Zisserman, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015 [cit. 2024-03-28]. Dostupné z: https://arxiv.org/abs/1409.1556.

[72] Simplilearn. *An Ultimate Tutorial to Neural Networks* [online]. 2024 [cit. 2024-04-8]. Dostupné z: https://www.simplilearn.com/tutorials/deep-learning-tutorial/neural-network.

[73] Simpson, B. N. J. . J. The tensions of deepfakes, Information, Communication & Society. *Information, Communication & Society*. Routledge. 2023, zv. 0, č. 0, s. 1–15, [cit. 2024-01-25]. DOI: 10.1080/1369118X.2023.2234980. Dostupné z: https://www.tandfonline.com/doi/full/10.1080/1369118X.2023.2234980.

[74] Somdev Sangwan, H. R. *Roop*. 2023 [cit. 2024-03-22]. Dostupné z: https://github.com/s0md3v/roop.

[75] Taigman, Y., Yang, M., Ranzato, M. a Wolf, L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, s. 1701–1708 [cit. 2024-06-23]. DOI: 10.1109/CVPR.2014.220.

[76] Team, D. A. *An Overview of State of the Art (SOTA) DNNs* [online]. 2023 [cit. 2024-05-03]. Dostupné z: https://deci.ai/blog/sota-dnns-overview/.

[77] Team, G. L. *Face Detection using Viola Jones Algorithm* [online]. 2023 [cit. 2024-04-29]. Dostupné z: https://www.mygreatlearning.com/blog/viola-jones-algorithm/.

[78] Turgay, G. *Two Most Common Similarity Metrics* [online]. 2022 [cit. 2024-07-02]. Dostupné z: https://towardsdatascience.com/two-most-common-similarity-metrics-39c37f3fe14d#ef62.

[79] Ty, U. *Face Embedding and what you need to know* [online]. 2023 [cit. 2024-04-30]. Dostupné z: https://uysim.medium.com/face-embedding-and-what-you-need-to-know-a623c7111b5.

[80] Vasilev, I. *Siamese networks* [online]. 2019 [cit. 2024-05-04]. Dostupné z: https://www.oreilly.com/library/view/advanced-deep-learning/9781789956177/d06332c0-0c39-413b-951e-51b95005eeda.xhtml.

[81] Vidhya, A. *Binary Cross Entropy/Log Loss for Binary Classification* [online]. 2021 [cit. 2024-06-21]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/.

[82] Viso.ai. *DeepFace: A Popular Open Source Facial Recognition Library* [online]. 2024 [cit. 2024-06-23]. Dostupné z: https://viso.ai/computer-vision/deepface/.

[83] Weisstein, E. W. *"L2-Norm."From MathWorld–A Wolfram Web Resource* [online]. 2024 [cit. 2024-07-02]. Dostupné z: https://mathworld.wolfram.com/L2-Norm.html.

[84] Wiki, A. *Accuracy and Loss* [online]. 2020 [cit. 2024-05-04]. Dostupné z: https://machine-learning.paperspace.com/wiki/accuracy-and-loss.

[85] Wikipedia. *Artificial neuron*. 2024 [cit. 2024-07-15]. Dostupné z: https://en.wikipedia.org/wiki/Artificial_neuron.

[86] Wikipedia. *Deepfake*. 2024 [cit. 2024-07-15]. Dostupné z: https://cs.wikipedia.org/wiki/Deepfake.

[87] Wikipedia. *Umělá neuronová síť*. 2024 [cit. 2024-01-25]. Dostupné z: https://cs.wikipedia.org/wiki/Um%C4%9Bl%C3%A1_neuronov%C3%A1_s%C3%AD%C5%A5.

[88] Zaragoza, A. *Facial Recognition: how it works and its safety* [online]. 2023 [cit. 2024-06-15]. Dostupné z: https://www.signicat.com/blog/face-recognition.

Príloha A

Obsah pamäťového média

Na priloženom pamäťovom médiu sa nachádza nasledujúca stromová štruktúra:

```
/
├── application ..... Zdrojové súbory aplikácie
│   └── app.py ..... Skript výslednej aplikácie
├── autocrop ..... Autocrop knižnica na orezávanie snímkov podľa tváre
├── dataset ..... Položka so všetkými snímkami použitých v práci
├── face_rec_experiments ..... Experimenty s algoritmami pre rozpoznanie tváre
├── metacentrum ..... Súbory pre prácu na Metacentre
├── neural_env ..... Prostredie obsahujúce python skripty
│   ├── meta-graphs.ipynb ..... Skript pre tvorbu grafov z výstupu Metacentra
│   ├── neural_start.ipynb ..... Skript pre siete s jedným vstupom
│   └── siamese.ipynb ..... Skript pre siamské siete
└── thesis_source ..... Zdrojové súbory LaTeX
```

Niektoré priečinky obsahujú readme.md, aby poskytli bližšie informácie o obsahu a taktiež jednotlivé skripty vyobrazené v adresárovom strome obsahujú komentáre.