VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCIA ŽIVOTNOSTI TVÁRE POMOCOU 2D KAMERY

FACE LIVENESS DETECTION USING 2D CAMERA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **ONDREJ VALO**
AUTHOR

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

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav inteligentních systémů (UITS) Akademický rok 2020/2021

Zadání bakalářské práce

Student: **Valo Ondrej**
Program: Informační technologie
Název: **Detekce živosti obličeje pomocí 2D kamery**
Face Liveness Detection Using a 2D Camera
Kategorie: Umělá inteligence

Zadání:

1. Seznamte se s problematikou prezenčních útoků s využitím 2D a 3D podvrhů obličeje.
2. Sumarizujte informace o dostupných řešeních pro detekci živosti obličeje.
3. Navrhněte algoritmus, který bude na základě dat z 2D kamery určovat, zdali se jedná o podvrh obličeje či nikoliv. Oblast obličeje můžete detekovat s využitím existujících algoritmů.
4. Navržené řešení implementujte v programovacím jazyce Python nebo C++.
5. Proveďte experimenty zaměřené na zhodnocení úspěšnosti detekce podvrhu a zhodnoťte robustnost vašeho řešení.

Literatura:

- DAMER, Naser, et al. Practical View on Face Presentation Attack Detection. In: *BMVC*. 2016.
- BENLAMOUDI, Azeddine, et al. Face spoofing detection from single images using active shape models with stasm and lbp. In: *Proceeding of the Troisime conference internationale sur la vision artificielle CVA*. 2015. p. 31.
- KIM, Sooyeon; BAN, Yuseok; LEE, Sangyoun. Face liveness detection using defocus. *Sensors*, 2015, 15.1: 1537-1563.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2

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

Vedoucí práce: **Goldmann Tomáš, Ing.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 11. listopadu 2020

Abstrakt

Rozpoznávanie tváre je jednou z najviac spoločensky akceptovaných foriem biometrického rozpoznávania. Nedávna dostupnosť veľmi presných a efektívnych algoritmov rozpoznávania tváre ponecháva zraniteľnosť voči prezentačným útokom ako hlavnú výzvu pre riešenia rozpoznávania tváre. Táto práca sa zaoberá vysvetlením problematiky týkajúcej sa s detekciou živosti tváre, ktorá pomôže pochopiť rôzne možnosti útokov a ich vzťah k existujúcim riešeniam. A implementáciu algoritmu, ktorý na základe videa rozoznáva živosť tváre.

Abstract

Facial recognition is one of the most socially accepted forms of biometric recognition. The recent availability of highly accurate and efficient face recognition algorithms leaves vulnerability to presentation attacks as a major challenge for face recognition solutions. This work deals with the explanation of the issues related to the detection of facial liveliness, which will help to understand the various possibilities of attack and their relationship to existing solutions. And the implementation of an algorithm that recognizes the liveliness of the face based on videos.

Kľúčové slová

životnosť tváre, konvolučná neurónová sieť, rozpoznávanie tváre, detekcia tváre, CNN, hlboké učenie

Keywords

liveness detection, convolutional neural network, face recognition,facial detection, CNN, deep learning

Citácia

VALO, Ondrej. *Detekcia životnosti tváre pomocou 2D kamery.* Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Goldmann,

Detekcia životnosti tváre pomocou 2D kamery

Prehlásenie

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

.....
Ondrej Valo
12. mája 2021

Poďakovanie

Chcel by som poďakovať Ing. Tomášovi Goldmannovi za pomoc a odborné rady pri riešení tejto bakalárskej práce. Zároveň by som chcel poďakovať mojej rodine, ktorá ma podporovala počas celého štúdia.

Obsah

Kapitola 1

Úvod

Biometria a biometrické systémy sa čoraz viac dostávajú do popredia ľudského záujmu. V poslednej dobe tvár viac a viac nahradzuje osobné identifikačné kódy a iné rôzne spôsoby overovania totožnosti vďaka jej výrazným znakom pre identifikáciu osoby. Neexistuje možnosť stratenia tváre, alebo potreba pamätania hesla. Pre tieto dôvody a ďalšie priemyselné odvetvia používajú rozpoznávanie tváre v bezpečnostných systémoch. Avšak aj tieto systémy sa môžu stať cieľom útoku, pri ktorom sa útočník snaží ovplyvniť, alebo zmeniť proces spracovania zachytených biometrických dát. Modul detekcie živosti tváre môže byť ako bezpečnostný prvok na vylepšenie efektivity a spoľahlivosti systémov rozpoznávania tváre proti útokom falšovania tváre. Falšovanie tváre je typ útoku kde sa útočník pokúša obísť systém rozpoznávania tváre. Útočník priamo pracuje so systémom rozpoznávania ako normálny užívateľ. Aby obišiel systém útočník môže priložiť fotku, video, alebo masku platného užívateľa pred kameru. Systém rozpoznávania tváre iba zaujíma či osoba je zaregistrovaná, alebo nie. V poslednej dobe kamery s vysokým rozlíšením sú ľahko dostupné za rozumnú cenu, tým pádom útoky falšovania tváre budú komplexnejšie s kvalitnejšími kamerami. Potenciálny útočníci môžu ľahko mať kamery s vyšším rozlíšením a vykonávať útoky s väčšou presnosťou.

Tému tejto práce som si vybral z dôvodu rozšírenosti a popularite užívania neurónových sietí, a ich aplikáciu na rôzne problémy s počítačovým videním. A taktiež pre popularitu využívania biometrických rozpoznávaní tváre ako bezpečnostný zámok. Cieľom tejto práce je objasniť problematiku spojenú s detekciou živosti tváre, navrhnutie funkčného modelu pre detekciu živosti tváre, jeho následná implementácia a testovanie.

Teoretická časť bude prechodom problematiky, kde v kapitole 2 sú uvedené najpopulárnejšie technológie pre detekciu tváre. Následne v kapitole 3 sú ako prvé definované živosti a typov útokov a následne možné existujúce riešenia možnej ochrany proti daným útokom.

Druhá časť práce za zaoberá praktickým riešením algoritmu pre detekciu živosti tváre. V kapitole číslo 4 je predstavený návrh riešenia a je rozdelený do určitých fáz. V kapitole 5 obsahuje stručný popis využitých knižníc pre implementáciu a popis samotnej implementácie algoritmu. Nakoniec kapitola 6 uvádza popis trénovania modelov a ich následné testovanie a hodnotenie testu, po ktorých sa prevádzali testy na kamerách. Na konci tejto kapitoly sa nachádza zhrnutie výsledkov testovania a možné návrhy úprav do budúcna.

Kapitola 2

Detekcia tváre

Detekcia objektu je určovanie či časť obrázku patrí, alebo nie do skupiny objektov ktoré sú predmetom záujmu, tým pádom všetko čo zvyšuje zložitosť rozhodnutia pre množinu obrázkov objektu zvyšuje náročnosť problému a prípadne zvyšuje počet chýb, ktoré detektor urobí. Predpokladajme, že chceme okrem tvár priamo pozerajúc do kamery zachytiť aj tváre naklonené pod uhlom z hľadiska kamery. Pridaním naklonených tvári do množiny obrázkov, ktoré chceme detektovať zvyšuje variabilitu množiny a môže zvýšiť zložitosť rozhodnutia. Ale je možné, že pridávanie nových obrázkov do množiny obrázkov objektu vyhladí rovinu rozhodnutia.

2.1 Viola Jones algoritmus

Paul Viola a Michael Jones v roku 2001 [35], predstavili prvý algoritmus s konkurencieschopnou rýchlosťou detekcie objektov využívaný strojové učenie, schopný detekcie objektov v reálnom čase ale je hlavne využívaný pre detekcii tvári. Jeho hlavný gól je odlišovanie tvárí od iných objektov nie ich rozpoznávanie, štyri hlavné kroky v tomto algoritme sú:

- Výber prvkov pomocou Haarovej bázovej funkcie, všetky ľudské tváre majú podobné vlastnosti, ako oblasť očí je tmavšia oproti oblasti kde sa nachádza nos a podobne. Takéto vlastnosti sú porovnávané pomocou Haarových prvkov.
- Vytvorenie integrovaného obrázku, ktorý sa tvorí výpočtom obdĺžnikov susediacich s obdĺžnikom nachádzajúcim sa na súradniciach (x, y) do jednej obrázkovej reprezentácie, ktorý pomáha pri urýchlení postupu.
- Trénovanie pomocou Adaboost, Adaboost je trénovaný algoritmus, ktorý sa využíva na postavenie kvalifikátora, ktorý sa bude trénovať. Adaboost pomáha pri hľadaní malých kritických vizuálnych prvkov z veľkej množiny potenciálnych prvkov.
- Kaskádový klasifikátor čo re predstavený proces kombinovania klasifikátor, ktorý zahadzuje oblasti pozadia, aby bolo možné vykonávať viacero výpočtov na oblastiach podobných tvári.

Napriek rýchlemu vyhodnoteniu Viola-Jones, nedokáže detektovať tváre z rôznych uhlov [39], tento problém bol pôvodne riešený použitím jedného kaskádového klasifikátora pre každý konkrétny pohľad tváre, alebo použitím rozhodovacieho stromu na odhad polohy a zodpovedajúcej kaskády na overenie detekcie. Ale tieto prístupy vyžadujú ďalšie dáta zatiaľ čo zložité kaskádové štruktúry zvyšujú výpočtové náklady [34].

2.2 Neurónové siete

V posledných rokoch sa na detekciu a rozpoznávanie tváre používali rôzne architektúry a modely neurónových sietí (*artificial neural network*) skratkovito ANN. ANN sa môže použiť na detekciu a rozpoznávanie tváre, pretože tieto modely môžu simulovať spôsob práce neurónov v ľudskom mozgu. To je hlavný dôvod jeho úlohy pri rozpoznávaní tváre [3].

Neurónová sieť definuje rodinu algoritmov prostredníctvom viacerých vrstiev vzájomne prepojených inšpirovaných neurónovou štruktúrou v mozgu. Kde každý kruhový uzol predstavuje umelý neurón a šípka predstavuje spojenie od výstupu jedného neurónu k vstupu druhého. Tieto siete sú určené pre aproximačné funkcie pre veľké množstvo všeobecne neznámych dát. Uzly v neurónovej sieti si vymieňajú dáta medzi sebou. Spojenia medzi uzlami majú prípadné číselné váhy, ktoré sú pozmeňované tak aby dosiahli výstup zodpovedajúci cieľovému výstupu v prípade učenia pod dohľadom, alebo sú menené neurónovou sieťou adaptívne na základe vstupu a schopné učenia. Na základe nasledujúcich kritérií možno sieť klasifikovať ako neurónovú: ak má súbor adaptívnych váh a je schopná aproximovať nelineárne funkcie ich vstupov. Tieto adaptívne váhy si môžeme predstaviť ako intenzity spojenia medzi neurónmi, ktoré sa upravujú počas trénovania [28].

Konvolučná neurónová sieť

Najvýhodnejším aspektom konvolučnej neurónovej (skratkovito CNN) siete je zníženie parametrov v umelej neurónovej siete. Najdôležitejší predpoklad o problémoch, ktoré CNN rieši, by nemala mať znaky, ktoré sú priestorovo závislé. Inými slovami, napríklad v aplikácií pre detekciu tváre nemusíme venovať pozornosť tomu kde sa tváre na obrázku nachádzajú. Jedinou starosťou je ich detektovať bez ohľadu na ich polohu na daných obrázkoch. Ďalším dôležitým aspektom CNN je získanie abstraktných prvkov, keď sa vstup šíri do hlbších vrstiev [4].

Konvolúcia

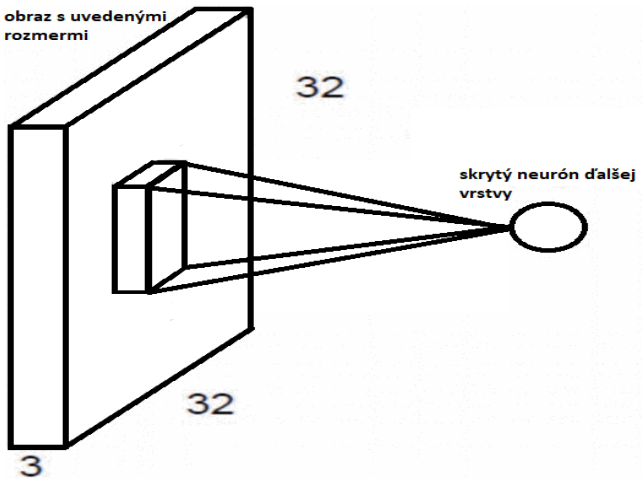
Konvolúcia je matematická kombinácia dvoch funkcií pre vytvorenie tretej funkcie. Spája dva sety informácií. Konvolúcia funkcií f a g sa napíše ako $f * g$, kde operátor $*$ označuje konvolúciu. Konvolúcia je definovaná ako integrál súčinu dvoch funkcií po tom, čo je jedna obrátená a posunutá. Ide o určitý druh integrálnej transformácie:

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau) g(t - \tau)\, d\tau. \tag{2.1}$$

Kde symbol t použitý vyššie, nemusí predstavovať časovú oblasť. V tejto súvislosti možno konvolučný vzorec označiť ako vážený priemer funkcie $f(\tau)$ v okamihu t, kde je váha daná $g(-\tau)$ jednoducho posunutá o čiastku t. Keď sa zmení t, funkcia bude zdôrazňovať rôzne časti vstupnej funkcie [13].

V prípade konvolučnej neurónovej siete. Vrstvy sú organizované v troch dimenziách, hĺbka, šírka a dĺžka. Inými slovami, napríklad máme obrázok ktorý má 32×32 pixlov a hĺbku o troch ako napríklad RGB, tým pádom pre prepojenie vstupnej vrstvy na jeden neurón ako napríklad v prípade skrytej vrstvy v viacvrstvovom perceptrone tak by sme mali $32 \times 32 \times 3$ vážených prepojení a ak by sme pridali ďalší neurón do skrytej vrstvy tak by sme potrebovali ďalších $32 \times 32 \times 3$ vážených prepojení čo by bolo viac ako 6 000 váhových parametrov by bolo použitých na prepojenie iba dvoch uzlov.

Vzhľadom na neefektívnosť riešenia bola vytvorená metóda kde namiesto kompletného prepojenia je lepšie prehľadávať po menších častiach takzvaných regiónoch oproti celému obrázku kde každý uzol nasledujúcej vrstvy má pripojený iba jeden región. Ako vidno v nasledujúcom obrázku 2.1.



Obr. 2.1: *Príklad konvolúcie [4].*

Aj keď množstvo parametrov kleslo, stále ostáva veľké množstvo na vyriešenie. Ďalšia možnosť pre zredukovanie je ponechať lokálne pripojené váhy nastavené pre celý jeden neurón na ďalšej vrstve. Toto prepojí susediace neuróny v ďalšej vrstve s rovnakou váhou na región ako v predchádzajúcej vrstve. Tým pádom zase zníži počet parametrov a počet váh. Je mnoho výhod pre tieto jednoduché operácie, za prvé počet prepojení klesol mnohonásobne a po druhé koncept stanovenia váh pre lokálne prepojenie je podobné ako posúvanie okna o veľkosti jedného regiónu na vstupných neurónoch a mapovanie vygenerovaného výstupu na konkrétne miesto. To poskytuje príležitosť detektovať a rozpoznanie prvku bez závislosti na jeho pozícií. Na zefektívnenie tejto metódy je možné pridať ďalšie vrstvy po

vstupnej vrstve, kde každá vrstva obsahuje rôzny filter tým pádom môžeme extrahovať konkrétny prvok z daného obrázku.

Zdieľanie hmotnosti prináša do modelu nezmeniteľné preklady. Pomáha filtrovať funkciu učenia bez ohľadu na priestorové vlastnosti. Spustením náhodných hodnôt pre filtre sa naučia detekovať hranicu, ktorá môže zlepšiť výkon. Je dôležité mať na pamäti, že ak potrebujeme vedieť, že niečo je v danom vstupe priestorovo dôležité, potom je mimoriadne zlý nápad použiť zdieľanú váhu [4].

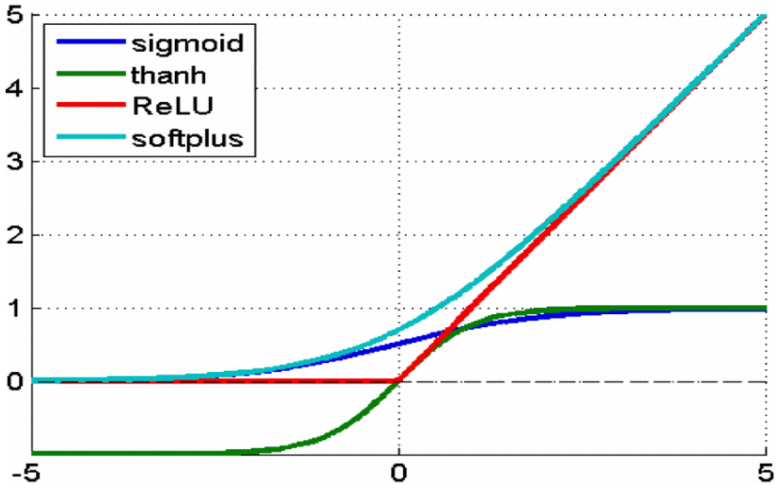
Stride Konvolučná neurónová sieť má mnoho príležitosti na zníženie parametrov viac a viac. Jedna z týchto možností je posun, alebo takzvaný *stride* určujúci veľkosť posúvania vstupnej matice. Vyššie sme jednoducho predpokladali, že uzol ďalšej vrstvy má vysoké prekrytie zo susedmi z hľadiska regiónov. Ale my môžeme manipulovať prekrývanie cez dĺžku krokov [4].

Padding Jedna z nevýhod konvolúcie je strata informácií, ktorá môže nastať na okrajoch obrázku, pretože sú zachytené iba keď sa posunie filter tým pádom nemajú nikdy šancu byť videné. Veľmi jednoduché riešenie na tento problém je pridať okraj takzvaný *padding*, ktorý bude obsahovať samé nuly tým pádom zachutíme okrajový pixel viacej krát [26].

Aktivačná funkcia

Ďalšou vrstvou po konvolúcii je aktivačná funkcia, podľa [4][37]. Aktivačná funkcia sa môže použiť na nastavenie, alebo prerušenie generovaného výstupu. Táto vrstva sa aplikuje s cieľom nasýtiť výstup alebo obmedziť vygenerovaný výstup.

Po mnoho rokov boli sigmoid a tanh najpoužívanejšie aktivačné funkcie. Obrázok 2.2 zobrazuje bežné typy aktivačných funkcií. Nedávno sa však rektifikovaná lineárna jednotka (ReLU) používala častejšie z nasledujúcich dôvodov.



Obr. 2.2: *Bežné typy aktivačných funkcií [4].*

1. ReLU má jednoduchšie definície ako vo funkcii, tak aj v gradiente.

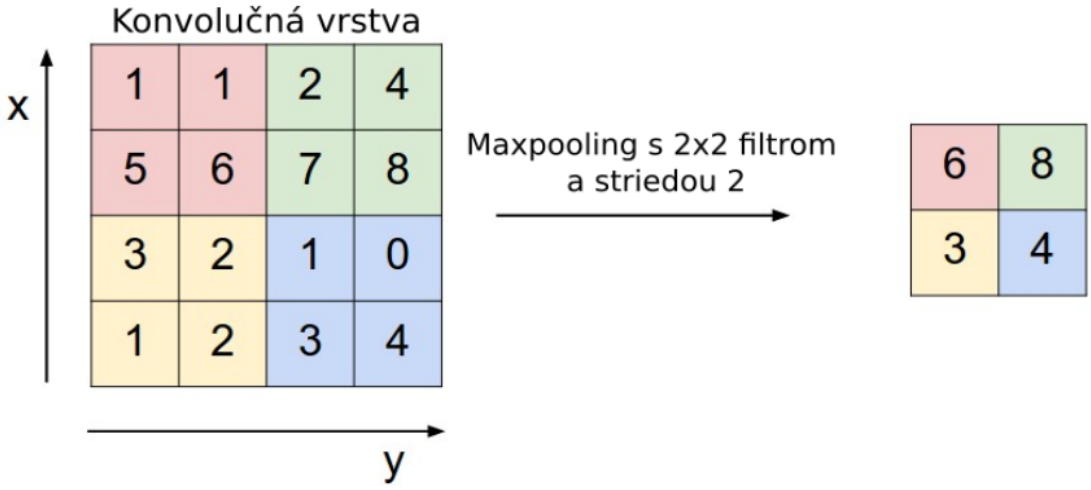
$$ReLU(x) = max(0, x) \tag{2.2}$$

$$\frac{d}{dx}(x) \ = \{1 \text{ ak } x > 0 \ \text{ináč } 0\} \tag{2.3}$$

2. Nasýtené funkcie ako sigmoid a tanh spôsobujú problémy so spätným šírením. Pretože je návrh neurónovej siete hlbší, gradientový signál sa začína strácať, čo sa nazýva *vanishing gradient*. To sa stáva, pretože gradient týchto funkcií je všade okrem stredu, veľmi blízko k nule. ReLU má však konštantný gradient pre pozitívny vstup. Napriek tomu že funkcia nie je diferencovateľná, čo môžeme pri skutočnej implementácii ignorovať.

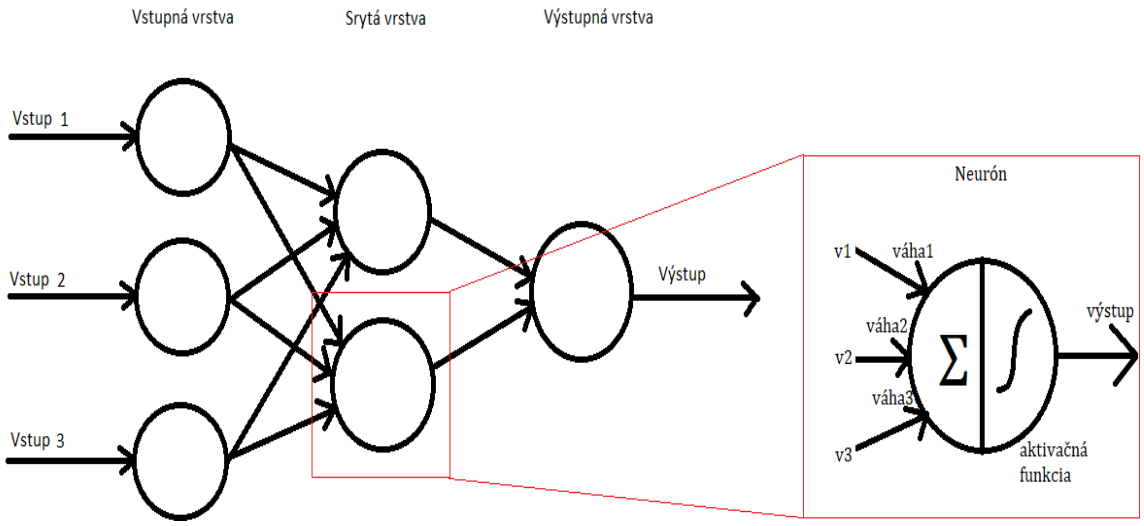
3. ReLU vytvára redšiu reprezentáciu. pretože nula v gradiente vedie k získaniu úplnej nuly. Avšak sigmoid a tanh majú vždy nenulové výsledky z gradientu, čo nemusí byť v prospech tréningu.

Pooling Hlavnou myšlienkou *poolingu* je znižovanie počtu vzoriek, aby sa znížila zložitosť pre ďalšie vrstvy. V doméne spracovania obrazu to možno považovať za podobné zníženiu rozlíšenia. Združovanie nemá vplyv na počet filtrov. Najpopulárnejšou formou operácie združovania je maximálne združovanie, ktoré extrahuje opravy z máp vstupných funkcií, na výstup vydáva maximálnu hodnotu v každej aktualizácii a všetky ostatné hodnoty zahodí. V praxi sa bežne používa maximálne združenie s filtrom o veľkosti 2×2 s krokom 2, piklad predstavený na obrázku 2.3. Toto prevzorkuje obraz v rovine rozmerov máp prvkov o faktor 2. Na rozdiel od výšky a šírky zostáva hĺbkový rozmer máp prvkov nezmenený [4].



Obr. 2.3: *Operácia maxpooling. Upravené z [4].*

Plne pripojené vrstvy



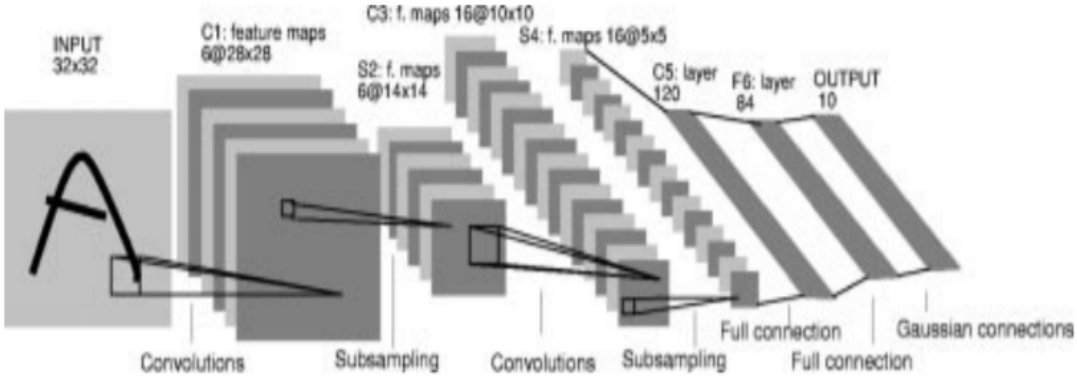
Obr. 2.4: *Zjednodušená štruktúra umelej neurónovej siete, kde neurón je kombináciou lineárneho súčtu vstupov a aktivačnej funkcie [4][8].*

Plne prepojená vrstva je podobná spôsobu, akým sú neuróny usporiadané v tradičnej neurónovej sieti. Preto je každý uzol v úplne spojenej vrstve priamo spojený s každým uzlom v predchádzajúcej aj v nasledujúcej vrstve, ako je znázornené na obrázku 2.4.

Hlavnou nevýhodou plne prepojenej vrstvy je to, že obsahuje veľa parametrov, ktoré si na cvičných príkladoch vyžadujú zložité výpočty. Preto sa snažíme eliminovať počet uzlov a pripojení. Odstránené uzly a pripojenie je možné uspokojiť použitím techniky *dropout*. Napríklad, LeNet a AlexNet navrhujú hlbokú a širokú sieť pri zachovaní konštantného výpočtového komplexu [4][26].

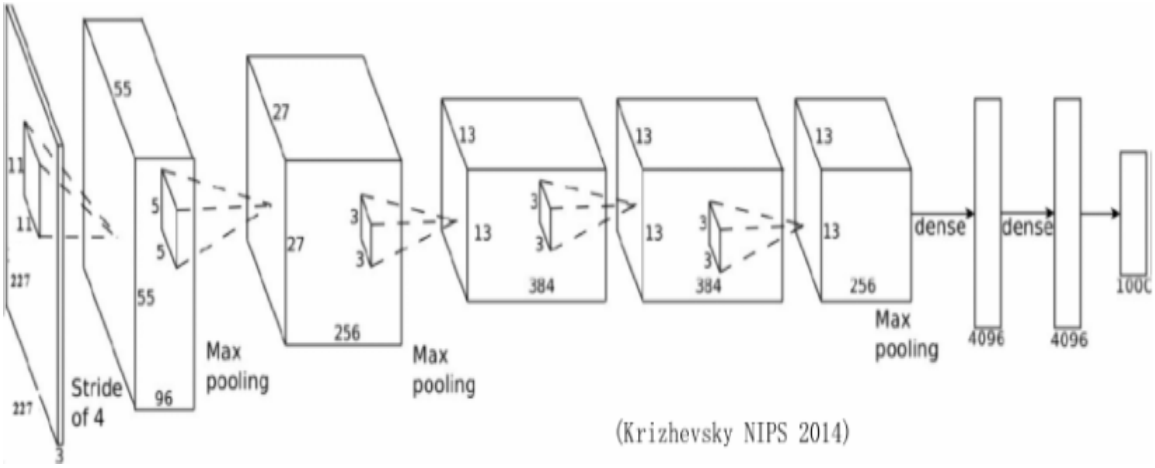
Populárne architektúry CNN

Lenet LeNet predstavil Yan LeCun na rozpoznávanie číslic. Obrázok 14. Obsahuje 5 konvolučných vrstiev a jednu úplne spojenú vrstvu [21][4].



Obr. 2.5: *Architektúra LeNet-5, konvolučnej neurónovej siete, používa sa na rozpoznávanie číslic. Každá rovina je mapou prvkov, takzvane množinou jednotiek, ktorých váhy sú obmedzené, aby boli identické. [21].*

Alexnet AlexNet obsahuje 5 konvolučných vrstiev a 2 úplne spojené vrstvy. Obr. 2.6, Má max-pooling po prvej, druhej a piatej konvolučnej vrstve. Celkovo má 650 000 neurónov, 60 miliónov parametrov a 630 miliónov spojení. AlexNet bol prvý, kto ukázal, že hlboké učenie je efektívne pri úlohách počítačového videnia [4].



Obr. 2.6: *Alexnet [4].*

VGG VGG je inovatívny model rozpoznávania objektov, ktorý podporuje až 19 vrstiev. VGG, postavené ako hlboká CNN, prekonáva mnoho existujúcich riešení počítačového videnia mimo siete ImageNet. VGG je v súčasnosti stále jednou z najpoužívanejších architektúr rozpoznávania obrazu.

Učenie neurónovej siete

Učenie s učiteľom Tento prístup využíva prístup učiteľa, ktorý je múdrejší ako sieť. Učiteľ definuje trénovaciu sadu vstupov a k nim príslušných výstupov. Pre dané vstupy sú následne vypočítané výstupy neurónovou sieťou. Pomocou rozdielu vypočítaných výstupov s výstupmi určených učiteľom môžeme vypočítať chybu, ktorá sa následne využije na upravenie váh neurónovej siete [40].

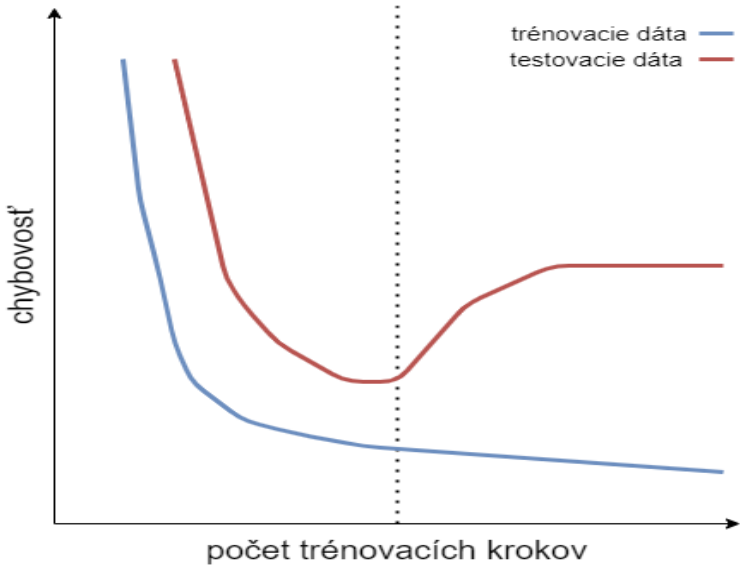
Učenie bez učiteľa Pri učení bez učiteľa nie sú známe hodnoty výstupov pre konkrétne vstupy. Neurónová sieť musí sama objaviť jednotlivé príznaky, podľa ktorých by mohla vznikať nejaká korelácia medzi dátami [40].

Algoritmus spätnej propagácie (backpropagation) Podla [2] funguje nasledovne. Na začiatku trénovania sú váhy jednotlivých neurónov určené náhodne. Následne sa zoberie sada vstupov a vypočítajú sa výstupy. Na základe vypočítaných a očakávaných výstupov môže stratová funkcia vypočítať stratu. Vypočítaná chyba sa následne spätne propaguje, teda začína na výstupnej vrstve a propaguje sa na všetky neuróny v skrytých vrstvách, pomocou ktorých bol výstup vypočítaný. Daná chyba sa rozloží pre jednotlivé neuróny v pomere,v akom neuróny prispeli ku výstupu. Nasleduje gradientný zostup, pomocou ktorého sa vie ako znížiť chybovosť pre každý neurón.

Posilňované učenie Pri posilňovanom učení je pre každú vstupnú hodnotu definovaná hodnota výstupná. Častokrát je ale pre veľa sekvencií vstupov definované iba, či je výsledný výstup správny, alebo nesprávny. Preto je tento prístup založený na pokusoch a omyloch [40].

Možné chyby učenia Pri učení neurónových sietí typicky platí pravidlo, že čím nižšia je hodnota stratovej funkcie, tým presnejší je detektor. To však platí iba pre kontrolujúce dáta. Pre veľmi nízke hodnoty (od 0,5 a nižšie) je detektor väčšinou veľmi nepresný pre detekciu na dátach, na ktorých nebol trénovaný ani kontrolovaný.

Je to tým, že jednotlivé parametre a váhy sa nastavia pre veľmi presnú detekciu na kontrolnej dátovej sade a kvôli tomu nedokáže odhaliť mierne odlišné objekty v testovacej dátovej sade. Tento jav nazývame pretrénovanie (overfitting) a je zobrazený na obrázku 2.7 Preto je v praxi lepšie zastaviť tréning skôr - okolo hodnoty 1 (napríklad v rozmedzí 1,2 až 0,8).



Obr. 2.7: *Chyba pre trénovacie dáta klesá až do hodnoty 0, ale pre testovacie dáta sa od určitého bodu začne znova zvyšovať.*

Opačný jav pretrénovania sa nazýva podtrénovanie. Podtrénovanie nastáva keď sa trénovanie zastaví pred dosiahnutím optimálnej hodnoty, respektíve model nie je dostatočne dlho trénovaný, alebo nie je dostatok dát pre trénovanie danej siete. Taktiež ako pretrénovanie tento jav znižuje presnosť daného modelu [40].

Tieto chyby môžu nastať z viacerých dôvodov kde najčastejším je chybná doba trénovania. Ďalej to môže ovplyvniť zlá augmentácia dát kde neurónovú sieť prepcháme veľmi podobnými dátami, alebo taktiež zle nastavené validačné dáta, kde dáta pre validáciu sú príliž podobné trénovacím dátam.

Kapitola 3

Detekcia živosti tváre

Táto kapitola je venovaná prehľadu publikovaných článkov, zaoberajúcim sa konkrétnymi metódami detekcie živosti tváre a kategorizáciou týchto metód a typov útokov na takéto systémy.

3.1 Kategorizácia typov útokov

Útoky môžeme zaradiť do dvoch oblastí, kde prvou je takzvaný *spoof*, kde sa útočník snaží dostať do systému cez predstavenie neživého objektu, ktorý vykazuje ľudské vlastnosti. Fotografie, videá, falošné bábiky, masky, to všetko sa radí do tohoto typu útoku. Tento typ útoku nazývame prezenčný útok, a vieme ho rozdeliť do troch úrovní:

1. Papierové a digitálne fotografie vo vysokom rozlíšení, videá s výzvami a odpoveďami vo vysokom rozlíšení a papierové masky. Aj keď sa tento typ útoku nemusí javiť ako hrozba pre jeho jednoduchosť, pri súčasne rýchlo rastúcom pokroku v rámci rozlíšenia obrazoviek, je možné že systém s dnes vysoko kvalitnou kamerou s kombináciou metódy detekcie slabého života, ako sú mrkanie, úsmev, otáčanie hlavy, blikajúce svetlá, vytváranie náhodných tvárí, hovorenie náhodných čísel a ďalšie, bude za rok alebo menej označený ako nespoľahlivý a náchylný tomuto typu útoku.

2. Digitálne vysoko realistické 3D modely. S detailnejšími obrazovkami prichádzajú aj uveriteľnejšie 3D modely vytvorené na základe fotografií kde systém vytvorí 3D model na základe 2D fotografie a s pomocou ručnej úpravy je možné vytvoriť vysoko realistické modely pripravené na animáciu.

3. Realistické 3D masky, vytvorené pomocou vosku. Vyrobené na mieru, alebo odliate na základe 3D modelu pomocou 3D tlačiarne. S príchodom 3D tlačiarní prišla aj možnosť vytvárania formy na základe už uvedených 3D modelov, na odlievanie masiek a ich manuálnej úpravy pre realistickosť.

Ako druhú oblasť berieme manipuláciu s biometrickými údajmi následným zachytením alebo dôjde k úplnému obídeniu fotoaparátu, nazýva sa to *bypass*. Tento útok zahrňuje možné presmerovanie vstupu z kamery alebo prístup k samotným dátam systému [15].

3.2 Kategorizácia živosti tváre

Systémy detekcie tváre môžu byť separované do rôznych kategórií na základe indikátoru živosti:

Pohybová analýza - Ploché objekty sa pohybujú rozdielne od skutočných tvárí. Tieto prístupy sú zvyčajne spojené s výpočtami optického toku medzi rôznymi snímkami vo video sekvencii. Predpokladá sa, že rôzne vzory polí optického toku predstavujú rozdiely medzi pohybom 3D a 2D plôch. Tu sa predpokladá, že skutočné tváre majú hlboké informácie a falošné tváre sa považujú za rovinné [19].

Analýza textúry - Založená na predpoklade, že vytlačené tváre obsahovali zistiteľné vzory textúr. Tu sa prvky textúry extrahujú z obrázkov tváre za predpokladu, že sa tlačia falošné tváre, a proces tlače alebo tlačený materiál vytvára určité textúry, ktoré na skutočných tvárach neexistujú [19].

Detekcia znakov života - Pokúša sa analyzovať známky života z obrázkov používateľov ako mrkanie očí či pohyb pier. Vyvinuté algoritmy v rámci sa zameriavajú na pohyb určitej identifikovanej časti tváre. Prístupy reagujúce na výzvy, kde je potrebná interakcia používateľa, sa tiež budú považovať za znamenie života [19].

3D metódy založené na tvare - Tieto metódy sú založené na skutočnosti, že skutočná tvár je trojrozmerná, zatiaľ čo falošná tvár je zvyčajne dvojrozmerná. Tieto metódy však zlyhajú pri zvládaní útoku 3D maskou [12].

3.3 Metóda založená na textúre a frekvenčnej analýze

Základným účelom je rozlíšiť medzi skutočnou a falošnou tvárou z hľadiska tvaru a detailnosti. Navrhnutá metóda využíva jeden obraz, a je založená na analýze frekvencií a textúr na rozlíšenie živých tvárí od falošných. Metóda frekvenčnej analýzy je založená na výkonovom spektre, ktorá využíva informácie o nízkej frekvencii a informácie nachádzajúce sa vo vysokofrekvenčných oblastiach. Okrem toho bola na analýzu textúr na daných obrázkoch tváre implementovaná metóda popisu založená na lokálnom binárnom vzore (LBP). LBP [25] je jednou z najpopulárnejších metód na opísanie textúry informácií. Ako je uvedené v rovnici 3.1, LBP priraďuje kód každému pixelu zohľadnením rozdielov relatívnej intenzity medzi pixelom a jeho susedmi.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \quad s(x) = \{1 \text{ pre} x \geq 0, 0 \text{ pre} x < 0\} \tag{3.1}$$

kde P je počet susedných pixelov, R je vzdialenosť od stredu k susedným pixelom. Zatiaľ čo g_c zodpovedá hodnote stredového pixelu, g_p zodpovedá hodnotám v šedej škále p pixelov rovnomerne rozložených na kružnici s polomerom R a $s(x)$ je prahová funkcia x.

Informácie o frekvencii sa používali z dvoch dôvodov. Prvým je rozdiel v existencii 3D tvarov, ktorý vedie k rozdielu v nízkofrekvenčných oblastiach, ktorý súvisí s komponentom osvetlenia generovaným celkovým tvarom tváre. Po druhé, rozdiel v podrobných informáciách medzi živými tvárami a maskami vytvára rozpor vo vysokofrekvenčných informáciách. Obrázky snímané z 2D objektov majú sklon k strate informácií o textúre v porovnaní so snímkami z 3D objektov. Na extrakciu sa implementuje frekvenčná extrakcia prvkov na základe textúry a extrakcia prvkov na základe fúzie. Pri extrakcii informácií o frekvencii sa najskôr transformuje obraz tváre do frekvenčnej oblasti pomocou 2D diskrétnej Fourierovej transformácie [13] uvedenej v 3.2. Potom je transformovaný výsledok rozdelený do niekoľkých skupín sústredných prstencov tak, že každý krúžok predstavuje zodpovedajúcu oblasť vo frekvenčnom pásme. Nakoniec sa získa 1D vektor funkcií kombináciou priemerných energetických hodnôt všetkých sústredných prstencov.

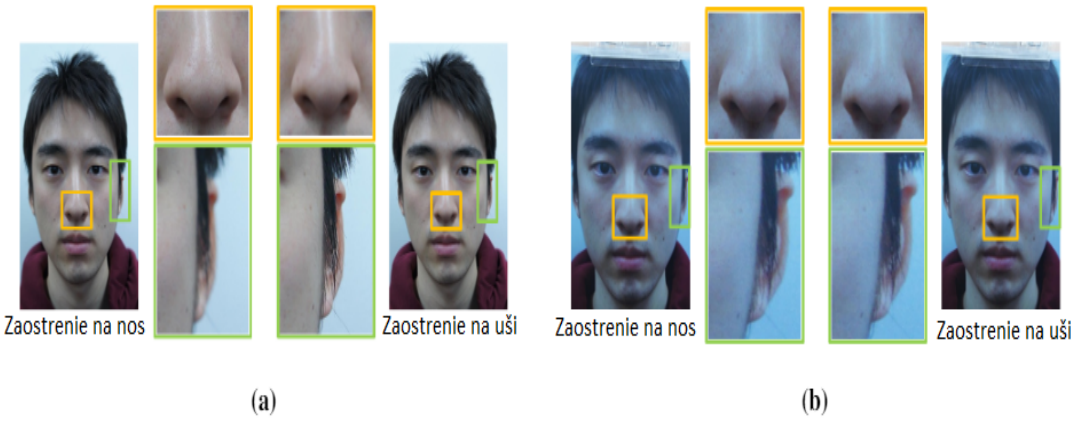
$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp^{-2\pi i(xu+yu)}\,dxdy \tag{3.2}$$

Na extrakciu znakov založených na textúre bol použitý lokálny binárny vzor (LBP) [7], ktorý je jednou z najpopulárnejších techník na opis textúr informácií obrázkov. Na poslednú, tak zvanú extrakciu funkcií založenú na kombinovanej metóde, sa používa klasifikátor (SVM) na učenie detektorov živosti pomocou vektorov funkcií generovaných metódami založenými na výkonovom spektre a LBP. Kombinovaná metóda extrahuje vektor funkcií kombináciou rozhodovacej hodnoty klasifikátora SVM ktoré sú trénované vektormi funkcií založenými na výkonovom spektre a ktoré sú trénované vektormi funkcií na báze LBP. Pre experimenty boli použité dva typy databáz: Databáza webových kamier a Databáza kamier z bankomatov predstavených v [11]. Všetky obrázky v databáze webových kamier boli zachytené za troch rôznych podmienok osvetlenia a falošné tváre boli zachytené z tlačeného papiera, časopisov a karikatúr. Experimentálne výsledky navrhovaného prístupu ukázali, že metóda založená na LBP vykazuje sľubnejšie výsledky ako metóda založená na frekvencii. Celkovo kombinovaná metóda vykázala najlepší výsledok s chybovosťou 4,42% v porovnaní s metódou založenou na frekvencii 5,43% a metódou založenou na LBP s chybovosťou 12,46% [16].

3.4 Metóda pomocou variabilného zaostrovania

Kľúčovým bodom tejto metódy je použitie jednej z funkcií fotoaparátu, variabilného zaostrovania. Ovládaním fotoaparátu sa dajú snímať fotografie zamerané na zložky tváre. Najvýznamnejším rozdielom medzi skutočnou a falošnou tvárou je existencia hĺbkovej informácie. Pravé tváre majú tri rozmery, pričom nos a uši sú relatívne ďaleko od seba. Túto vzdialenosť je možné použiť na adekvátne znázornenie hĺbkovej informácie. V závislosti od objektu alebo miesta zaostrenia môže byť oblasť uší jasná, alebo nie, ako je znázornené na obrázku 3.1. Táto metóda sa spolieha na stupeň hĺbky ostrosti, ktorý určuje rozsah variácií

zaostrenia v pixloch od postupne snímaných obrázkov. Ak je hĺbka ostrosti príliš široká na to, aby zaostril na tvár v uchu, alebo nose, nemohol by existovať žiadny rozdiel v meraniach SML medzi falošnými a skutočnými tvárami. Preto by vstupné údaje pre túto metódu mala získavať kamera, ktorá podporuje dostatočne úzku hĺbku ostrosti [18][17].



Obr. 3.1: *Čiastočne zaostrený obrázok reálnej tváre(a) a podvrhu(b) [18]*

V práci od Sooyeon Kim et. al. [18] Bola presnosť vyhodnocovaná na databázach vytvorených autormi, obrázkov čelnej tváre 24 subjektov pre potrebu zaostrenia na určité oblasti, jedna databáza bola zložená zo snímok bez zrkadlovej kamery a druhá snímaná web kamerou. rozdiel medzi databázami bol v možnosti presného a jemného ovládania zaostrenia. Pomocou nezrkadlovej kamery je možné presne zaostriť na oblasť nosa, alebo uší. Webová kamera však sťažuje podrobné zaostrenie a používatelia nie sú schopní určiť, na čo je zaostrené. Fotografie na falošné tváre boli tlačené v tlačiarni.

Celkový počet obrázkov v databáze bez-zrkadlovky je 5 968 (1 492 párov skutočných obrázkov a 1 492 párov falošných obrázkov). Fotografie sú rozdelené do štyroch skupín podľa rozsahu hĺbky ostrosti kde počet mužov je 17 a žien 7.

Pri testovaní bola do úvahy braná miera falošného prijatia (FAR) a mieru falošného odmietnutia (FRR). FAR je miera počtu falošných obrázkov nesprávne klasifikovaných ako skutočných a FRR je miera počtu skutočných obrázkov nesprávne klasifikovaných ako falošných. Experimentálne výsledky ukázali, že keď je hĺbka ostrosti (DoF) veľmi malá, FAR je 2,86% a FRR je 0,00%, ale keď je DoF veľká, priemerné FAR a FRR sa zvýšia. Výsledky teda ukázali, že táto metóda rozhodujúcim spôsobom závisí od DoF a pre lepšie výsledky je veľmi dôležité zmenšiť DoF.

3.5 Metóda využívajúca funkciu blesku

Metódu ktorá využíva výhody softvérových aj hardvérových metód. Na zvýšenie výkonu softvérovej metódy, ktorá zohľadňuje analýzu textúry a informácie o štruktúre, sa používa

ďalšie zariadenie, blesk. Základným princípom je zväčšenie rozdielov medzi skutočnou tvárou a falošnou tvárou zobrazenou v 2D médiách pomocou blesku. Počas detekcie sa pre objekt nasnímajú dva obrázky s bleskom a bez blesku. Identifikujú sa obdĺžnikové oblasti pre tvár a pozadie. Najprv sa stanoví oblasť tváre, a dva regióny pozadia sú určené na základe pozície regiónu tváre. Následne sa aplikuje klasifikátor Riedka sieť Winnows (SNoW) [24]. Navrhnuté deskriptory sú extrahované z oboch oblastí tváre a pozadia. Tieto deskriptory by mali byť schopné účinne, presne a dôrazne rozlíšiť legitímnych používateľov a bežný 2D spoofingový útok.

Na experimentovanie bola zhromaždená databáza obsahujúca 50 subjektov s útokmi 2D spoofing, vrátane papierových fotografií, fotografií mobilných zariadení, videa, 2D masky a útoku skrivenej masky. Na porovnanie s metódou termálneho obrazu sa zhromaždili aj termálne obrazy 21 subjektov so skutočným a piatimi typmi útokov. metóda sa tiež experimentálne porovnáva s piatimi softvérovými a jednou hardvérovou metódou detekcie živosti. Experimentálne výsledky ukazujú priemernú chybovosť pre všetky uvedené typy útokov 1,17% čo bolo porovnateľne lepšie oproti ostatným metódam taktiež vďaka kombinácií s hardvérovým prvkom bola efektívnejšia na dobu chodu. Nedostatok je ale však potreba blesku, ktorý nie je bežne inštalovaný na predné kamery mobilných zariadení a nepohodlnosť blesku pri každom možnom využití [6].

3.6 Metóda využívajúca morfologické operácie

Robustná technika proti spoofingu na detekciu živosti s morfologickými operáciami [32] na základe detekcie očí a ústnych pohybov, ktorú môže vykonať iba platný používateľ. Na zníženie výpočtovej zložitosti sa pri analýze pohybu podľa konvencie ľudskej tváre uvažuje iba s oblasťou očí a úst. V navrhovanej technike boli použité morfologické operácie pri snímaní pohybu na detekciu živosti tváre. Tieto operácie sú nelineárne operácie, ktoré zodpovedajú tvaru, alebo tvarosloviu prvkov v danom obraze. Namiesto číselných hodnôt sa morfologické operácie spoliehali na relatívne usporiadanie hodnôt pixlov. Morfologické operácie boli implementované umiestnením štruktúrujúceho prvku na všetky možné miesta v obraze a boli porovnané so zodpovedajúcimi susednými pixlami. Výkon navrhovanej techniky bol určený typom a veľkosťou štruktúrneho prvku použitého pri morfologických operáciách. Štruktúrny prvok *Disk* sa použil na efektívne zachytenie pohybu na detekciu živosti. Nový binárny obraz rovnakej veľkosti bol vytvorený v dôsledku morfologickej operácie erózie a dilatácie. Hranice každej oblasti sa vypočítali odpočítaním erodovaného obrázka od pôvodného obrázka.

Na overenie efektívnosti navrhovanej metódy boli použité tri rôzne súbory dát (databáza ZJU Eyeblink, databáza Print-Attack Replay a interná databáza). Pre zachytenie živosti vo videoklipoch pre prvé dva súbory údajov sa zvažoval jediný indikátor živosti, takzvané mrkanie, zatiaľ čo pre interný súbor údajov boli dva indikátory živosti, mrkanie a pohyb úst. Interný súbor údajov poskytuje spoľahlivejšie výsledky a vyššiu mieru presnosti pri detekcii živosti. Navrhovaná technika je jednoduchá a vyžaduje veľmi kratší čas na spracovanie. Spoľahlivosť detekcie tvárnosti tváre bola zistená umiestnením rôznych útokov z troch súborov údajov a systém úspešne registroval všetky útoky. Priemerná presnosť

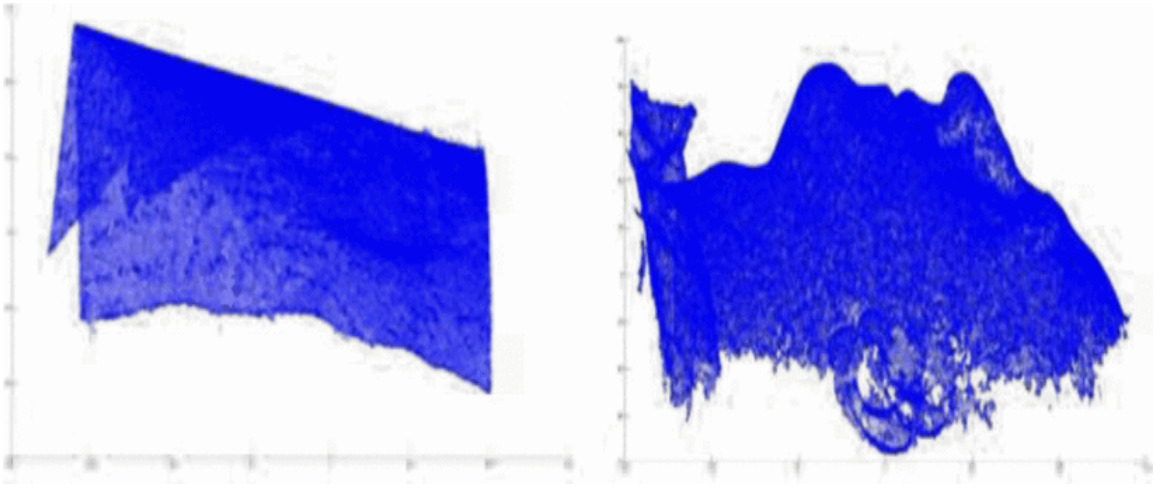
navrhovaného algoritmu na detekciu živosti tváre pomocou morfologických operácií bola získaná ako 98,89% [31].

3.7 Analýza založená na 3D tvare tváre

Cieľom navrhovanej techniky je zistiť, či podvodník využíva 2D obraz skutočného používateľa na oklamanie systému rozpoznávania tváre. Navrhovaná metóda počíta s 3D znakmi zaznamenaných údajov tváre a určuje, či bola ľudská tvár prezentovaná snímacej kamere.

Obrázok 3.2 (vľavo) zobrazuje príklad 3D získania z ohnutého 2D fotografického zdroja, to znamená projekciu bodu v rovine XY. Nedostatok povrchových variácií pri skenovaní (t.j. veľmi nízke zakrivenie povrchu) je jasným dôkazom toho, že akvizícia pochádza z 2D zdroja.

Obrázok 3.2 (vpravo) ilustruje získanie skutočnej 3D tváre. Na porovnanie dvoch 3D skenov na základe výpočtu stredného zakrivenia povrchu. Pri 3D skenovaní sa aproximácia skutočnej hodnoty zakrivenia v každom bode počíta z hlavných komponentov karteziánskych súradníc v danom susedstve. Dekompozícia singulárnej hodnoty sa počíta z kovariančnej matice karteziánskych súradníc všetkých bodov ležiacich v sférickom okolí polomeru [20].



Obr. 3.2: *3D získanie obrazu ľudskej tváre (vľavo) a skutočnej 3D tváre (vpravo) [20].*

Na overenie boli navrhnuté dva experimenty. V prvom boli použité databázy FS [5] a GVS [30].

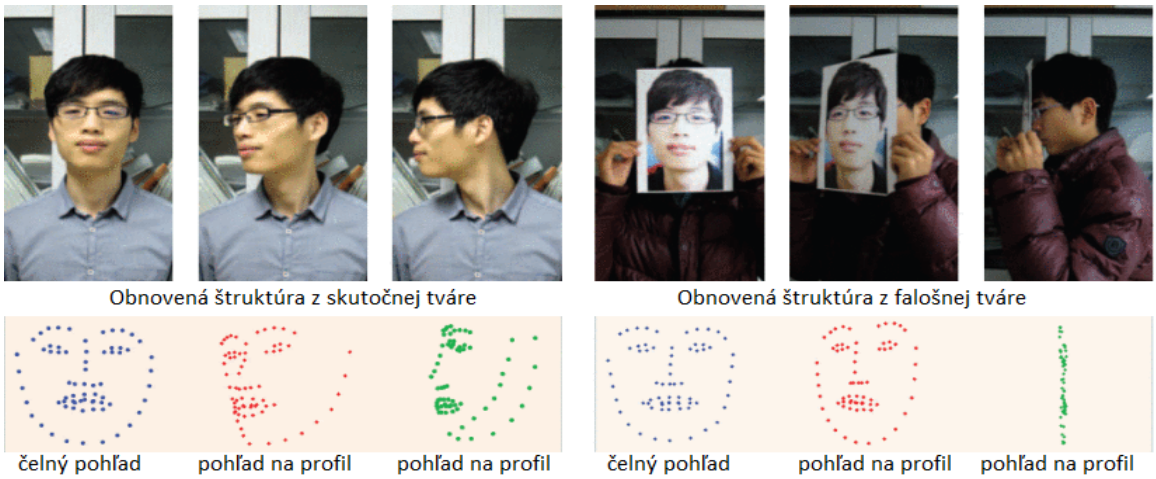
Pri testovaní bola do úvahy braná miera falošného prijatia (FAR) a mieru falošného odmietnutia (FRR). FAR je miera počtu falošných obrázkov nesprávne klasifikovaných ako skutočných a FRR je miera počtu skutočných obrázkov nesprávne klasifikovaných ako falošných.

Rozdelenie stredných hodnôt zakrivenia pre dve sady sa oddelilo a hodnota frekvencie falošného odmietnutia (FRR) sa vypočítala ako nula. V druhom experimente použili da-

tabázy FS a Bosforus. S cieľom určiť citlivosť algoritmu uskutočňujú rôzne experimenty s hodnotami od 4 do 20. Pre rôzne hodnoty polomeru je hodnota chybného odmietnutia (FRR) na 1. pozícii vždy rovná nule.

3.8 Detekcia živosti tváre pomocou 3D štruktúry obnovenej z jedného fotoaparátu

Táto metóda je založená na snímaní tváre pod rôznymi uhlami pohľadu z pevnej kamery pohybom, alebo obrátením tváre kde sa najprv využije algoritmus CLM [29] na lokalizáciu riedkych orientačných bodov tváre. V nasledujúcom kroku sa vyberú vhodné rámce pomocou metriky podobnosti grafu. Po získaní dvoch kľúčových snímok sa obnovia parametre fotoaparátu a počiatočná štruktúra tváre. Nakoniec sa urobí zdokonalenie štruktúry tváre, aby sa spresnili počiatočné obnovené výsledky, a do úpravy zväzku sa pridajú nové kľúčové snímky, pretože štruktúra získaná iba z dvoch snímok tváre môže byť ovplyvnená nepresnou detekciou orientačných bodov a nepresným odhadom parametrov kamery. Tieto štruktúry sa zarovnávajú a potom sa extrahujú prvky štruktúry na klasifikáciu. Po zarovnaní sú 3D súradnice riedkej štruktúry zreťazené do podoby vektora prvkov. Klasifikátor SVM je potom vyškolený na základe funkcií na klasifikáciu originálnych a falošných vzoriek tváre.



Obr. 3.3: *Porovnanie obnovených riedkych 3D štruktúr tváre medzi pravou a falošnou tvárou. Existujú významné rozdiely medzi štruktúrami [36].*

Na experimentovanie boli zhromaždené tri databázy používajúce fotoaparáty rôznej kvality na kontrolu výkonu proti falšovaniu na rôznych zariadeniach. Navrhovaný prístup dosahuje 100% výsledkov klasifikácie aj presnosti detekcie živosti tváre [36].

3.9 Metóda založená na analýze pohybu očí

Navrhovaný algoritmus na detekciu živosti je založený na analýze pohybu očí. Základným predpokladom je, že kvôli blikajúcim a nekontrolovaným pohybom zorničiek v ľudských očiach by mali existovať veľké variácie tvaru. To by malo byť vhodné na výpočet optického toku. Tento algoritmus je ďalej predstavený v nasledujúcom texte. Najskôr sa na vstupnom obrázku zistia stredové body oboch očí. Tento krok musí byť presný, pretože ide o počiatočný krok a má veľký vplyv na výkonnosť. Autori využili skutočnosť, že intenzita oblasti očí je nižšia. Na nájdenie kandidátov sa použije Gaussov filter a potom sa pomocou algoritmu gradientového zostupu zistia lokálne minimá intenzít. V ďalšom kroku sa kandidátske oblasti záujmu klasifikujú pomocou kvalifikátora Viola-Jones AdaBoost [35] a vo výsledku sa odstránia neplatné oči.

Po nájdení centier oboch očí je potrebná normalizácia oblasti tváre okolo očí. Nájdené oblasti sa normalizujú na jednu veľkosť a použije sa vysoko priepustný filter. Výsledkom sú obrázky s rozmermi 20×20 pixlov založené na stredoch očí. Potom sa oblasti prevádzajú do binárnej podoby pomocou prahu získaného zo stredných hodnôt pixlov oblasti oka. Očné oblasti od skutočných tvárí majú väčšie variácie tvaru ako oblasti získané od falošných tvárí.

Ďalším krokom je výpočet skóre živosti, tieto skóre sa vypočítali pomocou Hammingových vzdialeností. Autori vo svojom experimente porovnávali 5 ľavých a 5 pravých očí. Živé skóre medzi dvoma snímkami je počet rôznych pixlov medzi oboma regiónmi. Po výpočte desiatich skóre živosti oboch očí sa vezme priemerné skóre. Ak je toto priemerné skóre živosti väčšie ako prahová hodnota, potom je vstupným obrazom živá tvár, inak je to falošná.

Experimentálne výsledky ukázali, že keď sa skóre živosti meria pomocou Hammingovej vzdialenosti, priemerné skóre živej tváre je 30 a skóre falošnej tváre 17, čo ukazuje, že skóre živej tváre je zreteľne väčšie ako skóre falošných tvárí. Keď je teda hranica nastavená na 21, autori dosiahli najlepší výkon s FAR ako 0,01 a FRR ako 0,08 [14].

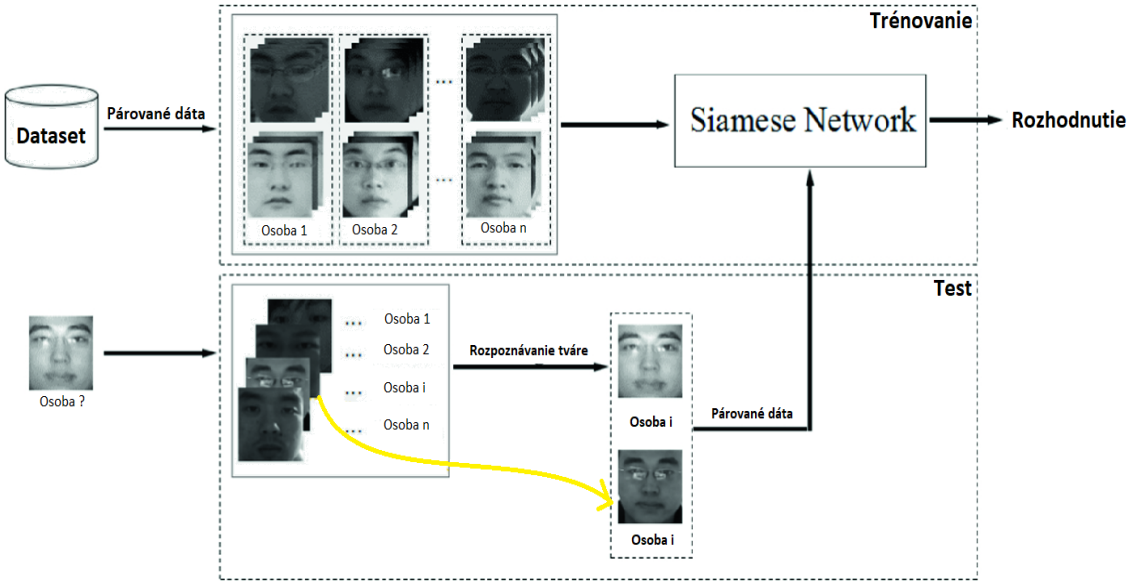
3.10 Metóda založená na základe identity klienta

Metódu detekcie živosti tváre založená na identite klienta pomocou siamskej siete.

Siamská sieť je trieda architektúr neurónových sietí, ktoré obsahujú dve, alebo viac podsietí. Obsiahnuté podsiete môžu byť rovnaké, alebo odlišné. V tomto prípade bola využitá s dvoma rovnakými podsieťami. Tieto dve podsiete boli konvolučné siete, ktoré mali rovnakú konfiguráciu s rovnakými parametrami a váhami a aktualizácia parametrov sa zrkadlila v oboch sieťach.

Živý prejav tváre zisťujú po rozpoznaní tváre namiesto pred rozpoznaním tváre, to znamená, že živosť tváre sa zisťuje pomocou informácií o totožnosti klienta. V tréningovej fáze sa zhromaždia páry obrázkov tváre, na precvičenie siamskej siete. Každý pár obrázkov sa skladá z dvoch obrázkov tváre. Dva obrazy tváre môžu byť obraz skutočnej tváre a falošný obraz tváre, alebo dva obrazy skutočnej tváre. Dva obrazy tváre v každej dvojici pochádzajú

od rovnakého klienta. Vyškolená siamská sieť môže klasifikovať dvojicu vstupných obrázkov ako *dva skutočné*, alebo *jeden falošný jeden skutočný*. V testovacej fáze je vstupný obraz testovacej tváre najskôr identifikovaný pomocou detekcie tváre a sú získané informácie o identite obrazu testovacej tváre. Potom sa získa obraz skutočnej tváre identifikovaného klienta. Získaný obraz skutočnej tváre a obraz testovacej tváre sú klasifikované trénovanou siamskou sieťou. Ak siamská sieť klasifikuje tieto dva obrázky ako *dva skutočné*, potom je vstupný testovací obraz tváre obrazom skutočnej tváre, inak ide o falošný obraz. Možnou nevýhodou tejto metódy je v prípade zlyhania rozpoznávanie tváre, to znamená, keď nie je správna totožnosť klienta, výkonnosť detekcie živosti tváre dramaticky poklesne.



Obr. 3.4: *Zobrazenie návrhu metódy[12]*

Pre preukázanie účinnosti tejto metódy, boli uskutočnené experimenty na dvoch verejných súboroch údajov: NUAA [33] a Replay-Attack [9].

NUAA je verejne dostupný súbor údajov, ktorý poskytuje Nanjingská univerzita pre letectvo a astronautiku a je široko používaný na vyhodnocovanie detekcie živosti tváre. Dátová sada obsahuje 12614 snímok 15 rôznych subjektov, vrátane snímok skutočnej aj falošnej tváre. Databáza je rozdelená na cvičnú súpravu s celkovým počtom 3 491 (reálne: 1 743 / falošné: 1 748) obrázkov a testovaciu súpravu s celkovým počtom 9 123 (skutočné: 3 362 / falošné: 5 761) obrázkov. Na tomto súbore chybovosť dosahovala 1,96%

Replay-Attack poskytuje IDIAP v roku 2012. Obsahuje 1 300 videoklipov s 50 rôznymi predmetmi. Tieto videoklipy sú rozdelené na 300 videí so skutočným prístupom a 1 000 falošných útočných videí. Súbor údajov zohľadňuje rôzne svetelné podmienky použité pri útokoch spoofing. Databáza Replay-Attack pozostáva z tréningovej sady, vývojovej sady a testovacej sady. Na tomto súbore chybovosť dosahovala 0,86% [12].

Kapitola 4

Návrh riešenia

Cieľom tejto kapitoly je popis navrhnutého modelu, ktorý využíva architektúru sekvenčného hlbokého učenia upravenej verzie štandardu VGG-11. VGG znamená skupina vizuálnej geometrie (Visual Geometry Group). Je odvodená od architektúry AlexNetu 2.2 kde rozdiel nastáva pri riešení hĺbky konvolučnej neurónovej siete. Na zachytenie diskriminačnej a všeobecnej mapy vlastností. V architektúre sú vrstvy konvolúcie a združovania skladané od vstupu k výstupu. Obrázok 4.1 zobrazuje navrhovanú architektúru systému, ktorý pozostáva z nasledujúcich etáp.

4.1 Fáza pred spracovania vstupných údajov

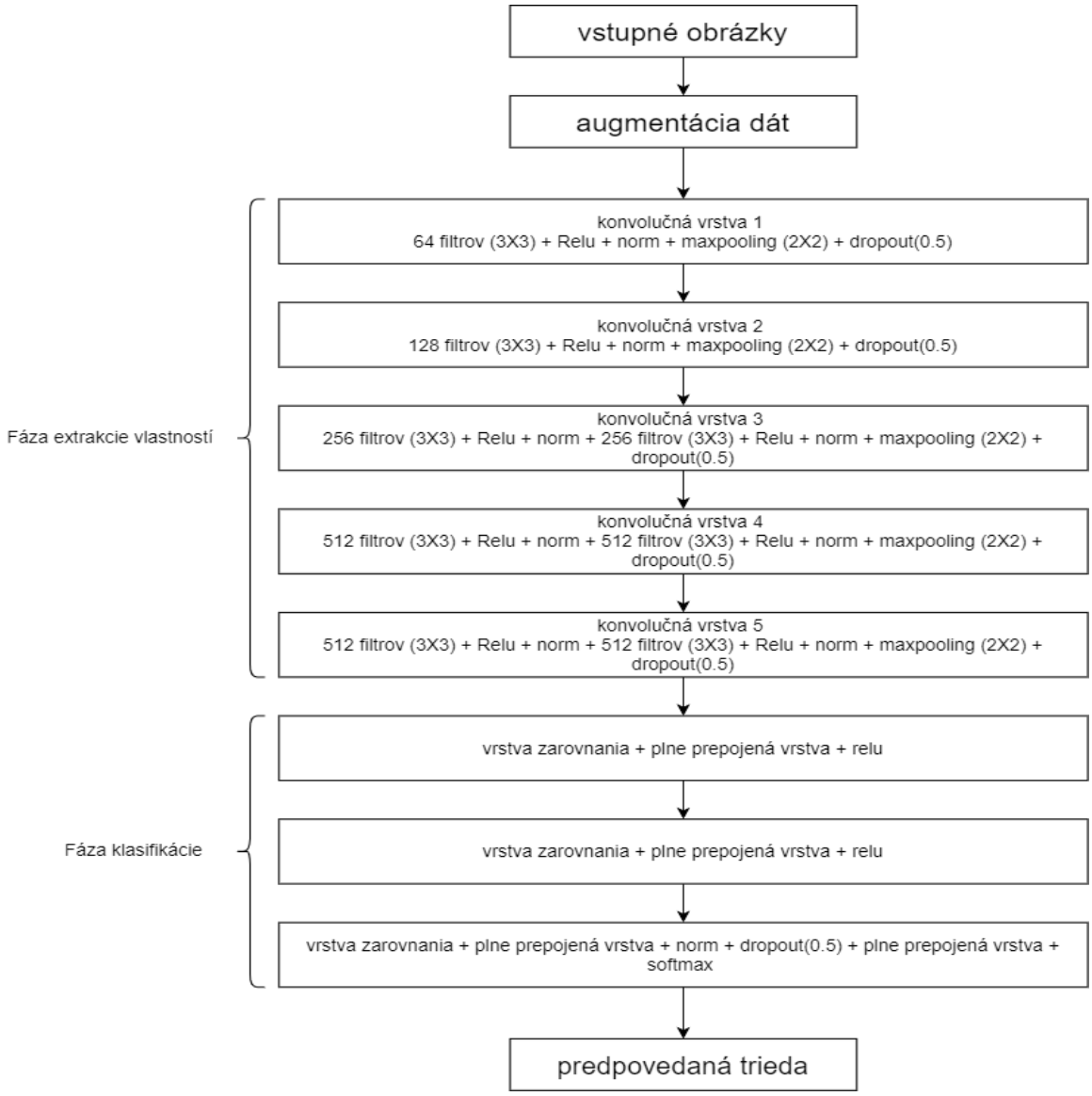
Augmentácia dát [27] je rozhodujúcou súčasťou trénovania neurónových sietí, pretože na dosiahnutie vysokej presnosti potrebujú veľké množstvo dát. S hľadaním dát pre skutočné tváre a podvrhy ako papierové fotky nebol problém, vzhľadom na už mnoho existujúcich databáz takéhoto typu. Jedine problém nastal so získavaním kvalitných dát masiek či už voskových, alebo z iného materiálu. Ale bolo možné rozšíriť tieto dátové sady vďaka rôznym technikám, vrátane rozsahu strihu, rozsahu zväčšenia a vodorovného preklopenia.

4.2 Fáza extrakcie vlastností

Navrhovaná architektúra konvolučnej neurónovej siete pozostáva z operácií: operácia konvolúcie, operácia združovania a lineárna aktivačná funkcia.

- Konvolučná operácia: Filtre veľkosti 3×3 sa prechádzajú cez vstupné obrázky veľkosti 300×300, aby sa získali vlastnosti, ktoré sú dôležité pre klasifikáciu vstupných obrázkov, a zahodia sa tie, ktoré nie sú. Inými slovami, filtre spájajú obrázky.
- Operácia združovania (maxpooling): Táto operácia vykoná redukciu vlastností na výstupe konvolučnej vrstvy pri zachovaní jedinečných vlastností obrázkov.

- Lineárna aktivačná funkcia (relu): Prevádza všetky negatívne vstupy na nuly, aby sa v manipulovaných obrázkoch javili ako bodky.
- Dávková normalizácia (norm): je technika vykonávaná medzi vrstvami neurónovej siete, slúži na urýchlenie tréningu a využitie vyššej rýchlosti učenia.



Obr. 4.1: *Navrhovaná architektúra systému.*

4.3 Fáza klasifikácie

- Flatten vrstva: funkciou tejto vrstvy je spracovávať mapy vlastností do jedného vektora vhodného na spracovanie ďalšou vrstvou neurónovej siete.

- Plne prepojená a relačná aktivačná funkcia: plne prepojená funkcia v kerase vytvára neurónovú sieť so špecifickým počtom skrytých uzlov a relačnou aktivačnou funkciou. Trvá to, že sploštený vektor je ako vstup do neurónovej siete.
- Dropout vrstva: jeho funkciou je zabrániť nadmernému pretrénovaní.
- Plne prepojená vrstva s softmax funkciou: predstavuje výstupnú vrstvu s sigmoidnou funkciou, pretože ide o problém s binárnou klasifikáciou (živá, alebo neživá tvár)

4.4 Zostavenie konvolučnej neurónovej siete

Na zostavenie konvolučnej neurónovej siete sa používa kompilácia funkcií s tromi parametrami: optimalizátorom je Adam, stratová funkcia a výkonovou metrikou je presnosť. Optimalizátor Adam [1] je stochastická gradientná zostupová metóda, ktorá je založená na adaptívnom odhade momentov prvého a druhého rádu. Kompletný vzorec stratovej funkcie v práci [22] určuje prestnosť detektoru 4.1

$$Strata = -\frac{1}{\text{výstup}} \sum_{\text{veľkosť}}^{\text{výstup}} y_i \times \log \hat{y}_i + (1 - y_i) \times \log(1 - \hat{y}_i) \tag{4.1}$$

kde $\hat{y}_i$ je i-tá skalárna hodnota na výstupe z modelu, y_i je zodpovedajúca cieľová hodnota a veľkosť výstupu je počet skalárnych hodnôt na výstupe z modelu. Presnosť sa počíta ako je uvedené v rovnici 4.2 [22].

$$Presnost = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{4.2}$$

- TP označuje počet objektov klasifikovaných ako správne, ktoré sú aj v skutočnosti správne.
- FP označuje počet objektov klasifikovaných ako správne, no v skutočnosti sú nesprávne.
- FN počet objektov klasifikovaných ako nesprávne, ktoré sú v skutočnosti správne.
- TN počet správne klasifikovaných nesprávnych objektov.

Rovnicu 4.1 môžeme zjednodušiť a vyjadriť podľa rovnice 4.3

$$total\_loss = confidence\_loss + \alpha * location\_loss \tag{4.3}$$

Kde $total\_loss$ je výsledná hodnota stratovej funkcie, $confidence\_loss$ je hodnota vyjadrujúca istotu detektora, že sa v danej ohraničenej oblasti nachádza objekt. A $location\_loss$ je hodnota vyjadrujúca vzdialenosť predikovanej oblasti detekcie a skutočnej oblasti. α určuje pomer, akým $location\_loss$ prispieva do celkovej straty.

Kapitola 5

Implementácia riešenia a trénovanie modelu

Táto kapitola obsahuje popis implementácie a jej najdôležitejších častí. Sekcia 5.1 sa zaoberá použitými knižnicami a ich popisom. Pri každej knižnici sú uvedené dôvody, prečo bola efektívnym výberom pri riešení tejto témy. Všeobecne boli preferované open-source knižnice, ktoré sú voľne dostupné aj na viacerých platformách. Ďalším kritériom bola ich rozšírenosť. Vysoká rozšírenosť umožňuje rýchlejšie riešiť vzniknuté problémy pri implementácií vlastného algoritmu tým, že existuje viac používateľov s podobným problémom a zároveň viac odborníkov, ktorý poznajú riešenie na daný problém. Takáto rozšírenosť taktiež svedčí aj o kvalite danej knižnice. Ďalšie kritéria pre výber boli kompletnosť dokumentácie či výskyt cvičných príkladov. V sekcii 5.3 sú popísané najdôležitejšie časti kódu pre vytvorenie a trénovanie neurónovej siete a pre jej použitie. Pri každom súbore je stručný popis funkčnosti.

5.1 Použité knižnice

Implementácia sa skladá z dvoch častí. Prvá časť sa zaoberá neurónovou sieťou. Konkrétne jej konštrukciou, trénovaním a ukladaním. Druhá časť je využitie danej siete pre detekciu tváre. Cieľom tejto sekcie je oboznámiť čitateľa s vybranými knižnicami. Boli použité nasledujúce technológie:

Tensorflow

Tensorflow[1] je jedna z populárne využívaných knižníc na problematiku strojového učenia. Vyvíjaný od roku 2011 tímom Google Brain vtedy známy ako DistBelief. Nie je viazaný na architektúry a tak dokáže pracovať na CPU, GPU ako aj na mobilných zariadeniach. Pracuje na základe grafov, kde každý uzol je matematickou operáciou a hrany sú viac di-

[1] https://www.tensorflow.org/api_docs

menzionálne úložné priestory, alebo takzvané tensory. Ponúka dostatočnú úroveň abstrakcie na jednoduchú implementáciu algoritmov strojového učenia [38].

OpenCV

OpenCV[2] je knižnica s voľne šíriteľným kódom, určená na prácu s počítačovým videním. Je šíriteľná za podmienok GPL a BSD licencií a je dostupná na viacerých platformách. Pôvodne bola napísaná v jazyku C++, dnes má však rozhrania aj pre jazyky Java a Python. Rieši problematiku počítačového videnia a momentálne má viac než 2500 algoritmov, pričom mnohé z nich implementujú najmodernejšie prístupy hlavne súvisiace s počítačovým videním v reálnom čase. Tieto algoritmy sa následne používajú napríklad v detekcii a klasifikácii objektov, sledovaní pohybu objektu, vyhľadávaní podobných objektov z databázy, rozpoznávaní scény, a majú mnoho ďalších možných použití, je kompatibilná s modernými viacjadrovými procesormi a grafickými kartami za využitia nástroju CUDA[3]. Zároveň je jednou z najpoužívanejších knižníc počítačového videnia.

Imutils

Je rada pohodlných funkcií, ktoré uľahčujú základné funkcie spracovania obrázkov, ako sú preklad, rotácia, zmena veľkosti a jednoduchšie zobrazovanie Matplotlib obrázkov s OpenCV a Python 2.7 aj Python 3.

scikit-learn

Je bezplatná softvérová knižnica strojového učenia pre programovací jazyk Python. Obsahuje rôzne algoritmy klasifikácie, regresie a zoskupovania vrátane metódy podporných vektorov, náhodných lesov, zosilňovania gradientov, k-means a DBSCAN a je navrhnutý na spoluprácu s numerickými a vedeckými knižnicami NumPy a SciPy v Pythone.

NumPy

NumPy je knižnica pre programovací jazyk Python, ktorá pridáva podporu pre veľké viac dimenzionálne polia a matice spolu s rozsiahlou zbierkou matematických funkcií na vysokej úrovni pre prácu s týmito poliami. Predchodcu NumPy bol Numeric vytvorený pôvodne Jim Hugunin-om s prispením niekoľkých ďalších vývojárov. NumPy je skratka pre Numerical Python. NumPy je softvér s otvoreným zdrojovým kódom a má veľa prispievateľov.

[2] https://docs.opencv.org/master/
[3] https://docs.nvidia.com/cuda/

Tkinter

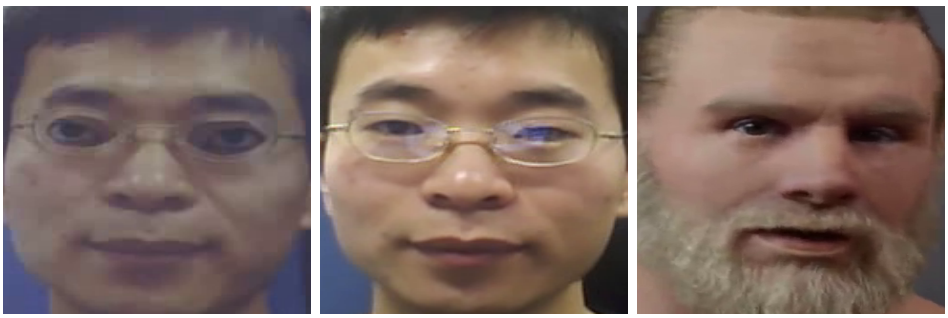
Tkinter je väzobná knižnica Pythonu na súbory nástrojov Tk GUI. Je to štandardné rozhranie Pythonu k sade nástrojov Tk GUI a je to vlastne Pythonovo štandardné GUI. Tkinter je súčasťou štandardných inštalácií Pythonu pre Linux, Microsoft Windows a Mac OS X. Názov Tkinter pochádza z rozhrania Tk.

Knižnica obrázkov Python (PIL)

Knižnica obrázkov Python je bezplatná a otvorená doplnková knižnica pre programovací jazyk Python, ktorá pridáva podporu pre otváranie, manipuláciu a ukladanie mnohých rôznych formátov obrazových súborov. Je k dispozícii pre Windows, Mac OS X a Linux.

5.2 Dátová sada

Na účely trénovania a testovania jednotlivých algoritmov bola vytvorená databáza, ktorej časť videí bola prebratá z dátovej sady CASIA[10] a časť zabezpečená z rôznych zdrojov zaoberajúcimi sa 3D maskami. Dátová sada CASIA bola vybraná z niekoľkých dôvodov. Prvým dôvodom je jej rozsiahlosť a rôznorodosť, kde obsahuje viac ako 500 videí pod rôznym osvietením, zachytenými rôznymi kamerami a rôzne typy 2D útoku, kde na jednu osobu je priemerne 12 natočených desať sekundových videí. Dáta 3D podvrhov boli pridané pre možnosť testovania útoku reálnejšími maskami. videá realistických masiek boli hlavné získavané od rôznych tvorcov, umelcov zaoberajúcich sa vytváraním takýchto masiek ako tvorba umenia. Boli vyberané prevažne masky, ktoré by vierohodne na prvý pohľad mohli prejsť ako skutočné.



Obr. 5.1: *Príklad obrázkov s dátovej sady. Najviac vľavo je papierový podvrh, v strede skutočná tvár a najviac v pravo realistická maska.*

Obrázky tvári boli zbierané pomocou python skriptu za využitia pretrénovaného modelu na detekciu tváre, kde je možno určiť medzeru medzi zachytenými rámcami pre väčšiu rozmanitosť dátovej sady, Ďalej tieto dáta boli manuálne prechádzané pre kontrolu správností zachytenia obrázku a prípadne zmazané, kde v tomto bode rovnako aj pri samotnej detek-

cií živosti ovplyvňuje detektor tvárií prestnosť riešenia. Obrázky sú pri začiatku trénovania normalizované na veľkosť 300×300 pre jednoduchšie trénovanie.

Obr. 5.2: *Príklad chybne zachyteného obrázku, detektorom tvárí.*

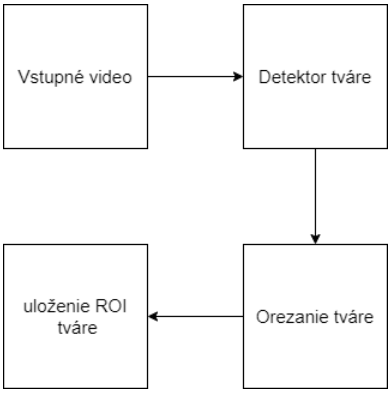
Ako vidno na 5.2, z určitého dôvodu detektor tvárí, pri chybe zachytáva iba spodnú krajnú časť tváre ako celú tvár. V každom prípade generovania obrázkov nastávala len táto chyba. Je možné sa domnievať že vybraný detektor, ktorý veľmi dobrú rýchlosť detekcie stráca na presnosti.

5.3 Dôležité súčasti implementácie

V tejto sekcií sú stručne popísane štyri hlavné implementačné časti a to extrahovanie tvári z videa, model detektora živosti, trénovanie detektora živosti a samotný detektor živosti tváre.

Extrahovanie tvári z videa - V sripte *gather_data.py* sa zachytávajú oblasti záujmu tváre (ROI) zo vstupných video súborov a pomáha nám vytvoriť dátovú sadu pre trénovanie modelu. Ako vstup berieme súbor s videami v ktorých hľadáme tvár osoby, pomocou pretrénovaného detektora[4], kde zadáme rozhodovaciu hranicu pre posunutie rámca ďalšiemu kroku, kde sa potom rámec oreže a ukladá do daného priečinku. Princíp je možno vidieť na obrázku 5.3. Vzhľadom k nie sto percentne presnému detektoru tváre je potrebné dané kontrolovať manuálne pre možne nevyžiadané dáta v dátovej sade.

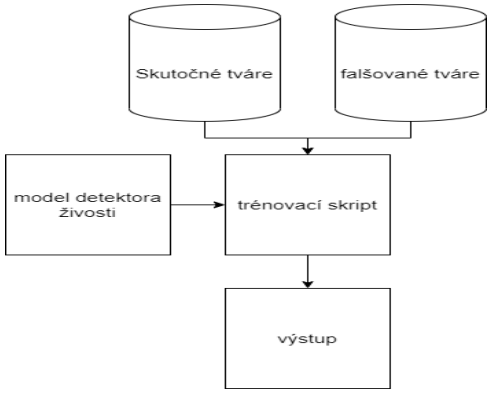
[4] https://github.com/spmallick/learnopencv/blob/master/FaceDetectionComparison/models/res10_300x300_ssd_iter_140000_fp16.caffemodel



Obr. 5.3: *Detekcia ROI tváre vo videách za účelom vytvárania dátovej sady.*

Model detektora živosti - V súbore *detektor_zivosti.py* je implementovaná konvolučná neurónová sieť podla modelu 4

Skript pre trénovanie - *train.py* slúži na trénovanie modelu. Proces trénovania modelu. Použitím *skutočných* a *falošných* obrázkov ako našej množiny údajov môžeme trénovať model detekcie živosti pomocou OpenCV, KERAS a hlbokého učenia. Ako prvé sú v skripte načítané vstupy ako cesta ku dátovej sade, modelu detekcie tváre, a výstupné cesty pre model, enkóder modelu a graf pre vizualizáciu tréningových krokov. Ďalej sú dáta spracovávané do požadovanej formy a je zostavený model detektora živosti. Následne sa dáta predávajú zostavenému modelu o ktorého priebeh tréningových krokov sa starajú funkcie v danej knižnici. Po dokončení trénovania sa sieť vyhodnotí numericky a graficky a model spolu s enkóderom sa uloží. V našom prípade énkóder predstavuje súbor typu *pickle* vytvorený za použitia modelu *pickle*[5].



Obr. 5.4: *Proces trénovania modelu.*

Detektor živosti - skript *detector_demo.py* implementuje už samotný detektor, ako prvé sa načítajú detektory tváre a živosti, ďalej sa pristúpi ku video kamere, z ktorej sa postupne čítajú rámce. Každý rámec prejde detektorom tváre, ktorý vráti tvár jej polohu a hodnotu rozhodnutia medzi 0 - 1. Pokiaľ je hodnota nad určitou hranicou tak sa tvár so súradnicami posúva detektoru živosti, ktorý určí či sa jedná o skutočnú

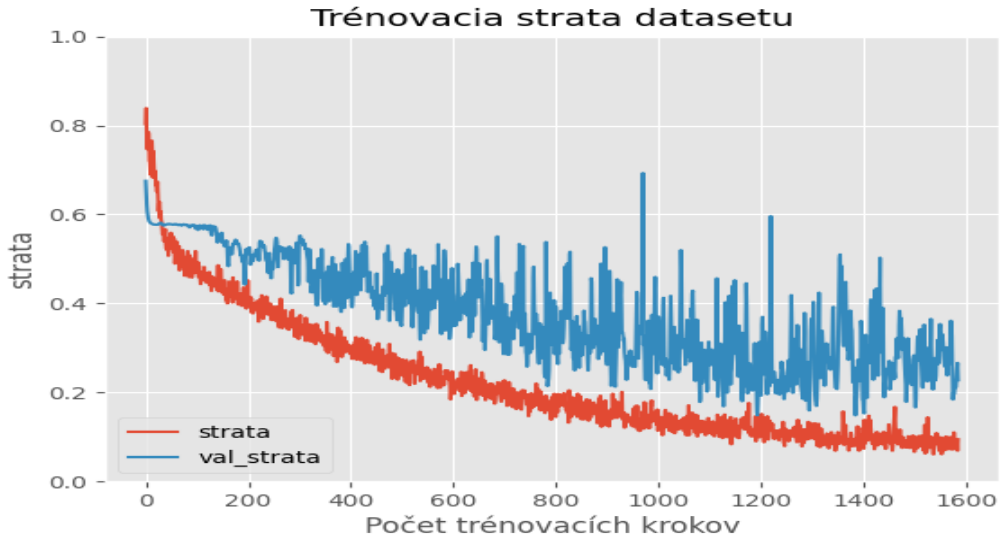
[5] https://docs.python.org/3/library/pickle.html#module-pickle

tvár, alebo podvrh, následne na základe súradníc zvýrazní tvár a označí jej triedu. Takýto rámec sa potom posiela na výstup. Z toho vypláva, že rýchlosť detektorov bude mať priamy vplyv na rýchlosť prehrávania videa. *detector.py* je upravená verzia s implementovaným jednoduchým GUI pre jednoduchšie testovanie.

5.4 Trénovanie modelu

V tomto modeli sa na tréning využila databáza CASIA bez úprav, ktorá obsahovala 360 krátkych videí ktoré sa rozsekali po rámcoch, z ktorých 75% slúžilo na trénovanie a 25% na testovanie.Pre návrh a priebeh testov boli ručne vyberané dáta z testovacích dát.

Tréning prebiehal na počítači Lenovo Y740. Po 1 hodine a 20 minútach sa hodnota stratovej funkcie nejako významne nemenila, tak bol tréning zastavený. Vykonalo sa 1589 trénovacích krokov a výsledná hodnota stratovej funkcie bola 0,1060. Vývoj stratovej a stratovej validačnej funkcie je zobrazený na grafe 5.5. *batch size* pre testovacie a validačné dáta bol 16 a *learning rate* bol $1e^{-4}$. Na trénovacie dáta sa aplikovala augmentácia dát pomocou preddefinovanej funkcie, ktorej parametre boli nastavené, 0,2 pre posun obrázka po dĺžke a šírke, rozsah možnej rotácie obrázka na 20, rozsah priblíženia obrázka na 0,15 a bola povolená možnosť horizontálneho otočenia obrázka. Taktiež bola nastavená funkcia *callback* ktorá zabezpečovala skoré ukončenie trénovania v prípade zhoršovania validačnej stratovej funkcie, kde v prípade stagnácie výsledku po 100 krokoch sa uloží posledný najefektívnejší model.



Obr. 5.5: *Priebeh stratovej funkcie prvého modelu*

Kapitola 6

Experimentálna časť

Cieľom tejto kapitoly je popísať použitý spôsob riešenia. Na začiatku sú špecifikácie jednotlivých výpočtových zariadení, na ktorých prebiehali experimenty, a následne je predstavená vytvorená databáza. Ďalej je popísaný spôsob hodnotenia úspešnosti použitých detekčných algoritmov. Nasledujú jednotlivé experimenty, u ktorých bude popísaný ich spôsob implementácie, použitá dátová sada a výsledky na danej dátovej sade. Prvá časť experimentov sa zaoberá testovaním daného detekčného algoritmu, jeho úspešnosť, silné a slabé stránky.

6.1 Použité zostavy

Keďže jeden z faktorov efektivity riešenia je rýchlosť, je nutné uviesť informácie ohľadom zariadení, na ktorých bolo experimentovanie vykonávané. Celkovo sa použilo 1 zariadenie a to: Notebook Lenovo Y740. Informácie o ňom sú uvedené nižšie.

- Operačný systém: Windows 10 Home 20H2
- Procesor: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
- Operačná pamäť: 16,0 GB DDR4
- Grafická karta: NVIDIA GeForce RTX 2070 Max-Q 6GB
- Pevný disk: SSD 1000GB

6.2 Spôsob hodnotenia a návrh jednotlivých testov

Pre hodnotenie jednotlivých modelov použijeme metódu pomocou Chybovej matice [23]. Jednotlivé stĺpce tejto matice ako možno vidieť na tabuľke 6.1 predstavujú triedy klasifikácie. Riadky predstavujú skutočnú triedu daného objektu.

- TP označuje počet objektov klasifikovaných ako správne, ktoré sú aj v skutočnosti správne.
- FP označuje počet objektov klasifikovaných ako správne, no v skutočnosti sú nesprávne.
- FN počet objektov klasifikovaných ako nesprávne, ktoré sú v skutočnosti správne.
- TN počet správne klasifikovaných nesprávnych objektov.

		Klasifikované hodnoty	
		Správne	**Nesprávne**
Skutočné hodnoty	**Správne**	TP	FN
	Nesprávne	FP	TN

Tabuľka 6.1: Chybová matica pre prvý natrénovaný model.

Na kvantifikovanie kvality detektora sa používajú tri hodnoty - presnosť, precíznosť a senzitivita. Presnosť určuje, koľko krát sa z celkových hodnotení podarilo objekt správne klasifikovať. Môžeme ju vyjadriť vzťahom 6.1. Precíznosť určuje úspešnosť klasifikátora pri určovaní správnych tried. Je vyjadrená v rovnici 6.2.Senzitivita vyjadruje pomer nájdených a nenájdených objektov. Je určená podľa rovnice 6.3.

$$\text{Presnosť} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

$$.\text{Precíznosť} = \frac{TP}{TP + FP} \tag{6.2}$$

$$\text{Senzitivita} = \frac{TP}{TP + FN} \tag{6.3}$$

Pre každý trénovaný model budú prebiehať tri sety testov. Kde prvý test bude hodnotenie prezenčného útoku papierovou fotografiou. Druhý bude testovať útoky fotografiou nachádzajúcou sa na obrazovke mobilného zariadenia. Nakoniec tretí test pre zhodnotenie odolnosti voči útokom realistických masiek. Všetky testy sa budú vykonávať viacero krát, a vyberie sa stredná hodnota výsledkov pre porovnanie. Taktiež ak nejaký test bude extrémne nevyhovujúci, testovanie modelu skončí na danom teste a posunie sa na ďalší možný model. Na základe vyhodnotených testov sa ďalej bude upravovať trénovanie modelu v prípade neuspokojenosti kvality modelu. Model ukazujúci najlepšie výsledky, bude ďalej testovaný v skutočných podmienkach, čo nám umožní objektívnejšie hodnotiť model, a pravdepodobne odhalí iné kvality ako len všeobecná presnosť a rýchlosť modelu.

6.3 Detekcia pomocou hlbokého učenia

Testovaný systém, kde model detekcie tváre bol použitý pretrénovaný model DNN[1] implementovaný pomocou caffe a využitý pomocou knižnice OpenCV[2]. Tento model bol využitý kvôli veľmi dobrej presnosti a rýchlosti klasifikácie. A samotný model na detekciu živosti podla návrhu v kapitole 4, bol implementovaný pomocou knižnice TensorFlow[3]. Trénovali sa 3 rôzne modely, ktoré sa navzájom líšili buď rožnou verziou použitej dátovej sady, alebo dĺžkou trénovaného času. Na trénovanie a testovanie bol využitá dátová sada CASIA a jeho rozšírená verzia a na validáciu bola extra využitá dátova sada NUAA [33]. Kde rozšírená verzia dátovej sady CASIA obsahovala dáta pre testovanie a trénovanie 3D útokov.

V tejto sekcií bude pri každom modeli uvedená hodnota stratovej funkcie (loss function) 4.3. Hodnota tejto funkcie dáva používateľovi najavo, ako dobre detektor funguje.

Prvý model

Prvý sa bude testovať trénovaný model predstavený v sekcií 5.4.

Prvý test

		Klasifikované hodnoty	
		Správne	**Nesprávne**
Skutočné hodnoty	**Správne**	TP = 90	FN = 20
	Nesprávne	FP = 20	TN = 81

Tabuľka 6.2: Chybová matica pre prvý test prvého modelu

$$\text{Presnosť} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{90 + 81}{90 + 81 + 20 + 20} = 0,81 \tag{6.4}$$

$$\text{Precíznosť} = \frac{TP}{TP + FP} = \frac{90}{90 + 20} = 0,81 \tag{6.5}$$

$$\text{Senzitivita} = \frac{TP}{TP + FN} = \frac{90}{90 + 20} = 0,81 \tag{6.6}$$

Chybová matica prvého testu 6.2 a vyhodnotenie testu v rovniciach 6.4, 6.5, a 6.6 ukázalo relatívne dobrú presnosť pri detekcii útoku pomocou papierovej fotky čo reprezentuje hodnota TP. Nepresnosť mohla nastať pri tvárach v rôznych polohách za čo s časti môže aj detektor tváre, ktorý nie vždy rozpozná tvár v zhoršených podmienkach.

[1] https://github.com/spmallick/learnopencv/tree/master/FaceDetectionComparison/models
[2] https://opencv.org/
[3] https://www.tensorflow.org/

Druhý test

		Klasifikované hodnoty	
		Správne	**Nesprávne**
Skutočné hodnoty	**Správne**	TP = 86	FN = 24
	Nesprávne	FP = 20	TN = 81

Tabuľka 6.3: Chybová matica pre druhý test prvého modelu

$$\text{Presnosť} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{86 + 81}{86 + 81 + 20 + 24} = 0,79 \tag{6.7}$$

$$\text{Precíznosť} = \frac{TP}{TP + FP} = \frac{86}{86 + 20} = 0,81 \tag{6.8}$$

$$\text{Senzitivita} = \frac{TP}{TP + FN} = \frac{86}{86 + 24} = 0,78 \tag{6.9}$$

Chybová matica druhého testu 6.3 a vyhodnotenie testu v rovniciach 6.7, 6.8, a 6.9 ukázalo relatívne dobrú presnosť pri detekcii útoku pomocou fotky zariadenia čo reprezentuje hodnota TP. Ne konzistentne však boli vyhľadávané tváre pod rôznym osvetlením a uhlom, kvôli vlastnostiam lesklosti obrazoviek mobilných zariadení.

Tretí test

		Klasifikované hodnoty	
		Správne	**Nesprávne**
Skutočné hodnoty	**Správne**	TP = 19	FN = 91
	Nesprávne	FP = 20	TN = 81

Tabuľka 6.4: Chybová matica pre tretí test prvého modelu

$$\text{Presnosť} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{19 + 81}{19 + 81 + 20 + 91} = 0,47 \tag{6.10}$$

$$\text{Precíznosť} = \frac{TP}{TP + FP} = \frac{19}{19 + 20} = 0,48 \tag{6.11}$$

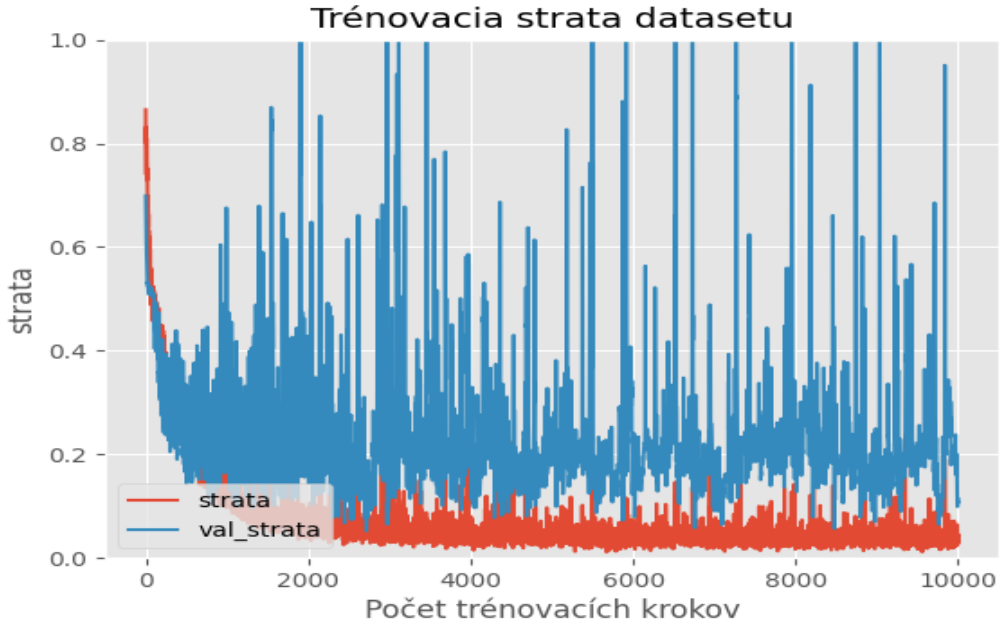
$$\text{Senzitivita} = \frac{TP}{TP + FN} = \frac{19}{19 + 91} = 0,17 \tag{6.12}$$

Chybová matica tretieho testu 6.4 a vyhodnotenie testu v rovniciach 6.10, 6.11, a 6.12 ukázalo relatívne chabú presnosť. Avšak je možno vidieť, že napriek neprítomnosti trénovaných dát pre 3D masky, určité množstvo bolo dochytené čo môže slúžiť ako referencia pre nasledujúci model s rozšírenejšou dátovou sadou.

Druhý model

V tomto modeli sa na tréning využila databáza CASIA s pridanými dátami pre útoky realistickými maskami, obsahovala 450 krátkych videí, ktoré sa rozsekali po rámcoch z ktorých 75% slúžilo na trénovanie a 25% na testovanie. V tomto modeli oproti prvému boli z videí vybrané všetky rámce pre dosiahnutie väčšej presnosti detekcie, a trénovanie prebiehalo za väčšieho množstva krokov. Pre návrh a priebeh testov boli manuálne vyberané dáta z testovacích dát.

Po 12 hodine a 18 minútach sa hodnota stratovej funkcie nejako významne nemenila, tak bol tréning zastavený. Vykonalo sa 10000 trénovacích krokov a výsledná hodnota stratovej funkcie bola 0,0160. Vývoj stratovej a stratovej validačnej funkcie je zobrazený na grafe 6.1. *batch size* pre testovacie a validačné dáta bol 64 a *learning rate* bol $1e^{-4}$. Na trénovacie dáta sa aplikovala augmentácia dát pomocou preddefinovanej funkcie, ktorej parametre boli nastavené, 0,2 pre posun obrázka po dĺžke a šírke, rozsah možnej rotácie obrázka na 20, rozsah priblíženia obrázka na 0,15 a bola povolená možnosť horizontálneho otočenia obrázka. Taktiež bola nastavená funkcia *callback* ktorá zabezpečovala skoré ukončenie trénovania v prípade zhoršovania validačnej stratovej funkcie, kde v prípade stagnácie výsledku po 1000 krokoch sa uloží posledný najefektívnejší model.



Obr. 6.1: *Priebeh stratovej funkcie druhého modelu*

Prvý test

		Klasifikované hodnoty	
		Správne	**Nesprávne**
Skutočné hodnoty	**Správne**	TP = 30	FN = 80
	Nesprávne	FP = 10	TN = 91

Tabuľka 6.5: Chybová matica pre prvý test druhého modelu

$$\text{Presnosť} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{30 + 91}{30 + 91 + 10 + 80} = 0,57 \tag{6.13}$$

$$\text{Precíznosť} = \frac{TP}{TP + FP} = \frac{30}{30 + 10} = 0,75 \tag{6.14}$$

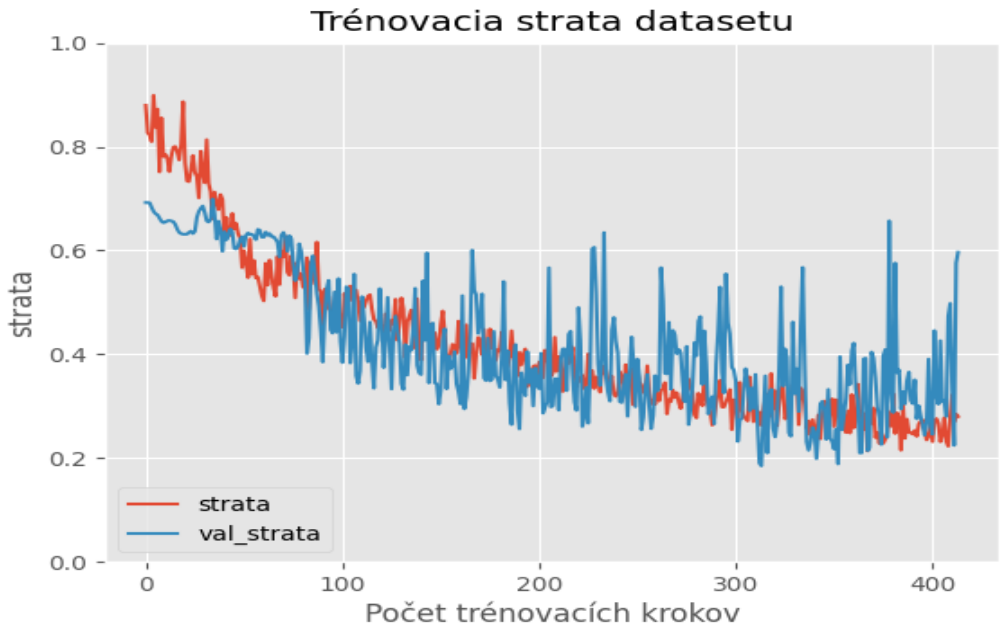
$$\text{Senzitivita} = \frac{TP}{TP + FN} = \frac{30}{30 + 80} = 0,27 \tag{6.15}$$

Oproti predchádzajúcemu modelu sa síce výrazne znížila stratová funkcia a počet nesprávnych detekcií (FP), ale zároveň zreteľne klesol aj počet správnych detekcií, napriek mnoho násobnej veľkosti databázy. Toto sa odrazilo vo zvýšení všetkých metrík, čo možno vidno v rovniciach 6.13, 6.14 a 6.15. Tieto fakty nasvedčujú tomu, že model bol pretrénovaný, a aj keď ukazoval nízku stratu pri trénovaní na kontrolujúcej dátovej sade, nie je taký efektívny na dátach, na ktorých netrénoval. Vzhľadom k pretrénovaní daného modelu nie je potreba na ňom vykonávať ďalšie testy.

Tretí model

V tomto modeli sa na tréning využila databáza CASIA s pridanými dátami pre 3D objekty, obsahovala 450 krátkych videí, ktoré sa rozsekali po rámcoch z ktorých 75% slúžilo na trénovanie a 25% na testovanie. V tomto modeli oproti druhému modelu bolo z videí vybrané menšie množstvo testov, ale stále značne viac ako z tretieho. Taktiež funkcia zabezpečujúca návrat (callback) bola upravená pre návrat presnejšieho modelu po prekročení hranice efektívnosti detektora.

Po 3 hodinách a 48 minútach sa hodnota stratovej funkcie nejako významne nemenila, tak bol tréning zastavený. Vykonalo sa 414 trénovaných krokov a výsledná hodnota stratovej funkcie bola 0,2341. Vývoj stratovej a stratovej validačnej funkcie je zobrazený na grafe 6.2. *batch size* pre testovacie a validačné dáta bol 8 a *learning rate* bol $1e^{-4}$. Na trénovacie dáta sa aplikovala augmentácia dát pomocou preddefinovanej funkcie, ktorej parametre boli nastavené, 0,2 pre posun obrázka po dĺžke a šírke, rozsah možnej rotácie obrázka na 20, rozsah priblíženia obrázka na 0,15 a bola povolená možnosť horizontálneho otočenia obrázka. Taktiež bola nastavená funkcia *callback* ktorá zabezpečovala skoré ukončenie trénovania v prípade zhoršovania validačnej stratovej funkcie, kde v prípade stagnácie výsledku po 100 krokoch sa uloží posledný najefektívnejší model.



Obr. 6.2: *Priebeh stratovej funkcie tretieho modelu*

Prvý test

		Klasifikované hodnoty	
		Správne	**Nesprávne**
Skutočné hodnoty	**Správne**	TP = 99	FN = 11
	Nesprávne	FP = 14	TN = 87

Tabuľka 6.6: Chybová matica pre prvý test tretieho modelu

$$\text{Presnosť} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{99 + 87}{99 + 87 + 14 + 11} = 0,88 \tag{6.16}$$

$$\text{Precíznosť} = \frac{TP}{TP + FP} = \frac{99}{99 + 14} = 0,87 \tag{6.17}$$

$$\text{Senzitivita} = \frac{TP}{TP + FN} = \frac{99}{99 + 11} = 0,9 \tag{6.18}$$

Z rovníc 6.16 až 6.18 a chybovej matice 6.6 vidíme, že oproti predchádzajúcim modelom nastalo k zlepšeniu špeciálne oproti druhému modelu.

Druhý test

		Klasifikované hodnoty	
		Správne	**Nesprávne**
Skutočné hodnoty	**Správne**	TP = 71	FN = 39
	Nesprávne	FP = 20	TN = 81

Tabuľka 6.7: Chybová matica pre druhý test tretieho modelu

$$\text{Presnosť} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{71 + 81}{71 + 81 + 20 + 39} = 0,72 \tag{6.19}$$

$$\text{Precíznosť} = \frac{TP}{TP + FP} = \frac{71}{71 + 20} = 0,78 \tag{6.20}$$

$$\text{Senzitivita} = \frac{TP}{TP + FN} = \frac{71}{71 + 39} = 0,64 \tag{6.21}$$

Oproti predchádzajúcemu testu na 3D masky v prvom modeli 6.3 sa výsledky značne zlepšili. Ale nepredstavujú veľké zlepšenie, čo hlavne ovplyvňuje relatívne menšia porcia dátovej sady pre detekciu 3D útokov.

6.4 Testy na videu

Táto sekcia sa bude zaoberať efektívnosťou modelu na videu. Pre tieto testy bol vybraný tretí model pre jeho najlepšie výsledky z troch. Testy budu vykonávane na 2 video kamerách, integrovanej kamere v zariadení Lenovo Y740 a externej kamere SpotLight Pro. Cieľom testov je zistenie efektívnosti a rýchlosti riešenia na reálnom prípade, a zistiť rozdiel funkčnosti na rozdielnych kvalitách vstupných zariadení.

Lenovo Y740:

- Rozlíšenie snímača 1280 × 720.
- Pevné zaostrovanie.

Webová kamera SpotLight Pro s LED svetlami:

- Rozlíšenie snímača 640 × 480.
- Manuálne zaostrovanie.

Prvý test

Cieľom tohoto testu bolo otestovať model detekcie živosti na prípade zo skutočného života a za druhé, poskytnúť referenčné dáta pre porovnávanie s ostatnými testami. Test prebie-

hal na zariadení Lenovo Y740 za využitia externej kamery SpotLight Pro. Test prebiehal pozorovaním voľným okom a zapisovaním dát.

		Klasifikované hodnoty	
		Správne	**Nesprávne**
Skutočné hodnoty	**Správne**	TP = 10	FN = 0
	Nesprávne	FP = 3	TN = 7

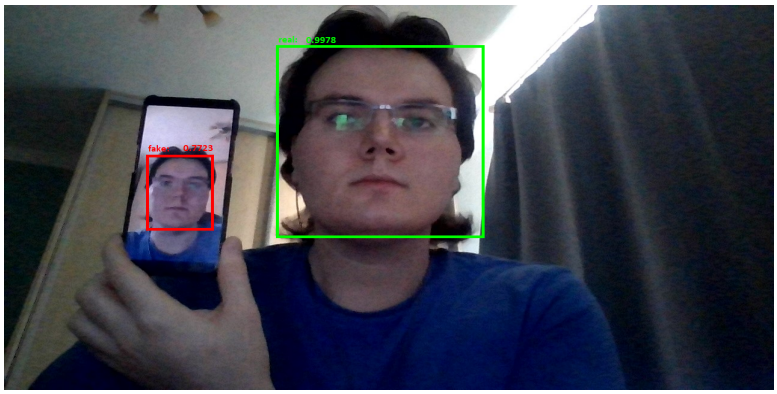
Tabuľka 6.8: Chybová matica pre druhý test tretieho modelu

$$\text{Presnosť} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{10 + 7}{10 + 7 + 3 + 0} = 0,85 \tag{6.22}$$

$$\text{Precíznosť} = \frac{TP}{TP + FP} = \frac{10}{10 + 3} = 0,76 \tag{6.23}$$

$$\text{Senzitivita} = \frac{TP}{TP + FN} = \frac{9}{9 + 0} = 1 \tag{6.24}$$

Chybovou maticou pre tento test je tabuľka 6.8 a jeho vyhodnotenie sa nachádza v rovniciach 6.22 až 6.24. Model dokázal správne označiť takmer všetky výskyty tváre, aj napriek malému množstvu vzoriek, ktoré vysoko skresľujú presnosť detektora je možno pozorovať iné anomálie. Model dokázal zaznamenať ohodnotiť všetky skutočné tváre správne, avšak rozpoznávanie podvrhov na kvalitnejšom displeji bol problematický. Taktiež osvetlenie pomáhalo pri detekcií kde, pri silnejšom osvetlení pracoval detektor výrazne lepšie. Na zostave Lenovo Y740 za využitia GPU bol čas na detekciu prakticky okamžitý.



Obr. 6.3: *Jeden rámec z detekcie pre lepšiu kvalitu videa*

Druhý test

Druhý test prebiehal v podobných podmienkach a s rovnakými jednotkami dát ako prvý test. Zariadenie taktiež Lenovo Y740 s rozdielom využitia integrovanej kamery, ktorá sa nachádza pod obrazovkou daného zariadenia, čiže okrem kvality videa bude rozdiel taktiež v uhle pohľadu.

		Klasifikované hodnoty	
		Správne	**Nesprávne**
Skutočné hodnoty	**Správne**	TP = 10	FN = 0
	Nesprávne	FP = 3	TN = 7

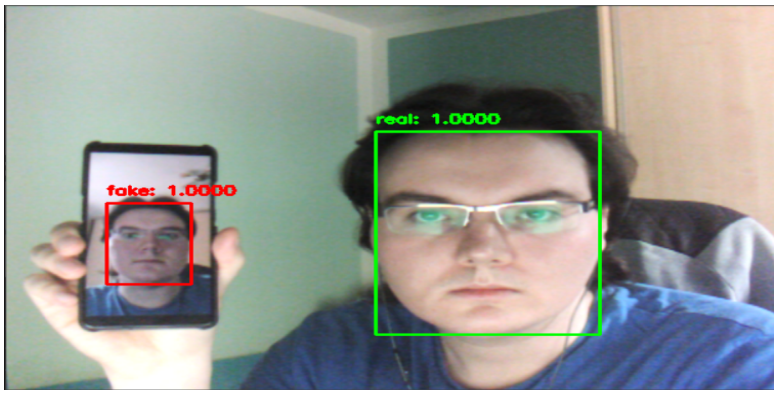
Tabuľka 6.9: Chybová matica pre druhý test tretieho modelu

$$\text{Presnosť} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{10 + 7}{10 + 7 + 3 + 0} = 0,85 \tag{6.25}$$

$$\text{Precíznosť} = \frac{TP}{TP + FP} = \frac{10}{10 + 3} = 0,76 \tag{6.26}$$

$$\text{Senzitivita} = \frac{TP}{TP + FN} = \frac{9}{9 + 0} = 1 \tag{6.27}$$

Z chybovej matice 6.9 a rovníc 6.25 - 6.27 môžeme pozorovať rovnaké výsledky ako v predchádzajúcom teste. Kvôli nízkemu počtu dát je možné skreslenie výsledkov, ale zo samotného priebehu testu bolo možné pozorovať, priemerne vyššiu rozhodovaciu hranicu v falošných prípadoch, aj keď nie dosť aby sa to dalo priradiť lepšej kvalite videa.



Obr. 6.4: *Jeden rámec z detekcie pre horšiu kvalitu videa*

6.5 Zhrnutie

Za účelom detekcie tváre z kamery bol navrhnutý model siete hlbokého učenia. Ktorý bol trénovaný vo viacerých inštanciách, z ktorých najlepšie výsledky v rozmedzí všetkých kategórií útokov dosiahol tretí model, ktorý po relatívne krátkej dobe trénovania 2 hodinách a 48 minútach za ktorých sa vykonalo 385 tréningových krokov. Tento model dokázal určiť takmer štyri z piatich testovaných tvár korektne a jeho celková presnosť činila medzi 72% - 88%, čo značí rozdiel medzi triedami útoku ako sú papierové fotky a silikónové masky. Jeho veľkým pozitívom je relatívne krátka doba potrebná na trénovanie, a vysoká presnosť pri skutočných tvárach. Ďalej bol tento model testovaný na reálnych prípadoch kde preukázal vysokú rýchlosť a uspokojujúcu efektivitu pri rozpoznaní živosti tváre. Čas na rozpoznanie tváre za vyžitia grafickej karty bol zanedbateľný ako pre videá z vyššou kvalitou tak aj s nižšou. Väčší rozdiel ako kvalita videa spôsobovala intenzita a uhol osvetlenia.

6.6 Možnosti rozšírenia a problémy pri implementácií

Predložený model bol trénovaný na viacero typov útokov, ktoré boli zaradené do jednej triedy čo môže spôsobovať nezhody v segmentálnej mape definujúcu triedu podvrhu, tým pádom by bolo dobré do budúcna vytvoriť viacero tried pre hlboké učenie. Taktiež pre zvýšenie efektivity riešenia bez ovplyvnenia rýchlosti je možné pridať rôzne podmienky, ako napríklad pre osvetlenie, kde pod určitou hranicou svetlosti detekcia neuspeje. Alebo ak by bol model aplikovaný na real-time detekciu živosti pomocou systémov CCTV, kde by detekcia upozornila na možné pokusy o nepovolený prístup do areálu s obmedzeným prístupom. V tejto práci mohli nastať rôzne nepresnosti z nasledujúcich dôvodov:

- Zostava Lenovo Y749 využívala pre knižnicu Tensorflow grafickú kartu, a tak nebola potvrdená rýchlosť modelu pri horších podmienkach.
- Využitá knižnica Tensorflow a jej Object Detection API je orientovaný skôr na výskum než rýchlosť. Preto v ňom chýbajú optimalizácie pre rýchlosť detekcie a sú vykonávané operácie, ktoré sú z hľadiska rýchlosti nevyžiadané.
- Algoritmus je zaťažený rôznymi diskovými operáciami načítania, alebo uloženia pomedzi detekciu modelov.
- Nebol presne definovaný vplyv osvetlenia na detekciu, len bol pozorovaný jav, ktorý naznačuje lepšiu efektivitu pri vyššej hodnote jasnosti.
- Na detekciu tváre bol testovaný iba jeden model, ktorý môže silno ovplyvniť výsledky testov. Bolo by vhodné testovanie za pomocí iných pretrénovaných modelov detekcie tváre pre porovnanie.
- Prvotne bola plánovaná implementácia algoritmu v jazyku C++, ale nastávali problémy s konvertovaním trénovaného modelu z TensorFlow do TensorFlow.js a jeho následná deserializácia v TensorFlow 2, kde nastával problém kvôli chybe deserializácii GRU vrstvy. A taktiež nastávali problémy s prácou knižníc nad grafickou kartou. Nakoniec som sa takýmto problémom vyhol pri konverzii celého projektu do jazyka Python.

Kapitola 7

Záver

V teoretickej časti sa táto práca zaoberala základnými charakteristikami detekcie tváre a najznámejšími spôsobmi metód spojených s detekciou tváre. Následne podrobnejším vysvetlením problematík spojených detekciou živosti tváre, ako sú typy útokov a všeobecná kategorizácia metód na detekciu. Prehľad podrobnejších vysvetlení podstaty jednotlivých techník pre detekciu životnosti ľudských tvárí a ich efektívnosť. Najbežnejší problém, ktorý bol pozorovaný v prípade mnohých metód sú dátové sady, ktoré hrajú dôležitú úlohu pri výkone riešení detekcie živosti.Dátové sady by mali byť informatívne a rozmanité, aby napodobňovali očakávané scenáre aplikácie. Neinteraktívne video sekvencie musia obsahovať interaktívne sekvencie, v ktorých používatelia vykonávajú určité úlohy. Modely detekcie živosti by mali očakávať možné komplexnejšie útoky, ako sú 3D napodobeniny tváre, a vylepšené informácie o textúre.

V druhej polovici práce sa zaoberala návrhom algoritmu pre detekciu živosti a jeho implementáciou. Výsledkom praktickej časti je algoritmus, ktorý na základe videa rozpoznáva skutočnú tvár od falošnej. Daný algoritmus bol testovaný, a jeho úspešnosť dosahovala 75% čo vzhľadom na rôzne možné vplyvy je úspešný výsledok. Je možno na testovaní pozorovať mnoho faktorov ovplyvňujúce výsledky, ktoré sa nepodarilo v tejto práci definovať a izolovať. Napriek nedostatkom má toto riešenie výhodu v jeho rýchlosti.

Vďaka tejto práci som získal vedomosti o tom čo predstavujú neurónové siete, ako fungujú a kde sa využívajú v rámci problémov spojených s počítačovým videním. Taktiež vedomosti o možnom nahradení neurónovej siete na základe potreby a požiadavkou daného problému - v tomto prípade, problém detekcie živosti tváre a možné techniky ako k takémuto problému pristupovať.

Literatúra

[1] *Optimalizátor Adam*. 2021. [Online; navštívené 9.5.2021]. Dostupné z: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam.

[2] Abu Mostafa, Y. S. Learning from hints in neural networks. *Journal of Complexity*. 1990. [Online; navštívené 9.5.2021], zv. 6, č. 2, s. 192–198. DOI: https://doi.org/10.1016/0885-064X(90)90006-Y. ISSN 0885-064X. Dostupné z: https://www.sciencedirect.com/science/article/pii/0885064X9090006Y.

[3] Al Allaf, O. a Aiden, J. Review of Face Detection Systems Based Artificial Neural Networks Algorithms. *International Journal of Multimedia and Its Applications*. Február 2014. [Online; navštívené 9.5.2021], zv. 6. DOI: 10.5121/ijma.2014.6101. Dostupné z: https://www.researchgate.net/publication/347888449_Review_of_Face_Detection_Systems_Based_Artificial_Neural_Networks_Algorithms.

[4] Albawi, S., Mohammed, T. A. a Al-Zawi, S. Understanding of a convolutional neural network. In: *2017 International Conference on Engineering and Technology (ICET)*. 2017. [Online; navštívené 9.5.2021], s. 1–6. DOI: 10.1109/ICEngTechnol.2017.8308186. Dostupné z: https://ieeexplore.ieee.org/document/8308186.

[5] Cadoni, M., Bicego, M. a Grosso, E. 3D Face Recognition Using Joint Differential Invariants. In: Tistarelli, M. a Nixon, M. S., ed. *Advances in Biometrics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. [Online; navštívené 9.5.2021], s. 279–288. ISBN 978-3-642-01793-3. Dostupné z: https://www.researchgate.net/publication/33680946_3D_Face_Recognition_Using_Joint_Differential_Invariants.

[6] Chan, P. P. K., Liu, W., Chen, D., Yeung, D. S., Zhang, F. et al. Face Liveness Detection Using a Flash Against 2D Spoofing Attack. *IEEE Transactions on Information Forensics and Security*. 2018. [Online; navštívené 9.5.2021], zv. 13, č. 2, s. 521–534. DOI: 10.1109/TIFS.2017.2758748. Dostupné z: https://ieeexplore.ieee.org/abstract/document/8055588.

[7] Chang, C.-C. a Lin, C.-J. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*. Acm New York, NY, USA. 2011. [Online; navštívené 9.5.2021], zv. 2, č. 3, s. 1–27. Dostupné z: https://dl.acm.org/doi/10.1145/1961189.1961199.

[8] Chen, Y., Chang, H., Jin, M. a Zhang, D. Ensemble Neural Networks (ENN): A gradient-free Stochastic method. *Neural Networks*. December 2018. [Online;

navštívené 9.5.2021], zv. 110. DOI: 10.1016/j.neunet.2018.11.009. Dostupné z: https://www.researchgate.net/publication/329367972_Ensemble_Neural_Networks_ENN_A_gradient-free_Stochastic_method/stats.

[9] Chingovska, I., Anjos, A. a Marcel, S. On the effectiveness of local binary patterns in face anti-spoofing. In: IEEE. *2012 BIOSIG-proceedings of the international conference of biometrics special interest group (BIOSIG)*. 2012. [Online; navštívené 9.5.2021], s. 1–7. Dostupné z: http://publications.idiap.ch/downloads/papers/2012/Chingovska_IEEEBIOSIG2012_2012.pdf.

[10] Dong, J., Wang, W. a Tan, T. CASIA Image Tampering Detection Evaluation Database. In: *2013 IEEE China Summit and International Conference on Signal and Information Processing*. 2013. [Online; navštívené 9.5.2021], s. 422–426. DOI: 10.1109/ChinaSIP.2013.6625374. Dostupné z: https://ieeexplore.ieee.org/document/6625374.

[11] Eum, S., Suhr, J. K. a Kim, J. Face recognizability evaluation for ATM applications with exceptional occlusion handling. In: *CVPR 2011 WORKSHOPS*. 2011. [Online; navštívené 9.5.2021], s. 82–89. DOI: 10.1109/CVPRW.2011.5981883. Dostupné z: https://ieeexplore.ieee.org/abstract/document/5981883.

[12] Hao, H., Pei, M. a Zhao, M. Face Liveness Detection Based on Client Identity Using Siamese Network. In: Lin, Z., Wang, L., Yang, J., Shi, G., Tan, T. et al., ed. *Pattern Recognition and Computer Vision*. Cham: Springer International Publishing, 2019. [Online; navštívené 9.5.2021], s. 172–180. ISBN 978-3-030-31654-9. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-030-31654-9_15.

[13] Hlaváč, V. Fourierova transformace v 1D a 2D. *VYSOKÉ UČENÍ TECHNICKÉ V PRAZE, české, Sv. Přednáška*. 2012. [Online; navštívené 9.5.2021]. Dostupné z: http://people.ciirc.cvut.cz/hlavac/TeachPresCz/11DigZprObr/12FourierTxCz.pdf.

[14] Jee, H.-K., Jung, S.-U. a Yoo, J.-H. Liveness Detection for Embedded Face Recognition System. *International Journal of Computer and Information Engineering*. World Academy of Science, Engineering and Technology. 2008. [Online; navštívené 9.5.2021], zv. 2, č. 6, s. 2142 – 2145. ISSN eISSN: 1307-6892. Dostupné z: https://publications.waset.org/vol/18.

[15] Kevin Alan Tussy, J. R. *Biometric Liveness Detection Explained* [online]. [cit. 2020-01-09]. Dostupné z: https://www.liveness.com/.

[16] Kim, G., Eum, S., Suhr, J. K., Kim, D. I., Park, K. R. et al. Face liveness detection based on texture and frequency analyses. In: *2012 5th IAPR International Conference on Biometrics (ICB)*. 2012. [Online; navštívené 9.5.2021], s. 67–72. DOI: 10.1109/ICB.2012.6199760. Dostupné z: https://ieeexplore.ieee.org/abstract/document/6199760.

[17] Kim, S., Yu, S., Kim, K., Ban, Y. a Lee, S. Face liveness detection using variable focusing. In: *2013 International Conference on Biometrics (ICB)*. 2013. [Online; navštívené 9.5.2021], s. 1–6. DOI: 10.1109/ICB.2013.6613002. Dostupné z: https://ieeexplore.ieee.org/abstract/document/6613002.

[18] Kim, S., Ban, Y. a Lee, S. Face Liveness Detection Using Defocus. *Sensors (Basel, Switzerland)*. Január 2015. [Online; navštívené 9.5.2021], zv. 15, s. 1537–1563. DOI: 10.3390/s150101537. Dostupné z: https://www.researchgate.net/publication/270962286_Face_Liveness_Detection_Using_Defocus.

[19] Kähm, O. a Damer, N. 2D face liveness detection: An overview. In: *2012 BIOSIG - Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG)*. 2012. [Online; navštívené 9.5.2021], s. 1–12. Dostupné z: https://ieeexplore.ieee.org/abstract/document/6313547.

[20] Lagorio, A., Tistarelli, M., Cadoni, M., Fookes, C. a Sridharan, S. Liveness detection based on 3D face shape analysis. In: *2013 International Workshop on Biometrics and Forensics (IWBF)*. 2013. [Online; navštívené 9.5.2021], s. 1–4. DOI: 10.1109/IWBF.2013.6547310. Dostupné z: https://ieeexplore.ieee.org/abstract/document/6547310.

[21] Lecun, Y., Bottou, L., Bengio, Y. a Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998. [Online; navštívené 9.5.2021], zv. 86, č. 11, s. 2278–2324. DOI: 10.1109/5.726791. Dostupné z: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=726791.

[22] Mohamed, A. A., Nagah, M. M., Abdelmonem, M. G., Ahmed, M. Y., El Sahhar, M. et al. Face Liveness Detection Using a sequential CNN technique. In: *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. 2021. [Online; navštívené 9.5.2021], s. 1483–1488. DOI: 10.1109/CCWC51732.2021.9376030. Dostupné z: https://ieeexplore.ieee.org/document/9376030.

[23] Narkhede, S. Understanding Confusion Matrix. *Towards Data Science*. 2018. [Online; navštívené 9.5.2021]. Dostupné z: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62.

[24] Nilsson, M., Nordberg, J. a Claesson, I. Face Detection using Local SMQT Features and Split up Snow Classifier. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. 2007. [Online; navštívené 9.5.2021], sv. 2, s. II–589–II–592. DOI: 10.1109/ICASSP.2007.366304. Dostupné z: https://ieeexplore.ieee.org/abstract/document/4217477.

[25] Ojala, T., Pietikainen, M. a Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2002. [Online; navštívené 9.5.2021], zv. 24, č. 7, s. 971–987. DOI: 10.1109/TPAMI.2002.1017623. Dostupné z: https://ieeexplore.ieee.org/abstract/document/1017623.

[26] O'Shea, K. a Nash, R. *An Introduction to Convolutional Neural Networks*. 2015. [Online; navštívené 9.5.2021]. Dostupné z: https://arxiv.org/pdf/1511.08458.pdf.

[27] Perez, L. a Wang, J. *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. 2017. [Online; navštívené 9.5.2021]. Dostupné z: https://arxiv.org/pdf/1712.04621.pdf.

[28] S, T. a N, M. Detection, Segmentation and Recognition of Face and its Features Using Neural Network. *Journal of Biosensors & Bioelectronics*. OMICS Publishing Group. 2016. [Online; navštívené 9.5.2021], zv. 7, č. 2. DOI: 10.4172/2155-6210.1000210. ISSN 2155-6210. Dostupné z: http://dx.doi.org/10.4172/2155-6210.1000210.

[29] Saragih J.M., C. J. Deformable Model Fitting by Regularized Landmark Mean-Shift. *International Journal of Computer Vision*. 2011. [Online; navštívené 9.5.2021], zv. 91, s. 200–215. DOI: 10.1007/s11263-010-0380-4. ISSN 1573-1405. Dostupné z: https://doi.org/10.1007/s11263-010-0380-4.

[30] Savran, A., Alyuz, N., Dibeklioglu, H., Celiktutan, O., Gokberk, B. et al. Bosphorus Database for 3D Face Analysis. In: Schouten, B., Juul, N. C., Drygajlo, A. a Tistarelli, M., ed. *Biometrics and Identity Management*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. [Online; navštívené 9.5.2021], s. 47–56. ISBN 978-3-540-89991-4.

[31] Singh, M. a Arora, A. A robust anti-spoofing technique for face liveness detection with morphological operations. *Optik*. 2017. [Online; navštívené 9.5.2021], zv. 139, s. 347 – 354. DOI: https://doi.org/10.1016/j.ijleo.2017.04.004. ISSN 0030-4026. Dostupné z: http://www.sciencedirect.com/science/article/pii/S0030402617303935.

[32] Srisha, R. a Khan, A. Morphological Operations for Image Processing : Understanding and its Applications. In:. December 2013. [Online; navštívené 9.5.2021]. Dostupné z: https://www.researchgate.net/publication/272484795_Morphological_Operations_for_Image_Processing_Understanding_and_its_Applications.

[33] Tan, X., Li, Y., Liu, J. a Jiang, L. Face liveness detection from a single image with sparse low rank bilinear discriminative model. In: Springer. *European Conference on Computer Vision*. 2010. [Online; navštívené 9.5.2021], s. 504–517. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-642-15567-3_37.

[34] Triantafyllidou, D. a Tefas, A. A Fast Deep Convolutional Neural Network for Face Detection in Big Visual Data. In: Angelov, P., Manolopoulos, Y., Iliadis, L., Roy, A. a Vellasco, M., ed. *Advances in Big Data*. Cham: Springer International Publishing, 2017. [Online; navštívené 9.5.2021], s. 61–70. ISBN 978-3-319-47898-2. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-319-47898-2_7.

[35] Viola, P. a Jones, M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 2001, sv. 1, s. I–I. DOI: 10.1109/CVPR.2001.990517. Dostupné z: https://ieeexplore.ieee.org/abstract/document/990517.

[36] Wang, T., Yang, J., Lei, Z., Liao, S. a Li, S. Z. Face liveness detection using 3D structure recovered from a single camera. In: *2013 International Conference on Biometrics (ICB)*. 2013. [Online; navštívené 9.5.2021], s. 1–6. DOI: 10.1109/ICB.2013.6612957. Dostupné z: https://ieeexplore.ieee.org/abstract/document/6612957#full-text-header.

[37] Yamashita R., D. R. e. a. Convolutional neural networks: an overview and application in radiology. *Insights Imaging*. 2018. [Online; navštívené 9.5.2021], zv. 9, s. 611–629. DOI: https://doi.org/10.1007/s13244-018-0639-9. ISSN 1869-4101. Dostupné z: https://doi.org/10.1007/s13244-018-0639-9.

[38] Yegulalp, S. What is TensorFlow? The machine learning library explained. *InfoWorld*. 2019. [Online; navštívené 9.5.2021]. Dostupné z: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html.

[39] Yimyam, W., Pinthong, T., Chumuang, N. a Ketcham, M. Face Detection Criminals through CCTV Cameras. In: *2018 14th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*. 2018, s. 351–357. DOI: 10.1109/SITIS.2018.00061. Dostupné z: https://ieeexplore.ieee.org/document/8705938.

[40] Zhang, H., Zhang, L. a Jiang, Y. Overfitting and Underfitting Analysis for Deep Learning Based End-to-end Communication Systems. In: *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. 2019. [Online; navštívené 9.5.2021], s. 1–6. DOI: 10.1109/WCSP.2019.8927876. Dostupné z: https://ieeexplore.ieee.org/abstract/document/8927876.

Príloha A

Obsah priloženého DVD

Súčasťou práce je aj priložené DVD s nasledujúcou štruktúrou:

- **README** - Používateľská príručka.
- **train.py** - Trénovací skript.
- **liveness.model** - Natrénovaný model detekcie živosti.
- **le.pickle** - Enkóder pre model detekcie živosti.
- **gather_data.py** - Skript pre zbieranie dát z videí.
- **detector_demo.py** - Demo verzia detekcie pre určité. testy
- **detector.py** - Kompletná verzia detekcie.
- **liveness_detection** - Adresár obsahujúci model. detekcie
- **face_detector** - Adresár obsahujúci detektor tváre.
- **bp_text** - Adresár obsahujúci zdrojové súbory pre generovanie technickej správy.
- **xvaloo00_BP.pdf**- Táto bakalárska práca v pdf. formáte.
- **videos**- Adresár s dátovou sadou videí.
- **dataset**- Adresár s dátovou sadou obrázkov.