



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**DETEKCIA A VYLEPŠENIE KVALITY OBJEKTU TVÁRE
OSOBY V NEKVALITNÝCH ZDROJOVÝCH SNÍMKACH**

DETECTION AND QUALITY IMPROVEMENT OF FACE OBJECTS IN LOW-QUALITY SOURCE
IMAGES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RICHARD ŠOLTIS

VEDOUCÍ PRÁCE

SUPERVISOR

Prof. Ing., Dipl.-Ing. MARTIN DRAHANSKÝ, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Šoltis Richard**
Program: Informační technologie
Název: **Detekce a vylepšení kvality objektu obličeje osoby v nekvalitních zdrojových snímcích**
Detection and Quality Improvement of Face Objects in Low-Quality Source Images

Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte literaturu týkající se detekce, zpracování a rozpoznání obličeje osob, dále algoritmů pro vylepšení kvality snímků obličeje. Seznamte se s dostupnými daty (obrázky a videa) s nekvalitními záznamy osob.
2. Navrhněte algoritmické řešení pro detekci a vylepšení kvality objektu obličeje osoby v nekvalitních zdrojových snímcích.
3. Vámi navržené řešení z bodu 2 implementujte.
4. Proveďte otestování Vašeho řešení. Shrňte dosažené výsledky a diskutujte možná vylepšení.

Literatura:

- ZANGENEH, Erfan; RAHMATI, Mohammad; MOHSENZADEH, Yalda. Low resolution face recognition using a two-branch deep convolutional neural network architecture. *Expert Systems with Applications*, 2020, 139: 112854.
- ZHOU, Yuqian; LIU, Ding; HUANG, Thomas. Survey of face detection on low-quality images. In: *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018. p. 769-773.
- SOUNDARARAJAN, Rajiv; BISWAS, Soma. Machine vision quality assessment for robust face detection. *Signal Processing: Image Communication*, 2019, 72: 92-104.
- MUTNEJA, Vikram; SINGH, Satvir. GPU accelerated face detection from low resolution surveillance videos using motion and skin color segmentation. *Optik*, 2018, 157: 1155-1165.

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: **Drahanský Martin, prof. Ing., Dipl.-Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 11. listopadu 2020

Abstrakt

Cieľom tejto práce bolo zostrojiť algoritmus na detekciu ľudskej tváre z nekvalitných zdrojových snímok a následne vylepšenie obrazu ľudskej tváre. Výsledkom práce je aplikácia s grafickým rozhraním, ktorá zo vstupných snímok detekuje objekty ľudskej tváre a následne tieto zdetekované tváre vylepšuje z hľadiska kvality a veľkosti. Pri vytváraní aplikácie boli použité súčasné techniky a algoritmy, ako sú neurónové siete, ktoré tvorili základ pre detekciu ako aj pre vylepšenie obrazu, *S3FD* na detekciu a v neposlednom rade GAN siete na vylepšenie obrazu. Súčasťou práce je testovanie jednotlivých častí aplikácie v preddefinovaných scenároch ako aj testovanie komplexného behu aplikácie.

Abstract

The aim of this thesis was to construct an algorithm for the detection of human face from poor quality source images and subsequently improving the image of human face. The result of the work is an application with a graphical interface which detects human face objects from the input images and then improves these inherited faces from the point of quality and size. When creating the application, current techniques and algorithms such as neuron networks were used. They formed the basis for detection and image improvement, *S3FD* detection and last but not least the GAN network to improve the image. Part of the thesis is testing the individual parts of the application in predefined scenarios as well as testing a comprehensive run application.

Klíčové slová

detekcia, vylepšenie obrazu, neurónové siete, *S3FD*, GAN

Keywords

detection, image enhancement, neural network, *S3FD*, GAN

Citácia

ŠOLTIS, Richard. *Detekcia a vylepšenie kvality objektu tváre osoby v nekvalitných zdrojových snímkach*. Brno, 2021. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Prof. Ing., Dipl.-Ing. Martin Drahanský, Ph.D.

Detekcia a vylepšenie kvality objektu tváre osoby v nekvalitných zdrojových snímkach

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne a pod odborným vedením pána profesora Martina Drahanského. Uviedol som všetky publikácie a zdroje, z ktorých som čerpal.

.....
Richard Šoltis
8. mája 2021

Podakovanie

Ďakujem môjmu školiteľovi pánovi profesorovi Martinovi Drahanskému za užitočné pripomienky, ochotu a usmernenie pri písaní bakalárskej práce. Taktiež ďakujem spoločnosti IT4BI s.r.o a MetaCentrum za výpomoc pri trénovaní modelu neurónovej siete v praktickej časti bakalárskej práce v podobe výpočtového servera.

Obsah

1	Úvod	3
2	Detekcia tváre	4
2.1	Prehľad existujúcich algoritmov	4
2.1.1	Viola-Jones detektor	4
2.1.2	CNN	6
2.1.3	Fast R-CNN	8
2.1.4	YOLO	9
2.1.5	SSSFD	10
3	Vylepšenie kvality objektu tváre	11
3.1	Prehľad dostupných technológií	11
3.1.1	Vyrovnanie histogramu	12
3.1.2	Adaptívne vyrovnanie histogramu	12
3.1.3	Interpolácia najbližšieho suseda	13
3.1.4	Bilineárna interpolácia	13
3.1.5	Bikubická interpolácia	14
3.1.6	Vylepšenie obrazu použitím CNN	15
3.1.7	Vylepšenie obrazu použitím GAN	16
4	Návrh a implementácia	18
4.1	Detekcia	18
4.1.1	Trénovacia sada	18
4.1.2	Anotácia testovacej sady	19
4.1.3	Trénovanie modelu	19
4.1.4	MetaCentrum	19
4.1.5	Priebeh tréovania	20
4.1.6	Konfigurácia tréovania	21
4.1.7	Spustenie tréovacieho procesu	21
4.1.8	Spustenie detekcie	21
4.2	Vylepšenie obrazu	22
4.2.1	Trénovacia sada	22
4.2.2	Zostavenie tréovacej sady	22
4.2.3	Trénovanie modelu	23
4.2.4	Konfigurácia tréovacieho procesu	24
4.2.5	Inicializácia tréovacieho procesu	24
4.3	Spustenie vylepšovania obrazu	24
4.4	Možné problémy pri tréovaní GAN sietí	24

4.5	Užívateľské rozhranie	25
4.5.1	Návrh užívateľského rozhrania	25
4.5.2	Grafické podporné knižnice	26
4.6	Implementácia	26
4.6.1	Detekcia a vylepšenie obrazu	26
4.6.2	Grafické rozhranie	27
5	Testovanie	28
5.1	Testovanie detektora	28
5.1.1	Testovacia sada č. 1	29
5.1.2	Testovacia sada č. 2	29
5.1.3	Testovacia sada č. 3	30
5.2	Testovanie vylešenia obrazu	31
6	Záver	33
	Literatúra	34

Kapitola 1

Úvod

Úlohou tejto bakalárskej práce je detekovať ľudské tváre zo zdrojových snímok a následné vylepšenie obrazu. Konkrétne sa zameriavame na nekvalitné zdrojové snímky, u ktorých je detekcia a následný proces vylepšenia obrazu náročnejší. Detekcia v dnešnej dobe dosahuje úctihodné výsledky, ale detekovanie ľudských tvárí v nízkom rozlíšení je oblasť, v ktorej je veľmi ťažko dosiahnuť milník, ktorý by určoval či je výsledok dostatočný alebo nie. V súčasnosti sa človek stretáva s čoraz väčším počtom snímok, ktorých zdroj je nekvalitný fotoaparát alebo kamera. To následne sťažuje proces detekovania, ktorý je neoddeliteľnou súčasťou ďalších procesov, ako napr. rozpoznávanie, identifikácia alebo vylepšenie obrazu.

V súčasnosti je nevyhnutné aby práca s obrazovými dátami z hľadiska kvality bola čo najjednoduchšia, čím sa zjednoduší celá práca s danými obrazovými dátami. Veľkým problémom, ktorý ostáva aj v dnešnej dobe, je používanie nie veľmi kvalitných zachytávajúcich zariadení, ktoré pridávajú snímke rôznorodé defekty, ako napr. šum, kontrast či rozmazanie.

Prínosom tejto práce je aj použitie výslednej aplikácie políciou Českej republiky na detekciu a rozpoznávanie tvárí potenciálnych páchatelov pri dokazovaní viny v trestných konaniach.

V kapitolách **2** a **3** sa čitateľ môže oboznámiť s dnešnými technikami a algoritmi, ktoré sa využívajú na detekciu a samotné vylepšenie obrazu. V kapitole **4** sa čitateľ zoznámi jednak s návrhom výslednej aplikácie, a to od návrhu techniky použitej na detekciu cez návrh algoritmu na vylepšenie obrazu až po návrh grafického rozhrania výslednej aplikácie. V tejto kapitole sú zahrnuté aj špecifikácie testovacích procesov ako pre detekciu tak aj pre vylepšenie obrazu. V poslednej časti tejto kapitoli sa čitateľ dozvie viac o aplikácií z hľadiska použitých knižníc, programovacieho jazyka a pod. V kapitole **5** je zhrnuté testovanie jednotlivých častí aplikácie, ako sú detekcia a vylepšenie obrazu. V kapitole **6** sú zhrnuté dosiahnuté výsledky, čo sa v práci nepodarilo a taktiež sa tam nachádza časť o možnom vylepšení aplikácie.

Kapitola 2

Detekcia tváre

Detekcia tváre je priebeh identifikácie tváre v danom prostredí. Pre ľudský mozog je táto úloha jednoduchá a prirodzená, nakoľko mozog dokáže rozoznávať a uchovávať vzhľad tváre už od narodenia. Na druhej strane, pre stroj, je proces detekcie a identifikácie tváre zložitý a komplikovaný, ale nie neriešiteľný. Aspektmi, ktoré stroju značne sťažujú tento proces sú predovšetkým mimika a nízka kvalita celého podkladu obrazu [38].

2.1 Prehľad existujúcich algoritmov

Metódy, ktoré sa používajú na detekciu tváre, je možné z hľadiska použitia neurónových sietí rozdeliť do dvoch základných kategórií:

- Tradičné metódy
- Metódy založené na hĺbkovom učení

Tradičné metódy

Základom tradičných metód sú ručne vytvorené funkcie. Tradičné metódy sú schopné dosiahnuť detekciu v reálnom čase na CPU (centrálne procesorová jednotka z angl. *central processing unit*), ale chýba im adaptácia na zmenu výrazu tváre a adaptácia na slabé osvetlenie tváre. Taktiež nie sú vhodným riešením na detekciu tváre pri nízkom rozlíšení. Typickým a najznámejším príkladom takýchto metód je Viola-Jones detektor [38].

2.1.1 Viola-Jones detektor

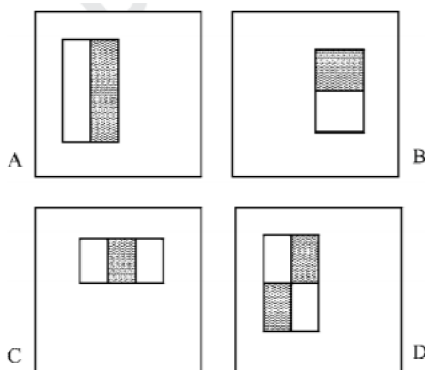
Viola-Jones detektor je rámec na detekciu tváre človeka, ktorý navrhli Paul Viola a Michael Jones v roku 2001. Celá funkčná logika algoritmu prebieha v štyroch krokoch [38]:

- Haar feature Selection
- Creating an integral image
- AdaBoost training
- Cascading classifiers

Haar feature Selection

Z hľadiska fyziognómie všetky ľudské tváre vykazujú určitú podobnosť.

Príkladmi bežných črt, ktoré sa používajú na vytvorenie Haar funkcie, sú: oblasť oka, ktorá je tmavšia ako špička nosa, oblasť horných líc, ktorá je tmavšia ako región očí a umiestnenie očí, úst a nosa. Tieto črty sa určujú prostredníctvom algoritmu a používajú sa ako súčasť vyhľadávania v obrázku, a to vo forme obdĺžnikov. Teda každá oblasť, resp. región, tváre je reprezentovaná jedným obdĺžnikom [25].



Obr. 2.1: Príklad obdĺžnikov. Prevzaté z [29].

Creating an integral image

Obdĺžnikové útvary, ktoré sú zobrazené na obrázku 2.1 sa efektívne vypočítavajú prostredníctvom metódy integrálnych obrázkov. Integrálny obrázok na mieste x, y obsahuje súčet pixelov hore a doľava z x, y vrátane:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (2.1)$$

kde $ii(x, y)$ je integrálny obrázok a $i(x, y)$ je pôvodný obrázok, prostredníctvom nasledujúcej dvojice opakovaní:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (2.2)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y), \quad (2.3)$$

kde $s(x, y)$ je kumulatívny súčet riadkov, $s(x, -1) = 0$ a $ii(-1, y) = 0$ integrálny obraz možno vypočítať jedným prechodom pôvodným obrazom [29].

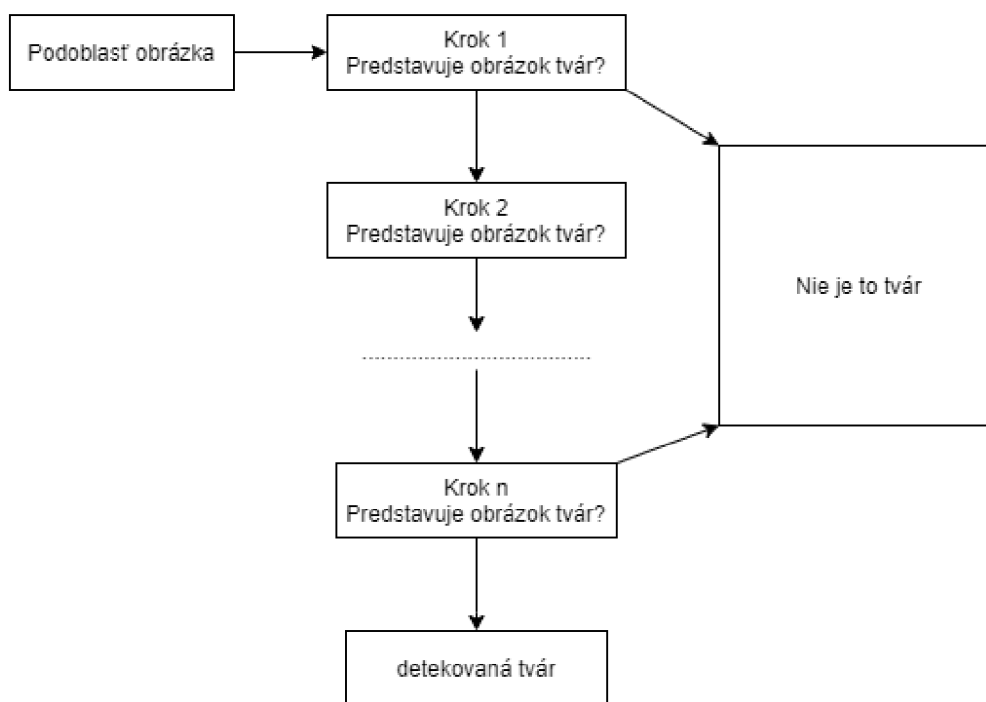
Adaboost training

Adaboost je klasifikátor, ktorý slúži na klasifikáciu obrázku. Adaboost klasifikátor sa skladá z malého počtu funkcií zredukovaných z veľkej skupiny klasifikátorov.

Všetky funkcie, ktoré sú vypočítané, nie sú nevyhnutne súčasťou tváre, a preto je nepotrebné, aby sa nadbytočné funkcie zahodili. To sa deje práve pomocou klasifikátora. Adaboost klasifikátor je veľmi efektívny, pretože všetky funkcie, ktorých je viac ako 160 000, dokáže zredukovať na niekoľko stoviek. Na základe tejto redukcie je možné AdaBoost zobrazit aj ako selektor funkcií [25].

Cascading classifiers

Obdĺžniky, ktoré reprezentujú jednotlivé významné črty tváre, sa určujú prostredníctvom Haar funkcie a následne sa vyhodnocujú pomocou súboru klasifikátorov. Každý v poradí po sebe nasledujúci klasifikátor je o niečo zložitejší. Kaskádovité klasifikátory slúžia na odhalenie falošnej detekcie a následne na ukončenie klasifikácie. V opačnom prípade priebeh klasifikácie kaskádovite pokračuje na nasledujúci klasifikátor. Princíp klasifikácie je znázornený na obrázku 2.2.



Obr. 2.2: Postup klasifikácie obdĺžnikov. Prevzaté z [28].

Metódy založené na hĺbkovom učení

Druhou skupinou metód, ktoré sa zameriavajú na detekciu tváre, sú metódy založené na hĺbkovom učení. Tieto metódy využívajú princíp neurónových sietí. V porovnaní s tradičnými metódami dokážu úspešne rozpoznať tvár aj pri nízkom rozlíšení obrázku. Typickými príkladmi metód založených na hĺbkovom učení sú [38]:

- CNN 2.1.2
- Faster R-CNN 2.1.3
- YOLO 2.1.4
- SSSFD 2.1.5

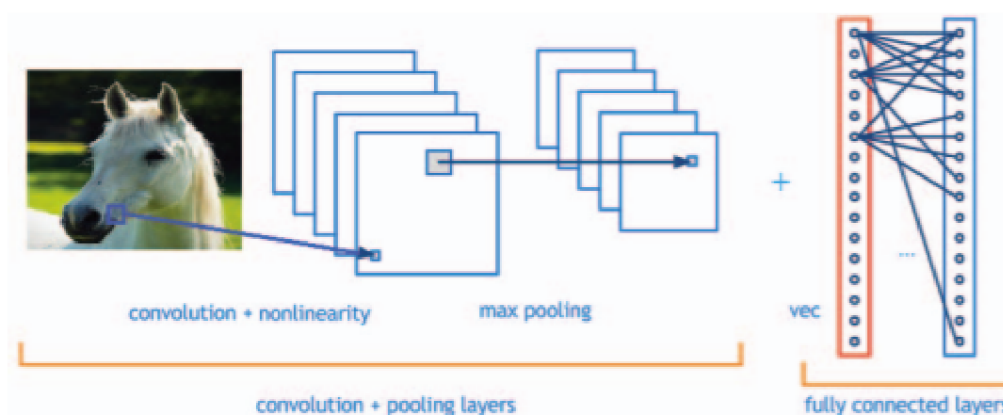
2.1.2 CNN

Konvolučná neurónová sieť, skrátene CNN (z angl. *convolutional neural network*), je kategóriou neurónovej siete, ktorá sa využíva v oblastiach spojených s klasifikáciou a rozpozná-

vaním obrazu. Typickými príkladmi architektúr CNN siete sú ResNet [12], VGG [36] a iné [4].

CNN patrí do skupiny tzv. *feed-forward* neurónových sietí, pre ktorú je typické, že vo svojej architektúre vytvára acyklický graf [35]. CNN sa ďalej skladá buď z filtrov, z jadier alebo z neurónov, ktoré obsahujú charakteristické hmotnosti alebo parametre a tzv. *biases*. Štruktúra CNN pozostáva z nasledujúcich častí [4]:

- konvolučné vrstvy,
- pooling vrstvy,
- rektifikovaná lineárna jednotka, skrátene ReLU (z angl. *Rectified Linear Unit*),
- plne prepojené vrstvy.



Obr. 2.3: Konvolučná neurónová sieť. Prevzaté z [4].

Konvolučná vrstva

Konvolučná vrstva predstavuje hlavný stavebný prvok konvolučnej siete, ktorý vykonáva väčšinu výpočtových úloh. Primárnym účelom konvolučnej vrstvy je extrakcia prvkov zo vstupných údajov, ktorými sú obrázky. Každá konvolučná vrstva sa skladá z niekoľkých filtrov. Filter slúži na detekovanie špecifickej vlastnosti zo vstupného obrázku. Je to matica o veľkosti $N \times N$, ktorá sa preloží cez vstupný obrázok a prostredníctvom skalárneho súčinu sa určia nové hodnoty pixelov. Z Výsledných hodnôt sa vytvoria aktivačné mapy, ktoré sa následne načítajú ako vstupné údaje do ďalšej konvolučnej vrstvy. Počet filtrov v konkrétnej konvolučnej vrstve udáva výsledný počet aktivačných máp [4, 17].

Pooling vrstva

Pooling vrstva znižuje rozmernosť každej aktivačnej mapy, pričom naďalej obsahuje najdôležitejšie informácie. Vstupné obrázky sa delia do množiny neprekrývajúcich sa obdĺžnikov. Každá oblasť je vzorkovaná nelineárnou operáciou. Typickým príkladom nelineárnej operácie je priemer alebo maximum. Pooling vrstva dosahuje lepšiu generalizáciu, rýchlejšiu konvergenciu, je odolná voči skresleniu a zvyčajne je umiestnená medzi konvolučné vrstvy [4].

Rektifikovaná lineárna jednotka

Rektifikovaná lineárna jednotka, skrátene ReLU, je operácia po prvkoch, čo znamená, že sa aplikuje na každý pixel a všetky negatívne hodnoty na mape prvkov sa zrekonštruujú na nulu. Aby sme pochopili ako funguje ReLU predpokladáme, že existuje neurónový vstup daný ako x , a že usmerňovač je v literatúre pre neurónové siete definovaný ako [4]:

$$f(x) = \max(0, x). \quad (2.4)$$

Plne prepojená vrstva

Plne prepojená vrstva zabezpečuje, že každý filter v predchádzajúcej vrstve je pripojený ku každému filteru v nasledujúcej vrstve. Výstupný obraz je výstupom z konvolučných, pooling a ReLU funkcií. Cieľom plne prepojenej vrstvy je využitie týchto funkcií na klasifikáciu vstupného obrazu do rôznych tried na základe súboru testovacích údajov. Plne prepojená vrstva využíva aktivačnú funkciu Softmax, ktorá zaisťuje, že súčet pravdepodobností výstupu z plne pripojenej vrstvy je 1. Funkcia Softmax vezme vektor ľubovoľných skóre so skutočnou hodnotou a roztláči ho na vektor hodnôt, ktoré sú z rozsahu 0 – 1, a ktoré dávajú svojím súčtom výsledok 1 [4].

2.1.3 Fast R-CNN

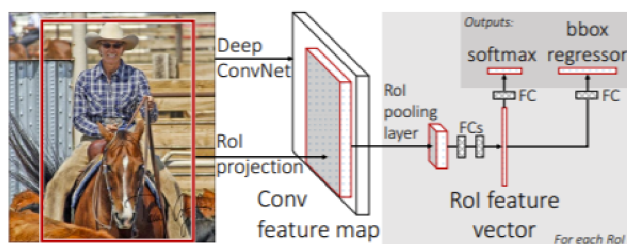
Fast R-CNN berie ako vstup celý obraz a súbor návrhov objektov. Sieť najskôr spracuje celý obraz s niekoľkými konvolučnými a maximálnymi pooling vrstvami a vytvorí konvolučnú mapu funkcií. Potom pre každý návrh objektu RoI (region záujmov z angl. *region of interests*) extrahuje z aktivačnej mapy vektor znakov s pevnou dĺžkou. Každý vektor je privádzaný do sekvencie úplne spojených vrstiev, ktoré sa nakoniec rozvetvujú do dvoch súrodeneckých výstupných vrstiev:

- vytvára odhady pravdepodobnosti funkcie softmaxu cez triedy K objektov, plus triedu „pozadia“ typu *catch-all*
- vydáva 4 reálne ohodnotené čísla pre každú z K tried objektov

Každá sada štyroch hodnôt kóduje pozície rafinovaného vymedzovacieho boxu pre jednu z tried K [8].

Pooling vrstva RoI

Pooling vrstva RoI využíva maximálny pooling na prevod prvkov vo vnútri ktorejkoľvek platnej oblasti záujmu na malú aktivačnú mapu s pevným priestorovým rozsahom $H \times W$, kde H a W sú hyperparametre vrstvy, ktoré sú nezávislé na ktorejkoľvek konkrétnej RoI. Každá RoI je definovaná štvoricou n -tíc (r, c, h, w) , ktorá určuje jej ľavý horný roh (r, c) a jej výšku a šírku (h, w) [8].



Obr. 2.4: Architektúra Fast R-CNN. Prevzaté z [8].

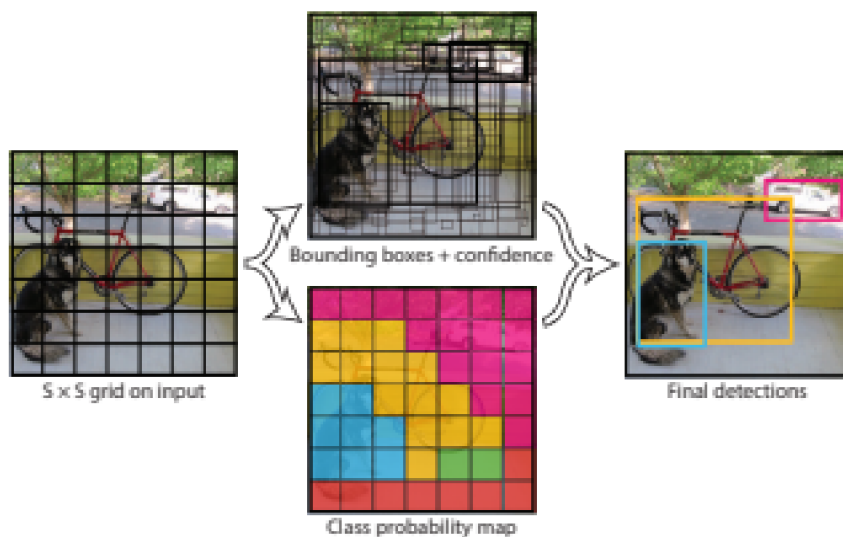
2.1.4 YOLO

You only look once, skrátene YOLO, sa využíva na detekciu objektov. YOLO vzhliada na detekciu cieľov ako na regresný problém [14, 32].

V súčasnosti sú k dispozícii dve verzie YOLO modelu. YOLOv1 je základným modelom, ktorý je znázornený na obrázku 2.5. YOLOv2 je novšou verziou, ktorej základ vychádza z YOLOv1. V porovnaní s predošlou verziou, nová verzia je schopná predpokladať objekt pri rôznom rozlíšení a dosahuje vyššiu úspešnosť pri detekcii objektov [6].

Princíp algoritmu

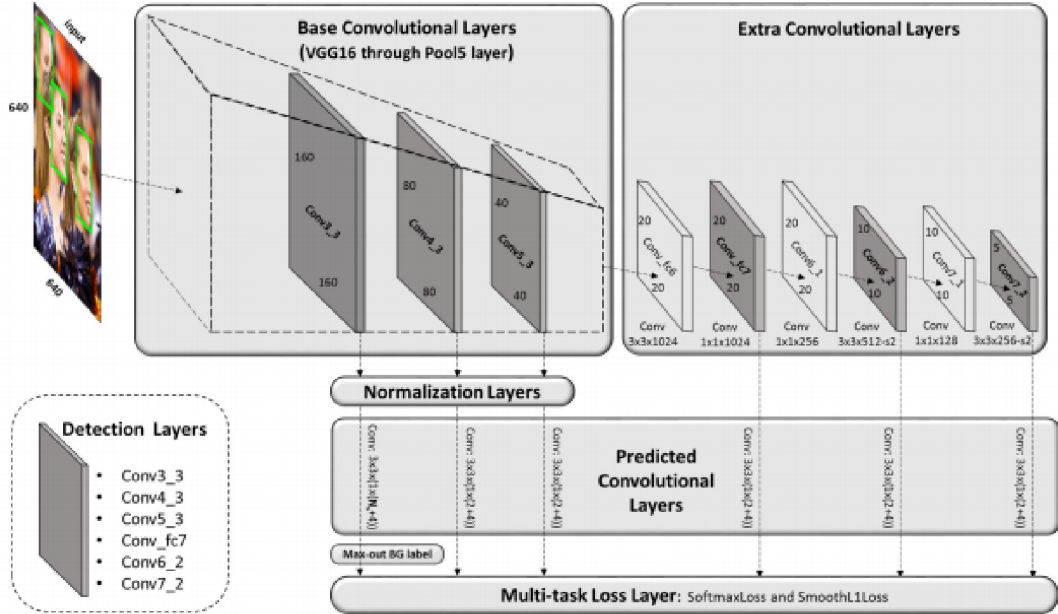
Algoritmus YOLO je možné rozdeliť do niekoľkých etáp. V prvej etape je vstupný obrázok rovnomerne rozdelený. Výstupom prvej etapy je obrázok, rozdelený na súbor malých štvorcov, ktorý vizualizuje sieť o veľkosti $S \times S$. V nasledujúcom kroku sa pre každý rámik S predikuje B počet ohraničených rámcov s pravdepodobnosťami ich náležitosti detekcie a pravdepodobnosťami vzťahovať sa do jednotlivých definovaných tried. V poslednom kroku sú vyselektované len také ohraničené rámce, ktoré dosahujú najvyššie pravdepodobnostné skóre. Ostatné ohraničené rámce sa zahadzujú [14].



Obr. 2.5: Priebeh Algoritmu YOLO. Prevzaté z [21].

2.1.5 SSSFD

Single shot scale invariant face detector, skrátene S^3FD , vychádza z modelu SSD, konkrétne z jeho viacúrovňového mechanizmu, a z Faster R-CNN, a to z jeho RMP časti [18, 22, 37].



Obr. 2.6: Architektúra S^3FD . Prevzaté z [37].

Hlavnou inováciou, vďaka ktorej sa podarilo posunúť detekciu na vyššiu úroveň, je riešenie detekcie tváre z hľadiska veľkosti, a to pre stredné až veľmi malé tváre [37].

Použitím sieťovej architektúry so širokým súborom vrstiev spojených s kotvami, z angl. *anchors*, ktorých veľkosť kroku sa postupne zdvojnásobuje od 4 do 128 pixelov, sa docielilo rozpoznanie aj menšie tváre. Prostredníctvom takejto sieťovej architektúry sa zaisťuje, aby rôzne rozpätie tvárí malo primerané vlastnosti na detekciu vo vhodných vrstvách spojených s kotvami [37].

Architektúra S^3FD je zobrazená na obrázku 2.6. Na obrázku sa takisto nachádza jedna CNN sieť 2.1.2, ktorej architektúra je VGGnet [36]. Nie je to ale klasická architektúra VGGnet [36]. Plne prepojené vrstvy v poradí 6 a 7 sa prostredníctvom podvzorkovania ich parametrov zmenia na konvolučné vrstvy. Bezprostredne za ne sa pridávajú ďalšie konvolučné vrstvy. Tie sa postupne zmenšujú, a tak vytvárajú viacrozmerne aktivačné mapy [37].

Kapitola 3

Vylepšenie kvality objektu tváre

Táto kapitola ponúka prehľad technológií, ktorých hlavnou úlohou je vylepšiť, resp. opraviť, zlú prezentáciu objektu tváre v obrázku, ktorá je nevyhnutným aspektom pre neskoršiu manipuláciu a prácu s ňou.

Medzi najčastejšie typy poškodenia obrázku, ktoré spôsobujú čiastočnú alebo celkovú deformáciu obrázku patria: rozmazanie, šum, nízka kvalita osvetlenia, kontrast, nízke rozlíšenie obrázka, oklúzia a mnoho ďalších. K spomenutým typom poškodení dochádza nevhodným zvolením systému, ktorý ponúka nízku kvalitu rozlíšenia; v dôsledku veľkej vzdialenosti snímajúceho objektu od objektívu fotoaparátu alebo kamery a v neposlednom rade prostredníctvom rôznych chýb v spracovávaní obrazu [20].

Kvalita objektu tváre ako aj celková kvalitatívna reprezentácia obrázku sú veľmi dôležité pre ďalšie kroky, ako napríklad identifikácia tváre, kamerové zábery či rekonštrukcia tváre. Na vyriešenie problémov, kedy je veľmi náročné pracovať so vstupným obrázkom, kvôli častým poškodeniam, sa používa tzv. super-rozlíšenie tváre [16].

Super-rozlíšenie tváre, skrátene SR (z angl. *super-resolution*), známe tiež ako tvárová halucinácia, z angl. *face hallucination*, sa zameriava na generovanie vysoko kvalitných, skrátene HR (z angl. *high resolution*) obrázkov, kde vstupnými dátami sú obrázky v nízkom rozlíšení, skrátene LR (z angl. *low resolution*) [2].

3.1 Prehľad dostupných technológií

Metódy, ktoré sa využívajú na vylepšenie tváre, je možné z hľadiska použitia neurónových sietí rozdeliť do týchto skupín:

- Metódy založené na histograme
- Interpolačné metódy
- Super resolution
- Metódy založené na neuronových sieťach

Metódy založené na histograme

Tieto metódy pracujú s kontrastom jednotlivých obrázkov, ktorý transformujú k dosiahnutiu kvalitnejšej vizualizácií obrázka. Medzi základné algoritmy, ktoré reprocessujú kontrast snímok sú:

- Vyrovnanie histogramu
- Adaptívne vyrovnanie histogramu

3.1.1 Vyrovnanie histogramu

Táto technika patrí k základným technikám na vylepšenie obrazu. Každému pixelu na obrázku je priradená hodnota z intervalu $\langle 0, 255 \rangle$, kde 0 predstavuje čiernu farbu a hodnota 255 predstavuje bielu farbu. Po tomto rozdelení sa vytvorí histogram s početnosťou jednotlivých hodnôt pixelov. Pre vysvetlenie, v prípade ak je obraz tmavej farby jednotlivé pixely budú mať nižšie hodnoty, tým pádom najväčšie zoskupenie pixelov bude v oblasti 0. Úlohou algoritmu vyrovnania histogramu je zvýšiť kontrast každému pixelu obrazu. Inými slovami, pre každú hodnotu pixelu j priradí novú hodnotu pixelu k pomocou rovnice 3.1.

$$k = \sum_{i=0}^j = \frac{N_i}{T}, \quad (3.1)$$

kde N je počet jednotlivých pixelov danej hodnoty kontrastu a T je počet všetkých pixelov [26].

Veľkým problémom tejto techniky je, že zvyšuje kontrast obrazu ako celku. To znamená, že oblasti s výskytom šumu budú spracované, čím sa zvýši nechcený jas v zašumenej oblasti ako je možné vidieť na obrázku 3.1.



Obr. 3.1: Spracovanie šumu. Zdroj¹.

3.1.2 Adaptívne vyrovnanie histogramu

Adaptívne vyrovnanie histogramu sa taktiež používa na vylepšenie kontrastu v obrázkoch. Líši sa od vyrovnania histogramu adaptívnou metódou, ktorá počíta niekoľko histogramov, kde každý histogram zodpovedá samostatnej časti obrázka. Adaptívne vyrovnanie histogramu na rozdiel od klasickej techniky vyrovnania histogramu vylepšuje obraz transformáciou každého pixelu pomocou transformačnej funkcie odvodenej od susednej oblasti. Znižuje tak časť šumu v regiónoch obrazu. Táto technika taktiež podporuje vylepšenie kontrastu v sivých a farebných obrazoch [11].

¹https://www.youtube.com/watch?v=poR-cFh3PDI&t=248s&ab_channel=DeepEigenDeepEigen



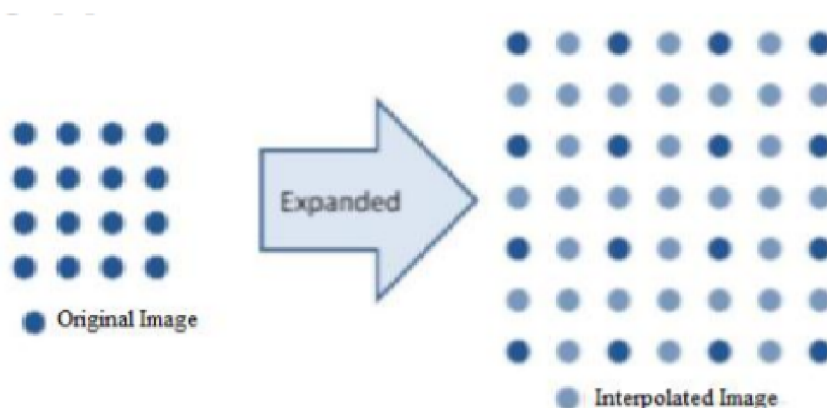
Obr. 3.2: Vylepšenie obrazu pomocou adaptívneho vyrovnania histogramu. Prevzaté z [11].

Interpoláčn  metody

Z kladom interpol čných met d je interpol chn  technika, u ktorej sa odhaduje nasleduj ca hodnota funkcie pomocou aproxim cie. Medzi z kladn  interpol chn  met dy patr  [27]:

- Interpol cia najbli šieho suseda
- Biline rna interpol cia
- Bikubick  interpol cia

Princ p interpol cie je zobrazen  na obr zku 3.3.



Obr. 3.3: Princ p interpol cie. Prevzat  z [27].

3.1.3 Interpol cia najbli šieho suseda

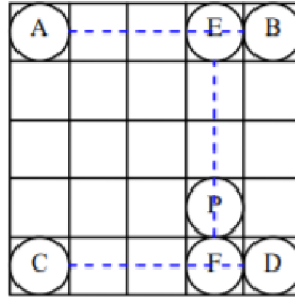
Interpol cia najbli šieho suseda je z kladnou technikou interpol cie. V porovnan  so v etk ymi interpol chn mi met dami je interpol cia najbli šieho suseda najjednoduch ia a na spracovanie d t  asovo najmenej n ro n . Princ pom interpol cie najbli šieho suseda je,  e interpol chn  pixel sa nahrad  najbli  im pixelom. T to met da je vhodn  na spracovanie obr zku, ktor  m  kont ry HR, ale pozdl  okrajov s  badateln  nedostatky [27].

3.1.4 Biline rna interpol cia

Biline rna interpol cia vyu iva line rnu interpol ciu na dopo itanie v slednej hodnoty pixelu. Princ p algoritmu je zn zornen  na obr zku 3.4 a mo no ho rozdel  do dvoch bodov.

Majme bod P , ktor ho poz cia je x a y . Hodnota bodu P sa vypo ita na z klade  tvorice bodov, ktor ch poz cia a hodnota s  zn me. V prvom kroku algoritmu sa vyr taju

lineárnou interpoláciou hodnoty nových dvoch bodov E a F , a to buď v horizontálnom alebo vertikálnom smere. V druhom kroku sa opäť lineárnou interpoláciou novovzniknutých bodov E a F vyjadří hodnota bodu P v opačnom smere. Poradie, v akom smere sa prevedie interpolácia, je ľubovoľné [10].



Obr. 3.4: Bilineárna interpolácia. Prevzaté z [10].

3.1.5 Bikubická interpolácia

Bikubická interpolácia, z angl. *Bicubic interpolation*, je v porovnaní s bilineárnou interpoláciou z hľadiska kvality výstupného obrázku podstatnejšie výkonnejšia. Princíp algoritmu bikubickej interpolácie je z väčšej časti založený prevažne na algoritme bilineárnej interpolácie. Hlavným rozdielom, ktorým vylepšuje kvalitu výstupu, je, že na výpočet neznámeho pixela použije 16 okolitých bodov. Pomocou kubického polynómu tretieho rádu je realizovaný výpočet, ktorý je znázornený v rovnici 3.2 [7]:

$$y = ax^3 + bx^2 + cx + d \quad (3.2)$$

Super resolution

Super resolution (SR) je jedna z mnohých metód, ktorá sa využíva na vylepšenie resp. na získanie čo najlepšej reprezentácie obrazu z hľadiska celkovej kvality obrazu. Na získanie požadovanej kvality obrazu sa používa jedna alebo súbor snímok nízkej kvality jednej a tej istej scenérie [24].

Predpokladáme, že I_L predstavuje obraz s nízkym rozlíšením (LR) a I_H predstavuje zodpovedajúci obraz s vysokým rozlíšením (HR), pričom I_L je možné získať degradačným procesom I_H , a to nasledovným spôsobom:

$$I_L = D(I_H; \delta), \quad (3.3)$$

kde D predstavuje degradačný proces a δ predstavuje parametre v tomto procese (zmenšená veľkosť, prírodný alebo umelý šum a faktory rozmazania). Takto získaný I_L stráca veľa detailov a vysokofrekvenčných informácií, a preto je nevyhnutné obnoviť tieto stratené informácie [31].

Avšak v ojedinelých prípadoch degradačný proces D je neznámy (nie je možné určiť, ako I_H degeneruje do I_L), z čoho vzniká kľúčový problém, ako pomocou daného I_L obrátiť proces D tak, aby reverzný výsledok \hat{I}_H mohol byť ekvivalentný voči definícii I_H . Tento proces je vyjadrený pomocou vzťahu:

$$\hat{I}_H = F(I_L; \theta), \quad (3.4)$$

kde F znázorňuje inverzný proces procesu D , inými slovami je to samotný SR a θ je reprezentáciou niektorých SR parametrov (ako napríklad násobenie prevzorkovania, parametre váh pri hĺbkovom učení a pod.) [31].

V súčasnosti veľkým problémom nielen pre oblasť počítačového videnia ale v niektorých prípadoch aj pre samotného človeka ostáva zaostrenie objektu, ktorý sa pohybuje veľmi rýchlo. V práci [23] je použitie tzv. časového super rozlíšenia jednou z metód ako toto zaostrenie prostredníctvom stroja možné docieľiť.

Metódy založené na neurónových sieťach

Metódy založené na neurónových sieťach sa líšia od interpolačných metód z hľadiska použitia technológií na vylepšenie obrazu. Interpolačné metódy využívajú na riešenie problému základné matematické metódy, ako interpoláciu, či Euklidovskú vzdialenosť. Metódy, ktoré sú postavené na neurónových sieťach dosahujú omnoho kvalitnejšie výstupné obrázky.

Neurónové siete patria do odvetvia hĺbkového učenia, ktorého hlavným cieľom je naučiť sa odvodiť vzťah medzi vstupom a výstupom priamo z dátovej prezentácie [1].

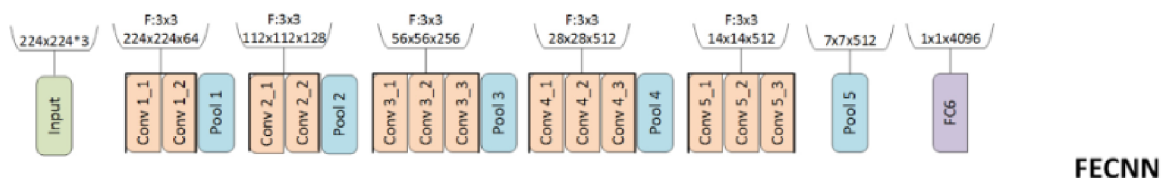
V súčasnosti je k dispozícii veľké množstvo metód založených na neurónových sieťach. Od základných CNN 2.1.2, GAN metód, až po rôzne vylepšenia týchto metód. V tejto sekcii sa pozrieme na prehľad existujúcich riešení postupov na vylepšenie obrázka [1].

3.1.6 Vylepšenie obrazu použitím CNN

CNN siete, ktoré sme definovali v sekcii 2.1.2, sú veľmi užitočné aj v oblasti zvýšenia kvality obrázka.

Príkladom použitia CNN v oblasti zvýšenia kvality ľudskej tváre z obrázku, je použitie dvoch samostatných CNN. Úlohou prvej CNN je premietiť HR obrázky do spoločnej oblasti. Úlohou druhej CNN je samotné mapovanie LR obrázkov do spomínanej spoločnej oblasti [34].

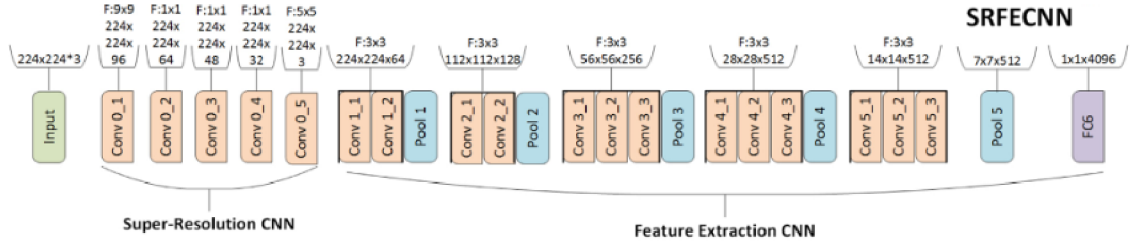
Na konštrukciu oboch CNN sa používa architektúra siete VGGnet [36], konkrétne jej verzia s trinástimi konvolučnými vrstvami a tromi plne prepojenými vrstvami. Prvú CNN, ktorá slúži na extrakciu vlastností tváre, možno nazvať *feature extraction* CNN, skrátene FECNN. Posledné dve plne prepojené vrstvy FECNN sú vynechané, ako je vidieť na obrázku 3.5. Vstupom pre spomínanú FECNN je obrázok, ktorý rozmerovo musí zodpovedať veľkosti 224×224 pixelov, a ktorý je vo vysokom rozlíšení. V prípade, že vstupný obrázok nemá dané rozmery, tak sa prostredníctvom bikubickej interpolácie 3.1.5 pretransformuje do požadovanej veľkosti [34].



Obr. 3.5: Architektúra FECNN. Prevzaté z [34].

Druhá sieť CNN pozostáva z dvoch častí. Prvá časť prvotnej siete je tvorená piatimi konvolučnými vrstvami, ktoré slúžia na vylepšenie vstupného obrazu. Vstupom tejto časti je LR obraz o veľkosti 224×224 , ktorý je interpolovaný tradičnými metódami 3.1.2. Výstupom

podsiete je obraz s rovnakou veľkosťou, ktorý putuje ako vstup do druhej podčasti. Druhá podčasť siete je z hľadiska funkcie a architektúry, ako je znázornené na obrázku 3.6, rovnaká ako FECNN. Takúto sieť možno jednotne nazvať ako SRFECNN, pretože v prvej podčasti vylepšuje obraz a v druhej podčasti už vykonáva činnosť ako sieť FECNN [34].

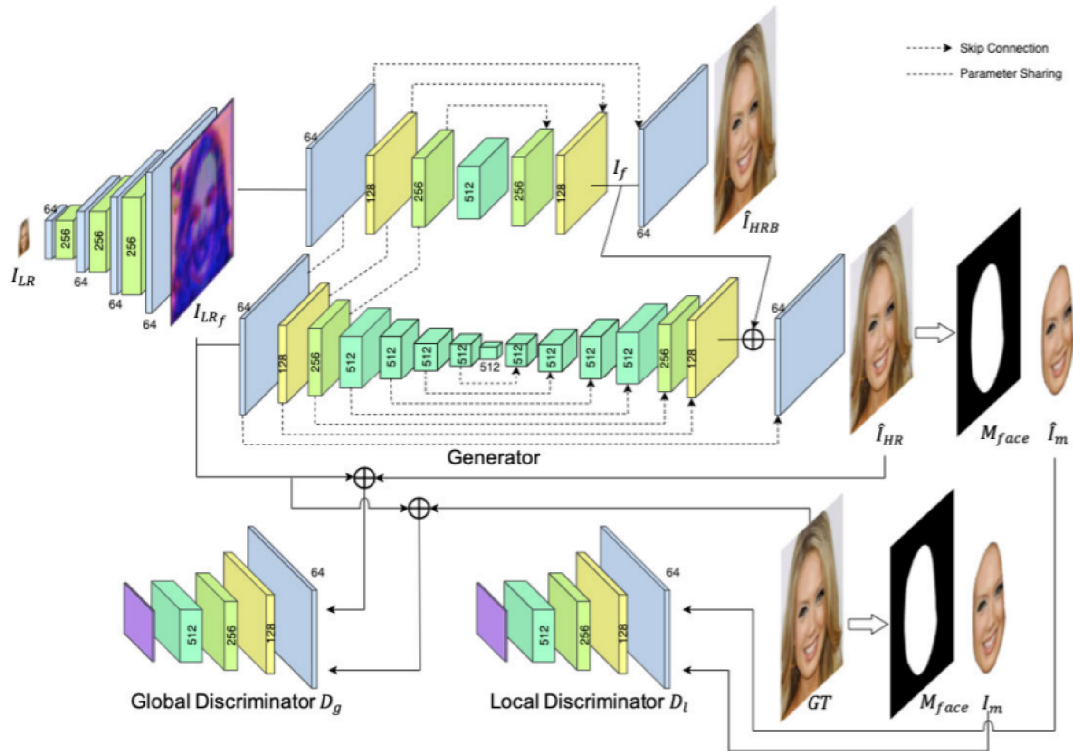


Obr. 3.6: Architektúra SRFECNN. Prevzaté z [34].

3.1.7 Vylepšenie obrazu použitím GAN

Generative adversarial network, skrátene GAN, je typ neurónovej siete, ktorú skonštruoval a zdizajnoval pán Ian Goodfellow [9].

GAN je využiteľná aj v oblasti vylepšenia obrazu. GAN syntetizuje HR obrázky použitím vstupných obrázkov, ktoré sú v nízkej kvalite. Táto sieť dokáže vytvárať obrázky tváre, ktoré dosahujú vysokú kvalitu. Hlavnou nevýhodou, ktorou GAN sieť disponuje, je, že vyžaduje veľké množstvo počítačového výkonu pri tréningu, a taktiež množstvo času potrebného na tréning výsledného modelu [33].



Obr. 3.7: Architektúra GAN. Prevzaté z [33].

Architektúru GAN siete možno rozdeliť do dvoch častí, čo je znázornené aj na obrázku 3.7:

- Generátor
- Diskriminátor

Každá z týchto dvoch častí tvorí samostatnú neurónovú sieť, inými slovami, každá časť GAN siete, má svoju vlastnú neurónovú sieť. Architektúra neurónovej siete u generátora a u diskriminátora je voliteľná. Súčasne ale prebiehajú rôzne experimenty, ktorých cieľom je určiť, aký typ architektúry je najvhodnejší na použitie. Nasledujúci popis vychádza z U-net siete [13].

Generátor

Generátor je časť siete, do ktorého pri vstupe prichádza LR obrázok. Použitím päťvrstvovej CNN sa extrahujú znaky priamo z LR obrázka a postupným zväčšovaním priestorového rozlíšenia až na 8-násobok sa docieli extrakcia aktivačných máp. Vďaka tejto extrakcii znakov na aktivačnej mape, sa predíde interpolačným defektom vlastností, ktoré vznikajú použitím interpolačných metód 3.1.2 [33].

Vygenerované obrazové vlastnosti sa paralelne použijú na generovanie HR tváre s rozmazaním, skrátene HRB, a na generovanie HR tváre bez použitia rozmazávania, skrátene HR. Potom sa takto vytvorená aktivačná mapa využije pred poslednou konvolúciou, viď obrázok 3.7, na vygenerovanie HR obrázku tváre.

Diskriminátor

Diskriminátor predstavuje klasifikátor, ktorého hlavnou úlohou je rozhodnúť, či výstupný obrázok vytvorený generátorom, je skutočným obrazom tváre alebo sa jedná o výtvor, ktorý nekorešponduje s tvárou, a ide teda o chybový obraz [33].

Prostredníctvom predtrénovaných architektúr neurónových sietí, ako sú napríklad ResNet [12] či rôzne variácie VGGnet [36], môžeme doceliť rôznorodé výsledky.

Vylepšením tejto metódy je SRGAN [15], alebo ESRGAN [30], ktoré už eliminujú problém z hľadiska spotrebovaného času na tréning modelu.

Kapitola 4

Návrh a implementácia

Táto kapitola je zložená z štyroch celkov. Prvý celok obsahuje výber a zdôvodnenie výberu jednotlivých algoritmov na riešenie časti detekcie tejto práce. Druhý celok taktiež obsahuje výber a zdôvodnenie výberu jednotlivých algoritmov na riešenie časti vylepšenia obrazu tejto práce. Tretí celok je tvorený grafickým návrhom výslednej aplikácie a posledný celok je tvorený popisom implementácie celej aplikácie.

Po preštudovaní dostupných riešení v oblasti detekcie ľudskej tváre a následnom vylepšení obrazu z nekvalitných zdrojových snímok som sa rozhodol použiť v oboch prípadoch neurónové siete. Neurónové siete ponúkajú spoľahlivejšie výsledky ako spomínané interpoláčne metódy 3.1.2, metódy založené na histograme 3.1 alebo tradičné metódy 2.1, ktoré nie sú založené na neurónových sietiach.

4.1 Detekcia

Na detekciu tváre som sa rozhodol použiť už existujúce riešenie. Jedná sa o detektor S^3FD ¹, ktorý dosahuje veľmi dobrú presnosť detekcie tváre aj z nekvalitných snímok. Tento detektor, resp. jeho model, dosahuje presnosť okolo 70 %. Mojm cieľom je doceliť vyššiu presnosť detektora natrenovaním modelu.

Ako trénovaciu sadu využijem *WIDERFACE*² dataset, ktorý obsahuje širokú škálu ľudských tvárí a okrem iného obsahuje aj ľudské tváre s rôznymi predmetmi, ako napr. šatku, okuliare, masku a pod.

4.1.1 Trénovacia sada

Trénovacia sada je jedna z najdôležitejších nevyhnutných prerekvizít. Je potrebná na trénovanie modelu danej neurónovej siete. Prostredníctvom výsledného modelu je neurónová sieť plne funkčná na plnenie rôznorodých úloh. Na detekciu ľudskej tváre je potrebná trénovacia sada, ktorá ponúka širokú škálu ľudských tvárí, a to aj takých, ktoré sú detektorom ťažko rozpoznateľné. Najväčším problémom, čo sťažuje výber trénovacej sady, je, že detekcia musí byť schopná detekovať tváre z nekvalitných zdrojových obrázkov. Ďalším problémom, ktorý vzniká, je schopnosť zdetekovať aj menšie tváre.

Po preštudovaní si dostupných trénovacích sád som sa rozhodol pre *WIDER FACE*² testovaciu sadu, ktorá má záber na rôzne rozmery ľudských tvárí. Táto sada taktiež obsahuje zmes obrázkov, na ktorých je tvár ťažko rozpoznateľná, t.j. je zakrytá šatkou, slnečnými

¹kód dostupný z <https://github.com/louis-she/sfd.pytorch>

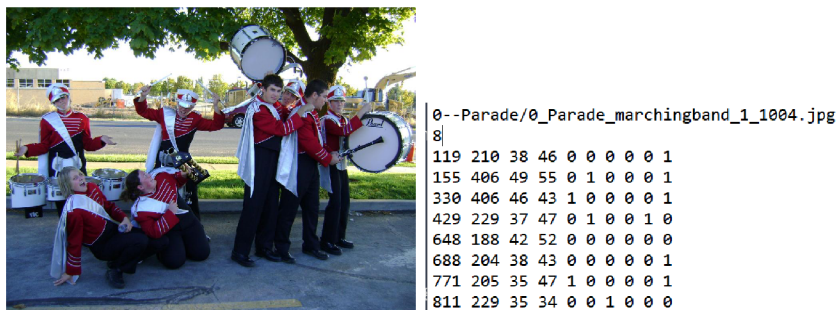
²dataset dostupný z <http://shuoyang1213.me/WIDERFACE>

okuliarmi, alebo tvárou prechádza dym, para a pod. Testovacia sada obsahuje viac než 32 000 trenovacích obrázkov, na ktorých sa nachádza viac ako 393 000 tvárí.

4.1.2 Anotácia testovacej sady

Anotácia testovacej sady je zápis, pomocou ktorého je daný objekt jednoznačne identifikovateľný. *WIDER FACE* využíva na identifikáciu objektu, teda ako anotáciu, textový formát. Jeho anotácia sa skladá z nasledovných parametrov:

- relatívna cesta k danej snímke,
- počet ľudských tvárí na danej snímke,
- atribúty identifikujúce pozíciu tváre v snímke: [pozícia x, pozícia y, šírka, výška, rozmazanie, výraz, iluminácia, validita, absorpcia, postoj].



Obr. 4.1: Ukážka obrázka a následnej textovej anotácie.

Na obrázku 4.1 je zobrazená anotácia daného obrázka s jednotlivými atribútmi, ktoré jednoznačne identifikujú ľudskú tvár na obrázku.

4.1.3 Trénovanie modelu

Nevyhnutnou súčasťou fungujúceho a spoľahlivého systému je natrénovanie modelu danej neurónovej siete. Zo všetkých etáp vývoja samotného detektora je táto etapa z časového hľadiska najnáročnejšia. Rozhodujúcim faktorom, ktorý určuje trvanie etapy (t.j. či bude trvať kratšie alebo dlhšie), je výpočtová sila počítača.

Na tréningovanie modelu som použil výpočtové stredisko MetaCentrum³. MetaCentrum je virtuálna organizácia, ktorej infraštruktúra ma záber po celej Českej republike. Ponúka výpočtový výkon bezplatne všetkým akademickým pracovníkom a taktiež študentom. Okrem iného ponúka aj veľkokapacitné úložné priestory.

4.1.4 MetaCentrum

Pre používanie tejto organizácie bola potrebná registrácia. Keďže VUT (Vysoké učení technické v Brne) je v úzkej spolupráci s touto organizáciou, prihlásenie prebehlo jednoducho, a to prostredníctvom verifikácie danej inštitúcie, v mojom prípade VUT. Po prihlásení môže užívateľ využívať všetky dostupné servery. Celá infraštruktúra sa skladá z dvoch častí:

³<https://metavo.metacentrum.cz/>

- čelné uzly,
- výpočtové uzly.

Čelné uzly slúžia na vzdialené prihlásenie, kde sú uložené všetky dáta daného užívateľa. Tieto uzly neslúžia na zložité výpočty, slúžia skôr na spojenie medzi užívateľom a výpočtovým uzlom. Na spustenie programu, resp. úlohy, sa využívajú výpočtové uzly. Užívateľ špecifikuje o aké výpočtové parametre má záujem a systém ich vyhodnotí na základe dostupnosti zdrojov. V prípade, že zdroje sú dostupné, systém odošle úlohu do výpočtového uzla. V opačnom prípade systém odošle úlohu na spracovanie do výpočtového uzla až v čase, kedy budú zdroje opäť dostupné. Týmto sa úloha odošle do tzv. *čakacej fronty* a systém naplánuje čas jej spustenia na danom výpočtovom uzle.

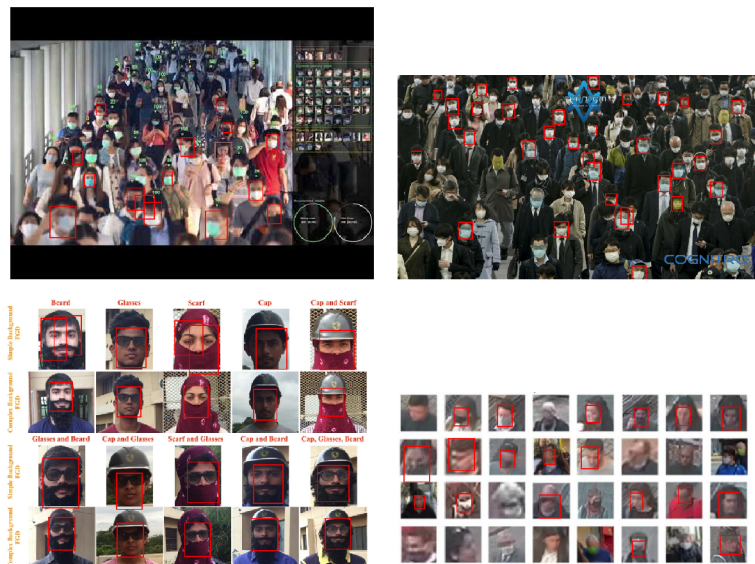
V MetaCentre sú rozlišované dve typy úloh:

- interaktívne úlohy,
- dávkové úlohy.

Rozdiel medzi týmito úlohami je, že interaktívne úlohy sa zadávajú prostredníctvom príkazového riadku (parametre ako sú veľkosť RAM, počet CPU, počet GPU a pod.). Dávkové úlohy sa píšú pomocou skriptu, kde skript s príslušnými informáciami sa odošle príkazom *qsub* [19].

4.1.5 Pribeh tréovania

Tréovanie bolo z časového hľadiska vskutku náročné. Ovplyvňovala to hlavne vyťaženosť GPU výpočtových uzlov. To spôsobilo čakanie a predlžovanie celého tréovania. Ďalším aspektom, ktorý ovplyvnilo dobu tréovania, bolo obmedzenie používania GPU výpočtových uzlov na maximálne 24 hodín. V dôsledku týchto problémov mi tréovanie trvalo celkovo 13 dní. Výsledný model po validácii dosahuje presnosť v priemere 75%, čo sa nemusí javiť ako dostatočné, ale aj napriek tomu dosahuje celkom uspokojujivé výsledky:



Obr. 4.2: Výsledky natrénovaného modelu. Obrázky dostupné z internetu.

Na súbore obrázkov 4.2 sú viditeľné početné defekty obrázkov, ako napr. slabá kvalita obrázka, prekrytie tváre nejakým predmetom alebo menšia veľkosť tváre, s ktorými si model poradil. V neposlednom rade sú viditeľné nepresnosti modelu ako napr. nepresné vyhodnotenie tváre alebo nedetekovanie tváre, ktoré sú spôsobené nedostatočným natrenovaním modelu.

4.1.6 Konfigurácia tréningovania

Konfigurácia tréningovania modelu je nevyhnutnou súčasťou tréningovania. Je to súbor parametrov, ktoré ovplyvňujú celý priebeh tréningovania. V priložených zdrojových súboroch sa nachádza v priečinku *detection* konfiguračný súbor *config.py*. Tento súbor využíva nastaviťelné parametre, ktoré sa odzrkadlia počas tréningovania. Nadčasovosť a volnosť tohto skriptu umožňuje užívateľom spúšťanie tréningu v dvoch variantoch - za použitia grafických výpočtových kariet alebo prostredníctvom RAM. Oba varianty sa líšia v kľúčovom aspekte, ktorý rozhoduje o časovej náročnosti daného tréningu.

4.1.7 Spustenie tréningového procesu

Tréningový proces sa spúšťa pomocou skriptu *main.py*. Všetky dodatočné parametre, ako sú *batch size* alebo rýchlosť učenia, sa definuje v spomínanom konfiguračnom skripte. Skript okrem samotnej inicializácie tréningového procesu umožňuje pokračovanie tréningu od už existujúceho natrénovaného modelu.

Nevyhnutnou úpravou je stiahnutie tréningovej sady. Ako tréningovú sadu som použil tréningovú sadu *WIDER FACE*, ktorá je popísaná v podkapitole 4.1.1. Samotné stiahnutie a usporiadanie celej tréningovej sady sú popísané v priloženom súbore *README.md*. Okrem iného je tu možnosť použitia podpornej knižnice *tensorboard*,⁴ ktorá vizualizuje samotné tréningovanie, a tak ponúka sledovanie vývoja tréningového procesu po jednotlivých iteráciách, resp. epochách.

Samotný priebeh tréningovania je tiež možné sledovať v textovom logovacom súbore, kde sú zaznamenané jednotlivé iterácie daného tréningu s príslušnými číselnými hodnotami, ako napr. stratovosť počas tréningovania. Po dokončení celej epochy sa vygeneruje aj priemerná presnosť identifikácie objektu.

4.1.8 Spustenie detekcie

Samotný proces spustenia detekcie na zadanej vstupnej snímke sa uskutočňuje prostredníctvom skriptu *detector.py*. Tento skript sa spúšťa pomocou nasledovných argumentov:

- *-model* názov modelu na použitie,
- *-image* názov vstupnej snímky,
- *-save_to* cesta uloženia výstupnej detekcie,
- *-face_save* cesta uloženia jednotlivých detekovaných ľudských hláv,
- *-keep* počet predpovedí na ponechanie. Implicitná hodnota je 150.

⁴dostupné z url: <https://pytorch.org/docs/stable/tensorboard.html>

4.2 Vylepšenie obrazu

Na vylepšenie obrazu som použil taktiež už existujúce riešenie. Priklonil som sa k použitiu GAN siete, konkrétne o variantu ESRGAN⁵, ktorá vylepšuje klasickú GAN sieť a taktiež urýchľuje trenovací čas modelu siete. Ako trénovaciu sadu použijem *Flickr-Faces-HQ Dataset*⁶, ktorý obsahuje širokú škálu ľudských tvári v priestore ale aj samostatnú sekciu, ktorej súčasťou sú len ľudské tváre.

4.2.1 Trénovacia sada

Prostredníctvom trénovacej sady pri vylepšovaní obrazu sa definuje ako sa daná neurónová sieť bude vo svojej finálnej fáze správať. Trénovacia sada ovplyvňuje či vylepšenie obrazu bude založené vylučne na skvalitnení obrazu, či bude založené len na zväčšení obrazu alebo kombináciou oboch týchto fenoménov. Problém, ktorý vzniká pri vylepšovaní obrazu v tejto práci je, že vstupné snímky budú v nízkej kvalite a zároveň väčšina vstupov bude mať malú veľkosť. Preto sa trénovacia sada musí zamerať na dva aspekty, ktoré musia byť zakomponované v trénovacej sade, aby daná sieť bola schopná jednoznačne interpretovať jednotlivé vstupné snímky. Týmito aspektami sú veľkosť a rozlíšenie.

Na základe týchto skutočností som sa rozhodol použiť trénovaciu sadu *Flickr Faces HQ Dataset*⁶, ktorá obsahuje širokú škálu ľudských tvári. Obsahuje aj predpripravenú sadu snímok ľudských tvári, ktoré sú vo vysokom rozlíšení a majú fixnú veľkosť, ktorá je 1024×1024 . Táto sada obsahuje vyše 70 000 rôznorodých ľudských snímok, t.j. od malých detí až po starších ľudí, obsahuje ľudské tváre s okuliarmi, šatkami, rôznymi výrazmi tváre a pod.



Obr. 4.3: Ukážka snímok z trénovacej sady *Flickr Faces HQ Dataset*.

Na obrázku 4.3 je možné vidieť úkážky snímok z príslušnej trénovacej sady. Všetky obrázky obsiahnuté v tejto sekcii trénovacej sady sú vo vysokom rozlíšení a majú pevnú veľkosť.

4.2.2 Zostavenie trénovacej sady

Pre zostavenie trénovacej sady pre samotný trénovací proces GAN siete 3.1.7 je potrebné pôvodný trénovací dataset rozdeliť do dvoch samostatných trénovacích sád. Prvá sada, označovaná ako *HR* sada, je tvorená snímkami z pôvodného testovacieho datasetu *Flickr Faces*

⁵zdrojové súbory dostupné z <https://github.com/BlueAmulet/ESRGAN>

⁶dataset dostupný z url: <https://github.com/NVlabs/ffhq-dataset>

HQ Dataset. Druhá sada, označovaná ako *LR* sada, je tvorená snímkami z novovzniknutej testovacej *HR* sady. Samotný proces úpravy *LR* snímkov sa odvíja od vstupu, ktorý sa očakáva pre danú sieť. Keďže sa na vstupe očakáva snímka malého formátu a nízkej kvality, bolo nevyhnutné tieto predispozície zakomponovať do samotného procesu.

Na zostavenie *HR* sady som použil skript *createHRDataset.py*. Keďže testovacia sada obsahuje veľké množstvo podpriechnikov a zároveň sú snímky o veľkosti 1024×1024 , tak sa pomocou už spomínaného skriptu premiestnia všetky snímky do jedného priechniku, pričom každá snímka sa zmenší na veľkosť 256×256 . Táto veľkosť sa upravuje z dôvodu zjednodušenia tréningového procesu.

Podobným spôsobom došlo k zostaveniu *LR* sady, na ktorú bol použitý skript *createLR-Dataset.py*. V tomto prípade sa ale veľkosť snímky znížila až štvornásobne oproti veľkosti snímky z *HR* sady. Týmto sa splnili aspekty spomínané v sekcii 4.2.1 a to, že výsledná snímka spracovaná touto sieťou bude štvornásobne väčšia oproti vstupnej snímke a z kvalitatívneho hľadiska vylepšená.



Obr. 4.4: Úprava datasetu na príslušné snímky s vysokým a nízkym rozlíšením

4.2.3 Tréningovanie modelu

Tréningové zdrojové súbory neboli súčasťou zdrojových súborov, ktoré sú definované v sekcii 4.2. Preto som použil tréningové zdrojové súbory⁷, ktoré sú predpripravené na tréningový proces.

Prvotné tréningovanie modelu bolo inicializované prostredníctvom organizácie MetaCentrum 4.1.4. Problémom ale bolo, že výpočtový uzol z neznámej príčiny nedokázal inicializovať tréningový proces. Po tomto probléme som si prostredníctvom firmy IT4BI⁸ zabezpečil súkromný výpočtový server bez GPU podpory. Vzhľadom na to, že podpora GPU nebola zabezpečená, bolo potrebné zredukovať dataset na necelých 1 000 snímkov. Zoznam parametrov, ktoré boli použité na inicializáciu tréningového procesu sú zaznačené v tabuľke 4.1. V dôsledku toho, že tréningovanie modelu tejto neurónovej siete nebolo možné inicializovať na servery s GPU podporou, celkový čas strávený na samotnom tréningovaní bol veľmi veľký. Priemerný čas strávený na jednej iterácii predstavoval 16 minút, čo spôsobilo, že tréningový proces zabral vyše 22 dní. Bolo vykonaných 2 000 iterácií čo v prepočte na epochy, znamená okolo 666 epoch. Počas tréningového procesu sa ukladali hodnoty váh periodicky po dokončení 200 epoch a samotný diskriminátor a generátor obdobne periodicky po dokončení 300 epoch.

⁷zdrojové súbory dostupné z <https://github.com/BlueAmulet/BasicSR>

⁸odkaz na webovú stránku <https://www.it4bi.com/>

4.2.4 Konfigurácia tréningového procesu

Súčasťou inicializácie tréningového procesu je konfiguračný súbor, ktorý obsahuje nastaviiteľné parametre, ktoré sa aplikujú na daný tréningový proces. V tabuľke 4.1 sú definované hodnoty základných parametrov, ktoré boli použité pri tréningu modelu ESRGAN siete. Celková konfigurácia tréningového procesu sa nachádza v súbore *train_template.yml*, ktorý je súčasťou zdrojových tréningových súborov.

Parameter	Hodnota
batch_size	32
learning_rate	0.0001
device	CPU
HR_size	256
iterations	10 000

Tabuľka 4.1: Konfigurácia tréningového procesu.

4.2.5 Inicializácia tréningového procesu

Tréningový proces sa spustí pomocou skriptu *train.py* s nasledujúcimi parametrami:

- *-opt* konfiguračný súbor,
- *-simple* spustí ukladanie logovacích hlášok do súboru.

4.3 Spustenie vylepšovania obrazu

Po úspešnom dokočení tréningového procesu ESRGAN siete, sa vytvoria dva modely - generátor a diskriminátor. Generátor je model, ktorý sa využije na samotné vylepšenie obrazu. Spustenie testovacieho procesu sa vykonáva prostredníctvom skriptu *test.py*, ktorý sa nachádza v priečinku *ESRGAN-test*.

Tento skript sa spúšťa s použitím nasledujúcich argumentov:

- *-name_of_model* názov natrénovaného modelu,
- *-input* priečinok so vstupnými snímkami,
- *-output* priečinok s výstupnými snímkami,
- *-cpu* s možnosťou použitia GPU alebo CPU.

Výsledné vylepšené snímky ľudských tvárí sa uložia do priečinka, ktorý sa zdefinuje prostredníctvom vstupného argumentu. V opačnom prípade sa tieto novovzniknuté snímky ľudských tvárí uložia do preddefinovaného priečinka *results*.

4.4 Možné problémy pri tréningu GAN sietí

Tréning GAN sietí so sebou prináša množstvo problémov, na ktoré doposiaľ neexistuje unikátne riešenie, a ktoré by tieto problémy úplne eliminovalo. Preto sa súčasné výskumy zaoberajú analýzou týchto problémov, čo by malo za následok ich odstránenie [5].

Medzi najčastejšie problémy, ktoré spôsobujú zlyhanie tréningového procesu, je, že diskriminátor je príliš dobre natrénovaný. To spôsobuje, že generátor môže pri tréningu zlyhať kvôli tzv. miznúcim gradientom. To má za následok, že aktualizácia váh na dosiahnutie najoptimálnejších hodnôt bude prebiehať veľmi malými zmenami a pri dokončení epochy neprinesie žiadne zlepšenie z hľadiska pokroku pri učení. Opakom je situácia, kedy aktualizácia váh bude prebiehať veľmi prudkými zmenami, čo spôsobí prudké zmeny hodnôt váh a náročnosť nájdenia optimálnej hodnoty váh [5, 3].

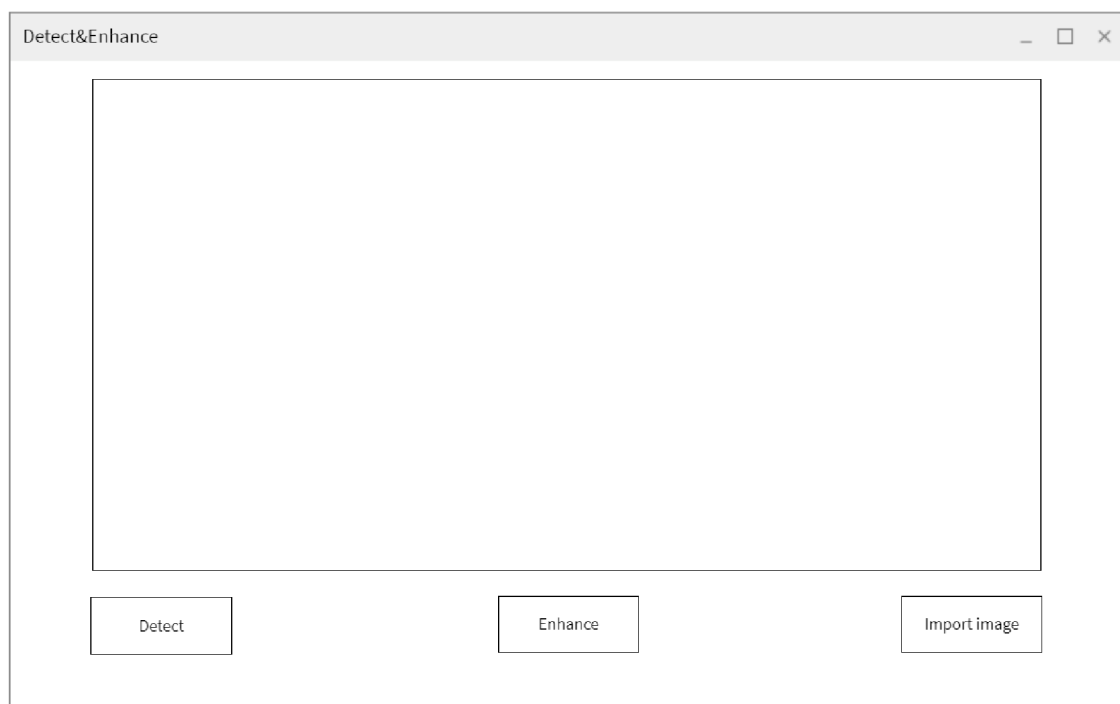
Možnými postupmi ako tomuto procesu zabrániť je použitie tzv. *Wasserstein loss*, ktorá zabraňuje postupnému miznutiu gradientu [5].

4.5 Uživatelské rozhranie

Na prepojenie detekcie a vylepšenia obrazu, ktoré tvoria majoritnú časť programu, som sa rozhodol vytvoriť grafické rozhranie. Hlavným cieľom pri vytváraní grafického rozhrania bolo vytvoriť a prispôbiť toto rozhranie tak, aby bolo jednoduché na ovládanie a na prácu s pomerne složitou tematikou pre užívateľa. Výsledkom je aplikácia s grafickým rozhraním, ktorá je jednoduchá na obsluhu a intuitívna na prípadnú analýzu medzivýsledkov, resp. výsledkov.

4.5.1 Návrh užívateľského rozhrania

Prioritou grafickej podoby rozhrania je poskytnúť užívateľom jednoduchosť, intuitívnosť ovládania a v neposlednom rade dostupnosť pre širokú škálu užívateľov. Podoba výsledného návrhu grafického rozhrania je zobrazená na obrázku 4.5.



Obr. 4.5: Grafické užívateľské rozhranie - hlavná obrazovka.

Ako je možné vidieť na obrázku 4.5, vstupom bude obrázok, ktorý sa vloží prostredníctvom aktivovania ikony *Import image* lokálne z počítača. Následne sa po spustení programu kliknutím na ikonu *Detect* vykoná detekcia ľudských tvárí, kde výsledná snímka zdetekovaných ľudských tvárí nahradí pôvodnú snímku. Po tejto operácii sa objaví nová ikona *Enhance*, ktorej úlohou je zdetekované ľudské tváre vylepšiť z hľadiska kvality. Táto ikona je po dobu, pokiaľ nie je vykonaná detekcia ľudských tvárí zo zdrojovej snímky, bez možnosti použitia.

Výsledkom operácie vylepšenia je okno s posuvným obsahom, ktoré pozostáva z výsledných snímok vylepšených zdetekovaných ľudských tvárí.

4.5.2 Grafické podporné knižnice

Grafické rozhranie aplikácie umožňuje užívateľom zjednodušiť použitie výslednej aplikácie, ktorá svojím obsahom a implementáciou sťažuje spustenie celého procesu.

Na implementáciu grafického rozhrania aplikácie sa využíva programovací jazyk *python*⁹, ktorý obsahuje rôzne podporné knižnice, ktoré uľahčujú vývoj daného grafického rozhrania

Medzi najfrekvencovanejšie knižnice, ktoré jazyk *python* poskytuje na implementáciu grafického rozhrania, patria¹⁰:

- *PyQT*
- *Tkinter*
- *Kivy*

PyQT je knižnica, ktorá je postavená na platforme QT¹¹. Prostredníctvom tejto platformy vzhľad aplikácie je skonštruovaný vo vývojovom grafickom prostredí, kde užívateľ si zostaví potrebný vzhľad a následne logiku aplikácie separátne vyvíja a napája na existujúci dizajn.

Tkinter je štandardné grafické rozhranie jazyka *python*. Práca prostredníctvom tejto knižnice je jednoduchá a intuitívna. Radí sa medzi najjednoduchšie grafické knižnice jazyka *python* z hľadiska náročnosti práce s ňou.

Kivy je knižnica jazyka *python*, ktorá je skonštruovaná na rýchly vývoj aplikácií. Je to multiplatformová knižnica, ktorá ponúka grafický návrh aplikácie prostredníctvom jazyka *Kivy*, ktorý svojou štruktúrou sa podobá na jazyk *css*¹². Najčastejšie sa používa pri vývoji hier.

4.6 Implementácia

4.6.1 Detekcia a vylepšenie obrazu

Oba algoritmy sú implementované programovacím jazykom *python*, ktorý ponúka širokú škálu knižníc. Tieto knižnice nám na jednej strane uľahčujú tvorbu neurónových sietí a na druhej strane obsluhu týchto sietí ako aj tréning modelu danej siete, samotné testovanie modelu a pod. Konkrétne sa jedná o verziu 3.8, ktorá bola použitá aj pre detekciu aj pre vylepšenie obrazu. Na prácu s neurónovými sieťami bola využitá knižnica *Pytorch*¹³ vo verzii

⁹<https://www.python.org/>

¹⁰<https://analyticsindiamag.com/8-python-gui-frameworks-for-developers/>

¹¹<https://www.qt.io/>

¹²<https://www.w3.org/Style/CSS/Overview.en.html>

¹³<https://pytorch.org/>

1.8.0. Táto knižnica sa využila aj na modelovanie samotných architektúr neurónových sietí. Ako hodnoty váh boli použité predtrénované hodnoty z VGG16 architektúry.

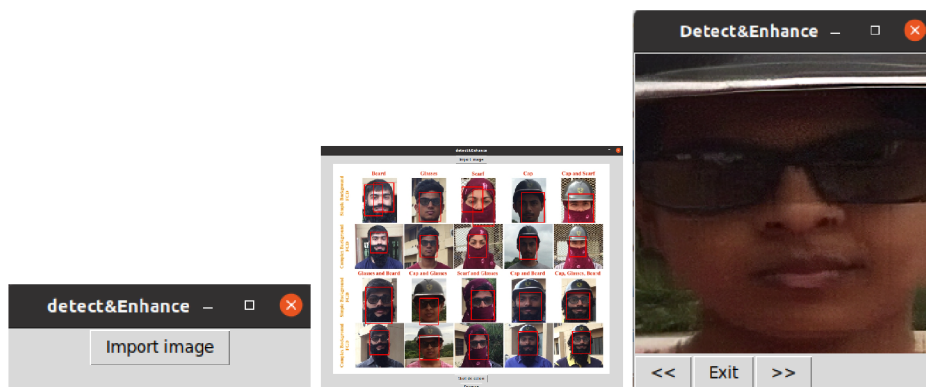
Práca s obrazom ako takým bola zabezpečená prostredníctvom knižnice *opencv*¹⁴ vo verzii 4.5.2.52 a pomocou knižnice *Pillow*¹⁵ vo verzii 8.1.2. Na zobrazenie detekovaných ľudských tvárí a zobrazenie výslednej detekcie sa použila *matplotlib*¹⁶ knižnica vo verzii 3.4.1.

Výsledkom sú dva priečinky, pričom každý priečinnok reprezentuje jednotlivú operáciu - *Detection* ako detekcia ľudských tvári a *ESRGAN* ako vylepšenie obrazu prostredníctvom GAN siete.

4.6.2 Grafické rozhranie

Grafické rozhranie je implementované pomocou jazyka *python*. Na túto časť algoritmu som sa rozhodol použiť grafickú knižnicu *tkinter*, a to z dvoch dôvodov. Prvým a zásadným dôvodom bola nenáročnosť údržby zdrojových kódov a fakt, že táto knižnica je súčasťou grafického rozhrania *python*. Inými slovami, túto knižnicu nie je potrebné externe inštalovať, pretože je obsiahnutá jazykom *python*. Druhým dôvodom bola moja skúsenosť s touto knižnicou a práca s ňou v rôznych operačných systémoch, napr. *Windows*, *Linux*, *Unix* a pod.

Aplikácia sa spúšťa pomocou skriptu *mainApp.py*, ktorý sa nachádza v hlavnom priečinku *Detect&Enhance*. Koncepcia aplikácie je založená na výbere vstupného obrázka. Po úspešnom načítaní obrázka je užívateľ schopný spustiť proces detekcie, ktorého výsledkom sú detekované ľudské tváre. Tie sa následne uložia do priečinku *faces* s príslušným názvom podpriečinku pre jednoduchú navigáciu. Po tomto procese sa užívateľovi spístupní finálna etapa aplikácie - vylepšenie obrazu. Po tejto etape sa užívateľovi zobrazia výsledne snímky ľudských tvárí s veľkosťou 256×256 ako separátne okno s posuvným algoritmom na prechádzanie celým zoznamom daných ľudských tvárí.



Obr. 4.6: Ukážka výstupov z aplikácie.

Na obrázku 4.6 sú ukázané jednotlivé prechody aplikácie, akými sú úvodná inicializácia aplikácie s následnou možnosťou výberu snímky na detekovanie, proces detekcie a následný výstup procesu vylepšenia obrazu v podobe koláže snímok.

¹⁴<https://pypi.org/project/opencv-python/>

¹⁵<https://pypi.org/project/Pillow/>

¹⁶<https://matplotlib.org/>

Kapitola 5

Testovanie

Na otestovanie praktickej časti bakalárskej práce bolo nevyhnutné zostaviť sadu testovacích snímok, ktoré čo najlepšie overia presnosť natrénovaných modelov neurónových sietí. Súčasťou testovacích dát boli snímky rozdelené na základe dvoch parametrov. Čo sa týka časti o detekcii ľudských tvárí, tak ich rozdeľovacími parametrami, ktoré vytvorili adekvátne testovacie sady, boli početnosť ľudských tvárí na danej snímke a kvalita danej snímky. Prvá časť testovacej sady bola prevažne zložená zo snímok, ktoré boli využité na tréning daného modelu s malou početnosťou výskytu ľudských tvárí a pomerne dostatočnou kvalitou snímok. Druhá časť testovacej sady bola založená na snímkach, ktoré z hľadiska početnosti ľudských tvárí, vykazovali nárast, čo malo za úlohu otestovať detektor z kvantitatívneho hľadiska a zároveň bola znížená kvalita daných snímok. Posledná sada bola založená na reálnych snímkach dostupných z internetu, ako napr. *Google Earth*¹, snímky z rôznych článkov a denníkov a časť snímok pozostávala z datasetu *QMUL-TinyFace*², ktorý je zložený explicitne z malých ľudských tvárí.

Druhou časťou práce bolo vylepšenie obrazu. Testovanie tejto časti prebiehalo v dvoch fázach. Prvou fázou bolo testovanie tejto časti práce pomocou tréningových sád, ktoré boli vytvorené na vyhodnotenie úspešnosti natrénovania modelu neurónovej siete tohto algoritmu. Druhou fázou bolo testovanie snímkami, ktoré prešli detekciou prostredníctvom detektora. Tým sa jednak otestovala funkčnosť a presnosť samotného detektora ako aj následného vylepšenia obrazu.

5.1 Testovanie detektora

Testovanie detektora prebiehalo prostredníctvom linuxovej distribúcie Ubuntu 20.4 a Hardwarového vybavenia, ktoré pozostávalo z: Intel i7-1065G7, 16GB RAM, integrovanej grafickej karte Intel IRIS PLUS GRAPHICS a dedikovanej grafickej karte NVIDIA GeForce MX250.

Testovanie prebiehalo v dvoch režimoch, a to pomocou integrovanej grafickej karte a prostredníctvom dedikovanej grafickej karte. Vo všeobecnosti sa výsledky z jednotlivých režimov nijak významne neodlišujú, ale rozdiel môžeme bádať v jednom aspekte, a tým je strojový čas. Kdežto použitím integrovanej grafickej karte sa strojový čas pohyboval priemerne okolo piatich sekúnd na jednu detekciu použitím dedikovanej grafickej karte sa

¹<https://www.google.sk/intl/sk/earth/>

²<https://qmul-tinyface.github.io/>

tento strojový čas znížil v priemere na dve sekundy, čo spôsobilo zníženie o 60 % a urýchlenie testovania.

5.1.1 Testovacia sada č. 1

Táto sada obsahovala okolo 500 náhodných snímok z tréningového datasetu, ktorý sa využil na tréning samotného modelu.



Obr. 5.1: Výsledna detekcia testovacej sady č. 1.

Na obrázku 5.1 sú viditeľné v ojedinelých prípadoch aj chybné detekcie, pri ktorých sa domnievam, že vznikli ako následok nedostatočného natrénovania modelu. Tieto chybné detekcie ale zatieňujú fakt, že model identifikoval v tejto testovacej sade 100 % ľudských tvárí. Výsledné štatistiky náhodne vybraných snímok z tejto sady sú zhrnuté v tabuľke 5.1.

Číslo snímky	Počet ľudských tvárí	Detekované ľudské tváre	Nadbytočné detekcie
1	4	4	3
2	3	3	2
3	3	3	1
4	2	2	1
5	3	3	1
6	3	3	0
7	4	4	3
8	4	4	2
9	2	2	2

Tabuľka 5.1: Výsledky testovacej sady č. 1.

Ako z tabuľky 5.1 vyplýva, sú pozorovateľné nadbytočné detekcie u niektorých snímok viac a u niektorých je detekcia vyhodnotená bez nadbytočných chybných detekcií.

5.1.2 Testovacia sada č. 2

Táto sada obsahovala menej testovacích snímok, ale dôraz bol kladený na snímky so zvýšeným počtom ľudských tvárí a v neposlednom rade aj kvalita samotných snímok bola zhoršená. Celkový počet snímok tejto testovacej sady predstavoval 400 snímok.



Obr. 5.2: Výsledna detekcia testovacej sady č. 2.

Na obrázku 5.2 vidíme dvojicu výsledných detekcií. Je možné vidieť, že detektor rozoznal pomerne ťažšie detekovateľné ľudské tváre. Časť tvárí aj napriek tomu zostala mimo obor detekovaných tvárí, čo má opäť za následok nedostatočné natrénovanie modelu. V tabuľke 5.2 sú opäť náhodné vybrané snímky z tejto testovacej časti pre vizualizáciu úspešnosti detektora.

Číslo snímky	Počet ľudských tvárí	Detekované ľudské tváre	Nadbytočné detekcie
1	10	6	2
2	8	3	0
3	11	4	0
4	25	16	5
5	10	9	1
6	23	19	11
7	8	8	0

Tabuľka 5.2: Výsledky testovacej sady č. 2.

Z tabuľky 5.2 vyplýva, že detektor ťažšie zvláda detekovanie väčšieho množstva ľudských tvárí z jednej snímky a taktiež diaľka, z ktorej bola snímka vytvorená, pridáva na náročnosti detekcie. Z tejto sady detektor vyhodnotil úspešne necelých 68 % ľudských tvárí.

5.1.3 Testovacia sada č. 3

Posledná testovacia sada bola zložená z náhodných snímok z reálneho života, kde dôraz sa kládol na nízku kvalitu snímky a početnosť ľudských tvárí na snímke. Celkový počet testovacích snímok bol 600.



Obr. 5.3: Výsledna detekcia testovacej sady č. 3.

Ako je možné vidieť na obrázku 5.3, vstupné snímky boli veľmi nejasné a tváre boli ťažko detekovateľné. Ale aj napriek tomu detektor vrátil uspokojivé výsledky. V tabuľke 5.3 sú znázornené nahodné snímky z testovacej sady pre znázornenie schopnosti detektora rozpoznať objekt ľudskej tváre.

Číslo snímky	Počet ľudských tvárí	Detekované ľudské tváre	Nadbytočné detekcie
1	9	8	3
2	3	2	2
3	7	5	1

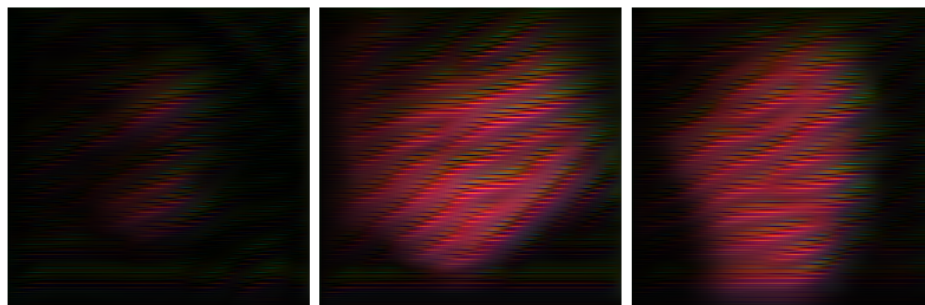
Tabuľka 5.3: Výsledky testovacej sady č. 3.

Prostredníctvom hodnôt z tabuľky 5.3 detektor dosiahol presnosť necelých 79 % a zároveň 73 % v rámci celej testovacej sady.

5.2 Testovanie vylešenia obrazu

Testovanie vylepšenia obrazu neprebehlo podľa stanovených scenárov a testovacích snímok. Výsledný model, ktorý vznikol tréningovým procesom v časti 4.2.3, pri spustení zakaždým vráti snímku čiernej farby. Usudzujem, že tento výsledok bol dosiahnutý z nízkeho počtu iterácií, ktoré boli vykonané počas tréningového procesu. Taktiež som vyskúšal aj model, ktorý vyšiel ako výsledok z tréningového procesu prvých 1000 iterácií, a ktorý dosahoval lepších čísel z hľadiska presnosti, pričom výsledok bol rovnaký.

Po tomto neúspechu som sa znova pokúsil inicializovať tréningový proces a po prvých 500 iteráciách som otestoval výslednú funkčnosť modelu.



Obr. 5.4: Výsledky modelu na vylepšenie obrazu.

Na súbore obrázkov 5.4 vidíme náznak siluet ľudských tvárí. Ale aj napriek tomu je počet iterácií, ktoré boli uskutočnené počas tréningového procesu, primálo. Parametre, ktoré boli použité na tréning tohto modelu boli identické ako parametre, ktoré boli použité na tréning prvotného modelu.

Kapitola 6

Záver

Cieľom tejto bakalárskej práce bolo vytvoriť algoritmus založený na dvoch rozličných ale zato nadväzujúcich časti, a to detekcia ľudských tvárí a vylepšenie obrazu týchto ľudských tvárí. Výsledkom práce je aplikácia, ktorá je schopná detekovať ľudské tváre na vstupných snímkach. Priemerná presnosť detekcie dosahuje okolo 70 %, čo aj odpovedá presnosti dosiahnutej z prebraných zdrojových súborov¹. Taktiež táto aplikácia podporuje vylepšenie obrazu vstupných ľudských tvárí, ale kvôli nedostatočnému natrénovanému modelu dosahuje táto časť neuspokojivé výsledky.

Toto správanie môžeme pripísať ako dôsledok použitia serveru bez GPU podpory, čo značne predĺžilo tréningový proces a znemožnilo otestovať model častejšie počas tréningového procesu, a tým zmeniť inicializačné parametre pre samotný tréningový proces. Následný druhý pokus o natrénovanie modelu dopadol o niečo lepšie s mierne zlepšenými výsledkami.

Najväčším prínosom tejto práce pre mňa bolo preskúmanie problémov detekcie ľudských tvárí na snímkach s nízkym rozlíšením, vylepšenie obrazu, práca s neurónovými sieťami, ktorých vyriešenie by prinieslo obrovský technologický posun v rozpoznávaní, detekcii a vylepšovaní obrazu.

Možným vylepšením, ktoré by túto aplikáciu posunuli na vyššiu úroveň, by mohlo byť skombinovanie dvoch tréningových sád na detekciu - *WIDER FACE* a *Tinyfaces*². To by zaistilo detekciu veľmi malých ľudských tvárí a taktiež detekcia by bola schopná detekovať tváre aj vo veľmi nízkej kvalite snímkov. Taktiež by mohla zaistiť detekciu a vylepšenie obrazu v reálnom čase vo videu.

¹<https://github.com/louis-she/sfd.pytorch>

²<https://qmul-tinyface.github.io/>

Literatúra

- [1] ANWAR, S., KHAN, S. a BARNES, N. *A Deep Journey into Super-resolution: A survey* [online]. 2020. Dostupné z: <https://arxiv.org/abs/1904.07523>.
- [2] CHEN, Y., TAI, Y., LIU, X., SHEN, C. a YANG, J. FSRNet: End-to-End Learning Face Super-Resolution with Facial Priors. In: IEEE. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, s. 2492–2501. DOI: 10.1109/CVPR.2018.00264. ISBN 978-1-5386-6421-6. Dostupné z: <https://ieeexplore.ieee.org/document/8578362>.
- [3] COMPANY, D. *Machine Learning & Deep Learning Fundamentals: Vanishing & Exploding Gradient Explained | A Problem Resulting From Backpropagation* [online]. 2018. 23.03.2018 [cit. 2021-04-18]. Dostupné z: https://deeplizard.com/learn/video/q0_NLVjD6zE.
- [4] COSKUN, M., UCAR, A., YILDIRIM, O. a DEMIR, Y. Face recognition based on convolutional neural network. In: IEEE. *2017 International Conference on Modern Electrical and Energy Systems (MEES)*. 2017, s. 376–379. DOI: 10.1109/MEES.2017.8248937. ISBN 978-1-5386-1751-9. Dostupné z: <https://ieeexplore.ieee.org/document/8248937>.
- [5] DEVELOPERS, G. *Common Problems* [online]. 2020 [cit. 2021-04-18]. Dostupné z: <https://developers.google.com/machine-learning/gan/problems>.
- [6] DU, J. Understanding of Object Detection Based on CNN Family and YOLO. *Journal of Physics: Conference Series*. 1. vyd. IOP Publishing. apr 2018, zv. 1004, č. 1004, s. 012029. DOI: 10.1088/1742-6596/1004/1/012029. Dostupné z: <https://doi.org/10.1088/1742-6596/1004/1/012029>.
- [7] GAO, S. a GRUEV, V. Bilinear and bicubic interpolation methods for division of focal plane polarimeters. *Opt. Express*. 1. vyd. OSA. Dec 2011, zv. 19, č. 27, s. 26161–26173. DOI: 10.1364/OE.19.026161. Dostupné z: <http://www.opticsexpress.org/abstract.cfm?URI=oe-19-27-26161>.
- [8] GIRSHICK, R. Fast R-CNN. In: IEEE. *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, s. 1440–1448. DOI: 10.1109/ICCV.2015.169. ISBN 978-1-4673-8391-2. Dostupné z: <https://ieeexplore.ieee.org/document/7410526>.
- [9] GOODFELLOW, I. J., POUGET ABADIE, J., MIRZA, M., XU, B., WARDE FARLEY, D. et al. Generative Adversarial Nets. In: ACM. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. Cambridge, MA, USA: MIT Press, 2014, s. 2672–2680. NIPS'14.

- [10] HAN, D. Comparison of Commonly Used Image Interpolation Methods. In: *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering*. Atlantis Press, 2013/03, s. 1556–1559. DOI: <https://doi.org/10.2991/iccsee.2013.391>. ISBN 978-90-78677-61-1. Dostupné z: <https://doi.org/10.2991/iccsee.2013.391>.
- [11] HANSPAL, R. K. a SAHOO, K. A Survey of Image Enhancement Techniques. In: Semantics scholar. *International Journal of Science and Research (IJSR)*. 2017, sv. 6, č. 5, s. 2467–2671. ISSN 2319-7064. Dostupné z: <https://www.semanticscholar.org/paper/A-Survey-of-Image-Enhancement-Techniques-Hanspal-Sahoo/1181a6505c9eef6869904f3cb72501088f7dab95?p2df>.
- [12] HE, K., ZHANG, X., REN, S. a SUN, J. Deep Residual Learning for Image Recognition. In: IEEE. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, s. 770–778. DOI: 10.1109/CVPR.2016.90. ISBN 978-1-4673-8852-8. Dostupné z: <https://ieeexplore.ieee.org/document/7780459>.
- [13] ISOLA, P., ZHU, J., ZHOU, T. a EFROS, A. A. Image-to-Image Translation with Conditional Adversarial Networks. *CoRR*. 1. vyd. 2016, abs/1611.07004, č. 1. Dostupné z: <http://arxiv.org/abs/1611.07004>.
- [14] LAN, W., DANG, J., WANG, Y. a WANG, S. Pedestrian Detection Based on YOLO Network Model. In: IEEE. *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*. 2018, s. 1547–1551. DOI: 10.1109/ICMA.2018.8484698. ISBN 978-1-5386-6076-8. Dostupné z: <https://ieeexplore.ieee.org/document/8484698>.
- [15] LEDIG, C., THEIS, L., HUSZÁR, F., CABALLERO, J., CUNNINGHAM, A. et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: IEEE. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, s. 105–114. DOI: 10.1109/CVPR.2017.19. ISBN 978-1-5386-0458-8. Dostupné z: <https://ieeexplore.ieee.org/document/8099502>.
- [16] LI, M., ZHANG, Z., YU, J. a CHEN, C. W. Learning Face Image Super-Resolution Through Facial Semantic Attribute Transformation and Self-Attentive Structure Enhancement. *IEEE Transactions on Multimedia*. 1. vyd. 2021, č. 23, s. 468–483. DOI: 10.1109/TMM.2020.2984092. Dostupné z: <https://ieeexplore.ieee.org/document/9055090>.
- [17] LIETAVCOVÁ, Z. *Konvoluční neuronové sítě*. Brno, CZ, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.fit.vut.cz/study/thesis/19975/>.
- [18] LIU, W., ANGUELOV, D., ERHAN, D., SZEGEDY, C., REED, S. et al. SSD: Single Shot MultiBox Detector. In: LEIBE, B., MATAS, J., SEBE, N. a WELLING, M., ed. *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, s. 21–37. ISBN 978-3-319-46448-0.
- [19] METACENTRUM. *Průvodce pro začátečníky: Gridové počítání: základní myšlenka* [online]. Uživatelský manuál, 1. vyd. Česká republika: MetaCentrum, október 2019, 17.03.2021 [cit. 08.04.2021]. Dostupné z: https://wiki.metacentrum.cz/wiki/Pruvodce_pro_zacatecniky.

- [20] RAJPUT, S. S., ARYA, K. V., SINGH, V. a BOHAT, V. K. Face Hallucination Techniques: A Survey. In: IEEE. *2018 Conference on Information and Communication Technology (CICT)*. 2018, s. 1–6. DOI: 10.1109/INFOCOMTECH.2018.8722416. ISBN 978-1-5386-8216-6. Dostupné z: <https://ieeexplore.ieee.org/document/8722416>.
- [21] REDMON, J., DIVVALA, S., GIRSHICK, R. a FARHADI, A. You Only Look Once: Unified, Real-Time Object Detection. In: IEEE. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, s. 779–788. DOI: 10.1109/CVPR.2016.91. ISBN 978-1-4673-8852-8. Dostupné z: <https://ieeexplore.ieee.org/document/7780460>.
- [22] REN, S., HE, K., GIRSHICK, R. a SUN, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1. vyd. 2017, zv. 39, č. 6, s. 1137–1149. DOI: 10.1109/TPAMI.2016.2577031. ISSN 1939-3539. Dostupné z: <https://ieeexplore.ieee.org/document/7485869>.
- [23] ROZUMNYI, D., OSWALD, M. R., FERRARI, V., MATAS, J. a POLLEFEYS, M. DeFMO: Deblurring and Shape Recovery of Fast Moving Objects. *CoRR*. 2020, abs/2012.00595. Dostupné z: <https://arxiv.org/abs/2012.00595>.
- [24] SHAH, A. J. a GUPTA, S. B. Image super resolution-A survey. In: IEEE. *2012 1st International Conference on Emerging Technology Trends in Electronics, Communication Networking*. 2012, s. 1–6. DOI: 10.1109/ET2ECN.2012.6470098. ISBN 978-1-4673-1628-6. Dostupné z: <https://ieeexplore.ieee.org/document/6470098>.
- [25] SRIVASTAVA, A., MANE, S., SHAH, A., SHRIVASTAVA, N. a THAKARE, B. A survey of face detection algorithms. In: IEEE. *2017 International Conference on Inventive Systems and Control (ICISC)*. 2017, s. 1–4. DOI: 10.1109/ICISC.2017.8068607. ISBN 978-1-5090-4716-1. Dostupné z: <https://ieeexplore.ieee.org/document/8068607>.
- [26] UK, I. A Review on Image Enhancement Techniques. *International Journal of Engineering and Applied Computer Science (IJEACS)*. 1. vyd. August 2017, zv. 02, č. 139, s. 232–235. DOI: 10.24032/ijeacs/0207/05. Dostupné z: https://www.researchgate.net/publication/319395669_A_Review_on_Image_Enhancement_Techniques.
- [27] VAISHALI PATEL1, P. K. M. A Review on Different Image Interpolation Techniques for Image Enhancement. In: IJETAE. *International Journal of Emerging Technology and Advanced Engineering*. 2013, sv. 3, s. 129–133. ISBN 2250-2459. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.443.5747&rep=rep1&type=pdf>.
- [28] VIKRAM, K. a PADMAVATHI, S. Facial parts detection using Viola Jones algorithm. In: IEEE. *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. 2017, s. 1–4. DOI: 10.1109/ICACCS.2017.8014636. ISBN 978-1-5090-4560-0. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/8014636>.

- [29] VIOLA, P. a JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. *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. ISBN 0-7695-1272-0. Dostupné z: <https://ieeexplore.ieee.org/document/990517>.
- [30] WANG, X., YU, K., WU, S., GU, J., LIU, Y. et al. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: LEAL TAIXÉ, L. a ROTH, S., ed. *Computer Vision – ECCV 2018 Workshops*. Cham: Springer International Publishing, 2019, s. 63–79. ISBN 978-3-030-11021-5.
- [31] YANG, Z., SHI, P. a PAN, D. A Survey of Super-Resolution Based on Deep Learning. In: IEEE. *2020 International Conference on Culture-oriented Science Technology (ICCST)*. 2020, s. 514–518. DOI: 10.1109/ICCST50977.2020.00106. ISBN 978-1-7281-8139-4. Dostupné z: <https://ieeexplore.ieee.org/document/9262829>.
- [32] YUAN, Y., WANG, Y. a CAO, F. Optimization approximation solution for regression problem based on extreme learning machine. *Neurocomputing*. 1. vyd. 2011, zv. 74, č. 16, s. 2475 – 2482. DOI: <https://doi.org/10.1016/j.neucom.2010.12.037>. ISSN 0925-2312. Advances in Extreme Learning Machine: Theory and Applications Biological Inspired Systems. Computational and Ambient Intelligence. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0925231211002505>.
- [33] YUN, J. U., JO, B. a PARK, I. K. Joint Face Super-Resolution and Deblurring Using Generative Adversarial Network. *IEEE Access*. 1. vyd. 2020, zv. 8, č. 8, s. 159661–159671. DOI: 10.1109/ACCESS.2020.3020729.
- [34] ZANGENEH, E., RAHMATI, M. a MOHSENZADEH, Y. Low resolution face recognition using a two-branch deep convolutional neural network architecture. *Expert Systems with Applications*. 1. vyd. 2020, zv. 139, č. 139, s. 112854. DOI: <https://doi.org/10.1016/j.eswa.2019.112854>. ISSN 0957-4174. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0957417419305561>.
- [35] ZELL, A. *Simulation of Neural Networks*. 1. vyd. Germany: Addison Wesley, 1994. ISBN 3-89319-554-8.
- [36] ZENG, D., CHEN, S., CHEN, B. a LI, S. Improving Remote Sensing Scene Classification by Integrating Global-Context and Local-Object Features. *Remote Sensing*. 10. vyd. Máj 2018, zv. 10, č. 734, s. 734. DOI: 10.3390/rs10050734.
- [37] ZHANG, S., ZHU, X., LEI, Z., SHI, H., WANG, X. et al. S^3FD : Single Shot Scale-Invariant Face Detector. In: IEEE. *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, s. 192–201. DOI: 10.1109/ICCV.2017.30. ISBN 978-1-5386-1033-6. Dostupné z: <https://ieeexplore.ieee.org/document/8237292>.
- [38] ZHOU, Y., LIU, D. a HUANG, T. Survey of face detection on low-quality images. In: IEEE. *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*. United States: Institute of Electrical and Electronics Engineers Inc., Jún 2018, s. 769–773. Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018. DOI: 10.1109/FG.2018.00121. ISBN 978-1-5386-2336-7. Publisher Copyright: © 2018

IEEE. Copyright: Copyright 2018 Elsevier B.V., All rights reserved.; 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018 ; Conference date: 15-05-2018 Through 19-05-2018. Dostupné z: <https://ieeexplore.ieee.org/document/8373914>.