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

REKONSTRUKCE NEKVALITNÍCH SNÍMKŮ OBLIČEJŮ

FACIAL IMAGE RESTORATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. MATÚŠ BAKO**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. MICHAL HRADIŠ, Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2019/2020

Zadání diplomové práce

Student: **Bako Matúš, Bc.**
Program: Informační technologie Obor: Inteligentní systémy
Název: **Rekonstrukce nekvalitních snímků obličejů**
Facial image restoration
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte základy konvolučních neuronových sítí a generative adversarial networks.
2. Vytvořte si přehled o současných metodách pro restauraci obrazu pomocí neuronových sítí.
3. Vyberte konkrétní metodu aplikovatelnou na snímky obličejů.
4. Obstarejte si databázi vhodnou pro experimenty.
5. Implementujte navrženou metodu a proveďte experimenty nad datovou sadou se zaměřením na odhad vlivu úprav na různé úlohy rozpoznávání obličeje.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Ian Goodfellow: NIPS 2016 Tutorial: Generative Adversarial Networks. NIPS, 2016.
- Goodfellow et al.: Generative Adversarial Networks, arXiv:1406.2661, 2014.
- Karras et al.: Progressive Growing of GANs for Improved Quality, Stability, and Variation, ICLR, 2018.

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

Vedoucí práce: **Hradiš Michal, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 3. června 2020
Datum schválení: 6. listopadu 2019

Abstrakt

V tejto diplomovej práci sa venujem superrezolúcii obrázkov tvárí pomocou konvolučných neurónových sietí so zameraním na zachovanie identity. Ako riešenie problému navrhujem metódu pozostávajúcu z architektúry DPNet a spôsobu trénovania, ktorá vychádza z moderných metód superrezolúcie pomocou neurónových sietí. Model architektúry DPNet je trénovaný na dátovej sade Flickr-Faces-HQ, kde pri štvornásobnom zväčšení dosahuje hodnotu SSIM 0.856, pričom najlepšia z vybraných moderných architektúr, nazývaná Residual channel attention network, dosahuje po natrénovaní hodnotu 0.858. Pri trénovaní modelov pomocou adversariálnej chyby sa v obrázkoch objavovali rôzne artefakty, pričom som experimentoval s viacerými metódami pre ich odstránenie, čo zatiaľ neviedlo k zlepšeniu. Pre porovnanie hodnotenia kvality s ľudským vnímaním som vyhodnotil dotazník, kde sú obrázky zoradené podľa kvality. Výsledky ukazujú, že navrhnutá architektúra sa kvalitou približuje najnovším metódam.

Abstract

In this thesis, I tackle the problem of facial image super-resolution using convolutional neural networks with focus on preserving identity. I propose a method consisting of DPNet architecture and training algorithm based on state-of-the-art super-resolution solutions. The model of DPNet architecture is trained on Flickr-Faces-HQ dataset, where I achieve SSIM value 0.856 while expanding the image to four times the size. Residual channel attention network, which is one of the best and latest architectures, achieves SSIM value 0.858. While training models using adversarial loss, I encountered problems with artifacts. I experiment with various methods trying to remove appearing artefacts, which weren't successful so far. To compare quality assessment with human perception, I acquired image sequences sorted by percieved quality. Results show, that quality of proposed neural network trained using absolute loss approaches state-of-the-art methods.

Kľúčové slová

superrezolúcia, počítačové videnie, neurónové siete, rekonštrukcia obrázkov

Keywords

super-resolution, computer vision, neural networks, image reconstruction

Citácia

BAKO, Matúš. *Rekonstrukce nekvalitních snímků obličejů.* Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Hradiš, Ph.D.

Rekonstrukce nekvalitních snímků obličejů

Prehlásenie

Prehlasujem, že som túto diplomovú práci vypracoval samostatne pod vedením Ing. Michala Hradiša, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Matúš Bako
3. júna 2020

Poďakovanie

Chcel by som poďakovať Ing. Michalovi Hradišovi, Ph.D. za pomoc, odborné konzultácie a svojej rodine za podporu pri štúdiu.

Obsah

Kapitola 1

Úvod

Superrezolúcia patrí medzi neustále napredujúce oblasti počítačového videnia. Nachádza využitie pri všeobecnej práci s obrazovými dátami, prípadne ako súčasť spracovania špecifických obrazových dát. Medzi takéto uplatnenia patrí zväčšovanie obrázkov tváre pri zachovaní identity [71], prípadne zväčšovanie rôznych biologických snímkov [29] pre ďalšie spracovanie.

Proces superrezolúcie je priradenie, ktoré k obrázku s nízkym rozlíšením priraďuje obrázok s vyšším rozlíšením. Na superrezolúciu sa dá preto pozerať ako na mapovanie 1 ku n, kde pre jeden vstup je viacero riešení a nájsť to najlepšie nie je triviálna úloha. Dôležité nie je iba zaoberať sa samotným zväčšovaním rozlíšenia, ale taktiež sa snažiť odstrániť šum, stopy po kompresii, prípadne rozmazanie.

Medzi prvé riešenia patria slovníkové metódy, ktoré sa snažia vytvoriť mapovanie a obrázok zväčšovať po menších častiach. Štatistické metódy pre superrezolúciu využívajú informácie o gradientoch v obrázku. Hranové metódy pracujú s informáciami ako je hrúbka, šírka hrán a iné. Najnovšie metódy pre riešenie tohto problému používajú rôzne variácie hlbokých neurónových sietí s prípadnou verifikáciou zachovania identity, prípadne Generatívne adversariálne siete.

Cieľom tejto diplomovej práce je navrhnúť metódu pre superrezolúciu obrázkov tvárí, pričom dôraz je kladený na zachovanie identity v obrázku. v rámci tejto diplomovej práce som preto navrhol metódu pre zväčšovanie obrázkov tvárí, ktorá vychádza z aktuálnych metód superrezolúcie obrázkov pomocou konvolučných neurónových sietí. Vybral som ďalších osem architektúr konvolučných neurónových sietí, natrénoval som ich pomocou dvojíc zväčšeného a pôvodného obrázku. Ako chyba pri trénovaní bola použitá absolútna chyba, ktorá počíta rozdiel medzi hodnotami každého pixelu. Navrhnutá metóda sa úspešnosťou približuje najlepšej z vybraných architektúr, *Residual Channel Attention Network* [72].

Taktiež som vykonal experimenty s trénovaním pomocou niekoľných rôznych chýb. *Adversarial loss* používa diskriminátor pre rozlíšenie zväčšených a pôvodných obrázkov. *Feature loss* si zakladá sa hľadaní rozdielov medzi rysmi extrahovanými zo zväčšeného a pôvodného obrázku. *Identity loss* počíta vzdialenosť medzi identitami získanými z obrázkov. Kombináciou týchto chýb sa však nepodarilo dosiahnuť lepšie výsledky ako pri použití samostatnej absolútnej chyby. Vyhodnotenie klasifikácie obrázkov zväčšených pomocou navrhnutej neurónovej siete ukázalo zlepšenie oproti zväčšovaniu bikubickou interpoláciou. Pre zistenie, či hodnotenie kvality zodpovedá ľudskému vnímaniu, som získal od respondentov postupnosti obrázkov zoradené podľa kvality. Výsledky ukázali, že medzi kvalitou obrázkov zväčšených navrhnutou neurónovou sieťou a inými modernými neurónovými sieťami sú iba malé rozdiely.

Kapitola 2

Obrazové dáta a ich kvalita

V oblastiach počítačového videnia sú požadované obrázky alebo videá s dostatočnou kvalitou. Princíp dvoch najčastejších aplikácii je interpretácia človekom alebo automatické strojové spracovanie.

Kvalita obrázku je popísaná jeho rozlíšením. Čím vyššie je rozlíšenie, tým vyššia je kvalita obrázku. Rozlíšenie obrázku má však viacero druhov. Pixelové rozlíšenie hovorí, koľko pixelov daný obrázok obsahuje. Udáva sa počtom pixelov na šírku a na výšku obrázku. Priestorové rozlíšenie popisuje hustotu pixelov udávanú v počte pixelov na jednotku plochy. Spektrálne rozlíšenie popisuje rozlíšiteľnosť farieb na úrovni vlnovej dĺžky. Poznáme taktiež temporálne rozlíšenie a rádiometrické rozlíšenie.

Mimo tohto sa však kvalita dá aj merať pomocou viacerých metrík. Všeobecne platí, že dobrá metrika pozitívne koreluje so subjektívnym vnímaním kvality človeka. Metriky kvality môžu taktiež nájsť chyby, ktoré sa propagujú v zreťazenom spracovaní obrázkov, prípadne porovnávať úspešnosť algoritmov na spracovanie obrazových dát. Metriky pre meranie kvality obrazu sa delia na metriky bez referencie, metriky s redukovanou referenciou a metriky s plnou referenciou. V prvom prípade sa kvalita určuje iba s pomocou degradovaného obrázku. Pri redukovanej referencii sa z originálneho obrázku extrahujú rysy, ktoré sú použité na určenie kvality. V prípade plnej referencie sa na výpočet kvality použije pôvodný obrázok a tiež degradovaný obrázok. V praktických aplikáciách pri práci s degradovaným obrázkom často nemáme k dispozícii jeho nepoškodenú verziu, preto v tomto prípade je jedinou možnosťou použiť metriky bez referencie popísané v nasledujúcej podkapitole.

2.1 Metriky bez referencie

V situácii, kedy nemáme k dispozícii rekonštruovaný obrázok a potrebujeme zistiť kvalitu iba z poškodeného obrázku, používajú sa metriky bez referencie. Tieto metriky pre výpočet používajú veličiny ako entropiu, gradient, prípade štandardnú odchýlku. Skúmaním metrík bez referencie [33] bolo zistené, že degradácia má často väčší vplyv na výsledok ako samotná informácia v obrázku. Zároveň rôzne typy degradácie obrazu môžu rôznym spôsobom ovplyvniť iné metriku. Metrika, ktorá vhodne hodnotila obrázky ovplyvnené istým typom degradácie preto nemusí rovnako dobre fungovať pre iné typy degradácie. Medzi bežne používané metriky bez referencie patria:

- Autokorelačná metóda: Pre výpočet kvality používa rozdiely medzi hodnotami autokorelácie v rôznych vzdialenostiach v horizontálnej aj vertikálnej osi. V prípade, že

obrázok je rozmazaný, hrany sú vyhladené a korelácia medzi susednými pixelmi je preto vysoká.

- Priemerný gradient [67]: Metóda zohľadňuje kontrast a ostrosť obrázku. Používa sa pre zmeranie rozlíšenia obrázku, pričom vyššia hodnota indikuje lepšie rozlíšenie.

- *Blind image quality index* [40]: Dvojkroková metóda založená na zohľadnení štatistických vlastností scény. Po natrénovaní nie je potrebná žiadna informácia o type skreslenia obrázku. Metóda je taktiež modulárna, takže je možné model rozšíriť o ďalšie druhy skreslenia.

- *Blind image quality assessment through anisotropy* [9]: Táto metóda dokáže odhaliť prítomnosť šumu v obrázku. Meria priemernú anizotropiu obrázku počítaním smerovej entropie pixelov. Smerová entropia je získaná zmeraním rozptylu očakávanej Rényiho entropie a normalizovanej Pseudo Wigner-Ville distribúcie obrázku pre dané smery.

- *Blind image integrity notator using discrete cosine transform statistics* [50]: Bayesovský model je použitý na na predikovanie kvality obrázku pomocou štatistických vlastností koeficientov lokálnej diskrétnej kosínusovej transformácie.

- *Blind/referenceless image spatial quality evaluator* [38]: Model využívajúci štatistické vlastnosti scény, ktorý je nezávislý na type skreslenia a pracujúci na priestorovej doméne. Pre výpočet kvality sa používajú štatistické vlastnosti lokálne normalizovaných koeficientov osvetlenia, pomocou ktorých sa vyhodnocuje možná strata „prirodzenosti“ obrázku kvôli prítomnosti skreslenia.

- *Edge intensity*: Na obrázok je aplikovaný Sobelov filter a následne je vypočítaný gradient.

- Laplaceova derivácia: Metriky derivácii prvého a druhého rádu sa chovajú ako vysokopriepustný filter vo frekvenčnej doméne. S rastúcou hodnotou metriky rastie ostrosť obrázku.

- Miera entropie: Zisťuje sa informačná hodnota v obrázku. Ak pravdepodobnosť výskytu hodnoty pixelu je malá, entropia je vysoká a rovnako to platí naopak.

- Priemer: Priemerovaním všetkých hodnôt obrázku je získaná priemerná hodnota jasu. Pre rovnaký obsah obrázku hodnota priemeru rastie s rastúcim jasom obrázku.

- *Quality aware clustering* [64]: Skreslené obrázky sú rozdelené na dlaždice a kvalita každej dlaždice je určená *Percentile pooling* metódou. Následne sú určené centroidy úrovní kvality a tie sú používané v slovníku pre vyhodnotenie kvality pre dlaždicu.

- Štandardná odchýlka: Hodnota je vypočítaná odmocnením rozptylu v obrázku. Štandardná odchýlka vyjadruje kontrast obrazu, pričom kontrast rastie s hodnotou odchýlky.

- Vychýlenie: Ako štatistická veličina vychýlenie popisuje smer a mieru, akou sa dáta odlišujú od distribúcie. Pre normálnu distribúciu vysoké vychýlenie indikuje nesymetrické dáta. V tomto prípade ale dáta obsahujú viac informácie.

Edge intensity metrika je odporúčaná pre obrázky vystavené rozmazaniu priemerným filtrom. *Edge Intensity*, *Blind Image Quality Assessment Through Anisotrophy* a priemer sú odporúčané pre ohodnotenie obrázkov, ktoré sú degradované Gausovským bielym šumom. Laplaceova derivácia a štandardná odchýlka sú vhodné pre vyhodnocovanie kvality obrázkov degradované rozmazaním jedným smerom. *Edge Intensity* je taktiež vhodné použiť pri vyhodnocovaní kvality obrázku, ktorý je degradovaný neznámym spôsobom.

2.2 Metriky s plnou referenciou

Pre určenie kvality obrázku metrikou v plnou referenciou [10] sa najčastejšie používa *Peak signal-tonoise ratio* (PSNR) a *Structural similarity* (SSIM). Ak máme dané dva obrázky x a y, pričom oba majú hodnoty pixelov normalizované do intervalu $<0, 1>$ a veľkosť $M \times N$, hodnotu PSNR vypočítame ako:

$$PSNR(x, y) = 10\ log_{10}(1/MSE(x, y)), \tag{2.1}$$

kde MSE predstavuje strednú kvadratickú chybu (*Mean square error*), vypočítanú nasledovne:

$$MSE(x, y) = \frac{1}{MN} \sum_{i}^{M} \sum_{j}^{N} (x_{ij} - y_{ij})^2. \tag{2.2}$$

Hodnota PSNR sa blíži k nekonečnu, keď sa hodnota MSE blíži k nule. Z toho vyplýva, že čím je menšia chyba, tým je väčšia hodnota PSNR.

Wang *et al.* [63] vytvorili *Structural similarity* (SSIM) metriku, ktorá koreluje s kvalitou vnímania ľudského zraku. Namiesto použitia tradičnej metódy sčítania chyby, SSIM modeluje skreslenie obrázku ako kombináciu straty korelácie, skreslenia iluminácie a skreslenia kontrastu. SSIM je definovaná ako

$$SSIM(x, y) = l(x, y)\ .\ c(x, y)\ .\ s(x, y), \tag{2.3}$$

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \tag{2.4}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_1}{\sigma_x^2 + \sigma_y^2 + C_1}, \tag{2.5}$$

$$l(x, y) = \frac{2\sigma_{xy} + C_3}{\sigma_x + \sigma_y + C_3}. \tag{2.6}$$

Prvá časť výrazu je funkcia porovnania jasu, ktorá meria podobnosť medzi strednou hodnotou jasu obrázkov. Maximálna hodnota tejto funkcie je 1, a to práve vtedy, keď $\mu_x = \mu_y$. Druhá funkcia meria podobnosť kontrastov obrázkov. Kontrast je vyjadrený štandardnou odchýlkou σ_x a σ_y. Rovnako ako pri prvej funkcii je maximálna hodnota rovná jednej, a to práve vtedy, keď $\sigma_x = \sigma_y$. Tretia funkcia sa zameriava na štruktúry v obrázkoch tým, že porovnáva korelačný koeficient medzi obrázkami. Konštanty $C_{n|n\ \in\{1,2,3\}}$ sú v zlomkoch kvôli tomu, aby v menovateli zlomku nikdy nebol rovný nule.

Neexistuje jednotné formálne pravidlo, podľa ktorého by bolo možné určiť, či je v danom momente lepšie použiť na vyhodnotenie kvality PSNR ale SSIM. Existujú štúdie [62], ktoré ukázali, že PSNR a teda aj MSE nedosahujú vhodné výsledky pri hodnotení štrukturálneho obsahu, pretože rôzne typy degradácii môžu byť ohodnotené rovnakým PSNR. Iné štúdie [2] zase ukazujú, že MSE a PSNR si vedú dobre pri určovaní kvality zašumených obrázkov.

Obr. 2.1: Vplyv rastúceho šumu na hodnotou PSNR a SSIM.

2.3 Degradácia kvality obrázkov

Pri zväčšovaní obrázkov je nutné snažiť sa odstrániť šum, kompresné artefakty, rozmazanie, prípadne ďalšie možné spôsoby degradácie. Nesprávne riešenie týchto problémov môže vytvoriť artefakty, ktoré ešte viac poškodia pôvodnú fotografiu.

Rozmazanie obrázku znamená, že znižujeme jeho kontrast a ostrosť. Jeden z hlavných problémov pri rekonštrukcii obrazu je, že väčšinou máme málo informácií o samotnom type rozmazania. To môže byť Gausovo rozmazanie, rozmazanie pohybom, priemerné rozmazanie. Samotné rozmazanie môže byť uniformné alebo neuniformné. V prípade neuniformného rozmazania sa jeho intenzita a smer môže meniť v rôznych miestach obrázku, čo robí rekonštrukciu tým zložitejšou. Väčšina techník je založená na odhadnutí jadra rozmazania dekonvolúciou rozmazaného obrázku. Proces odhadu jadra je však ovplyvnený šumom, takže výsledkom je zašumený zaostrený obrázok. Snahe o rekonštrukciu obrázkov bola už venovaná nemalá pozornosť. Liu *et al.* [48] navrhli model na detekciu rozmazaných regiónov a rozpoznanie typu rozmazania. Zo vstupného obrázku detekuje vlastnosti ako lokálne silové spektrum, histogram gradientov, saturáciu a lokálnu autokorelačnú kongruenciu. Prvé tri z nich používa na modelovanie charakteristík rozmazania a autokorelačnú kongruenciu na rozlíšenie pohybového rozmazania od rozostrenia. Pan *et al.* [45] prezentovali slepú metódu odstraňovania šumu pomocou *Lokálneho maximálneho gradientu*. Intuícia za touto metódou je, že gradient lokálnej dlaždice sa po rozmazaní znižuje, pričom túto informáciu zakomponovali do funkcie energie navrhnutého optimalizačného algoritmu. Nah *et al.* [41] navrhli konvolučnú neurónovú sieť, ktorá rekonštruuje rozmazané obrázky s neuniformným

Obr. 2.2: Príklady rôznych spôsobov degradácie obrázku.

rozmazaním v dynamickom prostredí. Z toho vyplýva, že rozmazanie nie je spôsobené iba pohybom kamery, ale aj pohybom objektov v scéne. Navrhnutá neurónová sieť iteratívne prevádza extrakciu rysov, odhad konvolučného jadra a následne odhad rekonštruovaného obrazu, pričom začína s hrubým odhadom, ktorý postupne vylepšuje. Pre ďalšie vylepšenie výsledkov je použitá *Adversarial loss*. Ramakrishnan *et al.* [46] navrhli generatívnu adversariálnu sieť, ktorej architektúra vychádza z reziduálnych sietí [19], avšak prináša niekoľko zmien. Vďaka tomu, že model nepredikuje konvolučné jadro rozmazania samostatne, rekonštrukcia je menej časovo náročná. Kupyn *et al.* [30] vytvorili architektúru generatívnej adversariálnej siete *DeblurGAN*. Kvalita tohto modelu bola vyhodnotená pomocou detekcie objektov na rekonštruovaných obrázkoch. Autori synteticky vytvárali rozmazané obrázky pomocou generovania trajektórii pre konvolučné jadrá na rozmazanie. Zhang *et al.* [70] navrhli niekoľko konvolučných neurónových sietí a integroval ich do optimalizačnej metódy pre redukovanie šumu v obraze. Lefkimmiatis *et al.* [32] natrénovali dve varianty konvolučnej neurónovej siete pre odstraňovanie šumu vo farebných obrázkoch a v obrázkoch s odtieňami šedej. Obe siete sú robustné a zvládnu rekonštruovať obrázky s rôznymi úrovňami zašumenia. Tang *et al.* [57] vytvorili konvolučnú neurónovú sieť na detekciu rozmazania rozostrením. Navrhnutá architektúra si zakladá na iteratívnom kombinovaní rysov a postupnom vylepšovaní mapy rozmazania pomocou špeciálnej vrstvy.

Kapitola 3

Superrezolúcia

Cieľom superrezolúcie vytvoriť z obrázku s nižším rozlíšením I^{LR} obrázok s vyšším rozlíšením I^{SR}. Existujú metódy, ktoré však referenčný obrázok nemusia mať k dispozícii. Súčasťou procesu môže byť taktiež odstraňovanie rozmazania, šumu, kompresných artefaktov a podobných defektov. Nasledujúce podkapitoly popisujú chyby používané na trénované neurónových sietí a niektoré z existujúcich architektúr konvolučných neurónových sietí.

3.1 Delenie metód

Metódy superrezolúcie sa delia Single image super-resolution (SISR), alebo Multiple image super-resolution (MISR), teda podľa toho, či zväčšujeme jeden obrázok, alebo z viacerých obrázkov vytvárame jeden zväčšený. SISR techniky sú teda viac limitované, pretože možných riešení ako zväčšiť daný obrázok je viacero a vybrať to najlepšie nie je triviálna úloha. Metódy tohto typu využívajú závislosti vrámci jedného obrázku [11], alebo sa snažia napodobniť mapovaciu funkciu pomocou dvojíc zmenšeného a zväčšeného obrázku.

SISR metódy sa dajú rozdeliť do dvoch kategórii. Kým metódy špecifické vzhľadom na určitú doménu sa zameriavajú na určitú triedu obrázkov napr. tváre [65] alebo text [69], generické SISR algoritmy sú vytvárané pre všetky druhy obrázkov. Metódy špecifické na danú doménu sú väčšinou úspešnejšie, pretože využívajú informácie, ktoré sú spoločné v danej skupine obrázkov.

Generické SISR algoritmy sa pri generovaní I^{SR} obrázkov z jedného vstupného obrázku zameriavajú na isté spoločné vlastnosti. Podľa toho, ako a na aké vlastnosti sa zameriavajú, ich môžeme rozdeliť do viacerých kategórii. **Predikčné modely** generujú I^{SR} obrázky pomocou preddefinovanej matematickej formule. Jednou z hlavných výhod tohto prístupu je, že nie je nutné trénovať žiadny model. Metódy založené na interpolácii vytvárajú I^{SR} obrázky váženým priemerovaním susedných pixelov. Algoritmy tohto typu vytvárajú oblasti s jemnými prechodmi, čo je výhodné pri vyhladzovaní šumu, ale problémové vo vysokofrekvenčných regiónoch s hranami. Pretože hrany hrajú veľkú rolu vo vizuálnom vnímaní, niektoré z existujúcich metód sú založené na skúmaní **vlastností hrán**. Medzi skúmané vlastnosti patrí napríklad šírka a hĺbka, prípadne parameter gradientného profilu [54]. Keďže model sa učí primárne z hrán, vytvorené I^{SR} obrázky majú kvalitné ostré hrany s málo artefaktami. Hranové metódy však nedosahujú dostatočnú úspešnosť pri rekonštrukcii iných vysoko-frekvenčných štruktúr, ako sú napr. textúry. **Štatistické metódy** využívajú vlastnosti ako distribúcia hustoty gradientov [51], prípadne rozptyl gradientov [21]. Metódy **založené na dlaždiciach** sa snažia vyrezať dlaždice vhodnej veľkosti z LR obrázku pre

naučenie mapovacej funkcie. Dlaždice môžu byť vygenerované z externých dátových sád, prípadne zo vstupného obrázku. Na trénovanie mapovacej funkcie bolo navrhnutých viacerých metód, medzi ktoré patrí vážený priemer [55], regresia jadrom [21], regresia podporným vektorom [42], regresia Gaussovským procesom [17], prípadne reprezentácia riedkym slovníkom [66]. Na vysporiadanie sa s prekrývajúcimi sa časťami dlaždíc bolo navrhnuté váhové priemerovanie [11], Markovské náhodné polia [7] a podmienené náhodné polia [60]. Veľkým pokrokom v oblasti superrezolúcie bolo použitie neurónových sietí, čo popisujú nasledujúce podkapitoly.

V prípade MISR metód sa na vytvorenie I^{SR} obrázku používajú obrázky jednej scény s minimálnymi výchylkami. Každý obrázok vyjadruje množinu istých obmedzení pre neznámy výsledný obrázok I^{SR}. Metódy na preloženie a zjednotenie obrázkov sú však náročné na výpočet a ich presnosť priamo ovplyvňuje výsledok. Táto diplomová práca sa zaoberá iba technikami pre zväčšovanie jedného obrázku, takže metódy tejto úlohy nie sú popísané.

3.2 Trénovanie neurónových sietí

Neurónové siete sú v poslednej dobe využívané pre riešenie problémov v rôznych odvetviach informatiky. V oblasti počítačového videnia boli úspešne aplikované na detekciu objektov [47], extrakciu štýlu z obrazu [22] a v mnohých iných prípadoch. Neurónové siete sa však používajú aj v iných oblastiach, napríklad v spracovaní zvuku [13], alebo pri predikcii na burze [39]. Medzi pokroky v úspešnosti konvolučných neurónových sietí patrí použitie reziduálnych blokov [19] a taktiež Generatívne adversariálne siete [12].

Natrénované modely neurónových sietí určených na superrezolúciu môžu byť použité ako súčasť komplexnejších modelov určených pre riešenie iného problému. Prakticky je takýto model možné použiť pri akejkoľvek úlohe, kde poskytnuté obrazové dáta nedosahujú dostatočnú kvalitu. Medzi konkrétne prípady patrí napr. rozpoznanie identity ľudí z obrázkov [6, 56].

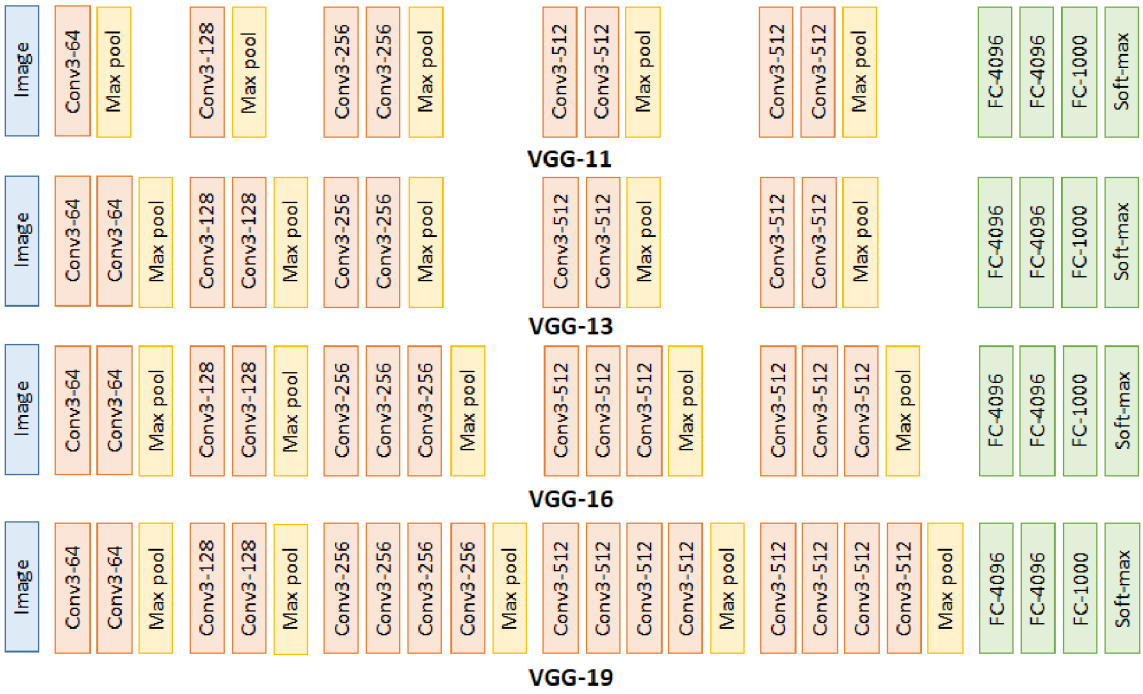
Konvolučné neurónové siete sú pri superrezolúcii trénované na pároch zmenšených fotografii I^{LR} a originálnych fotografii I^{HR} zvolenej veľkosti. Rôzne navrhnuté architektúry ale používajú rôzne chybové funkcie, prípadne váhovanú kombináciu viacerých chybových funkcii. Základným spôsobom pre výpočet chyby, ktorý používa väčšina popisovaných neurónových sietí [5, 16, 35, 26, 52, 34], je tzv. *Pixel loss* chyba. Táto chyba spočíva v počítaní rozdielu medzi hodnotami pixelov v originálnom obrázku a obrázku zväčšenom pomocou neurónovej siete. Na to sa používa funkcia MSE popisovaná v podkapitole 2.2, prípadne stredná absolútna chyba (MAE):

$$Loss_{MAE}(I^{SR}, I^{HR}) = \sum_{xy} \frac{|I^{HR}_{xy} - I^{SR}_{xy}|}{n}. \tag{3.1}$$

Tieto chyby môžu byť taktiež pomenované ako L2 a L1 vzdialenosť.

Pri použití MSE alebo MAE chyby sa neurónová sieť dokáže natrénovať iba do úrovne, kde zväčšeným obrázkom chýbajú detaily a ostré hrany ako môžeme vidieť na obrázku 3.2. Pre vylepšenie kvality rekonštrukcie je však možné použiť pri trénovaní ďalšie chybové funkcie.

Jednou z ďalších chýb je tzv. *Feature loss* [31, 61], ktorá počíta rozdiel medzi aktiváciami neurónovej siete určenej na extrakciu rysov z obrazových dát. Intuícia za použitím tejto chyby je v tom, že v prípade, že zväčšený obrázok je podobný tomu pôvodnému,



Obr. 3.1: Rôzne veľkosti neurónovej siete VGG. Neurónová sieť bola pôvodne vytvorená ako klasifikátor, ale ak vynecháme plneprepojené vrstvy a *Softmax* vrstvu, je možné ju použiť na extrakciu rysov z obrázku. (zdroj [53])

extrahované rysy by mali taktiež byť podobné. Na extrakciu rysov je použitá predtrénovaná konvolučná neurónová sieť. Medzi vhodných kandidátov sa radí Resnet [19], prípadne VGG [53] z obrázku 3.1. Neurónová sieť musí byť natrénovaná ako klasifikátor, avšak pre extrakciu rysov nie je použitý výstup poslednej vrstvy, ale výstup niektorej z konvolučných vrstiev. V neurónových sieťach SRGAN [31] a ESRGAN [61], ktoré použili pre extrakciu rysov neurónovú sieť VGG, boli pre výpočet chyby aktivácie použité skôr ako z posledného bloku, čím sa ušetrí čas na výpočet, ale extrahované rysy sú na nižšej úrovni abstrakcie. Pri trénovaní architekúry ESRGAN bol pri výpočte *Feature loss* chyby použitý výstup pred aktivačnou vrstvou. Problémom podľa autorov bolo, že výstupné aktivácie boli riedke, čo znižovalo ich kvalitu a taktiež spôsobovalo nekonzistentný zrekonštruovaný jas v porovnaní s pôvodným obrázkom.

V článku [59] autori použili tzv. *Total variance loss* [37]. Pri trénovaní pomocou *Adversarial loss* sa na vygenerovaných obrázkoch často vytvárajú farebné pravidelné artefakty 6.6. Na meniace farby sa dá pozerať aj ako na vysokofrekvenčnú informáciu, ktorú táto chyba využíva:

$$Loss_{TV}(I^{SR}) = \sum_{h,w} |I^{SR}_{h,w} - I^{SR}_{h,w+1}| + |I^{SR}_{h,w} - I^{SR}_{h+1,w}|. \tag{3.2}$$

Chyba počíta počíta rozdiely medzi susednými pixelmi horizontálne a vertikálne. Keďže očakávame, že susedné hodnoty nebudú príliš odlišné, trénovanie s touto chybou by malo vyhladzovať spomínané artefakty.

Obr. 3.2: Príklad výstupu neurónovej siete natrénovanej pomocou MSE chyby (vľavo) a originálny obrázok (vpravo). Z obrázku je viditeľné, že takto natrénovaná sieť nedokáže dostatočne reprodukovať detaily a výsledný obrázok sa javí ako neostrý.

Pri trénovaní architektúry *Super identity CNN* [71] bola použitá tzv. *Identity loss*. Chyba rozpoznania identity si zakladá na výpočte vzdialenosti identít v metrickom priestore hypergule. Vzorec pre výpočet chyby je:

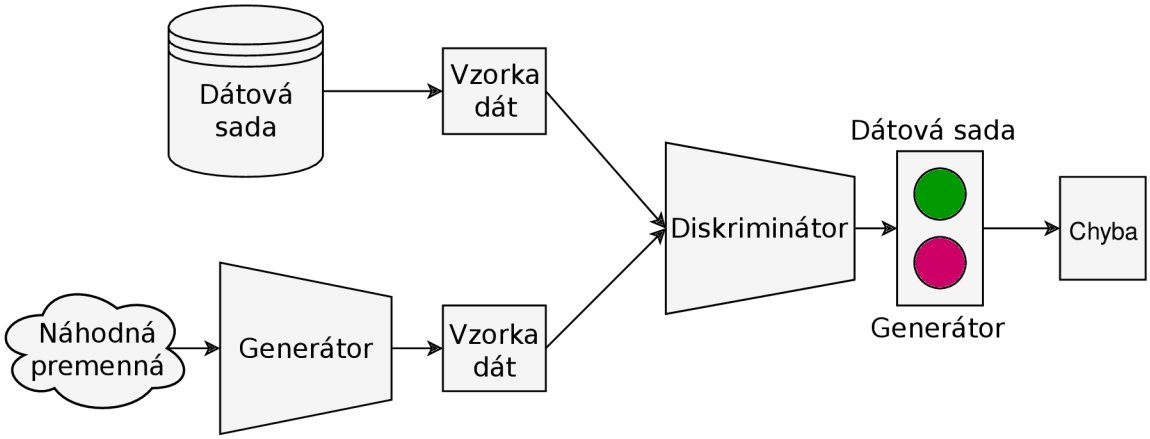
$$Loss^{SI}(I^{SR}, I^{HR}) = \|\widehat{CNN_R}(I^{SR}) - \widehat{CNN_R}(I^{HR})\|_2^2, \tag{3.3}$$

kde $CNN_R(x)$ je výstup neurónovej siete ResNet [19] natrénovanej na rozpoznávanie identít. Znamená to, že výstupom siete je vektor danej dĺžky, ktorý popisuje identitu obrázku na vstupe. Pre výpočet vzdialeností identít sa používa Euklidovská vzdialenosť. Keďže vzdialenosť má byť ale počítaná v metrickom priestore hypergule, vektory sú normalizované ako $\widehat{CNN_R}(x) = \frac{CNN_R(x)}{\|CNN_R(x)\|_2}$.

3.3 Trénovanie generatívnych adversariálnych sietí

Prelomovým míľnikom v oblasti trénovania neurónových sietí sú generatívne adversariálne siete (GAN) [12]. Trieda modelov tohto typu používa pri trénovaní dve architektúry, generátor a diskriminátor. Postup trénovania je možné prirovnať ku zlodejovi, ktorý sa snaží vytvoriť falšované bankovky a detektívovi, ktorý sa snaží rozlíšiť falošné a pravé bankovky. Tým, že sa zlepšuje jeden z nich, núti to zlepšovať sa toho druhého, čím sa obaja navzájom trénujú. Generátor dostáva na vstup šum, ktorý sa snaží namapovať na očakávaný výstup z danej dátovej distribúcie. Úlohou diskriminátoru je zistiť, či jeho vstup je výstupom generátora alebo je súčasťou dátovej sady. Oba modely sú trénované striedavo a snažia sa navzájom zlepšiť.

Pre natrénovanie distribúcie generátoru p_g na dátach x, definujeme vstupnú náhodnú premennú $p_z(z)$. Mapovanie reprezentujeme ako $G(z;\theta_g)$, kde G je funkcia reprezentovaná generátorom s parametrami θ_g. Taktiež definujeme diskriminátor $D(x;\theta_d)$ s analogickým



Obr. 3.3: Ilustrácia spôsobu trénovania generatívnych adversariálnych sietí.

značením. Diskriminátor trénujeme, aby toto číslo maximalizoval a zároveň trénujeme generátor, aby minimalizoval $log(1-D(G(z)))$. Generátor a diskriminátor hrajú hru minimax o dvoch hráčoch s funkciou ohodnotenia $V(G,D)$:

$$\begin{aligned}\min_G \max_D V(D,G) &= \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))] \\ &= \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{x\sim p_g(x)}[log(1-D(x))].\end{aligned} \tag{3.4}$$

V praxi je tento proces implementovaný iteratívnym prístupom. V rámci jednej iterácie diskriminátor a nasledovne generátor vykonajú jeden alebo viac optimalizačných krokov. Takto trénovaný diskriminátor dosahuje optimálnu úspešnosť, kým sa generátor mení dostatočne pomaly. Podľa Goodfellowa *et al.* [12] ale rovnica 3.4 nemusí poskytovať dostatočný gradient pre generátor. Na počiatku trénovania je úspešnosť generátoru malá. Diskriminátor odmieta vzorky dát s veľkou istotou, pretože sú jasne odlišné od dát z trénovacej dátovej sady. Rovnica $log(1-D(G(z)))$ v tomto prípade saturuje. Namiesto toho, aby sme generátor učili minimalizovať $log(1-D(G(z)))$, môžeme ho učiť maximalizovať $log(D(G(z)))$. Táto funkcia poskytuje vyššie hodnoty gradientu v počiatočnej fázy trénovania.

Pri trénovaní GAN-ov môže však nastať niekoľko problémov. Prvý z nich nastáva, keď diskriminátor nedokáže rozlíšiť dáta a poskytuje chybnú odozvu. V druhom prípade diskriminátor perfektne rozlišuje dáta, gradient sa blíži nule a trénovanie sa spomalí, prípadne až zasekne. Ďalší z problémov je tzv. kolaps modelu. V tomto prípade generátor produkuje rovnaký výstup nezávisle na vstupe. Napriek tomu, že tento výstup môže byť veľmi kvalitný a diskriminátor má problém ho rozoznať od reálnych dát, generátor tým prestáva reagovať na svoj vstup a je teda nepoužiteľný na reálne aplikácie. Medzi negatívne javy patrí taktiež explózia gradientov. Tým sa označuje moment, kedy sa kvôli veľkému parametru učenia a veľkej chybe v niektorej z iterácii spropagovali sieťou príliš veľké gradienty. Následne sú váhy parametrov zmenené natoľko, že chyba narastie o niekoľko rádov. V tomto momente je časovo efektívnejšie pokračovať od modelu uloženého pred explóziou, alebo začať trénovanie úplne odznova.

Pri superrezolúcii [31, 61] sa ako generátory používajú konvolučné neurónové siete, ktoré na vstup namiesto šumu dostávajú zmenšený obrázok. Na vstupný obrázok je vhodné aplikovať metódy degradácie, ako už bolo spomenuté v podkapitole 2.3. Diskriminátor v tomto

Algorithm 1 Algoritmus používaný na trénovanie generatívnych adversariálnych sietí. Parameter k predstavuje počet krokov trénovania diskriminátoru na jeden krok trénovania generátoru. (zdroj [12])

for počet iterácii **do**
  for k krokov **do**
    vyber podmnožinu vstupov pre generátor $z^{(1)}, ..., z^{(m)}$
    vyber podmnožinu dát z dátovej sady $x^{(1)}, ..., x^{(m)}$
    trénuj diskriminátor stochastickým gradientným výstupom:

$$\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \Big[log\, D(x^{(i)}) + log\,(1 - D(G(z^{(i)}))) \Big]$$

  end for
  vyber podmnožinu vstupov pre generátor $z^{(1)}, ..., z^{(m)}$
  trénuj generátor stochastickým gradientným zostupom:

$$\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} \Big[log\,(1 - D(G(z^{(i)}))) \Big]$$

end for

prípade určuje, či vstupný obrázok pochádza z dátovej sady, alebo je vytvorený generátorom.

Relativistic average GAN. S ďalším výpočtom chyby a tzv. Relativistickým diskriminátorom prišla metóda *Relativistic average GAN* [23]. Namiesto toho, aby chyba pracovala s informáciou či je obrázok reálny alebo vygenerovaný, porovnáva, ktorý z dvojice obrázkov je viac reálny. Relatívny rozdiel medzi obrázkami sa počíta ako:

$$L_{rel}(x, y) = \sigma(D(x) - \mathbb{E}[D(y)], \tag{3.5}$$

kde σ je *Sigmoid* funkcia, x_r a x_f označujú dáta z dátovej sady a dáta vytvorené generátorom, D označuje diskriminátor a $\mathbb{E}[D(x_f)]$ je priemerný výstup diskriminátoru. Výpočet adversariálnej chyby generátoru a diskriminátoru z algoritmu 1 teda vyzerá nasledovne:

$$L_D^{rel}(x_r, x_f) = -\mathbb{E}[\log(L_{rel}(x_r, x_f))] - \mathbb{E}[\log(1 - L_{rel}(x_f, x_r))], \tag{3.6}$$
$$L_G^{rel}(x_r, x_f) = -\mathbb{E}[\log(1 - L_{rel}(x_r, x_f))] - \mathbb{E}[\log(L_{rel}(x_f, x_r))]. \tag{3.7}$$

Výhodou tohto prístupu je taktiež to, že generátor je trénovaný gradientami z generovaných a aj z reálnych dát, čo zvyšuje stabilitu priebehu trénovania.

Wasserstein GAN. o zvýšenie stability trénovania generatívnych adversariálnych sietí sa taktiež snaží *Wasserstein GAN*[15]. Tento článok prichádza s novým spôsobom výpočtu chyby, ktorým je tzv. Wassersteinova vzdialenosť, nazývaná aj ako *Earth mover's distance*. Wassersteinova vzdialenosť, vyjadruje minimálnu cenu potrebnú na „presun" jednej pravdepodobnostnej distribúcie na druhú. Jednoduchým ekvivalentom môže byť minimálna energia

potrebná na presunutie jednej kopy materiálu na inú. Diskriminátor v tomto prípade nie je priamo trénovaný, aby rozoznával reálne a generované dáta, ale je trénovaný na funkciu, ktorá počíta Wassersteinovu vzdialenosť a spĺňuje podmienku *K-Lipschitz spojitosti* [73]. Autori pre zachovanie tejto podmienky navrhli ohraničenie váh do intervalu $[-0.01, 0.01]$ po každom prepočítaní gradientov. Tento návrh však nie je bezchybný a samotní autori článku sú si vedomí, že orezávanie váh nie je vhodný spôsob zachovania *K-Lipschitz* spojitosti. Problémom je taktiež pomalá konvergencia v prípade príliš veľkého intervalu orezania váh a miznúce gradienty, ak je interval orezania váh príliš malý.

Pre napravenie nedostatkov WGAN-u bolo navrhnuté použitie tzv. *Gradient penalty* [15]. Autori tohto článku narážajú na problémy s optimalizáciou neurónových sietí pomocou pôvodnej WGAN metriky s orezávaním váh, pričom experimentovali aj s inými spôsobmi modifikácie váh. Podľa nich diskriminátoru do istej miery pomáha *Batch normalization*, avšak pri hlbších architektúrach ani toto nie je dostačujúce. Autori taktiež prišli na to, že orezávaním váh sa diskriminátor učí iba jednoduché funkcie. Pre demonštráciu problémov s explodujúcimi a miznúcimi gradientami sa autori pokúsili natrénovať WGAN s rôznou veľkosťou konštanty orezania váh. Pri sledovaní gradientu diskriminátoru zistili, že gradient pri prechode cez vrstvy rastie alebo klesá exponenciálne. Vďaka použitiu *Gradient penalty* však gradient stabilizovali a zamedzili explózii a zániku.

Princípom riešenia je iný spôsob, akým sa dá vynútiť *K-Lipschitz* podmienka. Diferencovateľná funkcia je *1-Lipschitz* práve vtedy, keď má takmer všade normu gradientu rovnú 1. Z tohto dôvodu sa autori článku rozhodli priamo obmedzovať normu gradientu. Nová chybová funkcia vyzerá nasledovne:

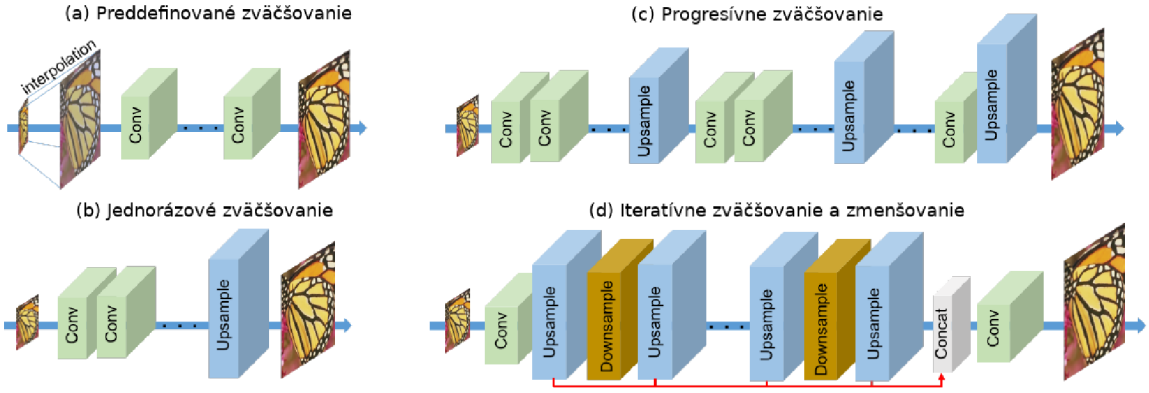
$$Loss_{WGAN\text{-}GP} = \mathbb{E}_{x\sim p_g}[D(x)] - \mathbb{E}_{x\sim p_{data}}[D(x)] + \lambda \mathbb{E}_{x\sim p_{interp}}[(\|\nabla_x D(x)\|_2 - 1)^2]. \quad (3.8)$$

Pravdepodobnostná distribúcia p_{interp} vo vzorci predstavuje body v metrickom priestore z distribúcii p_g a p_{data}, ktoré sú interpolované náhodným parametrom. Pretože vynucovanie podmienky pre normu gradientu všade je prakticky nemožné, vynútenie iba na spojeniach medzi bodmi v metrickom priestore je dostačujúce, čo potvrdzujú aj výsledky z článku[15].

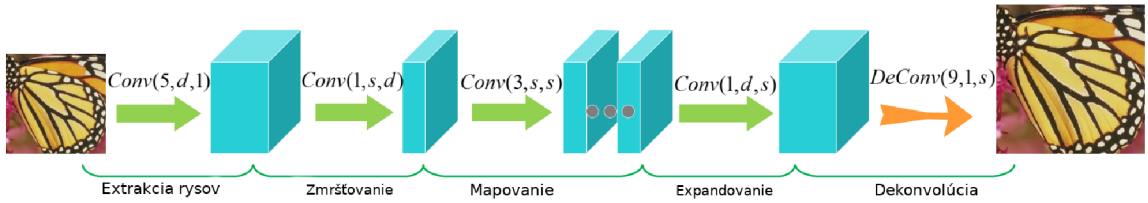
3.4 Architektúry neurónových sietí pre superrezolúciu

V predošlej podkapitole sú popísané chybové funkcie, ktorými sú neurónové siete trénované. Táto kapitola popisuje existujúce architektúry neurónových sietí použitých na superrezolúciu obrázkov. Jednotlivé architektúry sú približne zoradené od najjednoduchších až po komplikovanejšie. Samotné architektúry si prešli vývojom od trojvrstvových neurónových sietí až po hlboké neurónové siete s reziduálnymi blokmi a rôzne iné varianty. Hlavný rozdiel, ilustrovaný v obrázku 3.4, je v princípe samotného zväčšovania.

Najjednoduchšia neurónová sieť na zväčšovanie a rekonštrukciu obrázkov z existujúcich článkov je *Super-Resolution Convolutional Neural Network* (SRCNN) [5]. Architektúra tejto neurónovej siete vychádza z algoritmov, ktoré používajú natrénovaný riedky slovník dlaždíc. V algoritmoch, ktoré pracujú s dlaždicami, sa ako prvý krok dlaždice vyrežú zo vstupného obrázku. Tieto dlaždice sú zakódované a následne rekonštruované slovníkmi výrezov. Podľa tohto bola vytvorená neurónová sieť, ktorá z I^{LR} obrázkov vytvorí zväčšený I^{SR} obrázok. Namiesto slovníkov, ktoré mapujú dlaždice, používa neurónová sieť skryté konvolučné vrstvy. Neurónová sieť najprv extrahuje dlaždice a vhodným spôsobom zmení ich internú reprezentáciu na tzv. mapy rysov. Nasledujúca skrytá vrstva prevádza nelineárne mapovanie na vysoko-dimenzionálne vektory. Posledná skrytá vrstva agreguje I^{SR}



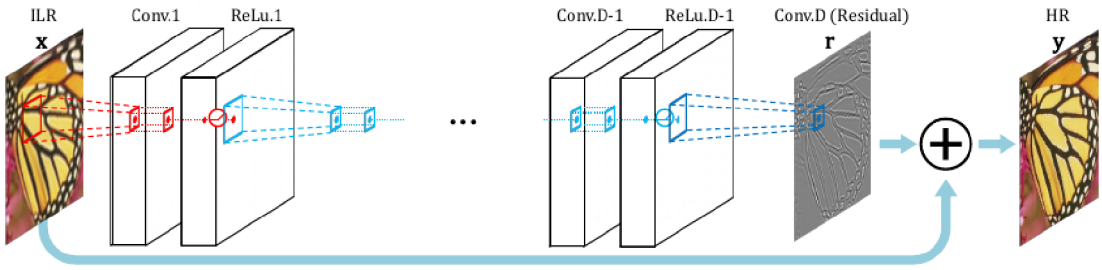
Obr. 3.4: Rozdiely v prístupoch ku zväčšovaniu obrázkov v jednotlivých architektúrach. (zdroj [16])



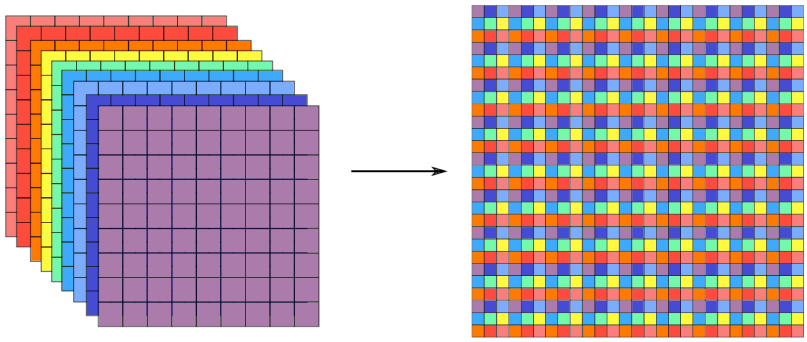
Obr. 3.5: Architektúra neurónovej siete FSRCNN. Parametre vrstiev vyjadrujú veľkosť konvolučného jadra, počet výstupných a vstupných kanálov jednotlivých vrstiev. Platí $s << d$, čím dochádza k zmenšeniu počtu kanálov. (zdroj [52])

dlaždice a vyprodukuje finálny I^{SR} obrázok. Napriek jednoduchosti architektúry, neurónová sieť prekonáva bikubickú interpoláciu a slovníkové metódy. Nevýhodou však je, že neurónová sieť sama o sebe nevykonáva zväčšenie, ale iba rekonštrukciu. Vstupný obrázok musí byť zväčšený na požadovanú veľkosť bikubickou interpoláciou. Toto však spomaľuje samotný výpočet, pričom výpočetné náklady pri faktore zväčšenia n sú n^2-krát väčšie, ako by boli pri použití I^{LR} obrázku. Pri trojnásobnom zväčšení obrázku o veľkosti 240×240 sa dostávame na rýchlosť 1.32 snímkov za sekundu, čo je ale ďaleko schopnosti vykonávať superrezolúciu v reálnom čase.

Architektúra, ktorá nadväzuje na SRCNN, sa nazýva *Fast Super-Resolution Convolutional Neutal Network* (FSRCNN) [6]. Hlavné vylepšenie oproti SRCNN je, že vstupný obrázok nemusí byť interpolovaný na požadovanú veľkosť, ale je zväčšený dekonvolučnou vrstvou [68]. Táto vrstva je umiestnená až na konci architektúry, aby samotné mapovanie prebiehalo na úrovni I^{LR} obrázku kvôli menšej výpočtovej náročnosti. Na dekonvolúciu sa dá pozerať ako na násobenie každého vstupného pixelu filtrom s posunom a následné sčítanie výsledných dlaždíc. Ďalšou zmenou je, že do architektúry sú pridané vrstvy, ktoré zmršťujú následne expandujú počet kanálov, čo taktiež znižuje výpočtovú náročnosť. Namiesto jednej mapovacej vrstvy architektúra obsahuje viacero mapovacích vrstiev s fixnou veľkosťou konvolučného jadra 3×3. Celkovo sa teda architektúra skladá z piatich častí: extrakcia rysov, zmršťovanie kanálov, mapovanie, expandovanie kanálov a dekonvolúcia. Prvé štyri časti sú reprezentované konvolučnými vrstvami. Nelineárne mapovanie je najdôležitejšia časť, pretože určuje úspešnosť zväčšenia a rekonštrukcie. Úspešnosť taktiež ovplyvňuje



Obr. 3.6: Architektúra neurónovej siete VDSR. Z obrázku je viditeľné, že väčšinu neurónovej siete tvoria reziduálne bloky. (zdroj [26])

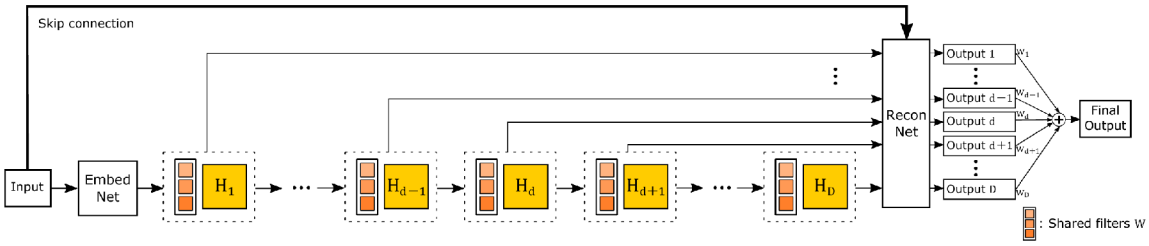


Obr. 3.7: Ilustrácia trojnásobného zväčšenia pomocou *Sub-pixel* konvolučnej vrstvy z architektúry ESPCN. (zdroj [52])

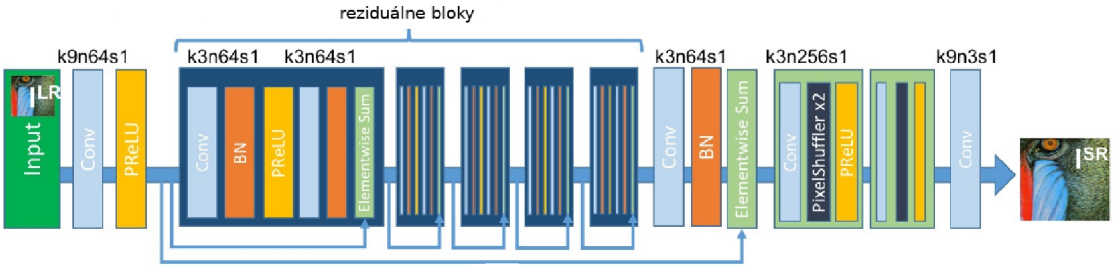
počet výstupných kanálov vrstiev a počet samotných vrstiev v mapovacej časti. Navrhnutá architektúra dosahuje oproti SRCNN [5] 40-násobnú rýchlosť, čo umožňuje pracovať s videom v reálnom čase.

Nasledujúca architektúra, ktorá sa pokúsila využiť reziduálne prepojenie [19], sa nazýva *Very Deep Super-Resolution* (VDSR) [26]. Vďaka tejto zmene konvolučné vrstvy počítajú diferenciálne zmeny nad obrázkom, čo prináša nový pohľad na tento problém. Výsledok je následne pripočítaný k vstupu neurónovej siete, čím vznikne zrekonštruovaný obrázok. Pri trénovaní architektúry je použitý vysoký parameter učenia a orezávanie gradientu, pretože hlboké neurónové siete konvergujú pomaly pri malej hodnote parametru učenia. Výhodou tejto architektúry je, že s jedným modelom je možné použiť viacero zväčšovacích koeficientov, čo znižuje pamäťové nároky na ukladanie modelov. Nevýhodou tejto architektúry však je, že rovnako ako pri architektúre SRCNN [5] je nutné vstupný obrázok najprv interpolovať na požadovanú veľkosť, čo zvyšuje výpočtovú náročnosť.

Ďalšou architektúrou, ktorá stavia na SRCNN, je *Efficient Sub-pixel Convolutional Neural Network* (ESPCN) [52]. Táto neurónová sieť dokáže pracovať s videom v reálnom čase s rozlíšením 1080p pri použití jedinej NVidia GRID K2 grafickej karty. Rovnako ako pri FSRCNN dostáva na vstup zmenšený obrázok, čím sú zmenšené výpočtové nároky. Neurónová sieť pozostáva zo striedajúcich sa konvolučných a aktivačných vrstiev, za ktorými nasleduje zväčšovacia vrstva. Zväčšovanie prebieha pomocou tzv. *Sub-pixel* konvolučnej vrstvy s posunom konvolučného jadra $\frac{1}{r}$, pričom r je koeficient zväčšenia obrázku. V praxi je táto metóda implementovaná konvolúciou, ktorá je nasledovaná tzv. *Periodic shuffling* operáciou. Konvolúcia vytvára z pôvodného množstva kanálov v tenzore r^2-násobný počet



Obr. 3.8: Architektúra neurónovej siete DRCN. v spodnej časti obrázku je zvýraznená opakovaná aplikácia rovnakej konvolučnej vrstvy. (zdroj [27])

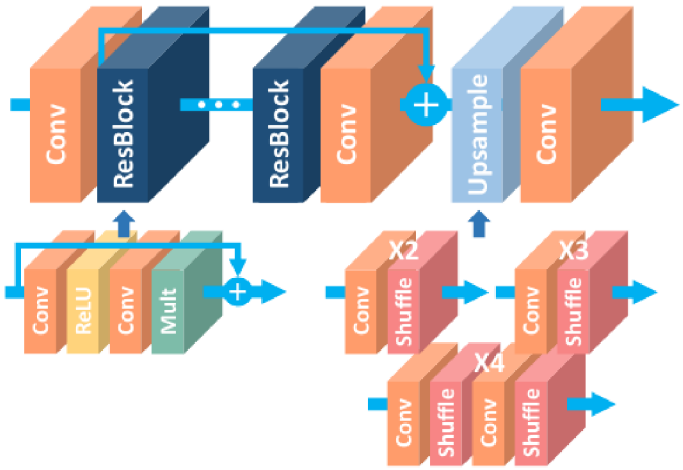


Obr. 3.9: Architektúra neurónovej siete SRResNet. Prvýkrát sa v architektúre vyskytujú reziduálne bloky, pričom zväčšovanie prebieha pomocou *Periodic shuffling* vrstvy. (zdroj [31])

kanálov. *Periodic shuffling* vrstva následne reorganizuje prvky tenzoru tak, že z tenzoru o rozmere $(r^2C \times H \times W)$ vznikne tenzor o rozmere $(C \times rH \times rW)$ ako je naznačené v obrázku 3.7.

Ďalší prístup k superrezolúcii využíva architektúra *Deep Back-Projection Network* (DBPN) [16], ktorá vstupný obrázok iteratívne zväčšuje a zmenšuje, ako je ilustrované v obrázku 3.4. Tento proces je je nazvaný ako (*Back-projection*) a pôvodne bol navrhnutý pre siete s viacerými zmenšenými obrázkami na vstupe. Vďaka tomu je zväčšenie rozdelené na viacero krokov, čo nabáda model ku vylepšeniu výsledkov predošlých zväčšení a o to viac minimalizuje chybu pri rekonštrukcii. Každé zmenšenie podľa autorov reprezentuje rozdielny typ degradácie obrázku. Takýmto spôsobom sa vytvorí viacero zväčšených vstupných obrázkov, ktoré sú skonkatenované a ďalšou rekonštrukciou vznikne výsledný obrázok.

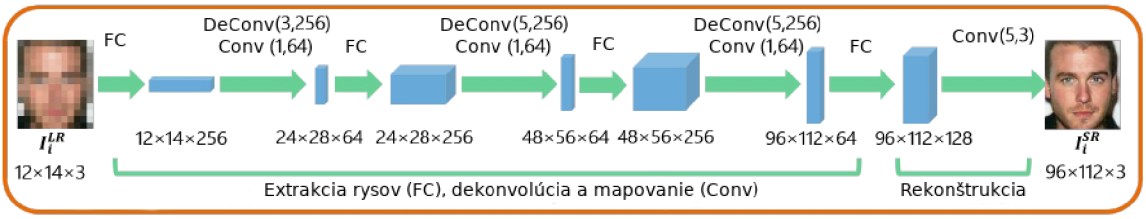
Opakovanú aplikáciu vrstiev na aktivácie počas výpočtu využíva architektúra *Deeply-Recursive Convolutional Network* (DRCN) [27]. Ako prvý krok je vstupný obrázok prevedený na vhodnú reprezentáciu s 256 kanálmi pomocou dvoch konvolučných vrstiev. Na vytvorené aktivácie je 16-krát aplikovaná konvolučná vrstva, pričom každý medzivýsledok je vždy uložený. Následne je na každý medzivýsledok aplikovaná rekonštrukčná vrstva, ktorá aktivácie prevedie na tri kanály. Výsledné aktivácie sú váhovane sčítané, pričom parametre váh sú trénovateľné. Rekurzívne modely však majú problém s explodujúcimi a zanikajúcimi gradientmi. Autori článku sa týmto problémom vyhli použitím všetkých medzivýsledkov rekurzie a Aalikovaním rekonštrukčnej vrstvy už na aktivácie extrahované zo vstupu. Architektúra však obrázky nezväčšuje, ale iba prevádza rekonštrukciu nad už vopred zväčšenými obrázkami.



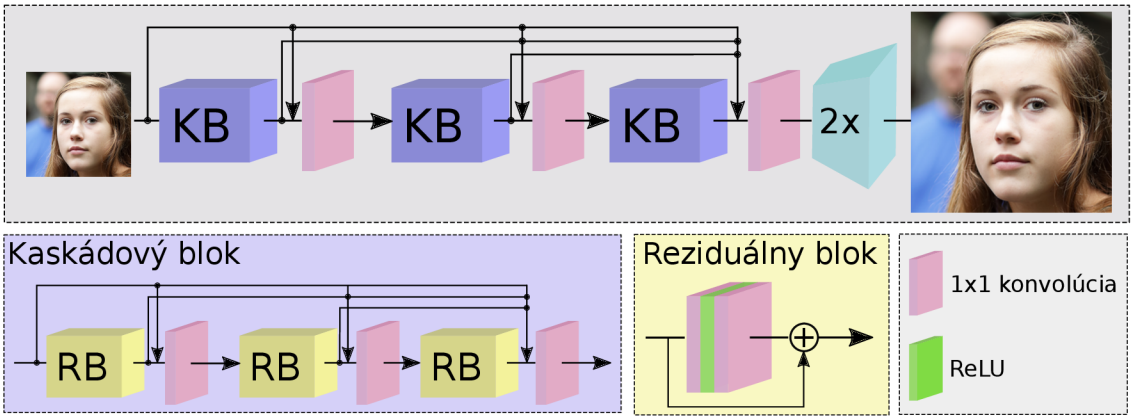
Obr. 3.10: Architektúra neurónovej siete EDSR. V obrázku je taktiež zobrazené zloženie reziduálneho bloku a rôzne možnosti zväčšenia. Blok „Mult“ predstavuje škálovanie konštantou 0.1, čo podľa autorov zlepšilo stabilitu trénovania pri rastúcom počte kanálov. (zdroj [35])

Architektúra, ktorá ako prvá spomedzi neurónových sietí na superrezolúciu používa tzv. reziduálne bloky, sa nazýva *Super-Resolution Residual Network* (SRResNet) [31]. Reziduálny blok obsahuje vrstvy, ktoré sú pri priechode sieťou aplikované na jeho vstupný tenzor. Výstup týchto vrstiev je následne pričítaný ku vstupu reziduálneho bloku. Nad vstupným tenzorom je teda spočítaná diferencia, ktorá je k nemu nakoniec pričítaná. V reziduálnom bloku konvolučných neurónových sietí sa najčastejšie sa používajú dve konvolučné vrstvy doplnené o aktivačnú funkciu, prípadne aj *Batch normalization* vrstvu. Za skupinou reziduálnych blokov sa nachádza reziduálne prepojenie, ktoré k výstupu z celej skupiny reziduálnych blokov pričítava vstup prvého z reziduálnych blokov. Reziduálne prepojenie je teda použité ako na úrovni každého reziduálneho bloku, tak na úrovni celej architektúry. Táto neurónová sieť však používa iba 16 reziduálnych blokov, ako bolo odporúčané v blogu o reziduálnych architektúrach [14]. Autori sa rozhodli použiť aktivačnú funkciu *Parametric rectified linear unit* (PReLU) [18] a taktiež *Batch normalization* vrstvu. Konkrétne zloženie reziduálneho bloku tejto architektúry je ilustrované v obrázku 3.9. Zväčšovanie na výslednú veľkosť prebieha pomocou *Sub-pixel* konvolúcie. Ak je na konci architektúry viacero zväčšovacích blokov paralelne vedľa seba s rôznym koeficientom zväčšenia, je možné zväčšiť vstupný obrázok na rôzne veľkosti pri jednom priechode architektúrou.

Ďalšou z architektúr, ktoré používajú reziduálne bloky [19], je *Enhanced Deep Super-Resolution* (EDSR) [35]. Konkrétne zloženie reziduálneho bloku je ilustrované na obrázku 3.10. Výstupný tenzor reziduálneho bloku je škálovaný konštantou 0.1, čo podľa autorov zlepšilo stabilitu trénovania s rastúcim počtom kanálov. Architektúra obsahuje konkrétne 32 reziduálnych blokov za sebou. Pred samotnými reziduálnymi blokmi a taktiež za nimi je jedna konvolučná vrstva naviac. Oproti architektúre VDSR, ktorá vyžaduje vstupný obrázok zväčšiť na požadovanú veľkosť interpoláciou, zväčšovanie prebieha na konci architektúry pomocou *Sub-pixel* konvolúcie, podobne ako v architektúre ESPCN [52] a SRResNet [31]. Autori pôvodného článku taktiež použili modifikáciu architektúry siete, aby zvládala vstupný obrázok zväčšiť na rôzne veľkosti pri jednom priechode, podobne ako pri SRResNet. To je dosiahnuté použitím viacerých zväčšovacích blokov vedľa seba, ako je naznačené v obrázku 3.10.



Obr. 3.11: Architektúra zväčšovacej neurónovej siete SICNN. Táto neurónová sieť ako jedna z mála z architektúr uvedených v tomto článku používa na zväčšovanie obrázkov dekonvolúciu. (zdroj [71])

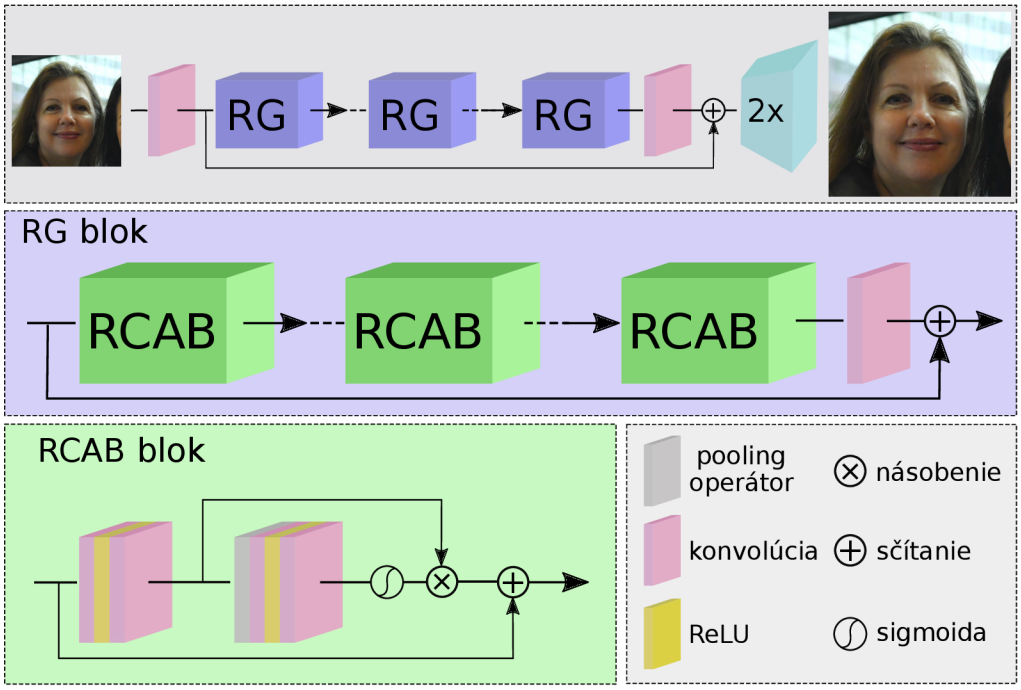


Obr. 3.12: Architektúra neurónovej siete CARN, ktorá ako prvá používa kaskádový priechod sieťou a to na dvoch úrovniach. (zdroj [34])

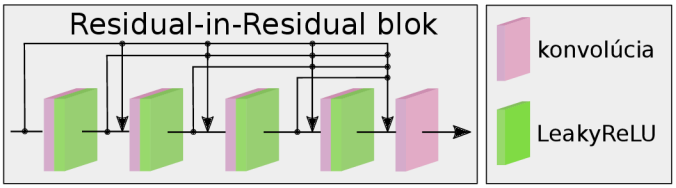
Pretože modely boli pamäťovo náročné, trénovanie bolo vykonávané na niekoľkých Titan X grafických kartách.

Architektúra, ktorá posúva princíp reziduálnych sietí ešte ďalej, sa nazýva *Cascading Residual Network* [34]. Táto neurónová sieť obsahuje kaskádové bloky, ktorých výstup konkatenuje s výstupmi predošlých kaskádových blokov pred vstupom do nasledujúcej vrstvy. Rovnako sú reťazené aj reziduálne bloky umiestnené v kaskádových blokoch. Tento princíp je ilustrovaný v obrázku 3.12. Každý kaskádový blok obsahuje reziduálne bloky striedané s konvolučnou vrstvou. k použitiu kaskádovej schémy viedli dva dôvody. Vďaka kaskádovému prepojeniu môžu informácie prúdiť viacerými cestami a pridaním ďalších konvolučných vrstiev sa model môže sám naučiť zvoliť vhodnú informáciu z viacerých ciest. Podľa autorov však použitie kaskádovej schémy iba na globálnej úrovni alebo v kaskádovom bloku nemá veľký prínos. Z toho dôvodu je kaskádová schéma prítomná ako na globálnej úrovni, tak v každom kaskádovom bloku. Druhá navrhnutá verzia tejto architektúry umožňuje zväčšovať vstupné obrázky na viacero rôznych veľkostí, podobne ako EDSR [35] a SRResNet [31] architektúry. V rámci iného článku s touto architektúrou bolo použité progresívne trénovanie [24], ktorého ideou je začať od jednoduchšieho modelu a postupne pridávať nové vrstvy. Tento prístup uľahčuje trénovanie a zvyšuje rýchlosť konvergencie na rozdiel od priameho prístupu.

Ďalšia architektúra, ktorá používa reziduálne bloky s vlastným vylepšením, sa nazýva *Residual Channel Attention Network* [72]. Navrhnutá architektúra bola vytvorená s dôrazom

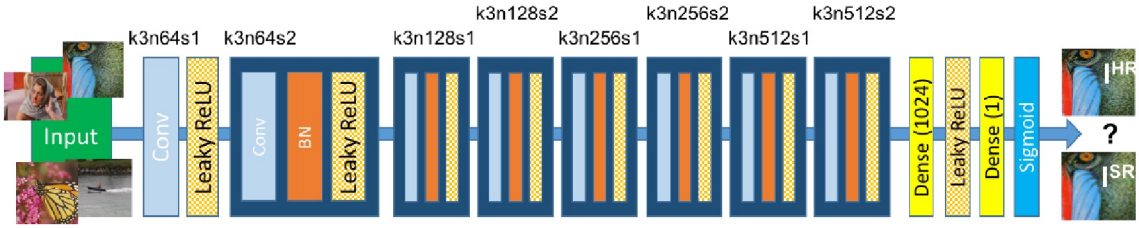


Obr. 3.13: Architektúra neurónovej siete RCAN a popis jednotlivých blokov, z ktorých je zložená. (zdroj [72])

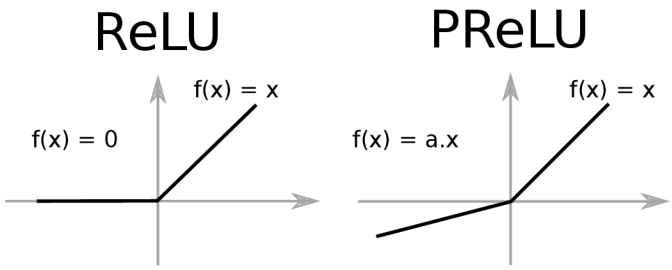


Obr. 3.14: *Residual-in-Residual* blok použitý v architektúre ESRGAN. Podobné prepojenie vrstiev môžeme vidieť v architektúre CARN na obrázku 3.12.

na možnosť jednoduchého trénovania hlbokých sietí. Ako aj iných architektúrach, je na začiatku konvolučná vrstva, ktorá extrahuje základné rysy. Za ňou nasledujú reziduálne bloky s reziduálnym prepojením, ďalšia konvolúcia a nakoniec vrstvy pre zväčšenie obrázku. Na najvyššej úrovni sú tzv. *Residual Group* (RG) bloky s dlhým reziduálnym prepojením, a tie obsahujú *Residual Channel Attention Block* (RCAB) bloky, pričom každý z nich má na konci vlastné reziduálne prepojenie. Tento princíp je vysvetlený na obrázku 3.13. Autori zistili, že iba skladanie RG blokov za seba nedosahuje dostačujúce výsledky. Z tohto dôvodu do architektúry pridali dlhé reziduálne prepojenie, ktoré pričítava extrahované základné rysy ku výstupu RG blokov, čo stabilizovalo trénovanie. RCAB blok ako prvý z doteraz spomenutých reziduálnych blokov obsahuje na najnižšej úrovni adaptívnu pooling operáciu, ktorá spriemeruje každý kanál aktivácii na veľkosť 1×1. Na začiatku bloku sa nachádzajú dve konvolúcie a ReLU funkcia pre prvotnú extrakciu rysov. Úlohou nasledujúcich vrstiev bolo naučiť sa nelineárne mapovanie medzi kanálmi a zdôrazniť viacero rysov v kanáloch. Pre toto boli použité dve konvolučné vrstvy, sigmoida a ReLU funkcia. Výstup sigmoidy funguje ako škálovací parameter pre každý kanál, ktorým sú vynásobené aktivácie pred



Obr. 3.15: Diskriminátor neurónovej siete SRGAN. Úlohou diskriminátoru je určiť, či vstupný obrázok pochádza z dátovej sady, alebo bol vygenerovaný neurónovou sieťou. (zdroj [31])



Obr. 3.16: Rozdiel medzi aktivačnými funkciami *ReLU* a *Parametric ReLU*, ktorá je podobná s funkciou *LeakyReLU*[43]. Rozdielom je, že parameter a je funkcii *LeakyReLU* predaný pri inicializácii, a pri použití *PReLU* je tento parameter trénovateľný.

pooling operáciou. Takto škálované aktivácie sú sčítané so vstupným tenzorom RCAB bloku. Autori v článku používali 64 kanálov, ktoré redukovali na 16 kanálov. Zväčšovanie na konci architektúry zabezpečuje už spomínaná *Sub-pixel* konvolúcia.

Generátor neurónovej siete s názvom *Enhanced Super-Resolution Generative Adversarial Network* (ESRGAN) [61] stavia na architektúre SRResNet[31]. Jednou zo zmien je nahradenie pôvodného reziduálneho bloku tzv. *Residual-in-Residual* blokom z obrázku 3.14. Podľa autorov má tento blok väčšiu kapacitu a je jednoduchšie trénovateľný. Prepojenie vrstiev v tomto bloku vychádza z kaskádového bloku architektúry CARN[34] z obrázku 3.12. Z generátora sú odstránené *Batch normalization* vrstvy, pretože ich použitie mení pôvodnú úlohu generátora z mapovania jedného vstupu na jeden výstup na mapovanie celej dávky. Taktiež platí, že keď sa testovacia a trénovacia dátová sada štatisticky odlišujú, normalizácia dávky môže vytvárať artefakty a znižovať schopnosť generalizácie. Podľa autorov článku sa tieto artefakty objavovali pri rôznych nastaveniach, čo znemožňovalo stabilné trénovanie. Odstránenie týchto vrstiev viedlo k lepším výsledkom, stabilnejšiemu trénovaniu a zmenšeniu výpočtovej zložitosti.

3.5 Architektúra diskriminátoru pre superrezolúciu

Konvolučné neurónové siete na zväčšovanie obrázkov trénované pomocou *Pixel loss* sa snažia o priamu rekonštrukciu daného obrázku, čo je stále problémové pri vysoko-frekvenčnej informácii, pretože tá v I^{LR} obrázku chýba a preto je jej rekonštrukcia obtiažna. Neurónové siete trénované adversariálnou chybou sa snažia si chýbajúcu informáciu vygenerovať. Vý-

počet adversariálnej chyby vyžaduje ďalšiu neurónovú sieť nazývanú diskriminátor. Použitie ďalšieho modelu však zvyšuje pamäťové nároky a čas potrebný na trénovanie architektúry.

Prvá neurónová sieť, určená na superrezolúciu a trénovaná adversariálnou chybou, je *Super-Resolution Generative Adversarial Network* (SRGAN) [31]. Diskriminátor v tejto metóde je konvolučná neurónová sieť s reziduálnymi blokmi, postupne rastúcim počtom kanálov, *LeakyReLU* aktiváciou a plne prepojenými vrstvami na konci. Výhodou *LeakyReLU* je, že pri záporných hodnotách na vstupe nesaturuje, ale prepúšťa záporné gradienty. Poslednou vrstvou je sigmoid funkcia, ktorá z aktivácii vytvára príslušné pravdepodobnosti, ktoré určujú pôvod vstupného obrázku. Rovnakú architektúru pri trénovaní používa aj metóda ESRGAN [61].

Kapitola 4

Metóda superrezolúcie

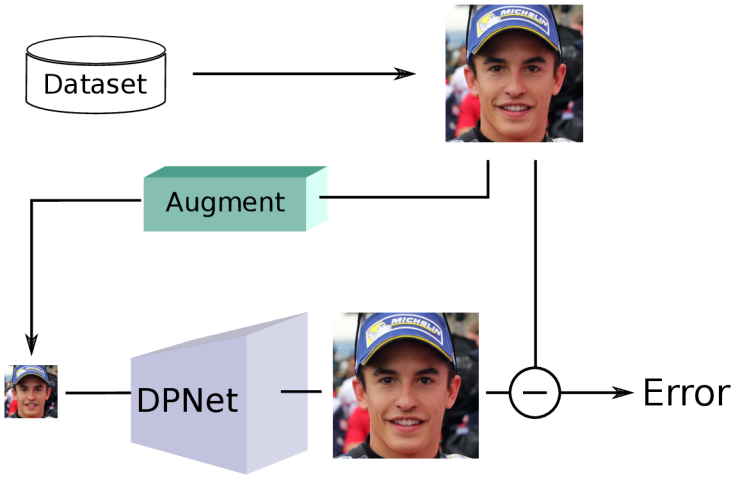
Ako súčasť tejto diplomovej práce som navrhol metódu pre superrezolúciu obrázkov tvárí. Samotná architektúra, nazvaná DPNet, si berie inšpiráciu z viacerých iných neurónových sietí a snaží sa kombinovať ich výhody. Základným stavebným prvkom sú reziduálne prepojenia a *Sub-pixel* konvolúcia. Nasledujúce kapitoly popisujú samotnú metódu, jednotlivé časti architektúry a v poslednej podkapitole 4.4 je popísaná architektúra ako celok.

4.1 Trénovanie modelu

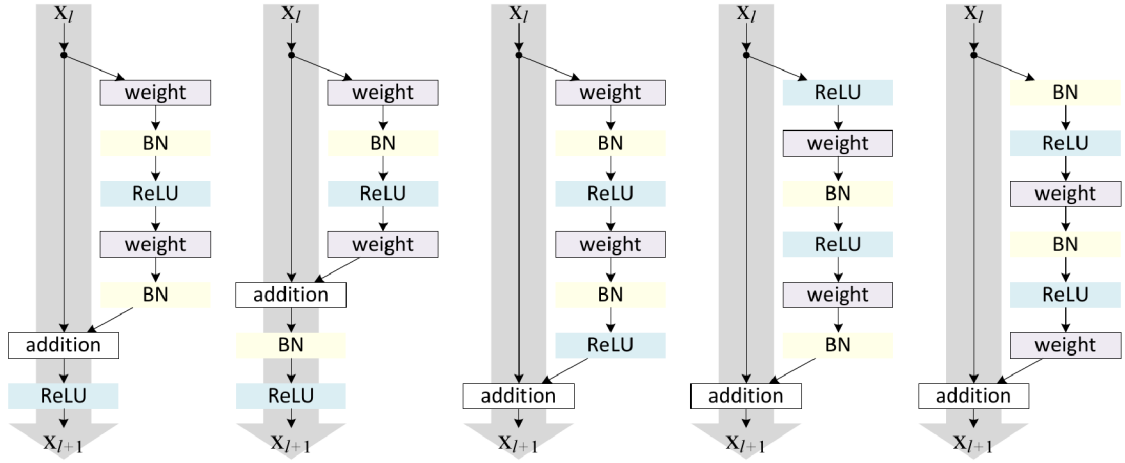
Model neurónovej siete je trénovaný pomocou dvojice zmenšeného a originálneho obrázku. Neurónové siete SRCNN [5] a VDSR [26] dostávajú ako vstup už zväčšený obrázok, čo ale zväčšuje výpočtové nároky. Ostatné metódy [34, 52, 72, 31, 16, 27], vrátane tejto, z toho dôvodu pracujú so zmenšeným obrázkom, ktorý zväčšujú pri priechode vrstvami architektúry. Pri trénovaní sú vstupné dáta augmentované, čím je umelo zväčšená dátová sada. Konkrétne spôsoby augmentácie sú popísané v podkapitole 5.1. Ako základná chyba na trénovanie je použitá *Pixel loss* z podkapitoly 3.2. Pri výbere medzi použitím MSE a MAE sa podľa článku [34] a aj podľa vykonaného experimentu 6.2 ako lepšia voľba ukázala chybová funkcia MAE. Taktiež boli vykonané experimenty s adversariálnou chybou kombinovanou s *Feature loss*, *Gradient penalty* a *Identity loss*. Pri použití týchto chýb sa však na výstupe vyskytovali rôzne artefakty.

4.2 Reziduálne prepojenia a reziduálne bloky

Jedným z komponentov, ktorý je použitý prvý krát vo VDSR [26] architektúre, je tzv. *Skip connection* prepojenie a využíva ho viacero neskôr navrhnutých architektúr [61, 31, 35, 34, 72]. V praxi to znamená, že na daný tenzor je aplikovaných niekoľko vrstiev a výsledok je pripočítaný ku ich vstupu. Výhodou tohto prístupu je, že úlohou použitých vrstiev je iba počítať diferenciu nad vstupným tenzorom, čo zjednodušuje ich úlohu. V návrhu siete na obrázku 4.4 je toto prepojenie viditeľné na dvoch úrovniach. Na vonkajšej úrovni je vstupný obrázok zväčšený na požadovanú veľkosť *Nearest-neighbour* interpoláciou a sčítaný so vstupným obrázkom po aplikovaní vnútornej časti neurónovej siete. *Nearest-neighbour* interpolácia bola zvolená, pretože bilineárna interpolácia by aktivácie vyhladila. Vďaka tomu neurónová sieť nemusí odvodzovať celý vstupný obrázok, ale iba zmenu nad ním. Podobné prepojenie sa v architektúre DPNet z obrázku 4.4 nachádza aj za prvou skupinou reziduálnych vrstiev. V tomto prípade je však tenzor pred pričítaním zmenšený a následne zväčšený



Obr. 4.1: Popis trénovania modelu pri použití *Pixel loss*. Počas experimentov bola ako *Pixel loss* pri trénovaní architektúry DPNet použitá MAE chyba.



Obr. 4.2: Rôzne verzie reziduálneho bloku. Všetky typy reziduálnych blokov pozostávajú z konvolúcie, normalizácie a aktivácie, ale mení sa ich poradie. (zdroj [20])

na pôvodnú veľkosť. To zabezpečí, že reziduálne bloky v tejto časti pracujú s menším tenzorom, pričom každý pixel vďaka tomu obsahuje informáciu o väčšom okolí z pôvodného obrázku. Po následnom zväčšení tenzoru a sčítaní s pôvodným tenzorom sa skombinuje lokálna informácia s globálnou informáciou.

Kľúčovou časťou architektúry sú reziduálne skupiny, ktoré obsahujú reziduálne bloky. Každý reziduálny blok predstavuje reziduálne prepojenie, pričom vnútro prepojenia obsahuje niekoľko vrstiev. Výstup týchto vrstiev je pričítaný ku ich vstupu, ako už bolo popisované pri reziduálnom prepojení. Každý tento blok teda počíta nejakú zmenu nad jeho vstupom. Reziduálne bloky v architektúre ResNet [19], ktorá ich prvý krát použila, sa skladajú z konvolúcie, aktivácie a konvolúcie. V generátore architektúry SRResNet [31] reziduálne bloky obsahujú aj *Batch normalization* vrstvu. Autori architektúry EDSR [35] však experimentálne dokázali, že vynechaním normalizácie dosiahnu lepšie výsledky, pre-

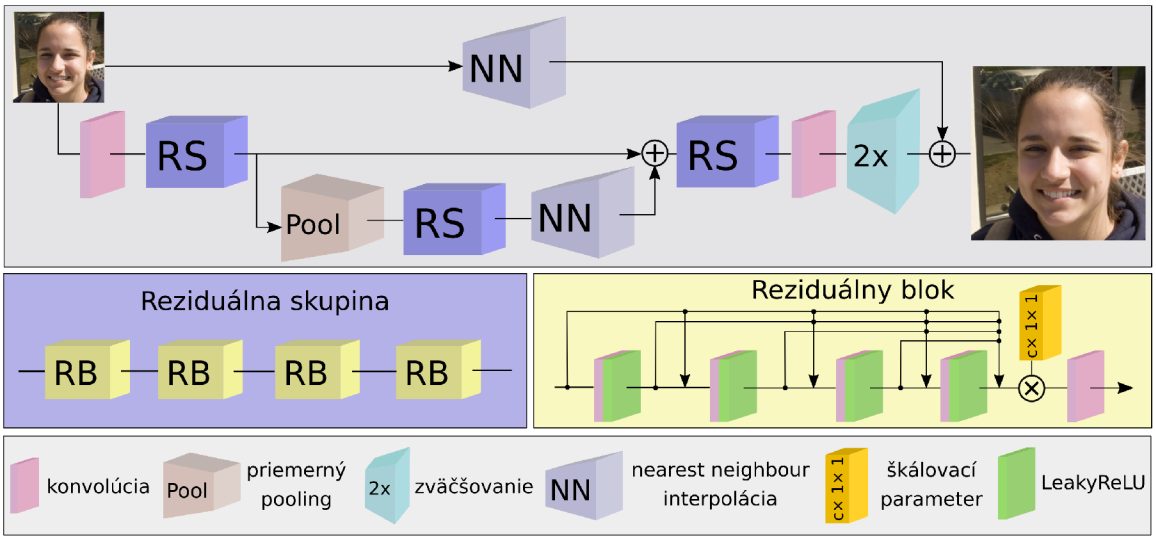
Obr. 4.3: Príklad artefaktov vytvorených pri trénovaní neurónovej siete, ktorá na zväčšovanie obrázku používa *sub-pixel* konvolúciu. Odlišné hodnoty v priľahlých kanáloch spôsobia veľké rozdiely v hodnotách susedných pixelov vo výslednom obrázku, čím vznikajú pravidelné artefakty.

tože normalizáciou sa stráca informácia o rozsahu hodnôt tenzorov. V článku popisujúcom architektúru ESRGAN [61] autori zistili, že ak sa štatistické vlastnosti trénovacej sady odlišujú od testovacej sady, normalizácia môže na výstupe spôsobovať artefakty a limitovať schopnosť generalizácie. Normalizácie má vyššiu tendenciu spôsobovať artefakty pri trénovaní s adversariálnou chybou pri rôznych nastaveniach, čo bráni stabilnému trénovaniu. Taktiež platí, že použitie *Batch normalization* vrstvy mení pôvodnú úlohu generátora z mapovania jedného vzorku dát na mapovanie celej dávky. Namiesto celej dávky je však možné normalizovať kanály každého vzorku samostatne pomocou tzv. *Instance normalization* [58] vrstvy, čo nevytvára závislosť na celej dávke. v architektúre DPNet bol navyše na koniec reziduálneho bloku pridaný trénovateľný parameter pre škálovanie výstupu.

4.3 Zväčšujúca vrstva

Pre zväčšenie aktivácii pred sčítaním so vstupným obrázkom som sa rozhodol použiť tzv. *Sub-pixel* konvolučnú vrstvu, ktorá bola prvýkrát použitá v architektúre ESPCN [52] a následne vo viacerých iných architektúrach [35, 61, 72, 34]. Táto konvolúcia používa parameter posunu menší ako jedna, čím vzniká obrázok, ktorý je väčší ako vstupný obrázok. V praxi je však táto metóda implementovaná pomocou jednej konvolučnej vrstvy a tzv. Periodic shuffling operácie. Konvolučná vrstva z c kanálov vytvorí $c \times r^2$ kanálov, ktoré následne reorganizuje, ako je naznačené v obrázku 3.7. V experimentoch pri trénovaní s adversariálnou chybou však táto vrstva vytvárala artefakty viditeľné na obrázku 4.3.

Podľa autorov článku [1] tento spôsob zväčšovania tenzoru môže vytvárať artefakty pri náhodnej inicializácii. V prípade, že sú hodnoty váh konvolúcie príliš odlišné, následné usporiadanie vytvorí pravidelné artefakty. Ako riešenie tohto problému autori článku navrhli použiť tzv. ICNR inicializáciu. Princípom je vytvoriť a inicializovať tenzor s rovnakým rozmerom, ako majú váhy konvolučnej vrstvy za Periodic shuffling vrstvou, ale namiesto pôvodného počtu kanálov c je vytvorený tenzor s c/r^2 kanálmi, kde r je koeficient zväčšenia. Následne sú váhy r-násobne zväčšené *Nearest-neighbour* interpoláciou a spätnou Periodic

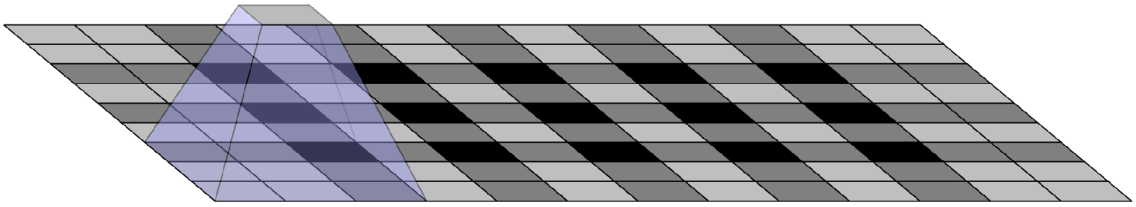


Obr. 4.4: Architektúra navrhnutej neurónovej siete DPNet. Použitý reziduálny blok sa nachádza v neurónovej sieti ESRGAN [61], avšak bol rozšírený o trénovateľný škálovací parameter.

shuffling operáciou sú váhy prevedené na pôvodný nižší rozmer s väčším počtom kanálov. Tým je zaručené, že hodnoty vo váhach, ktoré násobia jeden pixel vstupného tenzoru sú inicializované rovnakou hodnotou. Vďaka tomu sa po inicializácii zamedzí priveľkým rozdielom v $r \times r$ oblastiach vo výslednom obrázku, čím sa predíde tvorbe pravidelných artefaktov.

4.4 Navrhnutá architektúra generátoru

V rámci tejto diplomovej práce som navrhol vlastnú architektúru konvolučnej neurónovej siete, ktorá sa nazýva DPNet. Samotná architektúra, ilustrovaná na obrázku 4.4, je inšpirovaná neurónovými sieťami tvaru presýpacích hodín [49], pričom taktiež používa kaskádové bloky a reziduálne prepojenia. Idea za neurónovými sieťami tvaru presýpacích hodín je, že ich vstup sa postupne zmenšuje a následne zväčšuje po rovnakých krokoch na pôvodnú veľkosť. Spomínané kaskádové bloky, ktoré boli použité v architektúrach CARN [34] a ESRGAN [61], sú za sebou vždy v skupinách po štyroch blokoch. Prvá skupina sa nachádza hneď za úvodnou konvolučnou vrstvou, ktorej úlohou je zväčšiť počet kanálov. Za prvou skupinou je výsledný tenzor následne zmenšený na polovičné rozmery. Konvolvovaním zmenšeného tenzoru zabezpečíme, že každá súradnica v tenzore má informáciu o väčšej ploche z pôvodného obrázku. Väčší obrázok ale obsahuje detailnejšie informácie, čo po sčítaní kombinuje globálnu informáciu s detailami v obrázku. Zmenšený tenzor následne prejde ďalšou skupinou reziduálnych blokov, po ktorej je zväčšený *Nearest neighbour* interpoláciou [3] na pôvodné rozmery a pričítaný ku pôvodnému tenzoru, z ktorého bol zmenšený. Následne výsledný tenzor prejde poslednou skupinou reziduálnych blokov a je zväčšený pomocou *Sub-pixel* konvolúcie, ktorá bola použitá vo viacerých architektúrach [52, 35, 61] v podkapitole 3.4. Takto zväčšený tenzor je pripočítaný ku vstupnému obrázku neurónovej siete, ktorý je zväčšený na požadovanú veľkosť pomocou Nearest neighbour interpolácie. Kvôli problémom s tvorbou artefaktov som navrhol druhú verziu architektúry nazvanú DPNetNN. Rozdiel v tejto verzii je, že namiesto *Sub-pixel* konvolúcie je použitá na zväčšovanie



Obr. 4.5: Pri spätnom šírení gradientov v konvolučnej vrstve s veľkosťou konvolučného jadra 3×3 a posuvom o 2 pozície dochádza k nerovnomernému prekrývaniu a cez niektoré pozície sa propaguje väčší gradient. Použitie konvolúcii s takou konfiguráciou vedie k tvorbe artefaktov na výstupe generátoru. Rovnaký problém nastáva vždy, keď veľkosť konvolučného jadra a posuv sú nesúdeliteľné. (zdroj [44])

Nearest neighbour interpolácia nasledovaná konvolúciou. Podľa článku [44] tento spôsob zväčšovania nespôsobuje artefakty.

4.5 Zmeny v diskriminátore

Pri trénovaní pomocou *Adversarial loss* bol ako počiatočný diskriminátor zvolený už navrhnutý diskriminátor neurónovej siete SRGAN [31] z obrázku 3.15, avšak v jeho architektúre bolo nutné vykonať jednu zmenu. Problémom v architektúre boli konvolučné vrstvy, ktoré mali veľkosť konvolučného jadra 3×3 a posun o dve pozície. Úlohou tých vrstiev bolo zmenšiť výšku a šírku aktivácii na polovičný rozmer. Problémom však je, že konvolučné jadro sa posúva nerovnomerne a niektoré pozície v aktiváciách sú vstupom viacerých konvolúcii. Nerovnomerné pokrytie spôsobí nerovnomerné šírenie gradientov pri ich spätnej propagácii, čo opäť spôsobí artefakty ako na obrázku 4.5. Aby sa zabránilo prekrývaniu pozícií, jedna z možností je použiť konvolúciu s veľkosťou jadra 2×2 a posuv o dve pozície. V takomto prípade je každá pozícia použitá pri výpočte konvolúcii iba raz a gradienty sa pri spätnom priechode šíria rovnomerne.

Počas experimentov boli pri trénovaní s adversariálnou chybou použité obrázky s veľkosťou 256×256. Pre zmenšenie obrázku na veľkosť jedného pixelu však diskriminátor z architektúry SRGAN nestačí. Pre dodatočné zmenšenie boli pred aktivačné vrstvy na konci architektúry pridané dodatočné konvolučné vrstvy s normalizáciou a aktiváciou. Takýmto spôsobom vznikol na výstupe tenzor o rozmere 2×2 s 512 kanálmi. Na konci nasledovali dve plne prepojené vrstvy, medzi ktorými bola aktivácia. Podľa návrhu diskriminátora je na výstupe jediná hodnota.

Kapitola 5

Programová časť

Ako súčasť diplomovej práce bolo nutné vykonať niekoľko experimentov s rôznymi architektúrami neurónových sietí. Preto bolo nutné navrhnúť univerzálny trénovací proces, ktorý zvládne pracovať s rôznymi dátovými sadami, načíta konfiguráciu trénovania a podľa parametrov spustí trénovanie nad vybranou architektúrou. Podrobný popis konfiguračných súborov a uložených modelov sa nachádza v prílohe C. Nasledujúce podkapitoly popisujú objekty zaobalujúce samotné architektúry a dátové sady.

Trénovanie neurónových sietí bolo naprogramované v jazyku Python za pomoci knižnice Pytorch[1]. Táto knižnica je naprogramovaná v jazyku C++ a ponúka rozhranie pre jazyk Python, čím je programovanie značne zrýchlené a zjednodušené. Taktiež sú podporované nástroje CUDA pre trénovanie na grafickej karte a distribuované trénovanie. Knižnica je vyvíjaná na platformy Windows, Linux aj Mac OS.

5.1 Objektová skladba

Pre zaobalenie procesu trénovania a vyhodnocovania sa používa abstraktný `Solver` objekt. Metódy, ktoré objekt obsahuje, implementujú načítanie parametrov z konfiguračného súboru, trénovací algoritmus, ukladanie a načítanie modelu a taktiež vyhodnotenie. Keďže algoritmus trénovania sa líši podľa toho, či architektúra používa diskriminátor, bolo nutné vytvoriť dva rôzne objekty pre implementovanie mierne odlišných algoritmov trénovania. V prípade, že používame diskriminátor, potrebujeme `AbstractGanSolver` objekt, ktorý obsahuje abstraktné metódy `compute_generator_loss()` a `compute_discriminator_loss()` pre výpočet chyby pre generátor a diskriminátor. v opačnom prípade potrebujeme `AbstractCnnSolver` ktorý obsahuje abstraktnú metódu `compute_loss()` pre výpočet chyby modelu. Tieto chyby implementuje každá architektúra vo vlastnom `Solver` objekte, ktorý dedí z príslušného abstraktného objektu. Druhou abstraktnou metódou, ktorú musí `Solver` implementovať je `get_net_instance()`, ktorá vráti inštanciu modelu. Tým je zabezpečené, že každý `Solver` objekt spravuje vytvorenie inštancie objektu a vďaka tomu každý objekt môže predávať vlastnému modelu rôzne argumenty pri jeho vytvorení. Tieto argumenty môžu byť načítané z konfiguračného súboru. Ďalšiu metódu, ktorú môže `Solver` implementovať, je `post_backward()`. V abstraktnom `Solver`-i je táto metóda volaná po spočítaní gradientov a pred aktualizáciou váh, čo umožňuje orezávať gradient, prípade normu gra-

[1] https://pytorch.org/

dientu. Počas implementácie modelov CARN[2], DBPN[3], DRCN[4] a RCAN[5] som sa inšpiroval už existujúcimi implementáciami.

Objekt, ktorý sa stará o načítavanie a predspracovanie dát, dedí z abstraktnej triedy `torch.utils.data.Dataset` a implementuje funkcie `__getitem__(index: int)` a `__len__()`. To znamená, že objekt si určí počet položiek a spôsob, akým k nim bude pristupovať. Metóda pre prístup k položke s daným indexom načíta požadovaný obrázok a vykoná predspracovanie a augmentáciu.

[2] https://github.com/nmhkahn/CARN-pytorch
[3] https://github.com/alterzero/DBPN-Pytorch
[4] https://github.com/togheppi/pytorch-super-resolution-model-collection
[5] https://github.com/yulunzhang/RCAN

Kapitola 6

Experimenty

V tejto kapitole sú popísané experimenty vykonané nad navrhnutou architektúrou a vybranými architektúrami, ktoré sú bližšie popísané v podkapitole 3.4. Pre trénovanie boli použité datasety CelebA a Flickr-Faces-HQ popísané nasledujúcej podkapitole. Ako súčasťou základnej metódy bolo trénovanie pomocou *Pixel loss* chyby. Následne boli vykonané experimenty s *Adversarial loss* a jej kombináciou s ostatnými chybami, popísané v podkapitole 6.5. Experiment v podkapitole 6.6 bol vykonaný pre zistenie, či natrénovaný model naozaj pomáha pri klasifikácii. Taktiež som vykonal experiment, kde som od respondentov získal sekvencie obrázkov zoradené kvality. Podrobný popis sa nachádza v podkapitole 6.7.

6.1 Dátové sady použité pri trénovaní

Pre trénovanie neurónových sietí boli použité dátové sady CelebA [36] a Flickr-Faces-HQ (FFHQ) [25]. Pri vizuálnom porovaní dátových sád je však viditeľné, že obrázky v FFHQ majú vyššiu kvalitu. V dátovej sade CelebA je 202599 obrázkov s 10177 rôznymi identitami. Súčasťou sú taktiež súradnice piatich častí tváre v obrázku a 40 binárnych atribútov. Priložené atribúty popisujú napríklad prítomnosť okuliarov, fúzov, klobúku, a iné vlastnosti. Trénovacia sada obsahuje 162770 obrázkov, ktoré boli vopred zarovnané a následne zmenšené na veľkosť 208×176. Zostávajúce obrázky sú rozdelené približne na polovicu na testovaciu a validačnú dátovú sadu. Konkrétne rozdelenie je súčasťou dátovej sady. Keďže vstupné obrázky sú pred výpočtom zmenšované podľa zadaného koeficientu zväčšovania, je nutné aby výška aj šírka obrázku bola deliteľná dostatočne veľkou mocninou čísla dva.

Dátová sada FFHQ obsahuje 70000 obrázkov uložených vo formáte PNG v rozlíšení 1024×1024. Obrázky obsahujú dostatok variácii v podobe veku, etnicity, a pozadia. Pokryté sú taktiež doplnky ako napríklad okuliare, klobúky, náušnice a podobné predmety. Všetky obrázky boli automatizovane stiahnuté z webovej stránky Flickr[1], následne zarovnané a orezané pomocou knižnice dlib[2]. Zvyšné obrázky, ktoré obsahovali sochy, maľby, prípadne fotografie fotografii boli ručne odstránené.

V rámci predspracovania boli obrázky orezané tak, aby veľkosť bola deliteľná číslom 8 prípadne väčšou mocninou čísla 2, čo umožňovalo zmenšiť obrázok na polovicu minimálne tri krát. V prípade, že by toto nebolo dodržané, rozmery pôvodného a zväčšeného obrázku by nemuseli byť rovnaké. Doterajšie experimenty boli vykonané so zväčšovaním na maximálne štvornásobnú veľkosť. Tým mali neurónové siete priestor, aby svoj vstup mohli zmenšiť

[1] www.flickr.com
[2] http://dlib.net

Obr. 6.1: Príklady obrázkov z dátových sád CelebA (vľavo) a FFHQ (vpravo) použitých na trénovanie neurónových sietí. Z fotografii je viditeľné, že dátová sada FFHQ, pretože tváre zaberajú väčšiu časť obrázku, a samotné fotografie majú vyššie rozlíšenie. Nevýhodou je menší počet fotografii.

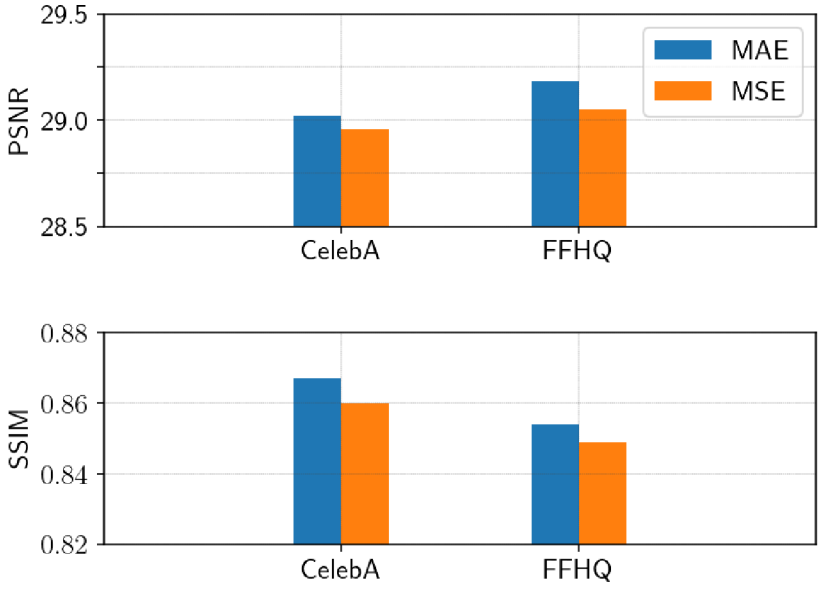
minimálne ešte raz, ako to bolo pri dátovej sade CelebA. Ako ukazujú experimenty, značné rozdiely v kvalite medzi pôvodným a zväčšeným obrázkom sú viditeľné už pri štvornásobnom zväčšení. Medzi použité metódy augmentácie pri trénovaní patrí horizontále pretočenie, JPEG kompresia, šum, rozmazanie rozostrením, rozmazanie pohybom, zmena saturácie, jasu a kontrastu.

6.2 Trénovanie absolútnou a kvadratickou chybou

Ako prvý experiment som vybral porovnanie úspešnosti architektúr pri trénovaní pomocou MAE chyby a MSE chyby. Z naštudovaných článkov z podkapitoly 3.2 bola ako *Pixel loss* pri trénovaní CARN [34], ESRGAN [61] a RCAN [72] architekúr použitá MAE chyba, pričom pri trénovaní ostatných spomenutých architektúr bola použitá MSE chyba. Autori článkov však uvádzajú, že MAE chyba v ich experimentoch dosahovala lepšie výsledky a rýchlejšiu konvergenciu. Toto som si chcel overiť a výsledky experimentu to následne potvrdili. Trénovanie bolo vyhodnotené na architektúre CARN s použitím oboch dátových sád. Z obrázku 6.2 je viditeľné, že napriek tomu, že výsledky nie sú príliš odlišné, zlepšenie pri použití MAE chyby dokazujú vyššie hodnoty PSNR aj SSIM vo všetkých prípadoch. Preto pre trénovanie navrhnutej neurónovej siete DPNet bola teda zvolená MAE chyba.

6.3 Porovnanie architektúr

Trénovanie modelov bez použitia adversariálnej chyby bolo vykonávané po 100000 iterácii s veľkosťou dávky o 16 obrázkoch. Parameter učenia bol na inicializovaný na hodnotu $1 * 10^{-4}$ a následne adaptívne zmenšovaný faktorom 0.2, ak sa nezlepšila trénovacia chyba behom posledných 5000 iterácií. Toto zaručilo dotrénovanie každej architektúry bez nutného manuálneho zásahu. Po viac ako dvoch zmenšeniach parametru trénovania však už nedochádzalo ku zlepšeniu. Pre trénovanie bol použitý optimalizačný algoritmus Adam [28].



Obr. 6.2: Porovnanie kvality rekonštrukcie architektúry CARN [34] trénovanej pomocou MAE a MSE chyby. PSNR a SSIM hodnoty nie sú príliš odlišné, ale trénovanie s MAE chybou dosahuje lepšie výsledky vo všetkých prípadoch.

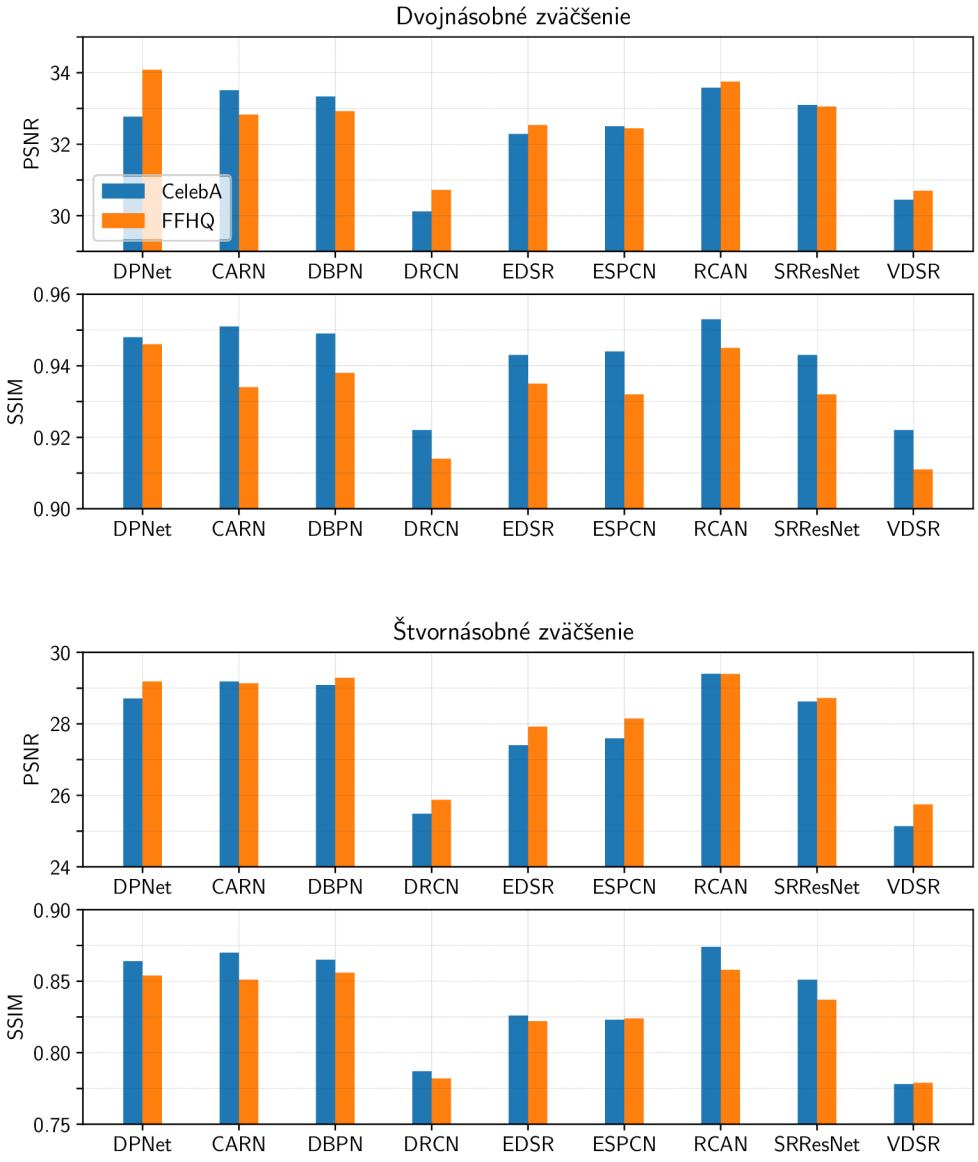
Spočiatku bolo testovaných viacero hodnôt, avšak pri vyšších hodnotách pri trénovaní väčšinou nastala už spomínaná explózia gradientov.

Následne boli vybrané architektúry, ktoré boli natrénované na evaluačnej časti dátových sád CelebA a FFHQ. Všetky tieto architektúry používali iba chybovú funkciu MSE alebo MAE podľa toho, ako boli trénované v príslušných článkoch. Úspešnosť takto natrénovaných architektúr bola následne vyhodnotená pri dvojnásobnom a štvornásobnom zväčšení. Niektoré z architektúr však museli byť zmenšené kvôli pamäťovým nárokom pri trénovaní. Počet kanálov v architektúre EDSR bolo nutné zmenšiť z pôvodných 256 na 160 a veľkosť dávky zo 16 na 8. Rovnako bola veľkosť dávky zmenšená aj pri trénovaní architektúry VDSR. v architektúre DRCN bol počet kanálov zmenšený z 256 na 64 a počet rekurzii zo 16 na 8. v architektúre DBPN boli použité iba dve zväčšovacie vrstvy a jedna zmenšovacia vrstva. Takto modifikované modely už bolo možné trénovať pri limite grafickej pamäte 8GB.

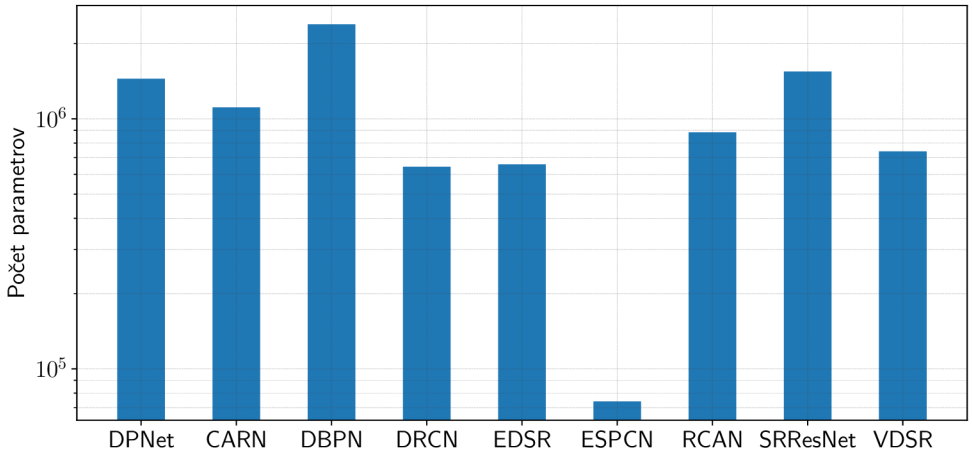
Podľa výsledkov z tabuliek 6.1 medzi najlepšie architektúry patria CARN, DBPN, RCAN za ktorými navrhnutá architektúra DPNet. Poradie však nie je jednoznačné, pretože v niektorých prípadoch dosahujú lepšie výsledky rôzne architektúry. Môžeme však povedať, že kvalita superrezolúcie navrhnutej architektúry sa približuje kvalite moderných neurónových sietí.

6.4 Porovnanie zväčšovacích vrstiev trénovaním adversariálnou chybou

Pri trénovaní architektúry DPNet s adversariálnou chybou som narazil na problémy s tvorbou artefaktov. Podľa odporúčaní z viacerých zdrojov [44, 1] som vytvoril dve verzie ge-



Obr. 6.3: Vyhodnotenie vybraných architektúr neurónových sietí na dátových sadách CelebA (modrá) a FFHQ (oranžová) pri dvojnásobnom (hore) a štvornásobnom (dole) zväčšení. z výsledkov je viditeľné, že navrhnutá neurónová sieť sa kvalitou rekonštrukcie približuje najlepším architektúram CARN a RCAN. Na nameraných hodnotách je taktiež vidieť väčšie rozdiely medzi architektúrami



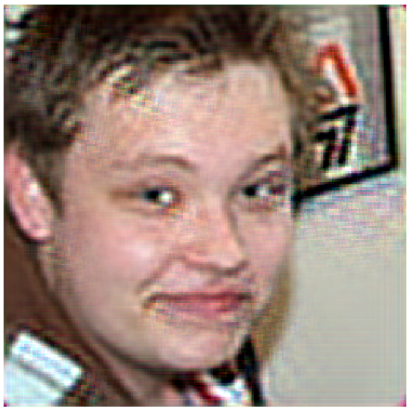
Obr. 6.4: Porovnanie počtu parametrov medzi vyhodnocovanými modelmi. Počet parametrov z veľkej časti koreluje s kvalitou rekonštrukcie, aj keď z vyhodnotenia vyplýva že modely s najlepšími výsledkami nemajú najviac parametrov.

4x zväčšenie	PSNR	SSIM	PSNR	SSIM
	CelebA		FFHQ	
DPNet	28.71	0.864	29.28	0.856
CARN [34]	29.28	0.870	29.18	0.854
DBPN [16]	29.08	0.865	29.13	0.851
DRCN [27]	25.48	0.787	25.87	0.782
EDSR [35]	27.55	0.824	27.55	0.821
ESPCN [52]	27.59	0.823	28.15	0.824
RCAN [72]	**29.39**	**0.874**	**29.39**	**0.858**
SRResNet [31]	28.62	0.851	28.72	0.837
VDSR [26]	25.13	0.778	25.74	0.779

2x zväčšenie	PSNR	SSIM	PSNR	SSIM
	CelebA		FFHQ	
DPNet	32.76	0.948	32.92	0.938
CARN [34]	33.51	0.951	34.08	0.946
DBPN [16]	33.33	0.949	32.83	0.934
DRCN [27]	30.11	0.922	30.72	0.914
EDSR [35]	32.29	0.943	32.54	0.935
ESPCN [52]	32.50	0.944	32.44	0.932
RCAN [72]	**33.58**	**0.953**	**33.75**	**0.945**
SRResNet [31]	33.09	0.943	33.05	0.932
VDSR [26]	30.44	0.922	30.69	0.911

Tabuľka 6.1: Porovnanie úspešností natrénovaných architektúr medzi úspešnosťami z príslušných článkov na dátových sadách CelebA a FFHQ pri dvojnásobnom a štvornásobnom zväčšení.

(a) *Instance normalization*, *Nearest neighbour* zväčšovanie



(b) žiadna normalizácia, *Nearest neighbour* zväčšovanie

(c) bez normalizácie, *Sub-pixel* konvolúcia, ICNR inicializácia

(d) *Instance normalization*, *Sub-pixel* konvolúcia, ICNR inicializácia

(e) bez normalizácie, *Sub-pixel* konvolúcia

(f) *Instance normalization*, *Sub-pixel* konvolúcia

Obr. 6.5: Obrázky vytvorené natrénovanou variáciou architektúry DPNet. Škálovací parameter pre *Adversarial loss* bol zvolený tak, aby nedochádzalo k vytváraniu artefaktov. Varianta (f) dosahuje spomedzi vybraných možností najvyššej kvality, avšak ani táto konfigurácia sa kvalitou nevyrovná trénovaniu iba pomocou *Pixel loss*. z obrázkov je ale viditeľné, že architektúra s *Instance normalization* vytvára kvalitnejšie obrázky.

nerátoru a modifikoval diskriminátor architektúry SRGAN [31]. Podrobný popis tvorby architektúry je v podkapitole 4.4 a modifikácie diskriminátoru sú bližšie popísané v podkapitole 4.5. V článkoch bolo odporúčané nepoužívať *Batch normalization* kvôli závislosti výstupu na celej dávke, napriek tomu že je tak možné dosiahnuť vyššie hodnoty PSNR. Alternatívou k tomu je použitie *Instance normalization*. Namiesto zväčšovania `Sub-pixel` konvolúciou bola ako ďalšia možnosť navrhovaná *Nearest neighbour* interpolácia nasledovaná konvolúciou. Preto som sa rozhodol natrénovať viacero kombinácii použitia normalizácie a spôsobu zväčšovania a sledoval tvorbu artefaktov. Pri trénovaní bola použitá iba MAE chyba a Relativistická adversariálna chyba. Pretože rôzne veľkosti obrázkov by vyžadovali odlišné architektúry diskriminátoru, pri trénovaní s adversariálnou chybou je použitá iba dátová sada FFHQ. Príklady obrázkov zväčšených natrénovanými modelmi sa nachádzajú v obrázkoch 6.5. Výsledky ukázali, že pri použití *Sub-pixel* konvolúcie vzniká najmenej artefaktov bez normalizácie pri použití ICNR inicializácie z podkapitoly 4.3. V obrázkoch sú však stále viditeľné biele horizontálne artefakty v oblastí očí a úst. Pri zväčšovaní pomocou *Nearest neighbour* interpolácie použitie *Instance normalization* zamedzilo vzniku artefaktov, avšak celková kvalita zväčšeného obrázku je stále nedostatočná.

6.5 Kombinovanie viacerých chybových funkcii

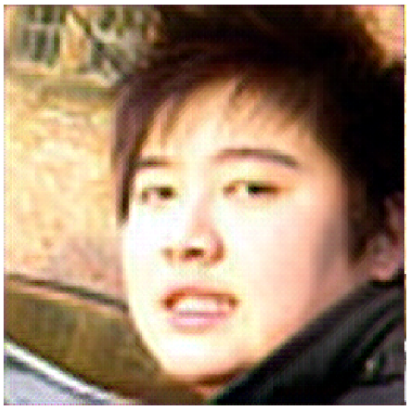
Ďalším z experimentov bolo trénovanie pomocou kombinácie viacerých chýb. Hlavnou súčasťou bola *Pixel loss* a *Adversarial loss*. Ako ďalšie chyby boli pridané *Identity loss*, *Feature loss* a *Gradient penalty*. Ako *Pixel loss* bola pri trénovaní použitá MAE chyba a pre adversariálne trénovanie bola použitá tzv. Relativistická adversariálna chyba. Pri použití *Feature loss* bol výstup braný zo štvrtého bloku extraktoru rysov. Ako extraktor rysov bol použitý model predtrénovanej VGG neurónovej siete z balíčku `torchvision`[3]. Z výsledkov v obrázkoch 6.7 a 6.6 je však viditeľné, že ani jedna z kombinácií nepomohla ku zlepšeniu kvality rekonštrukcie oproti použitiu samotnej MAE chyby.

6.6 Vyhodnotenie zlepšenia klasifikácie

Pre zistenie, či natrénovaná neurónová sieť naozaj pomáha pri rozpoznávaní identít, je nutné vyhodnotiť úspešnosť rozpoznania zväčšených obrázkov. Pre vyhodnotenie bol zvolené verifikácie na porovnávacom teste IARPA Janus Benchmark-A[4] (IJB-A). Táto dátová sada obsahuje 500 identít známych osobností, ktoré sú na 5712 obrázkoch a 2085 videách. Pred vyhodnocovaním sú vopred spočítané vektory popisujúce identitu tvárí na obrázkoch dátovej sady. Test je rozdelený na 10 prierezov (splits). Každý prierez má priradené páry skupín obrázkov, ktoré porovnáva. Prierezy delia obrázky každej identity na tzv. sondovaciu množinu (probe set) a galériu (gallery set). Galéria má reprezentovať obrázky uložené v operačnej databázy, ktoré slúžia na identifikáciu osoby. Sondovacia množina reprezentuje vyhotovené snímky, ktoré používame na vyhľadanie identity v databáze. Každý prierez má určených 10000 porovnaní rôznych identít. Počet porovnaní vrámci jednej identity je určený počtom jej sondovacích množín, pretože pre každú identitu existuje práve jedna galéria. Pri porovnaní rôznych identít platí, že porovnávané subjekty sú rovnakého pohlavia a ich odtieň pokožky sa neodlišuje viac ako o jeden zo šiestich definovaných stupňov. Každá identita má z príslušných obrázkov vytvorené skupiny obrázkov, tzv. templates. Vektory obrázkov

[3] https://pypi.org/project/torchvision
[4] https://www.nist.gov/programs-projects/face-challenges

(a) MAE, *Adversarial loss*



(b) MAE, *Adversarial loss*, *Feature Loss*

(c) MAE, *Adversarial loss*, *Identity Loss*

(d) MAE, *Adversarial loss*, *Gradient penalty*

Obr. 6.6: Výstupy neurónovej siete DPNet pri použití *Sub-pixel* konvolúcie s *Instance normalization* pri trénovaní kombináciou uvedených chýb. Žiadna z odskúšaných kombinácii však neviedla k lepším výsledkom a na výstupe vždy vznikol istý druh artefaktov.

patriace do skupiny sú spriemerované, čím vznikne jeden vektor popisujúci identitu skupiny obrázkov. Súčasťou porovnávacieho testu sú aj páry skupín, ktoré sú navzájom porovnávané. Pre porovnanie vektorov identít sa používa kosínusová podobnosť. Extrakcia identity z obrázkov bola vykonaná pomocou predtrénovaného klasifikátoru SE-ResNet-50[5].

Ako ukázali výsledky z obrázku 6.8, porovnávanie pomocou priemerných vektorov pre skupiny dosahuje vyššiu úspešnosť. Výsledky však taktiež ukazujú, že obrázky, ktoré boli zväčšované bikubickou interpoláciou, lepšie zachovávajú identitu na obrázku. Ku zlepšeniu výsledkov nepomohlo ani dotrénovanie modelu DPNet na dátovej sade IJB-A. Po preskúmaní obrázkov z IJBA dátovej sady som zistil, že obrázky majú nižšiu kvalitu ako dátová sada FFHQ, čo mohlo ovplyvniť schopnosť zväčšovania. Aby sa potvrdilo, že je to príčinou, vykonal som podobný test na dátovej sade VGGFace2.

Dátová sada VGGFace2 [4] obsahuje 3.3 milióna obrázkov zobrazujúcich 9131 identít známych osobností, z ktorých 500 identít je testovacích. Každá identita má v dátovej sade 80 až 843 obrázkov, pričom priemerný počet obrázkov na identitu je 362. Obrázky boli

[5] https://github.com/ox-vgg/vgg_face2

(a) MAE, *Adversarial loss*

(b) MAE, *Adversarial loss*, *Feature loss*

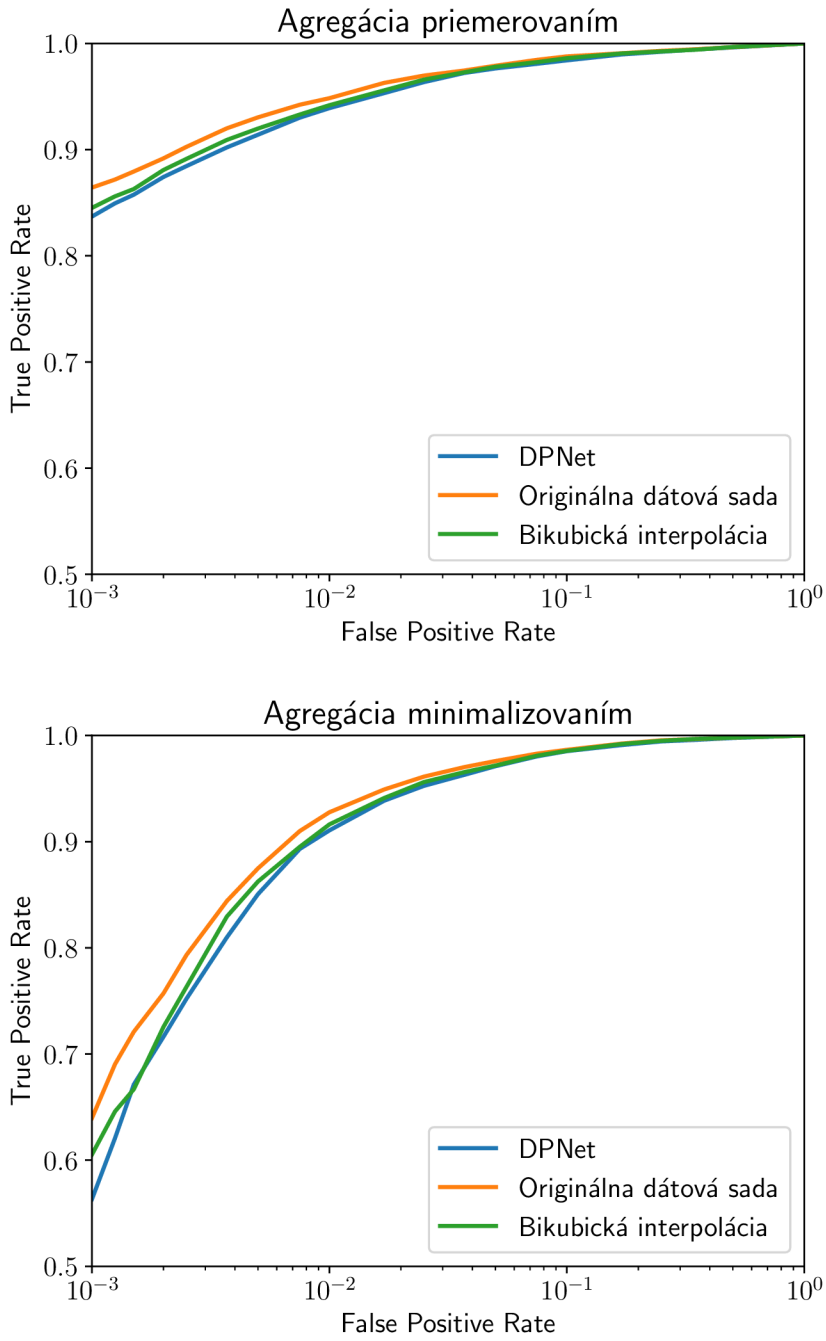
(c) MAE, *Adversarial loss*, *Identity loss*

(d) MAE, *Adversarial loss*, *Gradient penalty*

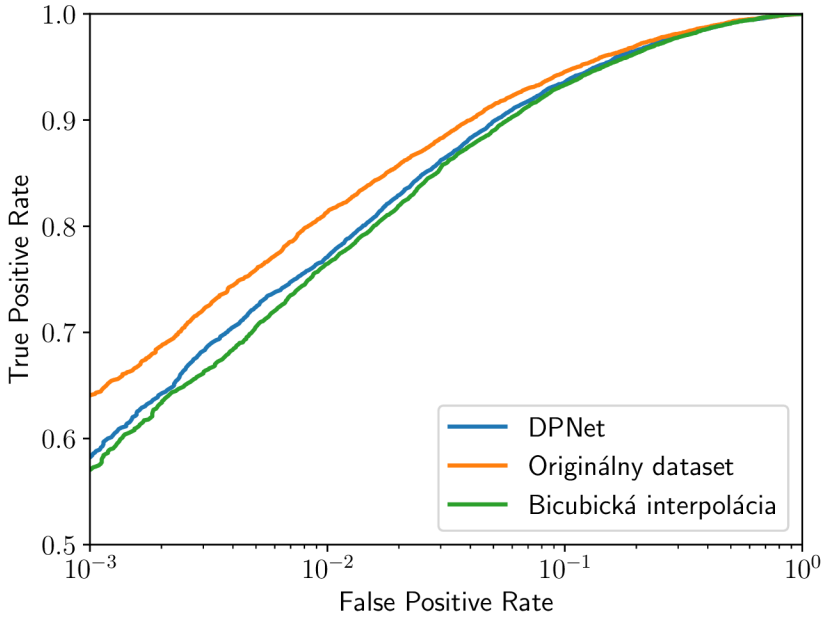
Obr. 6.7: Výstupy neurónovej siete DPNetNN, ktorá používa *Nearest neighbour* interpoláciu nasledovanú konvolúciou pre zväčšenie obrázku s *Instance normalization* a je trénovaná kombinácou chýb pod príslušnými obrázkami. Z daných konfigurácii je najkvalitnejší výstup modelu trénovaného iba pomocou MAE a adversariálnej chyby, avšak ani ten nedosahuje kvalitu ako model trénovaný iba pomocou MAE.

stiahnuté pomocou vyhľadávacieho nástroja Google a následne zarovnané a orezané. Tváre na obrázkoch sa líšia v natočení, veku osôb, osvetlení a pozadí. Z identít v dátovej sade je približne 59.3% mužov. Keďže táto dátová sada je rozdelená iba na trénovaciu a testovaciu, nemá definovaný protokol na verifikáciu.

Pre vyhodnotenie bolo preto vygenerovaných 10000 porovnaní obrázkov s rovnakými identitami a 100000 porovnaní obrázkov s rôznymi identitami. Z výsledkov, zobrazených na grafe 6.9 je viditeľné, že zväčšovanie obrázkov pomocou natrénovaného modelu navrhnutej neurónovej siete zlepšilo schopnosť klasifikátoru rozpoznať identity na obrázkoch v porovnaní s bikubickou interpoláciou rádovo o jednotky percent.



Obr. 6.8: Vyhodnotenie verifikácie na porovnávacom teste IJB-A meraním vzdialeností identít a následným zobrazením *Receiver operating characteristic* krivky. Pre porovnanie podobnosti medzi dvomi skupinami obrázkov boli vektory identít priemerované (hore) a v druhom prípade bola zo skupín bola vybraná podobnosť dvoch najpodobnejších vektorov (dole). Z výsledkov je viditeľné, že určenie identity priemerovaním vektorov identít z viacerých obrázkov výrazne zlepšuje schopnosť rozpoznania.



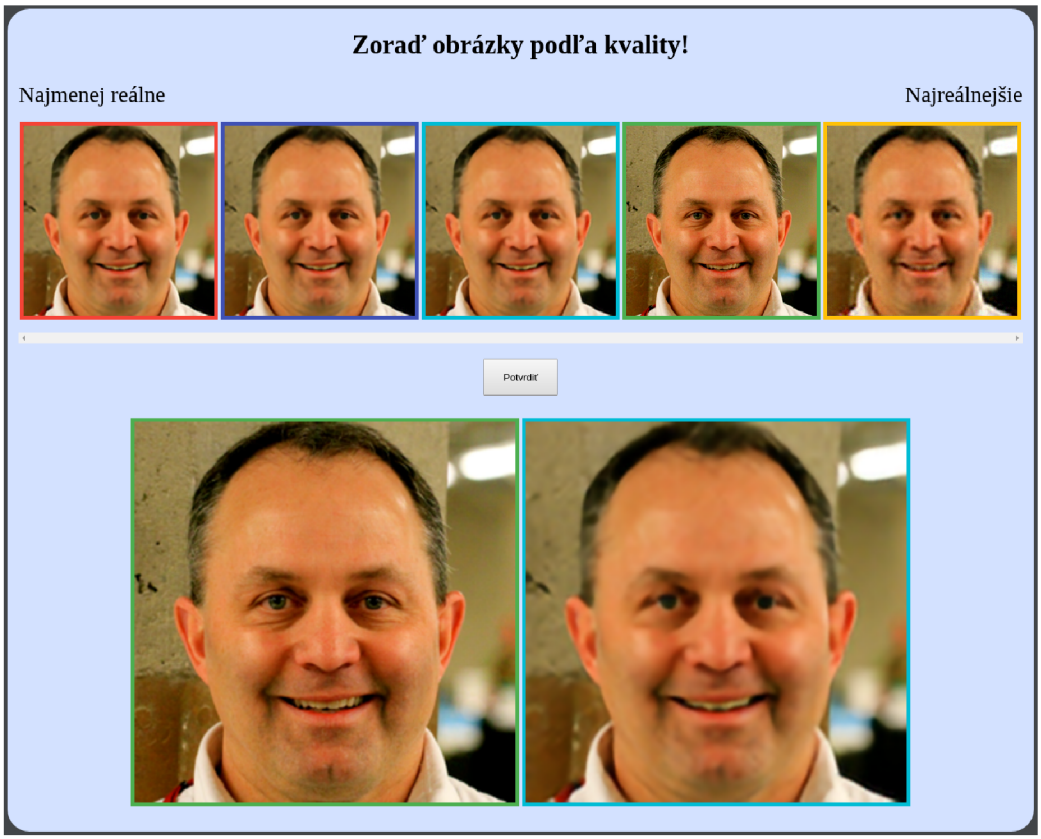
Obr. 6.9: Vyhodnotenie verifikácie na dátovej sade VGGFace2. Z výsledkov je viditeľné, že pri použití obrázkov zväčšených neurónovou sieťou DPNet sa schopnosť rozpoznania identity zlepšuje.

DPNet_adversarial	CARN	DPNet	RCAN	Dataset
1.022	2.837	2.904	3.25	4.987

Tabuľka 6.2: Priemerné skóre obrázkov pochádzajúcich z príslušných modelov neurónových sietí a z dátovej sady. Vyššie skóre označuje fotografiu s vyššou kvalitou.

6.7 Porovnanie hodnotenia kvality s ľudským vnímaním

Pre porovnanie, či hodnotenie kvality metrikami PSNR a SSIM naozaj zodpovedá ľudskému vnímaniu, som vykonal experiment, v ktorom zúčastnení respondenti porovnávali obrázky zväčšené natrénovanými modelmi. Úlohou bolo zoradiť vybrané obrázky do postupnosti od najmenej kvalitného až po najkvalitnejší. Pre porovnanie som vybral neurónové siete CARN [34], RCAN [72] a DPNet natrénované pomocou MAE chyby. Do porovnania som taktiež pridal neurónovú sieť DPNet natrénovanú pomocou MAE chyby a Relativistického adversariálneho diskriminátoru [23]. Poslednou súčasťou porovnania je originálny obrázok kvôli zisteniu, či sú zväčšené obrázky kvalitou rozlíšiteľné od zväčšených obrázkov. Pre získanie zoradení od respondentov som naprogramoval jednoduchú webstránku, ktorej rozhranie je ilustrované na obrázku 6.10. Rozhranie umožňuje presúvať jednotlivé obrázky a zväčšiť dva obrázky pre ich podrobnejšie porovnanie. Vyhodnotenie od jedného respondenta pozostáva celkovo z 50 zoradení. Každé zoradenie priradí každému obrázku unikátne skóre od 1 do 5. Celkovo som v rámci experimentu získal 300 zoradení. Následne som vypočítal test štatistickej významnosti nad kolekciou zoradení pomocou Friedman testu [8] a post-hoc Nemenyi testu. Pri zvolenej tolerancii chyby 1% nie je možné vylúčiť hypotézu,



Obr. 6.10: Užívateľské rozhranie webstránky určenej na zoradenie obrázkov podľa kvality. Rozhranie umožnuje meniť poradie obrázkov v hornej časti a zväčšiť dva obrázky pre podrobnejšie porovnanie.

že priemerné skóre sú reálne rozdielne iba pri skúmaní dvojice DPNet a CARN. Z výsledkov teda vyplýva, že s toleranciou 1% sú všetky ostatné dvojice v poradí, ktoré je uvedené v tabuľke 6.2. Výsledné poradie ukazuje, že obrázky zväčšené neurónovou sieťou RCAN sú viditeľne menej kvalitné ako originálne obrázky, takže rekonštrukcia na úroveň kvality pôvodného obrázku týmto spôsobom dosiahnutá nebola. Taktiež sa potvrdilo, že model trénovaný pomocou *Adversarial loss* produkuje menej kvalitné obrázky ako model trénovaný iba pomocou *Pixel loss*. Hodnoty skóre metód, kde modely neurónových sietí boli trénované pomocou *Pixel loss*, sú v rozpätí 0.5 čo značí iba malé rozdiely v kvalite.

Kapitola 7

Záver

Cieľom diplomovej práce bolo navrhnúť metódu superrezolúcie obrázkov tvárí s dôrazom na zachovanie identity. Navrhol som metódu pre superrezolúciu obrázkov, ktorú som porovnal s už existujúcimi riešeniami. Experimenty vykonané na dátových sadách CelebA a Flickr-Faces-HQ ukázali, že architektúry, ktoré nevyžadujú aby bol obrázok zväčšený pred tým, ako s ním pracujú, dosahujú lepšie výsledky a to za menší čas. Navrhnutá architektúra DPNet dosahuje na dátovej sade Flickr-Faces-HQ hodnotu SSIM 0.856, pričom najlepšia spomedzi všetkých porovnávaných architektúr, zvaná *Residual channel attention network*, dosahuje hodnotu 0.858. Pri použití adversariálnej chyby vznikali v obrázkoch artefakty a zatiaľ sa nepodarilo dosiahnuť lepšie výsledky, ako pri použití iba absolútnej chyby. Vyhodnotenie klasifikácie ukázalo zlepšenie pri použití obrázkov zväčšených pomocou architektúry DPNet oproti zväčšovaniu bikubickou interpoláciou. Porovnanie kvality rekonštrukcie respondentmi ukázalo iba malé rozdiely medzi navrhnutou architektúrou a inými modernými architektúrami. V rámci pokračovania budem pracovať na trénovaní architektúry pomocou adversariálnej chyby a snažiť sa vylepšiť doteraz dosiahnuté výsledky.

Literatúra

[1] Aitken, A. P., Ledig, C., Theis, L., Caballero, J., Wang, Z. et al. Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize. *CoRR*. 2017, abs/1707.02937.

[2] Avcibas, I., Sankur, B. a Sayood, K. Statistical evaluation of image quality measures. *Journal of Electronic imaging*. International Society for Optics and Photonics. 2002, zv. 11, č. 2, s. 206–224.

[3] Bovik, A. *Handbook of Image and Video Processing*. Elsevier, jún 2005. ISBN 9780080533612.

[4] Cao, Q., Shen, L., Xie, W., Parkhi, O. M. a Zisserman, A. VGGFace2: A dataset for recognising faces across pose and age. In: IEEE. *IEEE International Conference on Automatic Face & Gesture Recognition*. 2018, s. 67–74.

[5] Dong, C., Loy, C. C., He, K. a Tang, X. Image Super-Resolution Using Deep Convolutional Networks. In: *The European Conference on Computer Vision (ECCV)*. 2014.

[6] Dong, C., Loy, C. C. a Tang, X. Accelerating the Super-Resolution Convolutional Neural Network. In: Leibe, B., Matas, J., Sebe, N. a Welling, M., ed. *The European Conference on Computer Vision (ECCV)*. Springer, 2016, sv. 9906, s. 391–407. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-46475-6_25.

[7] Freeman, W. T., Jones, T. R. a Pasztor, E. C. Example-based super-resolution. *IEEE Computer graphics and Applications*. IEEE. 2002, zv. 22, č. 2, s. 56–65.

[8] Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*. Taylor & Francis. 1937, zv. 32, č. 200, s. 675–701. DOI: 10.1080/01621459.1937.10503522.

[9] Gabarda, S. a Cristóbal, G. Blind image quality assessment through anisotropy. *JOSA A*. Optical Society of America. 2007, zv. 24, č. 12, s. B42–B51.

[10] George, A. a Livingston, S. J. A survey on full reference image quality assessment algorithms. *International Journal of Research in Engineering and Technology*. Citeseer. 2013, zv. 2, č. 12, s. 303–307.

[11] Glasner, D., Bagon, S. a Irani, M. Super-Resolution from a Single Image. In: *International Conference on Computer Vision ICCV*. 2009.

[12] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D. et al. Generative Adversarial Nets. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*. 2014, s. 2672–2680.

[13] Graves, A., Mohamed, A. a Hinton, G. Speech recognition with deep recurrent neural networks. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. May 2013, s. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947. ISSN 1520-6149.

[14] Gross, S. a Wilber, M. *Training and investigating Residual Nets*. February 2016. Dostupné z: http://torch.ch/blog/2016/02/04/resnets.html.

[15] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. a Courville, A. C. Improved Training of Wasserstein GANs. In: Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R. et al., ed. *Advances in Neural Information Processing Systems NIPS*. Curran Associates, Inc., 2017, s. 5767–5777.

[16] Haris, M., Shakhnarovich, G. a Ukita, N. Deep Back-Projection Networks for Super-Resolution. In: *IEEE Conference on Computer Vision and Pattern Recognition CVPR*. IEEE Computer Society, 2018, s. 1664–1673. DOI: 10.1109/CVPR.2018.00179.

[17] He, H. a Siu, W. Single image super-resolution using Gaussian process regression. In: *CVPR 2011*. June 2011, s. 449–456. DOI: 10.1109/CVPR.2011.5995713. ISSN 1063-6919.

[18] He, K., Zhang, X., Ren, S. a Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In: *IEEE International Conference on Computer Vision, ICCV*. IEEE Computer Society, 2015, s. 1026–1034. DOI: 10.1109/ICCV.2015.123.

[19] He, K., Zhang, X., Ren, S. a Sun, J. Deep residual learning for image recognition. In: *Conference on Computer Vision and Pattern Recognition CVPR*. 2016, s. 770–778.

[20] He, K., Zhang, X., Ren, S. a Sun, J. Identity Mappings in Deep Residual Networks. In: Leibe, B., Matas, J., Sebe, N. a Welling, M., ed. *European Conference on Computer Vision ECCV*. Springer, 2016, sv. 9908, s. 630–645. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-46493-0_38.

[21] In Kim, K. a Kwon, Y. Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. IEEE. Júl 2010, zv. 32, č. 6, s. 1127 – 1133. DOI: 10.1109/TPAMI.2010.25.

[22] Johnson, J., Alahi, A. a Fei Fei, L. Perceptual losses for real-time style transfer and super-resolution. In: Springer. *European Conference on Computer Vision*. 2016, s. 694–711.

[23] Jolicoeur Martineau, A. The relativistic discriminator: a key element missing from standard GAN. In: *International Conference on Learning Representations ICLR*. 2019.

[24] Karras, T., Aila, T., Laine, S. a Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In: *International Conference on Learning Representations ICLR*. OpenReview.net, 2018.

[25] Karras, T., Laine, S. a Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In: *IEEE Conference on Computer Vision and Pattern Recognition CVPR*. Computer Vision Foundation / IEEE, 2019, s. 4401–4410. DOI: 10.1109/CVPR.2019.00453.

[26] Kim, J., Lee, J. K. a Lee, K. M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In: *IEEE Conference on Computer Vision and Pattern Recognition CVPR*. IEEE Computer Society, 2016, s. 1646–1654. DOI: 10.1109/CVPR.2016.182.

[27] Kim, J., Lee, J. K. a Lee, K. M. Deeply-Recursive Convolutional Network for Image Super-Resolution. In: *The IEEE Conference on Computer Vision and Pattern Recognition CVPR*. IEEE Computer Society, 2016, s. 1637–1645. DOI: 10.1109/CVPR.2016.181.

[28] Kingma, D. P. a Ba, J. Adam: A Method for Stochastic Optimization. In: Bengio, Y. a LeCun, Y., ed. *3rd International Conference on Learning Representations ICLR*. 2015.

[29] Köhler, T. Multi-Frame Super-Resolution Reconstruction with Applications to Medical Imaging. *CoRR*. 2018, abs/1812.09375.

[30] Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D. a Matas, J. DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks. In: *IEEE Conference on Computer Vision and Pattern Recognition CVPR*. IEEE Computer Society, 2018, s. 8183–8192. DOI: 10.1109/CVPR.2018.00854.

[31] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A. et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: *The IEEE Conference on Computer Vision and Pattern Recognition CVPR*. July 2017.

[32] Lefkimmiatis, S. Universal Denoising Networks : A Novel CNN Architecture for Image Denoising. In: *The IEEE Conference on Computer Vision and Pattern Recognition CVPR*. June 2018.

[33] Li, S., Yang, Z. a Li, H. Statistical Evaluation of No-Reference Image Quality Assessment Metrics for Remote Sensing Images. *ISPRS International Journal of Geo-Information*. 2017, zv. 6, č. 5. ISSN 2220-9964.

[34] Li, Y., Agustsson, E., Gu, S., Timofte, R. a Van Gool, L. CARN: Convolutional Anchored Regression Network for Fast and Accurate Single Image Super-Resolution. In: *The European Conference on Computer Vision (ECCV) Workshops*. September 2018.

[35] Lim, B., Son, S., Kim, H., Nah, S. a Lee, K. M. Enhanced Deep Residual Networks for Single Image Super-Resolution. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. July 2017.

[36] Liu, Z., Luo, P., Wang, X. a Tang, X. Deep Learning Face Attributes in the Wild. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, s. 3730–3738.

[37] Mahendran, A. a Vedaldi, A. Understanding deep image representations by inverting them. In: *IEEE Conference on Computer Vision and Pattern Recognition CVPR*. IEEE Computer Society, 2015, s. 5188–5196. DOI: 10.1109/CVPR.2015.7299155.

[38] Mittal, A., Moorthy, A. K. a Bovik, A. C. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*. IEEE. 2012, zv. 21, č. 12, s. 4695–4708.

[39] Moghaddam, A. H., Moghaddam, M. H. a Esfandyari, M. Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science*. 2016, zv. 21, č. 41, s. 89 – 93. DOI: https://doi.org/10.1016/j.jefas.2016.07.002. ISSN 2077-1886.

[40] Moorthy, A. K. a Bovik, A. C. A two-step framework for constructing blind image quality indices. *IEEE Signal processing letters*. IEEE. 2010, zv. 17, č. 5, s. 513–516.

[41] Nah, S., Kim, T. H. a Lee, K. M. Deep Multi-Scale Convolutional Neural Network for Dynamic Scene Deblurring. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[42] Ni, K. S. a Nguyen, T. Q. Image superresolution using support vector regression. *IEEE Transactions on Image Processing*. IEEE. 2007, zv. 16, č. 6, s. 1596–1610.

[43] Nwankpa, C., Ijomah, W., Gachagan, A. a Marshall, S. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *CoRR*. 2018, abs/1811.03378.

[44] Odena, A., Dumoulin, V. a Olah, C. Deconvolution and Checkerboard Artifacts. *Distill*. 2016. DOI: 10.23915/distill.00003.

[45] Pan, J., Sun, D., Pfister, H. a Yang, M. Blind Image Deblurring Using Dark Channel Prior. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, s. 1628–1636.

[46] Ramakrishnan, S., Pachori, S., Gangopadhyay, A. a Raman, S. Deep Generative Filter for Motion Deblurring. In:. Október 2017, s. 2993–3000. DOI: 10.1109/ICCVW.2017.353.

[47] 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*. 2017, zv. 39, č. 6, s. 1137–1149.

[48] Renting Liu, Zhaorong Li a Jiaya Jia. Image partial blur detection and classification. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008, s. 1–8.

[49] Ronneberger, O., Fischer, P. a Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., III, W. M. W. a Frangi, A. F., ed. *Medical Image Computing and Computer-Assisted Intervention - MICCAI*. Springer, 2015, sv. 9351, s. 234–241. Lecture Notes in Computer Science. DOI: 10.1007/978-3-319-24574-4_28.

[50] Saad, M. A., Bovik, A. C. a Charrier, C. A DCT statistics-based blind image quality index. *IEEE Signal Processing Letters*. IEEE. 2010, zv. 17, č. 6, s. 583–586.

[51] Shan, Q., Li, Z., Jia, J. a Tang, C.-K. Fast Image/Video Upsampling. *ACM Transactions on Graphics (SIGGRAPH ASIA)*. 2008.

[52] Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P. et al. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In: *IEEE Conference on Computer Vision and Pattern Recognition CVPR*. IEEE Computer Society, 2016, s. 1874–1883. DOI: 10.1109/CVPR.2016.207.

[53] Simonyan, K. a Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: Bengio, Y. a LeCun, Y., ed. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

[54] Sun, J., Xu, Z. a Shum, H.-Y. Image super-resolution using gradient profile prior. In: IEEE. *IEEE Conference on Computer Vision and Pattern Recognition CVPR*. 2008, s. 1–8.

[55] Sun, J., Zheng, N.-N., Tao, H. a Shum, H.-Y. Image hallucination with primal sketch priors. In: IEEE. *Computer Vision and Pattern Recognition*. 2003, sv. 2, s. II–729.

[56] Sun, Y., Chen, Y., Wang, X. a Tang, X. Deep Learning Face Representation by Joint Identification-Verification. Cambridge, MA, USA: MIT Press. 2014, s. 1988–1996. NIPS'14.

[57] Tang, C., Zhu, X., Liu, X., Wang, L. a Zomaya, A. DeFusionNET: Defocus Blur Detection via Recurrently Fusing and Refining Multi-Scale Deep Features. In: *IEEE Conference on Computer Vision and Pattern Recognition CVPR*. June 2019.

[58] Ulyanov, D., Vedaldi, A. a Lempitsky, V. S. Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR*. 2016, abs/1607.08022.

[59] Vu, T., Luu, T. M. a Yoo, C. D. Perception-Enhanced Image Super-Resolution via Relativistic Generative Adversarial Networks. In: Leal Taixé, L. a Roth, S., ed. *Computer Vision – ECCV 2018 Workshops*. Cham: Springer International Publishing, 2019, s. 98–113. ISBN 978-3-030-11021-5.

[60] Wang, Q., Tang, X. a Shum, H. Patch based blind image super resolution. In: *IEEE International Conference on Computer Vision (ICCV)*. Oct 2005, sv. 1, s. 709–716 Vol. 1. DOI: 10.1109/ICCV.2005.186. ISSN 1550-5499.

[61] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y. et al. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: *The European Conference on Computer Vision (ECCV) Workshops*. September 2018.

[62] Wang, Z. a Bovik, A. C. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*. Jan 2009, zv. 26, č. 1, s. 98–117. DOI: 10.1109/MSP.2008.930649. ISSN 1053-5888.

[63] Wang, Z., Bovik, A. C., Sheikh, H. R. a Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*. IEEE. 2004, zv. 13, č. 4, s. 600–612.

[64] Xue, W., Zhang, L. a Mou, X. Learning without Human Scores for Blind Image Quality Assessment. In:. Jún 2013, s. 995–1002. DOI: 10.1109/CVPR.2013.133.

[65] Yang, C.-Y., Liu, S. a Yang, M.-H. Structured face hallucination. In: *The IEEE Conference on Computer Vision and Pattern Recognition CVPR*. 2013, s. 1099–1106.

[66] Yang, J., Wright, J., Huang, T. S. a Ma, Y. Image super-resolution via sparse representation. *IEEE transactions on image processing*. IEEE. 2010, zv. 19, č. 11, s. 2861–2873.

[67] Yang, X.-H., Jing, Z.-L., Liu, G., Hua, L.-Z. a Ma, D.-W. Fusion of multi-spectral and panchromatic images using fuzzy rule. *Communications in nonlinear science and numerical simulation*. Elsevier. 2007, zv. 12, č. 7, s. 1334–1350.

[68] Zeiler, M. D., Taylor, G. W. a Fergus, R. Adaptive deconvolutional networks for mid and high level feature learning. In: IEEE. *IEEE International Conference on Computer Vision ICCV*. 2011, s. 2018–2025.

[69] Zhang, H., Liu, D. a Xiong, Z. CNN-based text image super-resolution tailored for OCR. In: IEEE. *IEEE Conference on Visual Communications and Image Processing VCIP*. 2017, s. 1–4.

[70] Zhang, K., Zuo, W., Gu, S. a Zhang, L. Learning Deep CNN Denoiser Prior for Image Restoration. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 2017, s. 2808–2817. DOI: 10.1109/CVPR.2017.300.

[71] Zhang, K., Zhang, Z., Cheng, C.-W., Hsu, W. H., Qiao, Y. et al. Super-Identity Convolutional Neural Network for Face Hallucination. In: *The European Conference on Computer Vision (ECCV)*. September 2018.

[72] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B. et al. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In: *The European Conference on Computer Vision (ECCV)*. September 2018.

[73] Zhou, Z., Liang, J., Song, Y., Yu, L., Wang, H. et al. Lipschitz Generative Adversarial Nets. In: Chaudhuri, K. a Salakhutdinov, R., ed. *International Conference on Machine Learning, ICML*. PMLR, 2019, sv. 97, s. 7584–7593. Proceedings of Machine Learning Research.

Príloha A

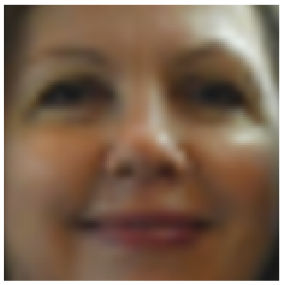
Zoznam skratiek

- **AdaIN** - Adaptive Instance Normalization
- **BN** - Batch normalization
- **CARN** - Cascading residual network
- **DBPN** - Deep back-projection network
- **DRCN** - Deep recursive convolutional network
- **FFHQ** - Flickr-Faces-HQ
- **GAN** - Generative adversarial network
- **IJB-A** - IARPA Janus Benchmark-A
- **MAE** - Mean absolute error
- **MISR** - Multiple image super-resolution
- **MSE** - Mean squared error
- **PSNR** - Peak signal-to-noise ratio
- **ResNet** - Residual network
- **RCAN** - Residual channel attention network
- **ReLU** - Rectified linear unit
- **SISR** - Single image super-resolution
- **SSIM** - Structural similarity

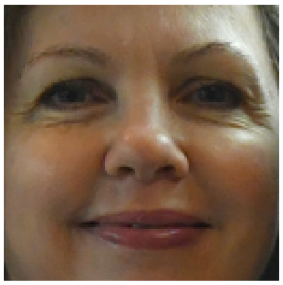
Príloha B

Ukážky zväčšených obrázkov

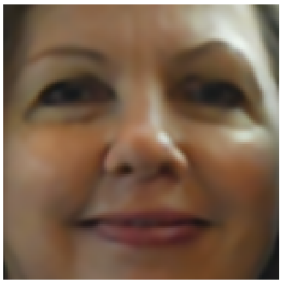
(a) Zmenšený obrázok



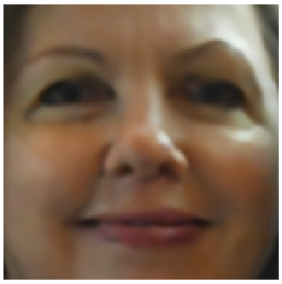
(b) Bikubická interpolácia



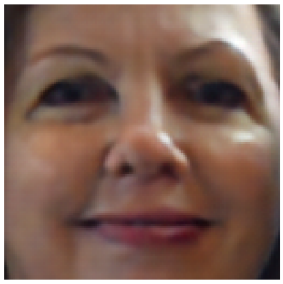
(c) Originálny obrázok



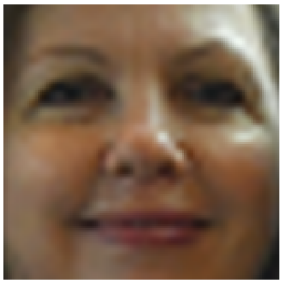
(d) DPNet (32.99)



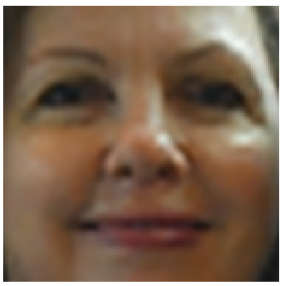
(e) CARN (32.78)



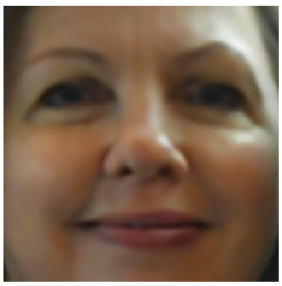
(f) DBPN (29.35)



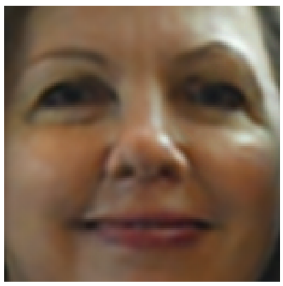
(g) EDSR (31.37)



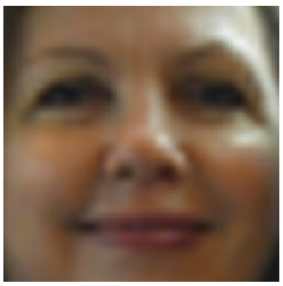
(h) ESPCN (31.73)



(i) RCAN (33.12)



(j) SRResNet (32.23)



(k) VDSR (29.606)

Obr. B.1: Výsledky štvornásobného zväčšenia obrázku o veľkosti 64×64 z datasetu FFHQ. Každý zväčšený obrázok má v zátvorke poznačenú hodnotu PSNR (db).

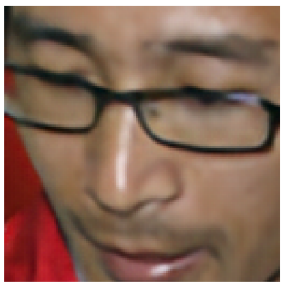
(a) Zmenšený obrázok

(b) Bikubická interpolácia

(c) Originálny obrázok

(d) DPNet (31.91)

(e) CARN (31.73)



(f) DBPN (28.48)

(g) EDSR (29.66)

(h) ESPCN (29.92)

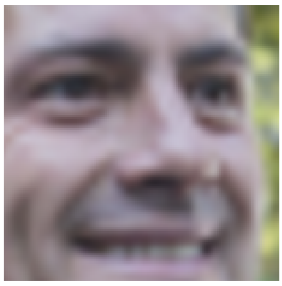
(i) RCAN (31.98)

(j) SRResNet (30.97)

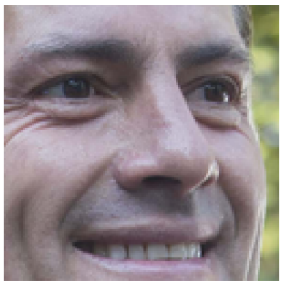
(k) VDSR (27.57)

Obr. B.2: Výsledky štvornásobného zväčšenia obrázku o veľkosti 64×64 z datasetu FFHQ. Každý zväčšený obrázok má v zátvorke poznačenú hodnotu PSNR (db).

(a) Zmenšený obrázok



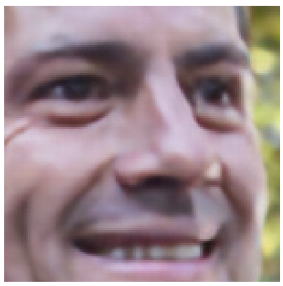
(b) Bikubická interpolácia



(c) Originálny obrázok

(d) DPNet (32.11)

(e) CARN (31.99)

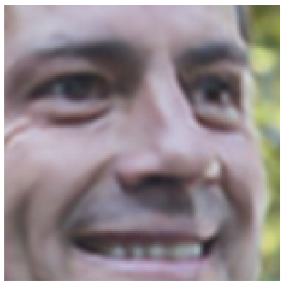


(f) DBPN (28.41)

(g) EDSR (31.15)

(h) ESPCN (31.20)

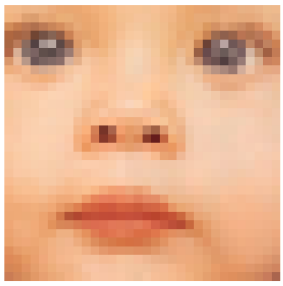
(i) RCAN (32.22)



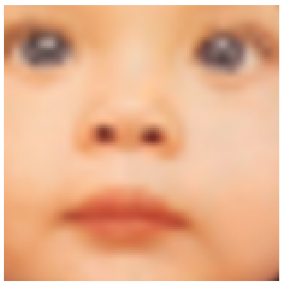
(j) SRResNet (31.55)

(k) VDSR (28.63)

Obr. B.3: Výsledky štvornásobného zväčšenia obrázku o veľkosti 64×64 z datasetu FFHQ. Každý zväčšený obrázok má v zátvorke poznačenú hodnotu PSNR (db).

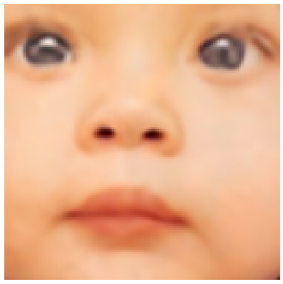


(a) Zmenšený obrázok



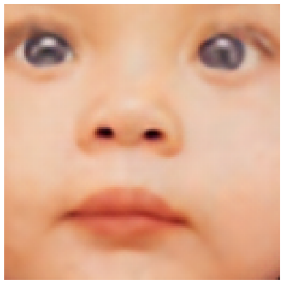
(b) Bikubická interpolácia

(c) Originálny obrázok



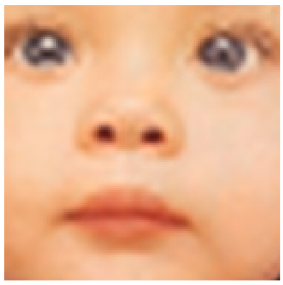
(d) DPNet (28.84)

(e) CARN (28.89)

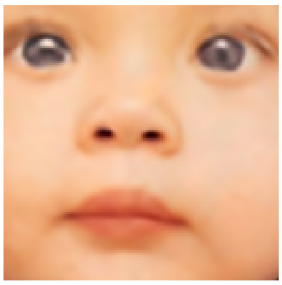


(f) DBPN (27.25)

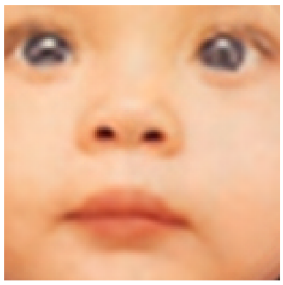
(g) EDSR (27.39)



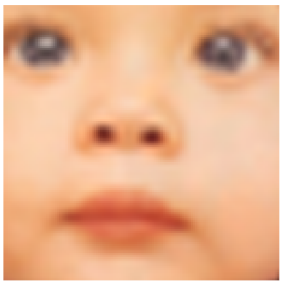
(h) ESPCN (27.89)



(i) RCAN (28.95)



(j) SRResNet (28.44)



(k) VDSR (24.39)

Obr. B.4: Výsledky štvornásobného zväčšenia obrázku o veľkosti 64×64 z datasetu Set5. Každý zväčšený obrázok má v zátvorke poznačenú hodnotu PSNR (db). Znížená hodnota PSNR je pravdepodobne zapríčinená

Príloha C

Popis a použitie repozitára

Súčasťou diplomovej práce je taktiež nástroj, ktorý umožnuje trénovať neurónové siete. Celkovo sa časti nástroja starajú o ukladanie natrénovavných modelov, priebežné vyhodnotenie, ukladanie zväčšených obrázkov počas trénovania, vykreslovanie grafu a ďalšie iné záležitosti. Podrobnosti o uložených objektoch a formátu konfiguračných súborov popisujú nasledujúce podkapitoly.

C.1 Obsah repozitára

Vytvorený kód je verejne prístupný na GitHub-e[1] so stručným popisom a návodom na použitie. Zloženie repozitára je nasledovné:

```
/
├── configs
├── datasets
│   ├── transforms.py
│   └── ...
├── feature_extractor
│   └── dlib_feature_extractor.py
├── models
│   ├── abstract_cnn_solver.py
│   ├── abstract_gan_solver.py
│   └── ...
├── utils
│   ├── drawer.py
│   ├── factory.py
│   └── logger.py
├── evaluate.py
├── cnn_plot.py
├── gan_plot.py
├── main.py
└── single_pass.py
```

[1] https://github.com/MatusBako/MakeFacesGreatAgain

Na koreňovej úrovni sa nachádzajú skripty ktoré obalujú kľúčovú funkcionalitu `Solver` objektov a skripty na vytváranie grafov z výsledkov trénovania. `Solver` je objekt, ktorý spravuje trénovanie a vyhodnocovanie neurónovej siete. Podrobný popis objektu je v samostatnej podkapitole 5.1. `main.py` je skript, ktorý sa volá pre trénovanie architektúry. Jediným parametrom tohto skriptu je cesta ku konfiguračnému súboru, ktorý obsahuje všetky potrebné parametre. Konkrétny formát tohto súboru je popísaný v sekcii C.2. Skript `evaluate.py` slúži na vyhodnotenie vybranej architektúry nad daným datasetom. Výstupom sú hodnoty PSNR, SSIM a priemerná vzdialenosť identít. Skript `single_pass.py` slúži na zväčšenie jedného obrázku natrénovaným modelom. Adresár `utils` obsahuje pomocné triedy a funkcie určené na priebežné ukladanie informácii a výsledkov počas trénovania.

C.2 Formát konfiguračného súboru

Pre nastavenie trénovania sa používa jeden súbor v `.ini` formáte spracovaný pomocou `ConfigParser` modulu. Jednotlivé sekcie súboru popisujú jednotlivé časti, ktoré zasahujú do samotného procesu. Pretože siete trénované s adversariálnou chybou používajú aj diskriminátor, parametre pre nastavenie trénovania sú v tomto prípade odlišné, a odlišuje sa aj názov prvej sekcie. V prípadne, že nepoužívame diskriminátor, sa prvá sekcia nazýva `CNN` a jej položky sú nasledovné:

- `OutputFolder <str>`: cesta k adresáru, v ktorom bude vytvorený adresár s výsledkami
- `BatchSize <int>`: počet obrázkov spracovaných v jednej iterácii
- `Device <'cuda', 'cpu'>`: výber medzi počítaním na procesore alebo grafickej karte
- `IterationLimit <int>`: počet iterácii trénovania
- `LearningRate <float>`: parameter trénovania
- `UpscaleFactor <int>`: faktor zväčšenia obrázku
- `ModelName <str>`: názov modelu z adresára `models`
- `IterationsPerSnapshot <int>`: počet iterácii, po ktorých bude uložený stav modelu na disk
- `IterationsPerImage <int>`: počet iterácii, po ktorých je na disk uložený výstupný obrázok
- `IterationsToEvaluation <int>`: počet iterácii, po ktorých je vykonané vyhodnotenie
- `EvaluationIterations <int>`: počet iterácii vykonaných počas vyhodnotenia

Použitie diskriminátoru znamená pre konfiguračný súbor dve zmeny v položkách a pridané parametre pre škálovanie chýb. Namiesto `LearningRate` pribudne `GeneratorLearningRate` a `DiscriminatorLearningRate`, pretože chceme mať možnosť nastavovať rôzny parameter učenia pre oba modely. Položku `ModelName` nahrádza `Generator` a `Discriminator`, pretože

podľa názvu modelu sa importuje modul architektúry. Týmto spôsobom máme možnosť použiť diskriminátor s jednej a generátor z inej architektúry. Niektoré z architektúr majú implementovanú možnosť škálovať jednotlivé časti výpočtu chyby. Názvy týchto parametrov sú `PixelLossParam`, `AdversarialLossParam`, `FeatureLossParam`, `GradientPenaltyParam`, `IdentityLossParam`, `VarianceLossParam`. Či daná architektúra používa niektoré z týchto parametrov je možné si overiť v príslušnom `solver.py` súbore.

Sekcia, ktorá popisuje používanú dátovú sadu, sa nazýva `Dataset`, a obsahuje nasledujúce položky:

- `Class <str>`: názov objektu, ktorý obaluje dátovú sadu
- `TrainData <str>`: cesta k adresáru, ktorý obsahuje trénovacie dáta
- `TestData <str>`: cesta k adresáru, ktorý obsahuje testovacie dáta
- `TrainLength <int>`: počet použitých položiek z trénovacej dátovej sady
- `TestLength <int>`: počet použitých položiek z testovacej dátovej sady

Objekt, ktorého názov je zadaný do položky `Class`, je implementovaný v niektorom zo súborov v adresári `datasets`. Do položiek `TrainLength` a `TestLength` je možné zadať hodnotu 0, pričom sú v tomto prípade použité všetky položky dátovej sady.

Nasledujúce dve sekcie, `Optimizer` a `Scheduler`, popisujú nastavenie optimalizátora a plánovača. Obe sekcia majú rovnaké polia:

- `Name <str>`: názov triedy
- `Args <list>`: argumenty
- `Kwargs <dict>`: slovník argumentov

Podľa názvu triedy je dynamicky importovaný optimalizátor, ktorému sú následne dodané argumenty z oboch zvyšných polí. Možné hodnoty sa nachádzajú v dokumentácii použitej knižnice. V prípade použitia diskriminátoru majú generátor aj diskriminátor vlastný optimalizátor aj plánovať, takže názvy sekcii dostávajú príslušnú predponu `Gen` alebo `Dict`. Rovnaké pravidlá platia aj pre plánovač.

Poslednou sekciou je `FeatureExtractor`, ktorá nastavuje extraktor rysov pre porovnávanie identít počas vyhodnotenia. Momentálne jediným funkčným extraktorom je model z knižnice `dlib`. Polia `ShapePredictor` a `Extractor` obsahujú cesty ku modelom detektoru a extraktoru.

C.3 Formát uloženého objektu

Pri trénovaní je možné v konfiguračnom súbore nastaviť parameter periodického ukladania po danom počte iterácii. V takom prípade sú na uloženie a prípadne následné načítanie použité funkcie `save()` a `load()` z priamo z knižnice PyTorch. Uložený objekt je slovník ktorý obsahuje tzv. stavové slovníky jednotlivých objektov získané zavolaním metódy `state_dict()` nad každým objektom. Konkrétne názvy položiek v slovníku sú nasledovné:

- `model_name <str>`: názov architektúry

- `model <dict>`: stav modelu
- `optimizer <dict>`: stav optimalizátora
- `scheduler <dict>`: stav plánovača
- `iteration <int>`: počet iterácii, po ktorých bol model uložený
- `upscale <int>`: faktor zväčšenia, pre ktorý bol model trénovaný