Univerzita Pavla Jozefa Šafárika v Košiciach

Prírodovedecká fakulta

ZVÝŠENIE ROZLÍŠENIA A REKONŠTRUKCIA OBRAZU POMOCOU GENERATÍVNYCH NEURÓNOVÝCH SIETÍ

DIPLOMOVÁ PRÁCA

Študijný odbor: **Informatika**
Školiace pracovisko: **Ústav informatiky**
Vedúci záverečnej práce: **Ing. Miron Kuzma**

Košice 2026

Bc. Roman Rapco

Poďakovanie

Rád by som poďakoval vedúcemu diplomovej práce Ing. Mironovi Kuzmovi a doc. RNDr. Ľubomírovi Antonimu, PhD., za cenné pripomienky, odborné rady a ochotu počas tvorby mojej diplomovej práce.

Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

36145806730369540

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Roman Rapco
Študijný program: informatika (jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: Informatika
Typ záverečnej práce: Diplomová práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Zvýšenie rozlíšenia a rekonštrukcia obrazu pomocou generatívnych neurónových sietí

Názov EN: Image Super-Resolution and Reconstruction by Generative Neural Networks

Cieľ:
1. Spracovať prehľad techník v počítačovom videní na zvýšenie rozlíšenia a rekonštrukciu obrazu.
2. Navrhnúť novú metódu na zvýšenie rozlíšenia obrazu a rekonštrukciu pomocou neurónových sietí.
3. Implementovať navrhnuté metódu a porovnať výsledky dosiahnuté na vybranej dátovej sade.

Literatúra:
1. Park, S. C., Park, M. K., & Kang, M. G. (2003). Super-resolution image reconstruction: a technical overview. IEEE signal processing magazine, 20(3), 21-36.
2. Dong, C., Loy, C. C., & Tang, X. (2016). Accelerating the super-resolution convolutional neural network. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14 (pp. 391-407). Springer International Publishing.

Vedúci: Ing. Miron Kuzma, PhD.
Oponent: RNDr. Rastislav Krivoš-Belluš, PhD.
Ústav : ÚINF - Ústav informatiky
Riaditeľ ústavu: doc. RNDr. Ondrej Krídlo, PhD.

Dátum schválenia: 30.04.2026

Abstrakt

V tejto diplomovej práci sa zaoberáme problematikou zvyšovania rozlíšenia obrazu pomocou generatívnych neurónových sietí. Cieľom je rekonštruovať vysokorozlíšený obraz z jeho nízkorozlíšenej verzie, čo predstavuje nejednoznačný problém so stratou detailov. Tradičné metódy založené na interpolácii často nedokážu obnoviť jemné textúry a realistické štruktúry, preto sa využívajú prístupy založené na hlbokom učení. Teoretická časť poskytuje prehľad klasických metód super-rezolúcie, konvolučných neurónových sietí a moderných generatívnych modelov. Praktická časť sa zameriava na návrh vlastnej metódy pre zvýšenie rozlíšenia obrazu pomocou generatívnych neurónových sietí. Navrhnuté riešenie je implementované a testované na vybranej dátovej sade. Výsledky sú porovnané pomocou rôznych metrík a vizuálneho hodnotenia kvality s cieľom dosiahnuť vysokú perceptuálnu kvalitu pri zachovaní kontextu a textúr.

Kľúčové slová: *zvyšovanie rozlíšenia obrazu, generatívne neurónové siete, rekonštrukcia obrazu, SRGAN, ConvNeXt, perceptuálna strata*

Abstract

This thesis addresses the problem of image super-resolution using generative neural networks. The goal is to reconstruct a high-resolution image from its low-resolution version, which represents an ill-posed problem involving loss of details. Traditional interpolation-based methods often fail to recover fine textures and realistic structures; therefore, deep learning–based approaches are employed. The theoretical part provides an overview of classical super-resolution methods, convolutional neural networks, and modern generative models. The practical part focuses on designing a novel method for image super-resolution using generative neural networks. The proposed solution is implemented and tested on a selected dataset. The results are compared using various quantitative metrics and visual quality assessment with the aim of achieving high perceptual quality while preserving contextual information and textures.

Keywords: *image super-resolution, generative adversarial networks, image reconstruction, SRGAN, ConvNeXt, perceptual loss*

Obsah

Úvod **15**

1 Základné princípy zvýšenia rozlíšenia a rekonštrukcie obrazu **17**

1.1 Digitálny obraz a jeho základné vlastnosti 17

1.1.1 Reprezentácia digitálneho obrazu 17

1.1.2 Pixel a priestorové rozlíšenie 18

1.1.3 Farebná reprezentácia obrazu 20

1.2 Degradácia obrazu 21

1.2.1 Model degradácie obrazu 21

1.2.2 Typické formy degradácie obrazu 22

1.2.3 Zníženie rozlíšenia a aliasing 23

1.2.4 Syntetická a reálna degradácia 24

1.3 Zvýšenie rozlíšenia a rekonštrukcia obrazu 24

1.3.1 Základná úloha zvyšovania rozlíšenia 25

1.3.2 Klasifikácia metód super-resolution 25

1.3.3 Rekonštrukcia obrazu ako inverzný problém 27

1.4 Klasické interpolačné metódy 27

1.5 Hodnotenie kvality rekonštruovaného obrazu 28

1.5.1 Pixelové metriky 29

1.5.2 Štrukturálna a perceptuálna kvalita 30

2 Súčasný stav v oblasti hlbokého učenia pre super-resolution **31**

2.1 Hlboké učenie v spracovaní obrazu 31

2.1.1 Neurónová sieť ako parametrická funkcia 31

2.1.2 Trénovanie neurónových sietí 32

2.1.3 Optimalizácia neurónových sietí 33

2.1.4 Praktické aspekty trénovania 34

2.2 Konvolučné neurónové siete . . . 34
2.2.1 Konvolučná vrstva . . . 35
2.2.2 Aktivačné funkcie . . . 36
2.2.3 Normalizačné vrstvy . . . 37
2.2.4 Priestorové rozlíšenie a upsampling . . . 38
2.3 Metódy super-resolution založené na CNN . . . 39
2.3.1 SRCNN a FSRCNN . . . 39
2.3.2 Hlboké siete a reziduálne učenie . . . 41
2.4 Generatívne adversariálne siete . . . 43
2.4.1 Princíp GAN . . . 44
2.4.2 GAN prístupy pre super-resolution . . . 45
2.5 Súčasné trendy v oblasti super-resolution . . . 47
2.5.1 Vision Transformery a ďalšie prístupy . . . 48
2.5.2 ConvNeXt a modernizácia konvolučných sietí . . . 48
2.5.3 ConvNeXt bloky v architektúrach pre super-resolution . . . 50

3 Architektúra a analýza modelu SRGAN 51
3.1 Architektúra generátora . . . 51
3.2 Architektúra diskriminátora . . . 54
3.3 Stratová funkcia . . . 55
3.3.1 Pixelová strata . . . 55
3.3.2 Perceptuálna obsahová strata . . . 56
3.3.3 Adversariálna strata . . . 56
3.3.4 Kompozitná stratová funkcia generátora . . . 57
3.4 Trénovací postup . . . 57
3.4.1 Predtrénovanie generátora . . . 58
3.4.2 Adversariálne trénovanie . . . 58
3.4.3 Trénovacie dáta a augmentácia . . . 58
3.4.4 Vyhodnotenie . . . 59
3.5 Silné stránky a obmedzenia SRGAN . . . 59
3.5.1 Silné stránky . . . 59
3.5.2 Obmedzenia . . . 60

4 Návrh a implementácia upravenej metódy 62
4.1 Cieľ navrhovanej úpravy . . . 62

4.2 Navrhnutá architektúra ConvNeXt-SRGAN . . . 63
4.2.1 Bloková štruktúra generátora . . . 63
4.2.2 ConvNeXt blok a jeho adaptácia pre SR . . . 64
4.2.3 Architektúra diskriminátora . . . 65
4.2.4 Porovnanie s pôvodným generátorom SRGAN . . . 66
4.3 Implementačné detaily . . . 67
4.3.1 Framework a hardvér . . . 68
4.3.2 Príprava dát . . . 69
4.4 Nastavenie experimentov . . . 69
4.4.1 Fázy trénovania . . . 69
4.4.2 Hyperparametre, optimalizátor a stratové funkcie . . . 70
4.5 Experimentálne varianty . . . 72
4.5.1 Variant s redukovaným bottleneck rozšírením . . . 72
4.5.2 Varianty veľkosti dávky a predtrénovania . . . 72
4.5.3 Varianty počtu ConvNeXt blokov . . . 73

5 Výsledky a vyhodnotenie 75
5.1 Kvantitatívne výsledky . . . 75
5.1.1 Porovnanie hodnôt PSNR . . . 76
5.1.2 Porovnanie hodnôt SSIM . . . 77
5.1.3 Perceptuálna metrika LPIPS . . . 78
5.2 Výpočtová náročnosť . . . 79
5.3 Vizuálne porovnanie . . . 80
5.3.1 Porovnanie na vybraných testovacích obrazoch . . . 80
5.4 Zhrnutie vyhodnotenia . . . 85

6 Diskusia 87
6.1 Interpretácia výsledkov . . . 87
6.2 Obmedzenia práce . . . 87
6.3 Možné smery ďalšej práce . . . 88

Záver 90

Zoznam obrázkov

1.1 Model degradácie obrazu: HR obraz je postupne rozmazaný konvolúciou s jadrom k, znížený o faktor s a zaťažený šumom n, čím vzniká LR obraz [11]. 21
1.2 Rozdelenie metód super-resolution podľa rekonštrukcii [14]. 26
1.3 Porovnanie interpolačných metód a) Najbližší sused, b) Bilineárna, c) Bikubická, d) Originálny HR obrázok [17] 28

2.4 Porovnanie priebehu vybraných aktivačných funkcií ReLU, Leaky ReLU, PReLU a GELU 37
2.5 Architektúra SRCNN a FSRCNN [15] 41
2.6 Porovnanie konfigurácie blokov ConvNeXt, ResNet a Swin Transformer [33] 49

3.7 Architektúra generátora v modeli SRGAN [5]. 54
3.8 Architektúra diskriminátora v modeli SRGAN[5]. 55

4.9 Bloková schéma generátora ConvNeXt-SRGAN. 64
4.10 Vnútorná štruktúra ConvNeXt bloku v generátore. 65

5.11 Porovnanie hodnôt PSNR jednotlivých modelov na 100 validačných obrázkoch datasetu DIV2K pri škálovacom faktore $r = 4$. 77
5.12 Porovnanie hodnôt SSIM jednotlivých modelov na 100 validačných obrázkoch datasetu DIV2K pri škálovacom faktore $r = 4$. 78
5.13 Porovnanie metód na obraze č. 7 z validačnej množiny DIV2K. Zľava: SRGAN, ConvNeXt-SRGAN, referenčný HR obraz. 81
5.14 Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na scéne s horským pozadím. 81
5.15 Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 68. . . . 82
5.16 Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 34. . . . 83

5.17 Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 13. . . . 83
5.18 Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 27. . . . 84
5.19 Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 57. . . . 84
5.20 Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 58 85

Zoznam tabuliek

2.1 Prehľad vybraných metód super-resolution založených na hlbokom učení. 47

4.1 Porovnanie reziduálneho bloku SRGAN a ConvNeXt bloku. 66
4.2 Kľúčové hyperparametre trénovania ConvNeXt-SRGAN. 71
4.3 Varianty konfigurácie ConvNeXt-SRGAN. 74

5.1 Porovnanie metód na validačnej množine DIV2K (100 obrazov, škálovací faktor $r = 4$). Delta Δ vyjadruje rozdiel oproti bikubickej interpolácii. Hlavný model je zvýraznený. 76
5.2 Porovnanie metód na validačnej množine DIV2K (100 obrazov, škálovací faktor $r = 4$). Pri metrike LPIPS nižšia hodnota znamená vyššiu perceptuálnu podobnosť. 79
5.3 Porovnanie výpočtovej náročnosti. 80

6.1 Porovnanie navrhnutej metódy s vybranými publikovanými prácami v oblasti super-resolution. 89

Úvod

V tejto práci sa venujeme zvýšeniu rozlíšenia a rekonštrukcii obrazu pomocou generatívnych neurónových sietí. V digitálnom svete, kde kvalita obrazu hrá kľúčovú rolu vo vede, medicíne, bezpečnosti aj vo videohernom priemysle, sa zvyšovanie rozlíšenia a rekonštrukcia obrazu stávajú čoraz dôležitejšími úlohami [1, 2]. Či už ide o obnovu poškodených historických fotografií, zlepšenie záberov z bezpečnostných kamier, zvýšenie kvality medicínskych snímok (napríklad MRI alebo CT), ktoré môže pomôcť pri diagnostike, alebo o satelitné snímky pre detailnejšiu analýzu terénu, schopnosť transformovať nekvalitný alebo rozmazaný obraz na detailnejší a ostrejší je vysoko žiadaná [1, 3].

Tradičné prístupy k zvýšeniu rozlíšenia sa opierajú o interpoláciu a deterministické modely, ktoré však často nedokážu zachytiť jemné textúry ani realistické detaily. V posledných rokoch sa preto do popredia dostávajú generatívne neurónové siete (GAN, z anglického Generative Adversarial Networks) [4], ktoré vďaka svojej schopnosti generovať detailné obrazy prinášajú pozoruhodné výsledky v oblasti generatívneho spracovania obrazu [5].

Hlavným cieľom tejto práce je poskytnúť komplexný pohľad na problematiku zvýšenia rozlíšenia obrazu. Ďalším cieľom je navrhnúť novú metódu na zvýšenie rozlíšenia a rekonštrukciu obrazu pomocou neurónových sietí. Posledným cieľom je túto metódu implementovať a porovnať dosiahnuté výsledky na vybranej dátovej sade.

Hlavným prínosom práce je modifikácia architektúry SRGAN využitím ConvNeXt blokov a experimentálne overenie ich vplyvu na kvalitu rekonštrukcie. Zámerom nie je len dosiahnuť vyššiu kvalitu výstupných obrazov z pohľadu tradičných metrík, ako sú PSNR alebo MSE, ktoré často nekorelujú s ľudským vnímaním [6], ale predovšetkým dosiahnuť vizuálne realistické výsledky, ktoré zodpovedajú ľudskému vnímaniu detailov a štruktúr [5, 7].

Práca je členená do šiestich kapitol. Prvá kapitola sa venuje základným princípom zvyšovania rozlíšenia a rekonštrukcie obrazu, reprezentácii digitálneho obrazu,

modelu degradácie a metrikám hodnotenia kvality. Druhá kapitola poskytuje prehľad súčasného stavu v oblasti hlbokého učenia pre super-resolution, od konvolučných sietí cez generatívne adversariálne siete až po moderné architektúry typu ConvNeXt. Tretia kapitola podrobne analyzuje architektúru a trénovací postup modelu SRGAN, ktorý tvorí základ navrhovanej úpravy. Štvrtá kapitola opisuje návrh a implementáciu modelu ConvNeXt-SRGAN vrátane experimentálnych variantov. Piata kapitola prezentuje kvantitatívne a vizuálne výsledky spolu s analýzou výpočtovej náročnosti. Šiesta kapitola výsledky interpretuje, rozoberá obmedzenia práce a načrtáva možné smery ďalšieho výskumu.

Kapitola 1

Základné princípy zvýšenia rozlíšenia a rekonštrukcie obrazu

V tejto kapitole sú predstavené základné princípy zvýšenia rozlíšenia a rekonštrukcie obrazu, teda pojmy súvisiace s digitálnym obrazom, jeho vznikom a reprezentáciou v počítači. Pochopenie týchto vlastností je dôležité pre ďalší opis degradácie obrazu, znižovania rozlíšenia a metód zvýšenia rozlíšenia (super-resolution).

1.1 Digitálny obraz a jeho základné vlastnosti

Digitálny obraz je diskrétnou reprezentáciou spojitého dvojrozmerného signálu. Definícia obrazu sa môže líšiť v závislosti od oblasti použitia, pričom jednotlivé disciplíny používajú definície zodpovedajúce ich potrebám. V kontexte spracovania obrazu a počítačového videnia možno obraz chápať ako špecifický prípad dvojrozmerného signálu, ktorého hodnota sa mení v závislosti od dvoch priestorových súradníc [2, 8].

1.1.1 Reprezentácia digitálneho obrazu

Obraz možno matematicky opísať ako dvojrozmernú funkciu

$$f(x, y), \tag{1.1}$$

kde:

- x a y predstavujú priestorové súradnice.
- hodnota funkcie $f(x, y)$ vyjadruje intenzitu alebo farebnú informáciu v danom bode.

Ak sú priestorové súradnice aj hodnoty obrazovej funkcie konečné a diskrétne, ide o digitálny obraz [9]. V digitálnom obraze sú súradnice bodov spravidla celočíselné. Počiatok obrazového súradnicového systému sa zvyčajne nachádza v ľavom hornom rohu obrazu, pričom súradnica x rastie smerom doprava a súradnica y smerom nadol [3].

V najjednoduchšom prípade obraz obsahuje jeden kanál, v ktorom každá hodnota reprezentuje jas, resp. odtieň sivej príslušného obrazového bodu od čiernej po bielu. Obraz s výškou H a šírkou W tak možno zapísať ako

$$I \in \mathbb{R}^{H \times W}, \tag{1.2}$$

kde jednotlivé prvky matice zodpovedajú hodnotám pixelov v príslušných súradniciach obrazu. Takáto reprezentácia je základom pre ďalšie digitálne spracovanie obrazu, keďže umožňuje nad obrazom vykonávať matematické operácie, napríklad filtráciu, transformácie alebo zmenu rozlíšenia [9, 10].

1.1.2 Pixel a priestorové rozlíšenie

Pixel

Základným prvkom digitálneho obrazu je obrazový bod, označovaný ako pixel. Názov pixel vznikol zo slovného spojenia *picture element*. Pixely sú usporiadané do pravidelnej dvojrozmernej mriežky, pričom každý pixel reprezentuje intenzitu alebo farebnú informáciu malej oblasti pôvodnej scény [9, 10].

Vzorkovanie a kvantovanie

Aby bolo možné obraz spracovať napríklad v počítači, musí sa spojitá obrazová informácia previesť do digitálnej podoby. Tento proces sa nazýva digitalizácia. Pri digitalizácii sa obraz najprv rozdelí na konečný počet obrazových bodov, teda pixelov. Tento krok sa označuje ako vzorkovanie. Následne sa každej zaznamenanej hodnote priradí konkrétna číselná hodnota z obmedzeného rozsahu. Tento krok sa označuje ako kvantovanie [9, 10].

Priestorové rozlíšenie

Priestorové rozmery obrazu v počítači sú určené počtom pixelov vo vertikálnom a horizontálnom smere. Celkový počet pixelov možno vypočítať ako

$$N = H \cdot W, \tag{1.3}$$

kde:

- H predstavuje výšku obrazu.
- W predstavuje šírku obrazu.

Priestorové rozlíšenie vyjadruje mieru detailov, ktoré je možné v obraze reprezentovať. Obraz s väčším počtom pixelov môže obsahovať podrobnejšiu reprezentáciu danej scény, pretože tú istú oblasť opisuje väčší počet pixelov. Samotný počet pixelov však nie je jediným faktorom ovplyvňujúcim kvalitu obrazu. Dôležité sú aj vlastnosti optickej sústavy, snímača, úroveň šumu, zaostrenie a ďalšie spôsoby spracovania obrazu [9, 1].

Bitová hĺbka

Ďalšou dôležitou vlastnosťou digitálneho obrazu je bitová hĺbka. Tá určuje počet bitov použitých na reprezentáciu jednej hodnoty pixelu alebo jedného farebného kanála. Ak sa na reprezentáciu použije b bitov, počet dostupných hodnôt je

$$L = 2^b. \tag{1.4}$$

Pri 8-bitovom monochromatickom obraze môže jeden pixel nadobúdať 256 hodnôt v rozsahu od 0 (čierna) do 255 (biela). Takýto obraz sa označuje ako obraz v odtieňoch sivej (*grayscale image*) [9].

Hodnoty digitálneho obrazu sú pri uložení často celočíselné, no pri trénovaní neurónových sietí sa zvyčajne normalizujú. Hodnotu pixelu p z 8-bitového rozsahu možno previesť do intervalu $[0, 1]$ podľa vzťahu

$$p_{\text{norm}} = \frac{p}{255}. \tag{1.5}$$

Normalizácia udržiava vstupné hodnoty v rozsahu vhodnom pre numerické výpočty a môže prispieť k stabilnejšiemu priebehu optimalizácie neurónových sietí.

1.1.3 Farebná reprezentácia obrazu

RGB

Na rozdiel od monochromatického obrazu obsahuje farebný obraz viacero kanálov, ktorých kombinácia určuje výslednú farbu jednotlivých pixelov. Najpoužívanejším spôsobom reprezentácie farebných obrazov je farebný model RGB (*Red, Green, Blue*), ktorý využíva tri farebné zložky: červenú, zelenú a modrú.

RGB je aditívny farebný model, v ktorom sa výsledná farba vytvára kombináciou intenzít jednotlivých farebných kanálov. Pri nulovej hodnote všetkých troch kanálov vzniká čierna farba, zatiaľ čo maximálne hodnoty všetkých zložiek vytvárajú bielu farbu [9, 3].

Z matematického hľadiska možno farebný obraz reprezentovať ako viacrozmerné pole hodnôt, ktoré sa v kontexte hlbokého učenia často označuje ako tenzor. Farebný obraz možno zapísať ako

$$I \in \mathbb{R}^{H \times W \times C}, \tag{1.6}$$

kde:

- H predstavuje výšku obrazu.
- W predstavuje jeho šírku.
- C počet farebných kanálov.

V prípade farebného modelu RGB platí $C = 3$, pretože obraz pozostáva z červeného, zeleného a modrého kanála. Pri 8-bitovom farebnom obraze RGB je každý z troch farebných kanálov reprezentovaný samostatnou hodnotou z rozsahu od 0 do 255. Jeden pixel je tak reprezentovaný 24 bitmi a môže nadobúdať viac ako 16 miliónov farebných kombinácií ($2^{24} = 16\,777\,216$) [9, 10].

YCbCr

Okrem RGB sa pri spracovaní obrazu často používa aj farebný priestor YCbCr. Na rozdiel od RGB oddeľuje jasovú informáciu od farebnej informácie. Zložka Y reprezentuje jas obrazu, zatiaľ čo zložky Cb a Cr nesú informáciu o farebných rozdieloch. V mnohých aplikáciách spracovania obrazu sa osobitná pozornosť venuje práve jasovej zložke Y, pretože obsahuje významnú časť informácií o hranách, štruktúrach a detailoch obrazu [3].

1.2 Degradácia obrazu

Degradácia obrazu označuje proces, pri ktorom dochádza k zníženiu kvality obrazu alebo k strate časti obrazovej informácie. Môže byť spôsobená viacerými faktormi, napríklad rozmazaním, šumom, kompresiou, nedokonalosťami optickej sústavy, pohybom fotoaparátu alebo samotným znižovaním rozlíšenia [9, 10].

Obraz s vysokým rozlíšením sa označuje skratkou HR (*high resolution*), obraz s nízkym rozlíšením skratkou LR (*low resolution*). HR obraz slúži ako referencia, označovaná aj ako *ground truth*.

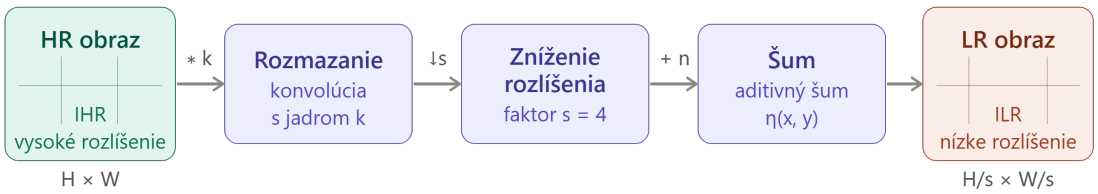
1.2.1 Model degradácie obrazu

Pri úlohách zvyšovania rozlíšenia sa často predpokladá, že LR obraz vznikol degradáciou pôvodného HR obrazu. Tento proces možno zjednodušene zapísať ako

$$I_{\mathrm{LR}} = (I_{\mathrm{HR}} * k) \downarrow_s + n, \tag{1.7}$$

kde:

- I_{HR} označuje obraz s vysokým rozlíšením,
- I_{LR} je obraz s nízkym rozlíšením,
- k je rozmazávacie jadro,
- $*$ označuje konvolúciu,
- $\downarrow_s$ predstavuje operáciu znižovania rozlíšenia so škálovacím faktorom s
- n je pridaný šum.



Obr. 1.1: Model degradácie obrazu: HR obraz je postupne rozmazaný konvolúciou s jadrom k, znížený o faktor s a zaťažený šumom n, čím vzniká LR obraz [11].

Tento model vyjadruje, že LR obraz nevzniká iba jednoduchým odstránením pixelov. Pred zmenšením môže byť obraz rozmazaný a počas snímania, prenosu alebo ukladania sa do neho môže pridať šum alebo iné možné artefakty [1, 11].

Dôsledkom degradácie je čiastočná strata obrazovej informácie, čo zvyšovanie rozlíšenia robí nejednoznačným problémom.

1.2.2 Typické formy degradácie obrazu

Degradácia obrazu môže mať viacero podôb. Medzi najčastejšie patria rozmazanie, šum a kompresné artefakty. Tieto javy môžu v obraze potláčať jemné detaily, meniť hodnoty pixelov alebo vytvárať dodatočné vizuálne skreslenia, ktoré sťažujú následnú rekonštrukciu obrazu.

Rozmazanie obrazu

Rozmazanie obrazu predstavuje jednu zo základných foriem degradácie, pri ktorej dochádza k potlačeniu jemných detailov a oslabeniu ostrých hrán. Rozmazaný obraz pôsobí menej ostro a hranice medzi objektmi, teda hrany sú v ňom horšie rozlíšiteľné. Tento jav môže vzniknúť napríklad nesprávnym zaostrením, pohybom fotoaparátu, pohybom snímaného objektu alebo aj spomínanými vlastnosťami optickej sústavy [9, 10].

V testovacích podmienkach možno rozmazanie aj umelo simulovať pomocou vhodne zvoleného rozmazávacieho jadra. Z matematického hľadiska sa rozmazanie často modeluje ako konvolúcia pôvodného obrazu s rozmazávacím jadrom. Tento vzťah možno zapísať ako

$$g(x, y) = k(x, y) * f(x, y), \tag{1.8}$$

kde:

- $f(x, y)$ predstavuje pôvodný obraz.
- $k(x, y)$ predstavuje rozmazávacie jadro.
- $g(x, y)$ je výsledný rozmazaný obraz.

Rozmazávacie jadro určuje, akým spôsobom sa hodnota jedného obrazového bodu rozloží do jeho okolia. V oblasti optického zobrazovania sa podobný pojem označuje aj ako bodová rozptylová funkcia, z anglického *point spread function* [3, 9].

Jedným z často používaných typov rozmazania je Gaussovo rozmazanie. Pri ňom sa hodnota pixelu nahrádza váženou kombináciou hodnôt z jeho okolia, pričom väčšiu váhu majú pixely bližšie k spracovávanému bodu. Gaussovo rozmazanie sa využíva aj ako vyhladzovací filter pred znižovaním rozlíšenia [3, 9].

Šum

Šum v obraze predstavuje nežiaducu zložku, ktorá mení pôvodné alebo očakávané hodnoty pixelov. Často vzniká napríklad pri nedostatočnom osvetlení, pri vysokej citlivosti snímača, v dôsledku elektronického šumu alebo počas prenosu obrazu [9, 10]. Jednoduchý model šumu možno zapísať ako

$$g(x, y) = f(x, y) + \eta(x, y), \tag{1.9}$$

kde:

- $f(x, y)$ predstavuje pôvodný obraz.
- $\eta(x, y)$ predstavuje šumovú zložku.
- $g(x, y)$ je výsledný zašumený obraz.

Tento model predpokladá, že šum sa k pôvodnej hodnote pixelu pripočíta.

Kompresné artefakty

Okrem šumu môžu kvalitu obrazu znižovať aj kompresné artefakty. Tie vznikajú najmä pri stratovej kompresii, ktorej cieľom je zmenšiť veľkosť obrazových dát za cenu čiastočnej straty informácie. Výsledkom môžu byť blokové artefakty, nepresnosti v okolí hrán alebo strata jemných textúr. Pri následnej rekonštrukcii môže byť náročné rozlíšiť, či určitá lokálna zmena predstavuje skutočnú obrazovú štruktúru alebo iba poruchu spôsobenú degradáciou. [9, 10].

1.2.3 Zníženie rozlíšenia a aliasing

Zníženie rozlíšenia

Znižovanie rozlíšenia, označované aj ako *downsampling*, je proces, pri ktorom sa znižuje počet pixelov obrazu. Ak má pôvodný obraz rozmery $H_{\mathrm{HR}} \times W_{\mathrm{HR}}$ a použije sa škálovací faktor s, rozmery výsledného obrazu s nízkym rozlíšením možno vyjadriť ako

$$H_{\mathrm{LR}} = \frac{H_{\mathrm{HR}}}{s}, \qquad W_{\mathrm{LR}} = \frac{W_{\mathrm{HR}}}{s}. \tag{1.10}$$

Pri znižovaní rozlíšenia dochádza k redukcii počtu pixelov. To znamená, že výsledný obraz obsahuje menej informácií o pôvodnej scéne.

Aliasing

Pri zmenšovaní obrazu môže vzniknúť aj aliasing. Ide o jav, pri ktorom nové nižšie rozlíšenie nedokáže správne reprezentovať jemné štruktúry pôvodného obrazu. Vizuálne sa môže prejaviť napríklad ako zubaté hrany, falošné vzory alebo moaré efekt. Aby sa aliasing zmiernil, pred samotným znížením rozlíšenia sa často používa vyhladenie obrazu. [3, 8].

1.2.4 Syntetická a reálna degradácia

Pri syntetickej degradácii sa obraz s nízkym rozlíšením vytvorí z referenčného obrazu s vysokým rozlíšením pomocou vopred známeho postupu, napríklad zmenšením obrazu. Výhodou je, že ku každému obrazu s nízkym rozlíšením existuje zodpovedajúci referenčný obraz s vysokým rozlíšením. Takéto dvojice obrazov sú vhodné na trénovanie a vyhodnocovanie metód pomocou metrík [1, 11].

Reálna degradácia je zložitejšia, pretože presný proces vzniku obrazu s nízkym rozlíšením zvyčajne nie je známy. Obraz môže byť ovplyvnený kombináciou viacerých faktorov, napríklad rozmazaním, šumom, stratovou kompresiou, neostrým zaostrením alebo neznámym spôsobom znižovania rozlíšenia. Tieto degradácie sa môžu navzájom kombinovať a ich parametre nemusia byť dostupné [11].

Rozdiel medzi syntetickou a reálnou degradáciou je dôležitý aj pri interpretácii výsledkov. Metóda, ktorá dosahuje dobré výsledky na synteticky vytvorených dátach, nemusí rovnako dobre fungovať na reálnych obrazoch, pretože tie môžu obsahovať zložitejšie a menej predvídateľné artefakty [11, 12].

1.3 Zvýšenie rozlíšenia a rekonštrukcia obrazu

Zvýšenie rozlíšenia obrazu, označované ako *super-resolution* (SR), predstavuje úlohu, ktorej cieľom je vytvoriť obraz s vyšším rozlíšením zo vstupného obrazu s nižším rozlíšením. Táto úloha priamo súvisí s degradačným modelom opísaným v predchá-

dzajúcej časti a jeho cieľom SR je vytvoriť čo najlepší odhad pôvodného HR obrazu zo vstupného LR obrazu [1, 11].

1.3.1 Základná úloha zvyšovania rozlíšenia

Formálne možno túto úlohu zapísať ako hľadanie zobrazenia, ktoré minimalizuje rozdiel medzi výstupom modelu a originálnym referenčným HR obrazom. Tento vzťah možno zapísať ako

$$I_{\mathrm{SR}} = F(I_{\mathrm{LR}}), \tag{1.11}$$

kde

- I_{LR} predstavuje vstupný obraz s nízkym rozlíšením.
- F označuje metódu alebo model zvyšovania rozlíšenia.
- I_{SR} je výsledný obraz so zvýšeným rozlíšením.

Cieľom je, aby sa I_{SR} čo najviac približoval referenčnému obrazu I_{HR}, pričom formálne $I_{\mathrm{SR}} \approx I_{\mathrm{HR}}$.

Pri zvyšovaní rozlíšenia sa často používa škálovací faktor s, ktorý určuje, koľkokrát sa majú zväčšiť priestorové rozmery obrazu. Ak má vstupný obraz rozmery $H_{\mathrm{LR}} \times W_{\mathrm{LR}}$, potom rozmery výsledného obrazu možno zapísať ako

$$H_{\mathrm{SR}} = s \cdot H_{\mathrm{LR}}, \qquad W_{\mathrm{SR}} = s \cdot W_{\mathrm{LR}}. \tag{1.12}$$

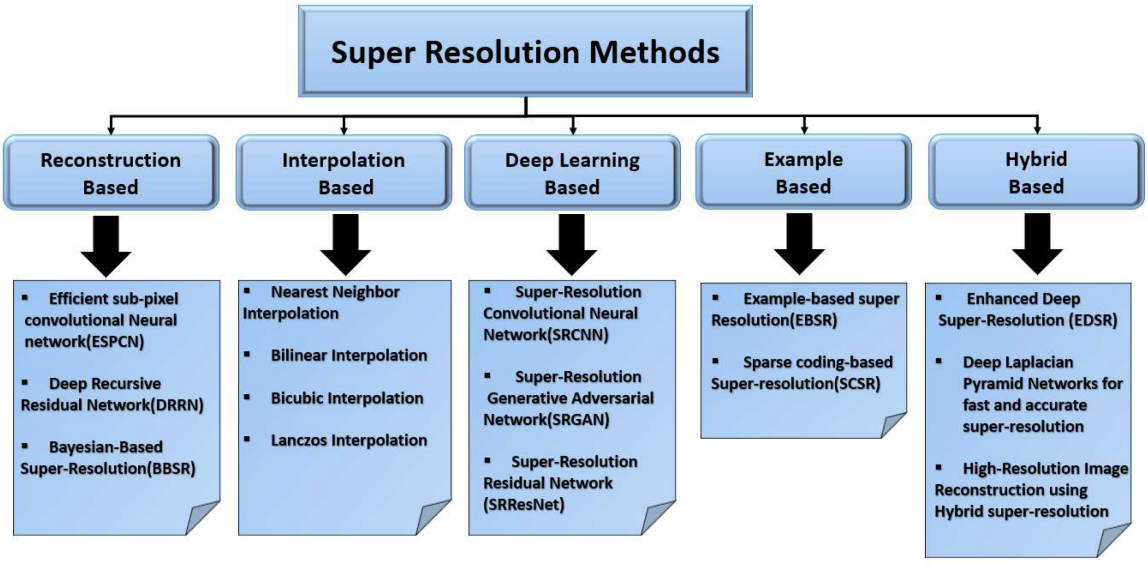
Úloha zvyšovania rozlíšenia teda nespočíva iba v geometrickom zväčšení obrazu, ale aj v rekonštrukcii detailov obrazu, ktoré pri degradácii zanikli [1, 11].

1.3.2 Klasifikácia metód super-resolution

Metódy super-resolution možno klasifikovať viacerými spôsobmi. Jednotlivé kategórie sa líšia typom vstupných dát, spôsobom spracovania obrazu aj prístupom k rekonštrukcii chýbajúcich detailov. Z hľadiska použitého prístupu k rekonštrukcii rozlišujeme metódy na:

- **Interpolačné metódy** (Interpolation Based) využívajú matematické vzťahy medzi hodnotami susedných pixelov na odhad nových hodnôt. Sú výpočtovo nenáročné, no nedokážu obnoviť jemné detaily stratené pri degradácii [9].

- **Rekonštrukčné metódy** (Reconstruction Based) formulujú zvyšovanie rozlíšenia ako optimalizačný problém s využitím apriórnych znalostí o degradačnom procese. Dosahujú lepšie výsledky ako interpolácia, sú však náročnejšie na výpočet [1].
- **Metódy založené na príkladoch** (Example Based) sa učia vzťah medzi LR a HR obrazmi z trénovacích dvojíc, napríklad pomocou riedkeho kódovania (SCSR). Dokážu generovať realistickejšie detaily ako interpolačné metódy [1].
- **Metódy hlbokého učenia** (Deep Learning Based) v súčasnosti dosahujú najlepšie výsledky. Hlboké neurónové siete sa naučia mapovanie medzi LR a HR obrazmi priamo z dát [5].
- **Hybridné metódy** (Hybrid Based) kombinujú viaceré prístupy, napríklad reziduálne učenie s pozornostným mechanizmom alebo konvolučné a transformerové architektúry [13].



Obr. 1.2: Rozdelenie metód super-resolution podľa rekonštrukcii [14].

Táto práca sa zameriava na metódy *single-image super-resolution* (SISR), pri ktorých neurónová sieť pracuje iba s jedným vstupným LR obrazom. Z uvedených kategórií sa práca opiera o interpolačné metódy ako referenčný základ a o metódy hlbokého učenia.

1.3.3 Rekonštrukcia obrazu ako inverzný problém

Rekonštrukcia obrazu sa vo všeobecnosti zaoberá obnovou obrazu zo znehodnoteného alebo neúplného vstupu. Super-resolution tak možno chápať aj ako špeciálny prípad rekonštrukcie obrazu, pri ktorom sa obnovuje najmä priestorové rozlíšenie a obrazové detaily [1, 11].

Keďže degradačný proces spôsobuje stratu časti obrazovej informácie, jeho spätné obrátenie nie je jednoznačné. K jednému obrazu s nízkym rozlíšením môže existovať viacero možných obrazov s vysokým rozlíšením, ktoré by po degradácii viedli k rovnakému alebo veľmi podobnému výsledku. Z tohto dôvodu sa zvyšovanie rozlíšenia označuje ako nejednoznačný, resp. *ill-posed*, inverzný problém [1].

Cieľom metódy zvyšovania rozlíšenia teda nie je nájsť presnú inverznú funkciu degradácie, ale vytvoriť čo najlepší odhad obrazu s vysokým rozlíšením na základe informácií dostupných vo vstupnom LR obraze. Formálne tento odhad zodpovedá vzťahu (1.11), pričom výstup I_{SR} možno chápať ako aproximáciu pôvodného HR obrazu, teda $I_{\mathrm{SR}} \approx I_{\mathrm{HR}}$.

1.4 Klasické interpolačné metódy

Pred rozšírením metód založených na hlbokom učení sa na zvyšovanie rozlíšenia obrazu používali najmä klasické interpolačné metódy. Ich cieľom je vytvoriť nový obraz s väčším počtom pixelov na základe hodnôt existujúcich pixelov vo vstupnom obraze [8, 9].

Medzi najčastejšie používané klasické metódy patria:

- **Interpolácia najbližšieho suseda:** ide o najjednoduchšiu metódu, pri ktorej sa hodnota nového pixelu určí podľa najbližšieho pixelu v pôvodnom obraze. Výhodou je nízka výpočtová náročnosť, no výsledný obraz môže pôsobiť blokovito a pri väčšom zväčšení sú viditeľné zubaté hrany [3, 5].

- **Bilineárna interpolácia:** hodnota nového pixelu sa vypočíta ako vážený priemer štyroch najbližších pixelov. V porovnaní s metódou najbližšieho suseda vytvára plynulejší výsledok, ale môže spôsobovať rozmazanie hrán a jemných detailov [8].

- **Bikubická interpolácia:** pri výpočte hodnoty nového pixelu využíva väčšie okolie, zvyčajne 4×4 pixelov. Vďaka tomu poskytuje kvalitnejšie a plynulejšie

výsledky ako bilineárna interpolácia. Bikubická interpolácia sa často používa ako jednoduchý referenčný základ pri porovnávaní metód zvyšovania rozlíšenia. Taktiež sa často používa aj pri syntetickom znižovaní rozlíšenia, napríklad pri tvorbe trénovacích dvojíc LR-HR v úlohách super-resolution. [7, 15].

- **Lanczosova interpolácia:** využíva interpolačné jadro založené na funkcii *sínus*. V porovnaní s jednoduchšími metódami môže lepšie zachovávať ostrosť hrán, no je výpočtovo náročnejšia a v niektorých prípadoch môže vytvárať artefakty v okolí výrazných prechodov [8, 16].



Obr. 1.3: Porovnanie interpolačných metód a) Najbližší sused, b) Bilineárna, c) Bikubická, d) Originálny HR obrázok [17]

Hoci sú tieto metódy jednoduché a rýchle, ich hlavné obmedzenie spočíva v tom, že nové hodnoty pixelov určujú iba na základe lokálneho okolia v pôvodnom obraze. Nevytvárajú nové obrazové detaily, ale iba odhadujú hodnoty medzi existujúcimi pixelmi. Preto pri vyšších faktoroch zväčšenia často vzniká rozmazanie, zubaté hrany alebo strata jemných textúr. Tieto obmedzenia priamo motivovali rozvoj metód hlbokého učenia, ktoré sa učia vzťah medzi LR a HR obrazmi z dát.

1.5 Hodnotenie kvality rekonštruovaného obrazu

Hodnotenie kvality výsledného obrazu je dôležitou súčasťou úloh zvyšovania rozlíšenia. Cieľom hodnotenia je určiť, do akej miery sa rekonštruovaný obraz podobá originálnemu referenčnému obrazu s vysokým rozlíšením. Pri experimentoch, v ktorých sú dostupné dvojice obrazov LR a HR, možno kvalitu výstupu hodnotiť pomocou kvantitatívnych metrík. Tieto metriky umožňujú porovnať rôzne metódy za rovna-

kých podmienok a objektívne vyjadriť rozdiel medzi rekonštruovaným a referenčným obrazom [3, 18].

1.5.1 Pixelové metriky

MSE

Jednou zo základných metrík používaných pri hodnotení rekonštruovaného obrazu je stredná kvadratická chyba, označovaná ako MSE [18] z anglického *mean squared error*. MSE vyjadruje priemernú štvorcovú odchýlku medzi hodnotami pixelov referenčného obrazu a rekonštruovaného obrazu. Pre obraz s rozmermi $H \times W$ možno MSE zapísať ako

$$\mathrm{MSE} = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(I_{\mathrm{HR}}(i,j) - I_{\mathrm{SR}}(i,j) \right)^2 , \tag{1.13}$$

kde

- $H \times W$ sú rozmery obrazu (výška a šírka v pixeloch),
- i, j sú indexy riadku a stĺpca pixelu,
- $I_{\mathrm{HR}}(i,j)$ je hodnota pixelu referenčného obrazu s vysokým rozlíšením na pozícii (i,j),
- $I_{\mathrm{SR}}(i,j)$ je hodnota pixelu rekonštruovaného obrazu na pozícii (i,j).

Nižšia hodnota MSE znamená menší rozdiel medzi porovnávanými obrazmi.

PSNR

Na základe MSE sa často počíta metrika PSNR [18], z anglického *peak signal-to-noise ratio*. PSNR vyjadruje pomer medzi maximálnou možnou hodnotou pixelu a chybou rekonštrukcie. Pre obraz s maximálnou hodnotou pixelu MAX_I možno PSNR zapísať ako

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{MAX_I^2}{\mathrm{MSE}} \right). \tag{1.14}$$

kde MAX_I je maximálna možná hodnota pixelu a MSE je stredná kvadratická chyba definovaná vzťahom (1.13). Pri 8-bitovom obraze je hodnota $MAX_I = 255$. Ak sú hodnoty pixelov normalizované do intervalu $[0, 1]$, potom platí $MAX_I = 1$.

Vyššia hodnota PSNR zvyčajne znamená menší rozdiel medzi rekonštruovaným a referenčným obrazom. PSNR je jednoduchá a často používaná metrika, no hodnotí najmä pixelovú podobnosť a nemusí vždy zodpovedať tomu, ako kvalitu obrazu vníma človek.

1.5.2 Štrukturálna a perceptuálna kvalita

SSIM

Keďže pixelové metriky nemusia dostatočne vystihovať vizuálnu kvalitu obrazu, často sa používa aj metrika SSIM, z anglického *structural similarity index*. SSIM neporovnáva obrazy iba na úrovni jednotlivých pixelov, ale zohľadňuje aj podobnosť jasu, kontrastu a štruktúry obrazu [18].

Hodnota SSIM sa zvyčajne nachádza v intervale od 0 do 1, pričom hodnota bližšia k 1 znamená vyššiu podobnosť medzi referenčným a rekonštruovaným obrazom. V porovnaní s PSNR dokáže SSIM lepšie zachytiť niektoré štrukturálne rozdiely, ktoré sú dôležité pre ľudské vnímanie obrazu [18].

Perceptuálne hodnotenie

Na hodnotenie percepčnej podobnosti slúžia metriky pracujúce v priestore príznakov neurónových sietí, napríklad LPIPS (*learned perceptual image patch similarity*). LPIPS meria vzdialenosť medzi príznakmi extrahovanými z vrstiev predtrénovanej neurónovej siete, napríklad VGG. Nižšia hodnota LPIPS znamená vyššiu perceptuálnu podobnosť medzi obrazmi [19].

Ďalšou metrikou používanou na hodnotenie perceptuálnej kvality je MOS, z anglického *mean opinion score*. Na rozdiel od predchádzajúcich metrík nejde o automatickú výpočtovú metriku, ale o subjektívne hodnotenie kvality obrazu skupinou ľudských hodnotiteľov. Výsledná hodnota MOS predstavuje priemer ich hodnotení na stupnici, zvyčajne od 1 do 5, pričom vyššia hodnota znamená lepšiu vnímanú kvalitu [5].

Generatívne metódy môžu vytvárať vizuálne presvedčivejšie detaily, no tieto detaily nemusia presne zodpovedať referenčnému HR obrazu. V takom prípade môže byť výsledok pre človeka vizuálne kvalitnejší, aj keď jeho PSNR alebo MSE nie sú najlepšie. Tento rozdiel medzi pixelovou presnosťou a vizuálnou kvalitou je jedným z dôležitých problémov pri hodnotení metód super-resolution a hľadanie rovnováhy medzi nimi predstavuje jednu z hlavných motivácií tejto práce [6].

Kapitola 2

Súčasný stav v oblasti hlbokého učenia pre super-resolution

Metódy založené na hlbokom učení sa v posledných rokoch stali jedným z hlavných prístupov pri riešení úloh spracovania obrazu. V porovnaní s klasickými metódami, ktoré sú často založené na pevne definovaných matematických pravidlách, umožňujú modely hlbokého učenia učiť sa vlastnosti obrazových dát priamo z trénovacích príkladov. V oblasti super-resolution je táto vlastnosť dôležitá najmä preto, že vzťah medzi obrazom s nízkym rozlíšením a jeho zodpovedajúcou vysokorozlíšenou podobou je zložitý a prakticky nejednoznačný.

2.1 Hlboké učenie v spracovaní obrazu

Hlboké učenie predstavuje oblasť strojového učenia, ktorá využíva neurónové siete s viacerými vrstvami na učenie zložitých reprezentácií dát. Model sa neučí iba priamo mapovať vstup na výstup, ale postupne si vytvára viacúrovňovú reprezentáciu vstupných dát kde nižšie vrstvy zachytávajú jednoduchšie vlastnosti, ako sú hrany alebo lokálne kontrastné zmeny, zatiaľ čo hlbšie vrstvy reprezentujú zložitejšie štruktúry a vzory [20, 21].

2.1.1 Neurónová sieť ako parametrická funkcia

Neurónovú sieť možno chápať ako parametrickú funkciu, ktorá transformuje vstupné dáta na požadovaný výstup [2, 21]. V prípade spracovania obrazu môže byť vstupom napríklad digitálny obraz alebo jeho časť a výstupom môže byť trieda objektu, odšumený obraz alebo obraz so zvýšeným rozlíšením. Takúto funkciu možno všeobecne

zapísať ako

$$\hat{y} = f_\theta(x), \tag{2.1}$$

kde

- x predstavuje vstup,
- $\hat{y}$ výstup modelu,
- f_θ parametrickú funkciu neurónovej siete,
- θ množinu trénovateľných parametrov siete.

Medzi tieto parametre patria najmä váhy a biasy (prahové jednotky) jednotlivých vrstiev. Počas trénovania sa tieto parametre postupne upravujú tak, aby výstup siete čo najlepšie zodpovedal očakávanému výsledku [20, 2].

Sieť je zložená z vrstiev, ktoré postupne spracúvajú vstupné dáta. V hlbokých sieťach tak vzniká postupnosť reprezentácií, v ktorej sa jednoduché lokálne vlastnosti postupne skladajú do zložitejších obrazových štruktúr. Tento princíp je dôležitý najmä pri obrazových úlohách, pretože obraz obsahuje priestorové vzťahy medzi pixelmi, hranami, textúrami a objektmi [21, 3].

2.1.2 Trénovanie neurónových sietí

Trénovanie neurónovej siete spočíva v hľadaní parametrov minimalizujúcich chybu na trénovacích dátach. Táto chyba sa vyjadruje pomocou stratovej funkcie, ktorá porovnáva výstup modelu s požadovaným výstupom:

$$\theta^* = \arg\min_\theta \mathcal{L}(f_\theta(x), y), \tag{2.2}$$

kde

- $\mathcal{L}$ označuje stratovú funkciu,
- y je požadovaný výstup,
- θ^* predstavuje nájdené parametre modelu.

Na minimalizáciu sa používajú optimalizačné algoritmy založené na gradientoch, ktoré iteratívne upravujú parametre v smere znižovania chyby [3, 21]. Dôležitou súčasťou tohto procesu je spätné šírenie chyby (*backpropagation*), vďaka ktorému možno vypočítať príspevok každého parametra k výslednej chybe a upraviť váhy vo všetkých vrstvách siete [21].

2.1.3 Optimalizácia neurónových sietí

Základnou variantou optimalizácie neurónových sietí je stochastický gradientový zostup (*Stochastic Gradient Descent*, SGD) [21], pri ktorom sa gradient počíta na základe náhodne vybranej podmnožiny trénovacích vzoriek. Veľkosť kroku, o ktorý sa parametre upravujú, určuje hyperparameter označovaný ako rýchlosť učenia (*learning rate*).

Rozšírením SGD je metóda gradientového zostupu s momentom (*SGD with Momentum*) [22]. Namiesto aktualizácie parametrov iba na základe aktuálneho gradientu akumuluje informáciu o predchádzajúcich gradientoch, čím urýchľuje konvergenciu v konzistentných smeroch a potláča oscilácie. Aktualizáciu parametrov možno zapísať ako

$$v_{t+1} = \mu v_t - \alpha \nabla_\theta \mathcal{L}(\theta_t), \qquad \theta_{t+1} = \theta_t + v_{t+1}, \tag{2.3}$$

kde

- v_t je rýchlostný vektor (akumulovaný moment),
- $\mu \in [0, 1)$ je koeficient momentu,
- α je rýchlosť učenia,
- $\nabla_\theta \mathcal{L}(\theta_t)$ je gradient stratovej funkcie vzhľadom na parametre θ_t.

V praxi sa často používa optimalizátor Adam (*Adaptive Moment Estimation*) [23]. Adam kombinuje výhody metód založených na odhade prvého momentu (priemer gradientov) a druhého momentu (rozptyl gradientov), vďaka čomu dokáže adaptívne upravovať veľkosť kroku pre jednotlivé parametre. Aktualizácia parametrov v optimalizátore Adam možno zapísať ako

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t, \tag{2.4}$$

kde

- α je rýchlosť učenia
- $\hat{m}_t$ je korigovaný odhad prvého momentu gradientu
- $\hat{v}_t$ je korigovaný odhad druhého momentu gradientu
- ϵ je malá konštanta zabraňujúca deleniu nulou.

Vďaka svojej robustnosti a rýchlej konvergencii je Adam štandardnou voľbou pri trénovaní architektúr používaných v oblasti super-resolution [5].

2.1.4 Praktické aspekty trénovania

Pri trénovaní hlbokých modelov sa dáta spracúvajú po dávkach (*batch*). Jedna epocha zodpovedá jednému prechodu cez celú trénovaciu množinu; počet epôch určuje, koľkokrát model tieto dáta spracuje.

Dôležitým problémom je preučenie (*overfitting*), keď sa model príliš prispôsobí trénovacím dátam a nedokáže dobre zovšeobecniť na nové vstupy. Opačným javom je podučenie (*underfitting*), keď model nedokáže zachytiť ani podstatu trénovacích dát [21]. Na zmiernenie týchto problémov sa používajú regularizačné techniky, vhodné rozdelenie dát a augmentácia. Bežnou regularizačnou technikou v klasifikačných sieťach je dropout [24]. Pri architektúrach pre super-resolution sa od nej spravidla upúšťa, keďže náhodné vypínanie neurónov narúša konzistenciu príznakových máp a negatívne ovplyvňuje kvalitu rekonštruovaného obrazu.

Rozvoj hlbokého učenia v spracovaní obrazu bol podmienený dostupnosťou veľkých dátových množín, ako je **ImageNet**, a výpočtovým výkonom grafických procesorov [20, 25].

2.2 Konvolučné neurónové siete

Konvolučné neurónové siete (*Convolutional Neural Networks*, CNN) patria medzi najdôležitejšie architektúry hlbokého učenia pri spracovaní obrazu. Na rozdiel od plne prepojených vrstiev využívajú lokálne spojenia a zdieľanie váh, vďaka čomu dokážu efektívne zachytávať priestorové štruktúry obrazu ako sú hrany, rohy, textúry a jednoduché vzory pri nižšom počte parametrov [20, 21]. Spracovanie prebieha hierarchicky: začiatočné vrstvy zachytávajú jednoduché lokálne štruktúry, hlbšie vrstvy ich skladajú do zložitejších tvarov. V oblasti super-resolution sú CNN dôležité najmä preto,

že umožňujú učiť sa zobrazenie medzi LR a HR obrazmi priamo z dát, namiesto ručne navrhnutých interpolačných pravidiel [21].

2.2.1 Konvolučná vrstva

Základným stavebným prvkom CNN je konvolučná vrstva, ktorá spracováva vstup pomocou malých filtrov — konvolučných jadier [21, 20]. Filter sa systematicky posúva po vstupe a v každej polohe vykonáva skalárny súčin medzi svojimi váhami a lokálnou oblasťou vstupu. Výstupom sú príznakové mapy (*feature maps*) reprezentujúce prítomnosť a intenzitu lokálnych vzorov na rôznych miestach obrazu[**gonzalez2018digital**]. Výstup konvolučnej vrstvy možno zapísať ako[26]

$$Y = X * W + b, \tag{2.5}$$

kde

- X je vstup
- W predstavuje konvolučné filtre (jadrá)
- b je bias
- Y predstavuje výstupnú príznakovú mapu.

Dôležitými parametrami konvolučnej vrstvy sú veľkosť jadra, krok posunu a padding. Veľkosť jadra určuje, aké veľké lokálne okolie sa pri výpočte berie do úvahy, napríklad 3×3 alebo 5×5 pixelov. Krok posunu, označovaný aj ako *stride*, určuje, o koľko pozícií sa filter posunie pri každom kroku. Padding dopĺňa hodnoty na okrajoch obrazu, aby bolo možné kontrolovať veľkosť výstupnej príznakovej mapy. Pri úlohách super-resolution je zachovanie priestorových rozmerov často dôležité, preto sa volí padding tak, aby nedochádzalo k nežiadúcej strate rozlíšenia vo vnútorných vrstvách [16, 27].

Dve významné vlastnosti konvolučných vrstiev sú lokálna konektivita a zdieľanie váh. Lokálna konektivita znamená, že každý výstupný prvok závisí iba od malej oblasti vstupu. Zdieľanie váh znamená, že rovnaký filter sa používa na rôznych miestach obrazu. Vďaka tomu môže sieť rozpoznávať rovnaký typ vzoru bez ohľadu na jeho polohu v obraze [9, 21].

2.2.2 Aktivačné funkcie

Po konvolučnej operácii sa na výstup každej vrstvy zvyčajne aplikuje aktivačná funkcia. Jej úlohou je zaviesť do siete nelinearitu bez ktorej by aj viacvrstvová sieť predstavovala iba postupnosť lineárnych transformácií a nedokázala by zachytávať zložité vzťahy medzi vstupmi a výstupmi.

Dlhodobo dominantnou voľbou boli sigmoidálne funkcie, ako napríklad hyperbolický tangens $\tanh(x)$ alebo logistická sigmoida. Napriek bežnému využívaniu trpia tieto funkcie takzvaným problémom miznúcich gradientov (*vanishing gradient*), kde pri hlbokých sieťach môžu gradienty v počiatočných vrstvách klesnúť takmer na nulu a trénovanie sa zastaví [28].

ReLU

Zásadný posun prinieslo zavedenie funkcie ReLU [29] z anglického *Rectified Linear Unit*, definovanej ako

$$f(x) = \max(0, x). \tag{2.6}$$

Kým je neurón aktívny, jeho gradient je konštantný a rovný jednej, čo odstraňuje problém miznúcich gradientov a podstatne urýchľuje konvergenciu trénovania. Krizhevsky et al. [25] ukázali, že siete s ReLU trénujú niekoľkonásobne rýchlejšie ako ekvivalentné siete so sigmoidálnymi funkciami. Nevýhodou ReLU je však tzv. problém mŕtvych neurónov z anglického *dying ReLU* kde neuróny, ktoré dostanú záporný vstup, majú nulový gradient a nemôžu sa ďalej učiť.

LReLU

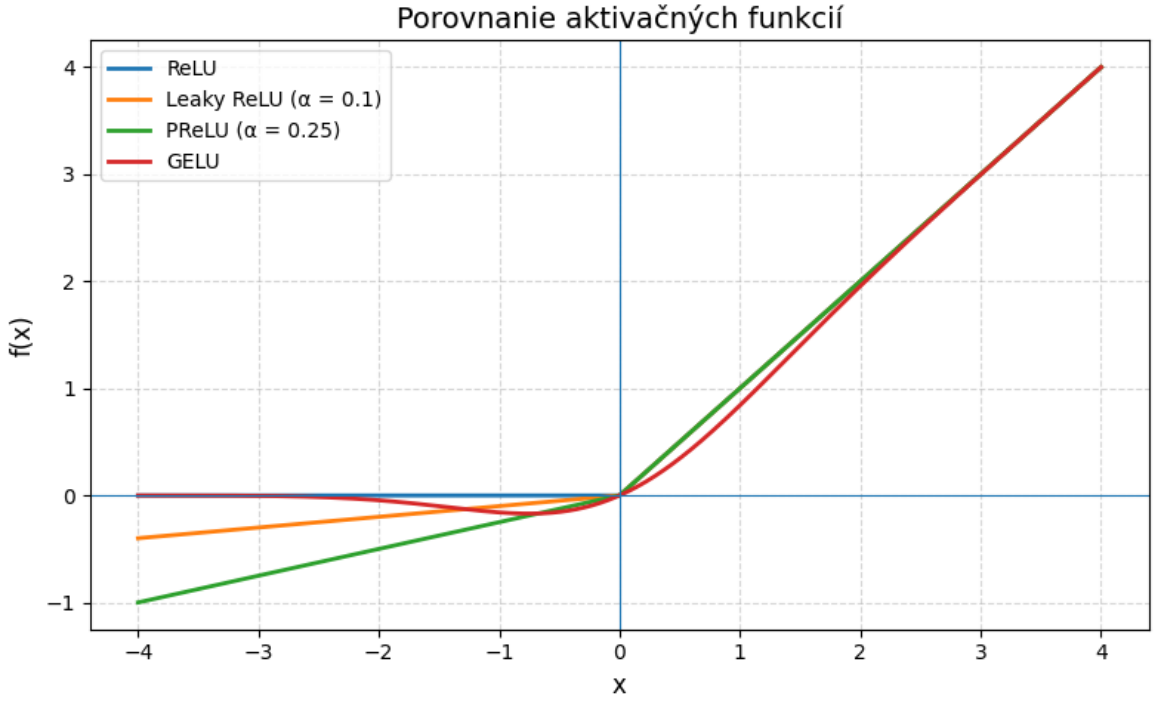
Tento problém čiastočne rieši Leaky ReLU (LReLU), ktorá ponecháva malý nenulový sklon aj pre záporné vstupy, konkrétne $f(x) = 0.01x$ pre $x \leq 0$ [28]. Ešte flexibilnejším riešením je Parametric ReLU (PReLU), kde je sklon zápornej časti a_i učiteľným parametrom trénovacieho procesu [26]:

$$f(y_i) = \begin{cases} y_i, & \text{ak } y_i > 0 \\ a_i y_i, & \text{ak } y_i \leq 0 \end{cases}. \tag{2.7}$$

He et al. [26] ukázali, že PReLU zlepšuje presnosť klasifikácie s takmer nulovými nákladmi na výpočet a bez znateľného rizika pretrénovania.

GELU

Moderné architektúry inšpirované transformermi čoraz častejšie využívajú funkciu GELU z anglického *Gaussian Error Linear Unit*, ktorá je hladšia ako ReLU a pre záporné hodnoty nevykazuje nulový gradient [30]. Voľba aktivačnej funkcie je preto jedným z prakticky relevantných rozhodnutí pri návrhu alebo modifikácii architektúry pre super-resolution, keďže rôzne rodiny blokov (reziduálne, husto prepojené, bloky inšpirované transformermi) sú historicky spájané s odlišnými aktivačnými funkciami.



Obr. 2.4: Porovnanie priebehu vybraných aktivačných funkcií ReLU, Leaky ReLU, PReLU a GELU

2.2.3 Normalizačné vrstvy

Normalizačné vrstvy upravujú rozdelenie aktivácií v sieti s cieľom stabilizovať a urýchliť trénovanie. Bez normalizácie môžu aktivácie v hlbokých sieťach nadobúdať extrémne hodnoty, čo sťažuje optimalizáciu a spomaľuje konvergenciu[31].

Najrozšírenejšou technikou je dávková normalizácia (*Batch Normalization*, BN) [31], ktorá normalizuje aktivácie v rámci každej trénovacej dávky zvlášť pre každý kanál a vstupný vektor aktivácií x sa normalizovaný výstup $\hat{x}$ vypočíta ako:

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \tag{2.8}$$

kde

- $\mu_{\mathcal{B}}$ a σ_B^2 sú priemer a rozptyl aktivácií v dávke
- B je dávka
- ε je malá konštanta zabraňujúca deleniu nulou.

Výstup sa následne škáluje a posúva pomocou dvojice učiteľných parametrov γ a β, čo sieti umožňuje obnoviť pôvodnú distribúciu aktivácií, ak je to pre danú úlohu výhodné [31].

Alternatívou je vrstvová normalizácia (*Layer Normalization*, LN) [32], ktorá normalizuje aktivácie naprieč všetkými kanálmi v rámci jedného príkladu, a preto nie je závislá od veľkosti dávky. V moderných konvolučných architektúrach, ako je ConvNeXt [33], sa vrstvová normalizácia používa ako priama náhrada dávkovej normalizácie a prispieva k lepšej stabilite trénovania.

Voľba normalizačnej vrstvy má priamy dopad na kvalitu rekonštrukcie: dávková normalizácia je závislá od štatistík celej dávky, čo môže vnášať do výsledného obrazu nežiaduce artefakty [7]. Vrstvová normalizácia sa tejto závislosti vyhýba a ukazuje sa ako stabilnejšia voľba v architektúrach pre super-resolution.

2.2.4 Priestorové rozlíšenie a upsampling

Poolingové vrstvy sa v CNN tradične používajú na znižovanie priestorových rozmerov príznakových máp. V kontexte super-resolution majú však zásadnú nevýhodu a to, že informácia o jemných detailoch kľúčová pre rekonštrukciu vysokého rozlíšenia sa pri podvzorkovaní nevratne stráca [3, 21]. Z tohto dôvodu architektúry pre super-resolution spravidla nepoužívajú pooling vo vnútorných vrstvách a priestorové rozmery zachovávajú vhodným paddingom.

Zvýšenie rozlíšenia výstupného obrazu sa dosahuje dvoma hlavnými prístupmi. Prvým je bilineárna alebo bikubická interpolácia nasledovaná konvolučnou vrstvou [16, 27]. Druhým, výpočtovo efektívnejším prístupom je sub-pixelová konvolúcia (*pixel shuffle*), ktorá generuje veľký počet príznakových máp v pôvodnom rozlíšení a reorganizuje ich do výstupu s vyšším rozlíšením. Celý výpočet tak prebieha v nízkej rozlišovacej doméne a priestorové zväčšenie nastáva až v poslednom kroku [13, 15].

Zachovanie priestorového rozlíšenia počas celého spracovania je kľúčovým návrhovým princípom CNN architektúr pre super-resolution.

2.3 Metódy super-resolution založené na CNN

Klasické interpolačné metódy opísané v predchádzajúcej kapitole majú spoločné zásadné obmedzenie a to že nevytvárajú nové obrazové informácie, ale iba odhadujú hodnoty medzi existujúcimi pixelmi na základe pevne definovaných matematických pravidiel. Vzťah medzi LR a HR obrazom je však príliš zložitý na to, aby ho bolo možné zachytiť takýmto spôsobom. Práve toto obmedzenie viedlo k rozvoju prístupov založených na učení z dát.

Prelomovú zmenu priniesol nástup konvolučných neurónových sietí. Na rozdiel od interpolačných metód, ktoré každý pixel spracúvajú nezávisle podľa vopred stanoveného pravidla, CNN sa naučia komplexné nelineárne zobrazenie $F_\theta : I_{\mathrm{LR}} \mapsto \hat{I}_{\mathrm{HR}}$ priamo z dát, kde θ označuje množinu trénovateľných parametrov, ktoré sa postupne upravujú minimalizáciou stratovej funkcie na trénovacích dátach. Najrozšírenejšou voľbou v začiatočných prácach bola stredná kvadratická chyba (MSE), ktorej minimalizácia je ekvivalentná maximalizácii metriky PSNR [5, 27].

Hoci PSNR zostáva dominantnou kvantitatívnou metrikou pri porovnávaní metód, siete trénované výhradne voči MSE produkujú štatisticky optimálne, no perceptuálne nevýrazné výstupy s nadmernou hladkosťou a stratou jemnej textúry. MSE totiž penalizuje priemernú pixelovú odchýlku naprieč priestorom všetkých možných riešení, čím potláča vysokofrekvenčné detaily. Táto diskrepancia medzi numerickými metrikami a vnímanou vizuálnou kvalitou je jednou z ústredných tém výskumu v oblasti super-resolution a priamo motivuje vývoj pokročilejších stratových funkcií [5, 6].

2.3.1 SRCNN a FSRCNN

Prvým modelom hlbokého učenia schopným priamo mapovať obraz s nízkym rozlíšením na obraz s vysokým rozlíšením bola sieť SRCNN (*Super-Resolution Convolutional Neural Network*), ktorú predstavili Dong et al. [27]. Napriek svojej jednoduchosti predstavovala SRCNN zásadný posun od predchádzajúcich prístupov: namiesto ručne navrhovaných príznakov a interpolačných pravidiel dokázala jediná neurónová sieť naučiť sa celé zobrazenie z LR na HR obrazy priamo z dát.

Architektúra SRCNN

Autori ukázali, že celý proces super-resolution (extrakcia príznakov, ich zobrazenie do priestoru HR obrazu a finálna rekonštrukcia) možno realizovať jedinou konvolučnou neurónovou sieťou, bez potreby ručne navrhovaných medzikrokov. Trojvrstvová architektúra SRCNN tento pohľad priamo odráža a každá vrstva zodpovedá jednej fáze rekonštrukcie.

Pred vstupom do siete sa LR obraz I_{LR} bikubicky interpoluje na cieľové rozlíšenie a označuje sa Y. Sieť sa následne učí zobrazenie $F : Y \mapsto \hat{I}_{\text{HR}}$, ktoré pozostáva z troch operácií:

1. **Extrakcia a reprezentácia príznakov:** prvá konvolučná vrstva s n_1 filtrami veľkosti $f_1 \times f_1$ extrahuje lokálne príznaky z interpolovaného obrazu:

$$F_1(Y) = \max(0,\, W_1 * Y + B_1), \tag{2.9}$$

   kde W_1 a B_1 označujú váhy a biasy tejto vrstvy.

2. **Nelineárne zobrazenie:** druhá vrstva zobrazuje každý n_1-dimenzionálny príznakový vektor na n_2-dimenzionálny vektor reprezentujúci zodpovedajúcu oblasť HR obrazu:

$$F_2(Y) = \max(0,\, W_2 * F_1(Y) + B_2). \tag{2.10}$$

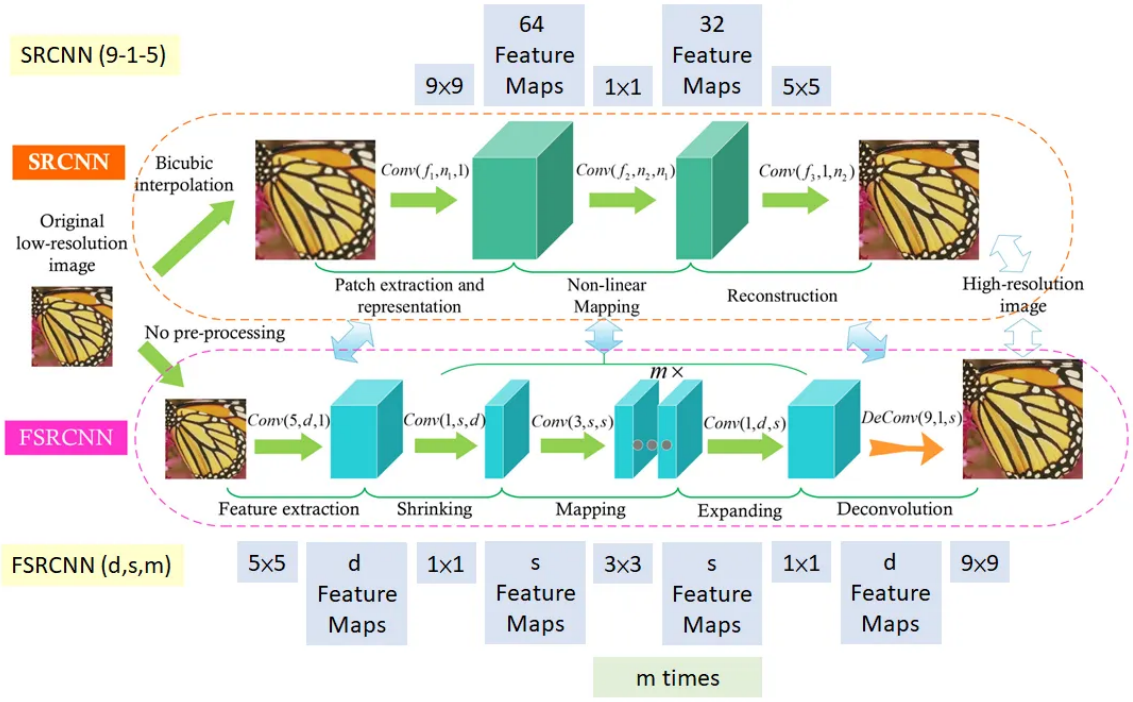
3. **Rekonštrukcia:** posledná vrstva zlučuje príznakové mapy a produkuje výsledný SR obraz:

$$F(Y) = W_3 * F_2(Y) + B_3. \tag{2.11}$$

Typická konfigurácia používa filtre veľkostí $f_1 = 9$, $f_2 = 5$ a $f_3 = 5$ s počtami filtrov $n_1 = 64$ a $n_2 = 32$. Celková sieť tak obsahuje iba 57 184 parametrov, napriek čomu prekonala vtedajšie najlepšie dostupné metódy, označované ako *state-of-the-art* (SOTA), vo viacerých štandardných metrikách [27]. Sieť sa trénuje minimalizáciou MSE straty pomocou stochastického gradientného zostupu so spätným šírením chyby.

Hlavným obmedzením SRCNN bola nutnosť bikubickej interpolácie pred vstupom do siete. Konvolúcie prebiehali nad obrazom v HR rozlíšení, čo znamenalo kvadratický nárast výpočtovej záťaže so škálovacím faktorom. Dong et al. tento problém adresovali v nadväzujúcej práci FSRCNN (*Fast Super-Resolution Convolutional Neural Network*) [15], ktorá presunula upsampling na koniec siete vo forme trénovateľnej dekonvolučnej vrstvy. Celý výpočet tak prebieha v LR doméne a priestorové zväčšenie nastáva až

v poslednom kroku, čo prinieslo viac ako 40× vyššiu rýchlosť pri porovnateľnej kvalite. Ako aktivačná funkcia sa používa PReLU namiesto ReLU, čo zabraňuje vzniku mŕtvych neurónov pri hlbších konfiguráciách [15]. Zásada spracovania v LR doméne s upsamplingom na konci sa stala štandardom pre väčšinu nasledujúcich architektúr.



Obr. 2.5: Architektúra SRCNN a FSRCNN [15]

2.3.2 Hlboké siete a reziduálne učenie

SRCNN a FSRCNN ukázali, že trénovanie konvolučných sietí prekonáva klasické metódy. Pri vyšších škálovacích faktoroch je rekonštrukcia jemných detailov podmienená kontextom z oveľa väčšieho okolia, než dokáže pokryť trojvrstvová architektúra. Prirodzenou odpoveďou bol prechod k hlbším sieťam, no jednoduché zväčšovanie hĺbky prináša vlastné problémy, ako sú miznúce gradienty a pomalú konvergenciu. Tieto problémy si vyžiadali nové trénovacie riešenie a stratégie.

VDSR a reziduálne učenie

Kim et al. [16] predstavili sieť VDSR (*Very Deep Super-Resolution*), inšpirovanú architektúrou VGG, ktorá pomocou 20 konvolučných vrstiev s filtrami 3×3 a 64 kanálmi dosahuje receptívne pole 41 × 41 pixelov. Kľúčovou inováciou VDSR nie je samotná

hĺbka, ale technika *reziduálneho učenia*. Namiesto priamej predikcie HR obrazu sa sieť učí odhadovať reziduálny obraz

$$r = I_{\mathrm{HR}} - Y, \tag{2.12}$$

teda rozdiel medzi cieľovým HR obrazom a bicubicky interpolovaným vstupom Y. Výsledný SR obraz sa získa jednoduchým súčtom

$$\hat{I}_{\mathrm{HR}} = F_\theta(Y) + Y, \tag{2.13}$$

kde $F_\theta(Y)$ je výstup siete. Keďže LR a HR obraz zdieľajú väčšinu nízkofrekvenčnej informácie, reziduálny signál obsahuje prevažne hodnoty blízke nule a sieť sa môže sústrediť výlučne na obnovu vysokofrekvenčných detailov. Na umožnenie rýchlej konvergencie takto hlbokej siete autori zaviedli orezávanie gradientov (*gradient clipping*), ktorá obmedzuje gradienty do intervalu $[-\theta/\gamma, \theta/\gamma]$, kde γ je aktuálna rýchlosť učenia. Táto adaptívna schéma umožňuje použiť počiatočnú rýchlosť učenia 10^4-krát vyššiu ako SRCNN bez rizika nekontrolovaného rastu gradientov, čím sa trénovanie 20-vrstvovej siete skráti na niekoľko hodín [16].

SRResNet

Ledig et al. [5] preniesli do oblasti super-resolution koncept reziduálnych blokov so skip-spojeniami. Sieť SRResNet pozostáva z $B = 16$ reziduálnych blokov, pričom každý obsahuje dve konvolučné vrstvy 3×3 so 64 kanálmi, dávkovú normalizáciu a aktiváciu PReLU. Výstup bloku sa sčíta so vstupom pomocou skip-spojenia:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l;\, \theta_l), \tag{2.14}$$

kde:

- $\mathbf{x}_l$ je vstup l-tého bloku
- $\mathcal{F}(\mathbf{x}_l;\, \theta_l)$ predstavuje transformáciu naučenú vnútri bloku.

Skip-spojenie zabezpečuje, že gradient môže prúdiť priamo späť cez sieť bez útlmu, čo odstraňuje problém miznúcich gradientov aj pri veľkej hĺbke. Upsampling sa realizuje pomocou dvoch sub-pixelových konvolučných vrstiev na konci siete, v súlade s princípom spracovania v LR doméne zavedeným v FSRCNN [15]. SRResNet trénovaná voči MSE dosiahla v čase publikácie najvyššie hodnoty PSNR spomedzi vtedy dostupných metód na štandardných benchmarkoch [5].

EDSR

Hoci SRResNet priniesla výrazné zlepšenie, priamo prevzala reziduálnu architektúru navrhnutú pre klasifikačné úlohy bez hlbšej analýzy toho, ktoré komponenty sú pre super-resolution skutočne potrebné. Lim et al. [13] tento nedostatok systematicky adresovali v sieti EDSR (*Enhanced Deep Residual Networks*).

Hlavným zistením bolo, že dávková normalizácia je v kontexte super-resolution kontraproduktívna: normalizáciou aktivácií podľa štatistík celej dávky obmedzuje flexibilitu rozsahu príznakov a zbytočne zužuje reprezentačnú kapacitu siete. Jej odstránením autori ušetrili až 40 % pamäte GPU, čo im umožnilo výrazne rozšíriť model na $B = 32$ blokov s $F = 256$ kanálmi (43 miliónov parametrov).

Pri takom počte filtrov sa trénovanie stáva numericky nestabilným, preto autori zaviedli reziduálne škálovanie: výstup každého bloku sa pred sčítaním so skip-spojením vynásobí malou konštantou $\lambda = 0.1$:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \lambda \cdot \mathcal{F}(\mathbf{x}_l;\, \theta_l). \tag{2.15}$$

Stratová funkcia MSE bola nahradená stratou ℓ_1, čo viedlo k lepšej konvergencii a mierne vyšším hodnotám PSNR aj SSIM [13]. EDSR zvíťazil na súťaži NTIRE 2017 Super-Resolution Challenge a jeho hlavný prínos spočíva v systematickom demonštrovaní toho, že návrhové rozhodnutia prevzaté z klasifikačných architektúr nemusia byť optimálne pre super-resolution.

2.4 Generatívne adversariálne siete

Všetky predchádzajúce opísané metódy minimalizujú stratu MSE, čo je ekvivalentné maximalizácii PSNR [5, 13]. Táto optimalizačná cieľová funkcia má zásadný perceptuálny nedostatok. Keďže k jednému LR obrazu existuje viacero možných HR riešení, MSE minimalizuje ich priemer v pixelovom priestore. Výsledkom sú rekonštrukcie, ktoré sú síce štatisticky optimálne, no vizuálne nevýrazné, napríklad hrany sú rozmazané a jemné textúry potlačené. Tento jav sa prejavuje tým výraznejšie, čím vyšší je škálovací faktor [5, 16].

Riešením nie je lepšia architektúra siete, ale iná stratová funkcia a to taká, ktorá hodnotí kvalitu výstupu nie na úrovni jednotlivých pixelov, ale na úrovni perceptuálne relevantných vlastností. Práve tento posun umožnili generatívne adversariálne siete.

2.4.1 Princíp GAN

Generatívnu adversariálnu sieť (GAN) navrhli Goodfellow et al. ako framework pre trénovanie generatívnych modelov prostredníctvom súťaže dvoch sietí [4]. *Generátor* G_{θ_G} sa snaží produkovať výstupy nerozoznateľné od reálnych dát, zatiaľ čo *diskriminátor* D_{θ_D} sa snaží rozlíšiť reálne vzorky od generovaných [4, 3]. Obe siete sa trénujú súčasne podľa minimax kritéria:

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{x \sim p_{\text{data}}} \Big[\log D_{\theta_D}(x) \Big] + \mathbb{E}_{y \sim p_G} \Big[\log \Big(1 - D_{\theta_D}(G_{\theta_G}(y)) \Big) \Big], \tag{2.16}$$

kde

- $x \sim p_{\text{data}}$ je reálna vzorka pochádzajúca z distribúcie skutočných dát,
- $y \sim p_G$ je vstup generátora pochádzajúci z distribúcie generátora,
- $G_{\theta_G}(y)$ je výstup generátora - syntetická vzorka,
- $D_{\theta_D}(x)$ je výstup diskriminátora - odhadovaná pravdepodobnosť, že vzorka x pochádza z reálnych dát,
- $\mathbb{E}_{x \sim p_{\text{data}}}[\log D_{\theta_D}(x)]$ je očakávaný logaritmus pravdepodobnosti, že diskriminátor správne rozpozná reálnu vzorku,
- $\mathbb{E}_{y \sim p_G}[\log(1 - D_{\theta_D}(G_{\theta_G}(y)))]$ je očakávaný logaritmus pravdepodobnosti, že diskriminátor odhalí syntetickú vzorku generátora [4].

Diskriminátor sa snaží maximalizovať túto hodnotu, teda čo najspoľahlivejšie rozlišovať reálne a generované vzorky [4, 3]. Generátor sa snaží ju minimalizovať, teda produkovať vzorky, ktoré diskriminátor nedokáže odhaliť. V ideálnom prípade trénovanie konverguje do bodu, v ktorom generátor produkuje vzorky nerozoznateľné od reálnych dát a diskriminátor nie je schopný ich spoľahlivo klasifikovať.

V kontexte super-resolution zastáva úlohu generátora sieť mapujúca LR obraz na SR obraz, zatiaľ čo diskriminátor sa učí rozlišovať rekonštruované SR obrazy od skutočných HR fotografií [5]. Adversariálna strata núti generátor produkovať výstupy konzistentné s distribúciou prirodzených obrazov, nie iba s pixelovým priemerom možných riešení kde výsledkom sú rekonštrukcie s ostrejšími hranami a vierohodnejšími textúrami.

Stratová funkcia generátora v SR kontexte kombinuje obsahovú stratu $\mathcal{L}_{\text{content}}$, ktorá zabezpečuje základnú vernosť rekonštrukcie, s adversariálnou stratou $\mathcal{L}_{\text{adv}}$, ktorá tlačí výstup smerom k distribúcii prirodzených obrazov [5]:

$$\mathcal{L} = \mathcal{L}_{\text{content}} + \lambda \cdot \mathcal{L}_{\text{adv}}, \tag{2.17}$$

kde λ je hyperparameter vyvažujúci oba príspevky. Výraznejší perceptuálny posun ako MSE v pixelovom priestore prináša *perceptuálna strata*, ktorá meria vzdialenosť v priestore príznakov predtrénovanej klasifikačnej siete, typicky VGG [34]. Príznaky hlbších vrstiev VGG [34, 35] zachytávajú sémantický obsah obrazu invariantne voči malým pixelovým odchýlkam, čo umožňuje generátoru produkovať textúry perceptuálne verné referenčnému obrazu aj napriek pixelovým rozdielom [5, 35]. Táto základná výmena medzi PSNR a perceptuálnou kvalitou, formalizovaná ako *perception-distortion tradeoff* [6], predstavuje jedno z ústredných napätí v modernom výskume super-resolution.

2.4.2 GAN prístupy pre super-resolution

SRGAN

Prvou prácou, ktorá aplikovala framework GAN na super-resolution so štvornásobným škálovacím faktorom, bola sieť SRGAN od Ledig et al. [5]. SRGAN kombinuje hlbokú reziduálnu generátorovú sieť SRResNet, ktorá je postavená na reziduálnych blokoch s dávkovou normalizáciou a PReLU aktivačnou funkciou. Sieť je trénovaná adversariálne s perceptuálnou stratovou funkciou definovanou v priestore príznakov VGG19. Rozsiahle používateľské hodnotenie (MOS) potvrdilo, že výstupy SRGAN sú vnímané ako výrazne fotorealistickejšie než výstupy metód optimalizovaných voči MSE, a to aj napriek nižším hodnotám PSNR. Keďže SRGAN tvorí základ modelu analyzovaného a upraveného v tejto práci, jeho architektúra a stratové funkcie sú podrobne rozobrané v nasledujúcej kapitole.

ESRGAN

Wang et al. [7] predstavili ESRGAN (*Enhanced Super-Resolution GAN*), ktorý adresoval niekoľko konkrétnych slabín pôvodného SRGAN. Prvou zmenou bolo odstránenie vrstiev dávkovej normalizácie z reziduálnych blokov generátora, čím sa predišlo normalizačným artefaktom, ktoré sa pri SRGAN prejavujú ako neprirozené jasové posuny

a nežiaduce vzory v rekonštruovanom obraze [7]. Na ich miesto autori navrhli hustejšie prepojené bloky RRDB (*Residual-in-Residual Dense Block*), v ktorých sú reziduálne spojenia vnorené aj vo vnútri jednotlivých blokov, čím sa zvyšuje tok príznakov a gradientov počas trénovania.

Druhou kľúčovou inováciou bola náhrada štandardného diskriminátora za *relatívny diskriminátor* (Relativistic GAN), ktorý nehodnotí absolútnu pravdepodobnosť, že daný obraz je reálny, ale porovnáva, o koľko je jeden obraz reálnejší ako druhý. Táto formulácia poskytuje generátoru informatívnejší gradientový signál aj v neskorších fázach trénovania, keď je diskriminátor schopný ľahko rozoznať rekonštruované obrazy [7]. Treťou zmenou bola úprava perceptuálnej straty. Namiesto aktivácií po ReLU sa využívajú príznaky pred aktivačnou funkciou, čo vedie k ostrejším a detailnejším výstupom. ESRGAN zvíťazil na súťaži PIRM 2018 SR Challenge a stal sa referenčnou architektúrou pre ďalší výskum v oblasti perceptuálnej super-resolution.

Real-ESRGAN

Spoločným predpokladom SRGAN aj ESRGAN je, že LR obraz vznikol bikubickým znižovaním rozlíšenia referenčného HR obrazu. Tento predpoklad výrazne zjednodušuje trénovanie, no v praxi vedie k výraznému poklesu výkonu, pretože reálne obrazy trpia kombináciou rôznych degradácií ako napríklad neznámym rozmazaním, šumom, kompresnými artefaktmi a neznámym spôsobom znižovania rozlíšenia, ktoré jednoduché bikubické zníženie rozlíšenia nezachytí.

Wang et al. [12] navrhli Real-ESRGAN ako riešenie v rámci tzv. slepej super-resolution (*blind super-resolution*), kde degradačný proces nie je vopred známy. Kľúčovým prínosom je degradačný model náhodne kombinujúci viaceré typy degradácií ako Gaussovo rozmazanie, šum, JPEG kompresiu a bikubické znižovanie rozlíšenia vo viacerých po sebe nasledujúcich krokoch, čím lepšie aproximuje komplexnosť reálnych degradácií. Architektúra preberá generátor RRDB z ESRGAN a rozširuje diskriminátor na U-Net architektúru poskytujúcu generátoru podrobnejšiu spätnú väzbu o kvalite jednotlivých častí rekonštruovaného obrazu [12].

Tabuľka 2.1: Prehľad vybraných metód super-resolution založených na hlbokom učení.

Metóda	**Rok**	**Kľúčová inovácia**	**Stratová funkcia**	**Doména**
SRCNN	2014	Prvá CNN architektúra pre super-resolution; vstup sa najprv zväčší bikubickou interpoláciou.	MSE	HR
FSRCNN	2016	Spracovanie obrazu prevažne v LR priestore a upsampling až na konci siete pomocou dekonvolučnej vrstvy.	MSE	LR
VDSR	2016	Hlbšia CNN s reziduálnym učením; model predikuje rozdiel medzi interpolovaným vstupom a HR obrazom.	MSE	HR
SRResNet	2017	Reziduálne bloky a sub-pixelová konvolúcia; generátor použitý v SRGAN.	MSE / obsahová	LR
EDSR	2017	Vylepšená reziduálna architektúra bez dávkovej normalizácie.	ℓ_1	LR
SRGAN	2017	GAN prístup a perceptuálna strata pre fotorealistickejšie výstupy.	VGG + adversariálna.	LR
ESRGAN	2018	RRDB bloky bez dávkovej normalizácie, relatívny diskriminátor a vylepšená perceptuálna strata.	perceptuálna + adversariálna	LR
Real-ESRGAN	2021	Rozšírenie ESRGAN pre reálne degradácie pomocou syntetického degradačného modelu.	pixelová + perceptuálna + adversariálna	LR / blind

V tabuľke 2.1 je uvedený prehľad vybraných metód super-resolution založených na hlbokom učení. Stĺpec *Metóda* označuje názov konkrétnej architektúry alebo prístupu a stĺpec *Rok* uvádza rok publikovania danej metódy. Stĺpec *Kľúčová inovácia* stručne opisuje hlavný prínos metódy oproti predchádzajúcim prístupom. Stĺpec *Stratová funkcia* označuje typ stratovej funkcie použitej pri trénovaní modelu, napríklad MSE, ℓ_1 stratu, percepčnú stratu alebo adversariálnu stratu.

Stĺpec *Doména* vyjadruje, v akej obrazovej doméne prebieha hlavná časť výpočtov modelu. Označenie *blind* znamená, že metóda nie je navrhnutá iba pre vopred známy degradačný proces, napríklad bikubické zníženie rozlíšenia, ale snaží sa pracovať aj s neznámymi alebo reálnymi degradáciami obrazu.

2.5 Súčasné trendy v oblasti super-resolution

Architektúry opísané v predchádzajúcich sekciách sú postavené výlučne na konvolučných operáciách. Napriek svojej efektívnosti majú konvolučné siete jedno štrukturálne obmedzenie: každý neurón vidí iba lokálne okolie dané veľkosťou jadra a dlhodobé

priestorové závislosti sa budujú až postupným skladaním vrstiev. Toto obmedzenie sa stalo zrejmým po nástupe mechanizmu *self-attention*, ktorý dokáže modelovať vzťahy medzi ľubovoľnými dvoma pozíciami v obraze v rámci jednej operácie [3, 33]. Zároveň sa ukázalo, že mnohé návrhové rozhodnutia transformerových architektúr možno preniesť späť do konvolučného rámca a získať tak modely kombinujúce efektívnosť konvolúcií s výkonnosťou moderných architektúr [33].

2.5.1 Vision Transformery a ďalšie prístupy

Dosovitskiy et al. [36] predstavili ViT (*Vision Transformer*), ktorý spracúva obraz odlišne od CNN: obraz sa rozdelí na malé neprekrývajúce sa výrezy, každý sa transformuje na vektor pevnej dĺžky (token) a vzniknutá sekvencia sa spracúva mechanizmom self-attention. Kľúčovou vlastnosťou je schopnosť modelovať závislosti medzi ľubovoľnými dvoma tokenmi bez ohľadu na ich priestorovú vzdialenosť.

Čistý ViT má však dve praktické nevýhody pre rekonštrukciu obrazu: výpočtová náročnosť self-attention rastie kvadraticky s počtom tokenov a obraz sa spracúva na jedinej úrovni bez hierarchického skladania príznakov. Tieto problémy adresoval Swin Transformer [37], ktorý obmedzuje self-attention na malé lokálne okná posúvané medzi vrstvami, čím dosahuje lineárnu výpočtovú zložitosť. V oblasti super-resolution sa presadil SwinIR [38] kombinujúci vstupné konvolučné vrstvy so sériou Swin Transformer blokov. SwinIR preukázal, že transformerový prístup dokáže prekonať konvolučné metódy vrátane EDSR na štandardných testovacích množinách, najmä pri rekonštrukcii opakujúcich sa textúr a pravidelných štruktúr.

Okrem transformerových prístupov sa objavila nová trieda generatívnych modelov, difúzne modely. Tieto modely dosahujú výnimočnú perceptuálnu kvalitu. Ich aplikácia na super-resolution je aktívnou oblasťou výskumu, no vysoká výpočtová náročnosť zatiaľ obmedzuje praktické nasadenie v porovnaní s CNN a transformerovými modelmi [21, 3].

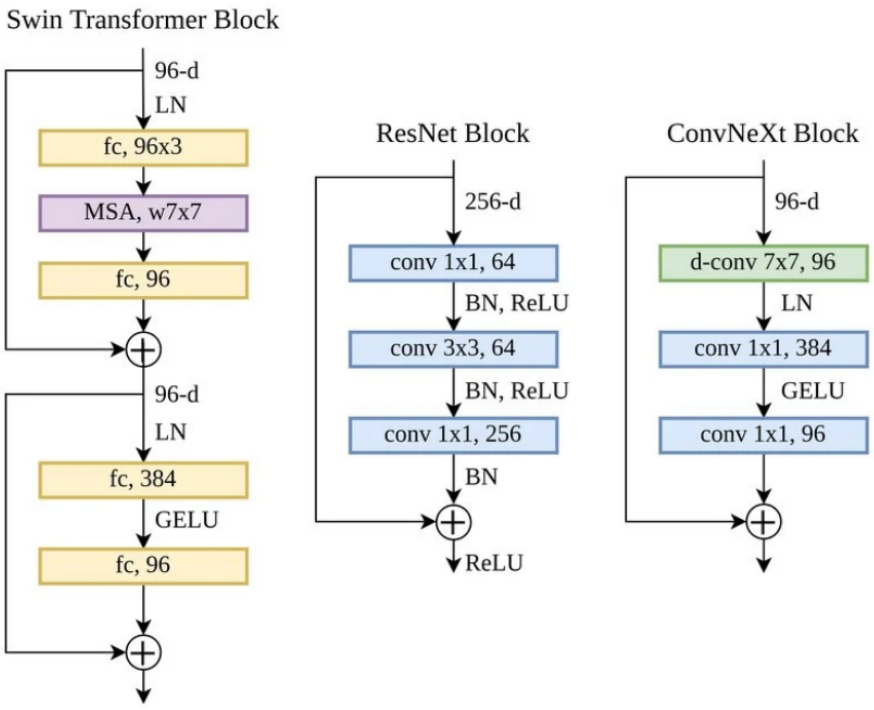
2.5.2 ConvNeXt a modernizácia konvolučných sietí

Nástup Vision Transformerov vyvolal otázku, či je výkonnostná prevaha transformerov daná samotným mechanizmom self-attention, alebo skôr súborom návrhových rozhodnutí, ktoré transformery štandardne prijali väčšie receptívne polia, iné normalizačné vrstvy, odlišné aktivačné funkcie a zmenená štruktúra blokov. Liu et al. [33] túto otázku systematicky preskúmali postupnou modernizáciou štandardnej ResNet archi-

tektúry a dosiahli výkonnosť porovnateľnú so Swin Transformerom [37] pri zachovaní jednoduchosti konvolučného prístupu.

Modernizácia zahŕňala štyri kľúčové zmeny. Štandardné konvolúcie 3×3 boli nahradené väčšími konvolúciami 7×7 aplikovanými po jednotlivých kanáloch (*depthwise convolution*), čím sa zväčšilo receptívne pole bez proporcionálneho nárastu počtu parametrov. Po priestorovej konvolúcii nasledujú bodové konvolúcie 1×1, ktoré najprv rozšíria počet kanálov a následne ho redukujú späť na pôvodnú hodnotu. Táto bottleneck štruktúra umožňuje modelovať vzťahy medzi kanálmi pri zachovaní relatívne nízkej výpočtovej náročnosti. Aktivačná funkcia ReLU bola nahradená funkciou GELU [30], pričom aktivácia sa používa iba raz v rámci bloku. Nakoniec dávková normalizácia ustúpila vrstvovej normalizácii [32], ktorá nie je závislá od štatistík celej trénovacej dávky.

Skip-spojenie zabezpečuje tok gradientov analogicky k reziduálnym blokom. Z pohľadu super-resolution sú ConvNeXt bloky zaujímavou alternatívou k štandardným reziduálnym blokom z dvoch dôvodov: depthwise konvolúcia 7×7 umožňuje sieti zachytiť širší kontext pri rekonštrukcii každého pixelu a vrstvová normalizácia odstraňuje závislosť od štatistík dávky, čím predchádza nežiaducim artefaktom [7]. Napriek nástupu transformerových architektúr zostávajú konvolučné siete praktickou voľbou vďaka výpočtovej efektívnosti a jednoduchosti trénovania.



Obr. 2.6: Porovnanie konfigurácie blokov ConvNeXt, ResNet a Swin Transformer [33]

2.5.3 ConvNeXt bloky v architektúrach pre super-resolution

Niekoľko súčasných prác skúmalo možnosti integrácie ConvNeXt blokov do architektúr pre super-resolution.

Zhang et al. [39] navrhli výpočtovo nenáročnú reziduálnu sieť pre super-resolution bez adversariálneho trénovania, ktorej základnými stavebnými prvkami sú bloky s depthwise konvolúciou, inverted bottleneck štruktúrou a aktiváciou GELU. Autori experimentálne potvrdili, že GELU v kombinácii s ConvNeXt blokmi dosahuje lepšie výsledky ako ReLU alebo Leaky ReLU.

Ma et al. [40] aplikovali ConvNeXt bloky priamo v generátore GAN pre super-resolution podvodných sonarových obrazov a preukázali ich zlučiteľnosť s adversariálnym trénovaním, hoci ich prístup nebol hodnotený na štandardných benchmarkoch pre prirodzené obrazy.

Park et al. [41] navrhli NeXtSRGAN ako rozšírenie ESRGAN, v ktorom je diskriminátor nahradený architektúrou inšpirovanou ConvNeXt, zatiaľ čo generátor si zachováva pôvodné RRDB bloky. Autori zaviedli relatívny diskriminátor a metódu interpolácie parametrov siete umožňujúcu plynulé vyváženie medzi perceptuálnou kvalitou a hodnotami PSNR. NeXtSRGAN dosiahol priemerné zlepšenie PSNR o 1,58 dB oproti ESRGAN na štandardných testovacích množinách.

Integrácia ConvNeXt blokov **priamo** do generátora v rámci **SRGAN** architektúry s adversariálnym trénovaním na prirodzených obrazoch však v čase začatia tejto práce nebola systematicky preskúmaná, čo predstavuje hlavnú motiváciu a pôvodný príspevok praktickej časti tejto práce.

Kapitola 3

Architektúra a analýza modelu SRGAN

Predchádzajúca kapitola sledovala vývoj CNN architektúr pre super-resolution od jednoduchých neuronových sietí až po generatívne prístupy. Na jej konci bolo ukázané, že metódy optimalizované voči MSE dosahujú vysoké hodnoty PSNR, no za cenu perceptuálne nevýrazných výstupov s potlačenými textúrami a rozmazanými hranami. SRGAN [5] predstavoval prvý model, ktorý tento kompromis riešil prostredníctvom adversariálneho trénovania a perceptuálnej stratovej funkcie, a stal sa základnou architektúrou pre celú triedu perceptuálne orientovaných metód super-resolution.

Táto kapitola poskytuje detailnú analýzu architektúry a trénovacieho postupu SRGAN, ktorá tvorí základ pre návrh upraveného modelu. Kapitola je uzavretá analytickým zhodnotením silných stránok a obmedzení pôvodného modelu, ktoré priamo motivujú neskôr navrhovanú úpravu.

3.1 Architektúra generátora

Generátor SRGAN je hlboká reziduálna konvolučná sieť, označovaná ako SRResNet, ktorej úlohou je zo vstupného LR obrazu $I_{\text{LR}} \in \mathbb{R}^{H \times W \times C}$ produkovať SR obraz $\hat{I}_{\text{HR}} \in \mathbb{R}^{rH \times rW \times C}$, kde r je škálovací faktor. Architektúra sa skladá zo štyroch funkčne odlišných častí: vstupnej konvolučnej vrstvy, bloku reziduálnych vrstiev, postreziduálnej časti s upsamplingom a výstupnej konvolučnej vrstvy [5].

Vstupná konvolučná vrstva

Vstupná časť siete pozostáva z jedinej konvolučnej vrstvy s veľkými filtrami 9×9 a 64 výstupnými kanálmi, po ktorej nasleduje aktivačná funkcia PReLU [5, 26]. Úlohou tejto vrstvy je extrahovať počiatočné príznaky zo vstupného LR obrazu a transformovať ho do priestoru príznakových máp, v ktorom operuje zvyšok siete. Väčšia veľkosť jadra oproti štandardným 3×3 filtrom zabezpečuje širší počiatočný kontext pri extrakcii príznakov. Výstup vstupnej vrstvy sa ukladá pre neskoršie skip-spojenie na úrovni celého generátora. Treba poznamenať, že referenčná implementácia použitá autormi v tejto práci nahrádza vstupnú konvolúciu 9×9 štandardným jadrom 3×3, čím sa zjednocuje veľkosť filtrov v celom generátore. Výstupná konvolúcia je taktiež redukovaná na jadro 1×1 [13].

Reziduálne bloky

Jadro generátora tvorí $B = 16$ reziduálnych blokov, ktoré majú rovnakú vnútornú štruktúru [5]. Každý blok obsahuje dve konvolučné vrstvy s filtrami 3×3 a 64 kanálmi, pričom po každej konvolúcii nasleduje dávková normalizácia. Medzi nimi je vložená aktivačná funkcia PReLU. Výstup bloku sa sčíta so vstupom pomocou skip-spojenia:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l;\, \theta_l), \tag{3.1}$$

kde

- $\mathbf{x}_l$ je vstup l-tého bloku,
- $\mathcal{F}(\mathbf{x}_l;\, \theta_l)$ predstavuje transformáciu naučenú vnútri bloku,
- θ_l sú jeho trénovateľné parametre.

Skip-spojenie plní dve funkcie a to:

1. Umožňuje gradientu prúdiť priamo späť sieťou bez útlmu, čím zmierňuje problém miznúcich gradientov,
2. Zbavuje blok nutnosti modelovať identické zobrazenie každý blok sa sústredí výlučne na reziduálnu korekciu vstupu[3].

Voľba dávkovej normalizácie v reziduálnych blokoch SRGAN bola v čase publikácie štandardnou praxou prevzatou z klasifikačných architektúr [5]. Práve táto voľba sa neskôr ukázala ako jeden z hlavných zdrojov vizuálnych artefaktov v rekonštruovaných obrazoch.

Post-reziduálna časť a upsampling

Za posledným reziduálnym blokom nasleduje konvolučná vrstva 3×3 s dávkovou normalizáciou, ktorej výstup sa sčíta s výstupom vstupnej konvolučnej vrstvy prostredníctvom globálneho skip-spojenia. Toto spojenie obchádza celý blok reziduálnych vrstiev a zabezpečuje, že nízkofrekvenčná informácia obsiahnutá vo vstupnom obraze je priamo dostupná pri rekonštrukcii [5, 16].

Upsampling na cieľové rozlíšenie sa realizuje pomocou dvoch sub-pixelových konvolučných vrstiev, z ktorých každá zvyšuje priestorové rozmery dvojnásobne. Každá upsamplingová vrstva pozostáva z konvolúcie 3×3, ktorá generuje až $r^2 \cdot C$ príznakových máp, po ktorej nasleduje operácia pixel shuffle [42] reorganizujúca tieto mapy do obrazu s vyšším rozlíšením:

$$\mathrm{PS}(F)_{x,y,c} = F_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, C \cdot r \cdot \mathrm{mod}(y,r) + C \cdot \mathrm{mod}(x,r) + c}, \tag{3.2}$$

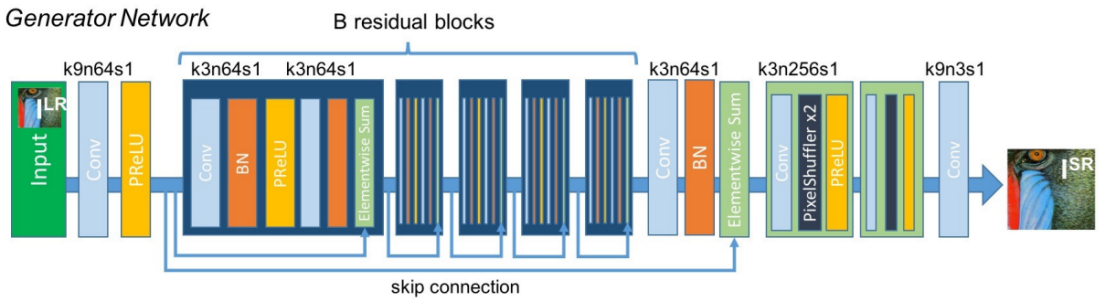
kde:

- $F \in \mathbb{R}^{H \times W \times r^2 C}$ je vstupná príznaková mapa s $r^2 \cdot C$ kanálmi,
- (x, y) sú priestorové súradnice výstupného obrazu s rozmermi $rH \times rW$,
- $c \in \{0, \ldots, C-1\}$ je index výstupného kanála,
- r je škálovací faktor,
- $\mathrm{mod}(x, r)$ a $\mathrm{mod}(y, r)$ určujú subpixelovú pozíciu v rámci $r \times r$ bunky,
- kanálový index $C \cdot r \cdot \mathrm{mod}(y, r) + C \cdot \mathrm{mod}(x, r) + c$ vyberá príslušný subpixel spomedzi r^2 kandidátov uložených v kanálovej dimenzii [42].

Celý upsampling tak prebieha v LR doméne a priestorové zväčšenie nastáva až v poslednom kroku, čo výrazne znižuje výpočtovú záťaž oproti prístupu SRCNN, kde konvolúcie prebiehali nad bicubicky interpolovaným obrazom. Po každej upsamplingovej vrstve nasleduje aktivačná funkcia PReLU [5, 15].

Výstupná vrstva

Poslednou vrstvou generátora je konvolúcia 9×9 s C výstupnými kanálmi s aktivačnou funkciou tanh, ktorá produkuje výsledný SR obraz $\hat{I}_{\mathrm{HR}}$. Väčšie jadro na výstupe zodpovedá vstupnej vrstve a umožňuje sieti pri finálnej syntéze zohľadniť širší priestorový kontext [5].



Obr. 3.7: Architektúra generátora v modeli SRGAN [5].

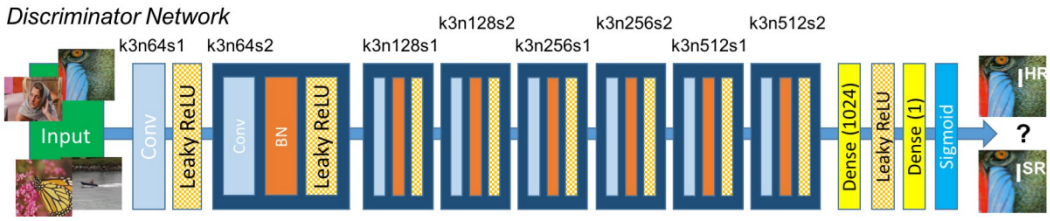
3.2 Architektúra diskriminátora

Diskriminátor D_{θ_D} plní v rámci SRGAN úlohu adversariálneho kritika kde sa počas trénovania učí rozlišovať reálne HR obrazy od SR obrazov produkovaných generátorom, čím poskytuje generátoru gradientový signál tlačiaci jeho výstupy smerom k distribúcii prirodzených fotografií. Architektúra diskriminátora je inšpirovaná sieťou VGG [34] a pozostáva z ôsmich konvolučných vrstiev s postupne rastúcim počtom kanálov, po ktorých nasledujú dve plne prepojené vrstvy a sigmoid aktivácia produkujúca skalárnu pravdepodobnosť.

Konvolučné vrstvy diskriminátora používajú filtre 3×3 a počet kanálov sa zdvojnásobuje v každom druhom bloku v poradí $64, 128, 256, 512$. Na znižovanie priestorových rozmerov príznakových máp sa striedavo používa krok (*stride*) $s = 1$ a $s = 2$, čím sa eliminuje potreba poolingových vrstiev. Po každej konvolúcii nasleduje dávková normalizácia [31] a aktivačná funkcia LeakyReLU s koeficientom zápornej časti $\alpha = 0.2$:

$$f(x) = \begin{cases} x, & \text{ak } x > 0 \\ 0.2\,x, & \text{ak } x \leq 0, \end{cases} \tag{3.3}$$

Voľba LeakyReLU namiesto ReLU je pri diskriminátoroch štandardnou praxou [4]: nenulový gradient pre záporné vstupy zabraňuje odumieraniu neurónov a zabezpečuje stabilnejší tok gradientov počas adversariálneho trénovania [21]. Dávková normalizácia sa neaplikuje na prvú konvolučnú vrstvu, ktorá spracúva priamo vstupný obraz.



Obr. 3.8: Architektúra diskriminátora v modeli SRGAN[5].

Výsledný tenzor sa sploští a spracuje dvoma plne prepojenými vrstvami s 1024 neurónmi a LeakyReLU aktiváciou. Výstupom siete je jediný skalár $D_{\theta_D}(x) \in (0, 1)$ produkovaný sigmoid aktiváciou, ktorý možno interpretovať ako pravdepodobnosť, že vstupný obraz x je reálna HR fotografia:

$$D_{\theta_D}(x) = \sigma(W_2 \, \text{LeakyReLU}(W_1 \, \text{Flatten}(F) + b_1) + b_2) \,, \tag{3.4}$$

kde F označuje výstup konvolučnej časti, W_1, W_2, b_1, b_2 sú parametre plne prepojených vrstiev a σ označuje sigmoid funkciu [4, 9].

Celková hĺbka diskriminátora je osem konvolučných vrstiev spolu s dvoma plne prepojenými. Tá mu poskytuje dostatočnú kapacitu na rozlišovanie jemných štatistických rozdielov medzi rekonštruovanými a reálnymi obrazmi. Diskriminátor musí byť dostatočne výkonný, aby generátoru poskytoval informatívny gradientový signál aj v pokročilých fázach trénovania [5, 21]

3.3 Stratová funkcia

Trénovanie SRGAN je riadené kompozitnou stratovou funkciou generátora, ktorá kombinuje tri čiastkové straty sledujúce odlišné aspekty kvality rekonštrukcie. Na rozdiel od predchádzajúcich metód optimalizovaných výlučne voči pixelovej vernosti, SRGAN zavádza perceptuálnu obsahovú stratu a adversariálnu stratu, ktoré spoločne tlačia generátor smerom k výstupom s prirodzeným vzhľadom [5].

3.3.1 Pixelová strata

Pixelová strata meria priamu odchýlku medzi rekonštruovaným obrazom $\hat{I}_{HR} = G_{\theta_G}(I_{LR})$ a referenčným HR obrazom I_{HR} na úrovni jednotlivých pixelov pomocou strednej

kvadratickej chyby, formálne zhodnej s metrikou MSE definovanou v rovnici (1.13):

$$\mathcal{L}_{MSE} = \frac{1}{r^2 HWC} \sum_{x=1}^{rW} \sum_{y=1}^{rH} \sum_{c=1}^{C} \left(I_{HR}^{(x,y,c)} - G_{\theta_G}(I_{LR})^{(x,y,c)} \right)^2, \tag{3.5}$$

kde H, W sú priestorové rozmery LR obrazu, C počet farebných kanálov a r škálovací faktor. Minimalizácia MSE vedie k vysokým hodnotám PSNR, a tým pádom výsledné obrazy trpia nadmerným vyhladzovaním textúr a hrán. Pixelová strata sa v rámci SRGAN využíva predovšetkým počas predtrénovania generátora [5, 27].

3.3.2 Perceptuálna obsahová strata

Perceptuálna obsahová strata [35] meria odchýlku nie na úrovni pixelov, ale na úrovni príznakových máp hlbokej siete predtrénovanej na klasifikácii obrazov. Táto reprezentácia zachytáva sémantický obsah a štrukturálne vlastnosti obrazu, ktoré sú perceptuálne relevantné, no pixelová metrika ich nedokáže vyjadriť.

V SRGAN sa ako extrakčná sieť používa VGG19 [34] predtrénovaná na databáze ImageNet [25]. Perceptuálna strata je definovaná ako MSE medzi aktiváciami konkrétnej vrstvy $\phi_{i,j}$ tejto siete pre rekonštruovaný a referenčný obraz:

$$\mathcal{L}_{\text{VGG}} = \frac{1}{W_{i,j} H_{i,j} C_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \sum_{c=1}^{C_{i,j}} \left(\phi_{i,j}(I_{\text{HR}})^{(x,y,c)} - \phi_{i,j}(G_{\theta_G}(I_{\text{LR}}))^{(x,y,c)} \right)^2, \tag{3.6}$$

kde $\phi_{i,j}$ označuje príznakové mapy pred j-tou aktivačnou funkciou po i-tej maxpoolingovej vrstve siete VGG19, a $W_{i,j}$, $H_{i,j}$, $C_{i,j}$ sú ich priestorové rozmery a počet kanálov. Autori experimentovali s rôznymi vrstvami VGG19 a zistili, že hlbšie vrstvy (označované ako $\text{VGG}_{5,4}$, teda pred štvrtou aktiváciou po piatom maxpoolingu) poskytujú lepšiu perceptuálnu kvalitu ako plytšie vrstvy, ktoré sa správajú podobnejšie k pixelovej strate [5].

3.3.3 Adversariálna strata

Adversariálna strata vyplýva priamo z princípu GAN a tvorí mechanizmus, ktorým diskriminátor poskytuje generátoru spätnoväzbový signál o perceptuálnej vierohodnosti jeho výstupov [4]. Generátor je trénovaný tak, aby maximalizoval pravdepodobnosť, že diskriminátor jeho výstupy považuje za reálne HR obrazy:

$$\mathcal{L}_{\text{adv}} = -\log D_{\theta_D}(G_{\theta_G}(I_{\text{LR}})), \tag{3.7}$$

kde $D_{\theta_D}(G_{\theta_G}(I_{\mathrm{LR}})) \in (0,1)$ je pravdepodobnosť priradená diskriminátorom rekonštruovanému obrazu. Minimalizácia $-\log D$ je ekvivalentná maximalizácii tejto pravdepodobnosti, teda klamaniu diskriminátora [5].

Stratová funkcia diskriminátora je štandardná binárna krížová entropia, ktorá ho trénuje rozlišovať reálne HR obrazy (cieľová hodnota 1) od SR výstupov generátora (cieľová hodnota 0) [3, 4]:

$$\mathcal{L}_D = -\log D_{\theta_D}(I_{\mathrm{HR}}) - \log(1 - D_{\theta_D}(G_{\theta_G}(I_{\mathrm{LR}}))) \,. \tag{3.8}$$

3.3.4 Kompozitná stratová funkcia generátora

Celková stratová funkcia generátora je vážená kombinácia perceptuálnej obsahovej straty a adversariálnej straty:

$$\mathcal{L}^{\mathrm{SR}} = \mathcal{L}_{\mathrm{VGG}} + \lambda \, \mathcal{L}_{\mathrm{adv}}, \tag{3.9}$$

kde $\lambda = 10^{-3}$ je váhový koeficient určený experimentálne [5]. Dominantným členom je perceptuálna obsahová strata $\mathcal{L}_{\mathrm{VGG}}$, ktorá riadi celkový charakter rekonštrukcie. Adversariálna zložka s malým koeficientom λ jemne koriguje štatistické vlastnosti výstupu smerom k distribúcii prirodzených fotografií. Blau a Michaeli [6] neskôr teoreticky preukázali, že zvyšovanie váhy adversariálnej straty na úkor pixelovej vernosti leží na hranici kompromisu medzi perceptuálnou kvalitou a distorziou, čo vysvetľuje, prečo je voľba koeficientu λ jedným z najcitlivejších rozhodnutí pri návrhu perceptuálne orientovaných SR modelov.

3.4 Trénovací postup

Trénovanie SRGAN prebieha v dvoch po sebe nasledujúcich fázach, ktoré reflektujú zásadný rozdiel medzi optimalizáciou voči pixelovej vernosti a optimalizáciou voči perceptuálnej kvalite. Priame trénovanie celého GAN systému od náhodnej inicializácie je nestabilné keďže generátor produkuje spočiatku náhodné výstupy, ktoré diskriminátor ľahko rozozná, čo vedie k degenerovaným gradientom a zlej konvergencii. Dvojfázový postup tento problém rieši tým, že generátor pred adversariálnym trénovaním získa solidný základ v podobe schopnosti produkovať pixelovo verné rekonštrukcie.

3.4.1 Predtrénovanie generátora

V prvej fáze sa generátor SRResNet trénuje samostatne, bez diskriminátora, minimalizáciou pixelovej MSE straty. Cieľom tejto fázy nie je dosiahnuť finálnu perceptuálnu kvalitu, ale naučiť generátor základné zobrazenie z LR do HR priestoru tak, aby jeho výstupy boli štrukturálne konzistentné s referenčnými HR obrazmi.

Predtrénovanie prebieha s optimalizátorom Adam [23] s parametrami $\beta_1 = 0.9$, $\beta_2 = 0.999$ a počiatočnou rýchlosťou učenia $\eta = 10^{-4}$ po dobu 10^6 aktualizačných krokov. Trénovacie dáta pozostávajú z náhodne vybraných 350 000 obrazov z databázy ImageNet [25], z ktorých sa pre každú mini-dávku orezáva 16 náhodných HR záplat veľkosti 96×96 pixelov. Zodpovedajúce LR vstupy sa generujú bikubickým znižovaním rozlíšenia so škálovacím faktorom $r = 4$ [5]. Referenčná implementácia použitá v tejto práci nahrádza optimalizátor Adam optimalizátorom SGD s momentom $\beta = 0.9$.

3.4.2 Adversariálne trénovanie

V druhej fáze sa predtrénovaný generátor použije ako inicializačný bod pre adversariálne trénovanie. Stratová funkcia generátora sa prepne z MSE na kompozitnú perceptuálnu stratu z rovnice (3.9) a do trénovania sa zapojí diskriminátor. Parametre diskriminátora sa inicializujú náhodne.

Trénovanie GAN systému prebieha striedavou aktualizáciou generátora a diskriminátora. V každom kroku sa najprv aktualizuje diskriminátor minimalizáciou straty $\mathcal{L}_D$ z rovnice (3.8), a následne sa aktualizuje generátor minimalizáciou kompozitnej straty $\mathcal{L}^{\mathrm{SR}}$. Tento postup zodpovedá štandardnej schéme $k = 1$ navrhnutej Goodfellowom et al. [4], kde sa diskriminátor aktualizuje raz za každú aktualizáciu generátora.

Adversariálne trénovanie prebieha v dvoch po sebe nasledujúcich etapách s rôznymi rýchlosťami učenia. V prvej etape sa trénuje 10^5 krokov s rýchlosťou učenia $\eta = 10^{-4}$, v druhej etape ďalších 10^5 krokov so zníženou rýchlosťou $\eta = 10^{-5}$. Zníženie rýchlosti učenia v druhej etape stabilizuje trénovanie v jeho záverečnej fáze a zabraňuje oscilácii parametrov v okolí lokálneho minima. Optimalizátor Adam s rovnakými parametrami β_1 a β_2 sa používa pre oba modely [5].

3.4.3 Trénovacie dáta a augmentácia

Trénovacia množina pozostáva z 350 000 obrazov náhodne vybraných z databázy ImageNet [25], ktorá pokrýva širokú škálu prirodzených scén, objektov a textúr. Táto

diverzita je kľúčová pre generalizáciu modelu. Sieť sa musí naučiť rekonštruovať rozmanité obrazové štruktúry, nie iba úzku triedu scén.

Pre každú mini-dávku o veľkosti 16 sa z každého trénovacieho obrazu náhodne oreže HR záplata veľkosti 96×96 pixelov. Zodpovedajúca LR záplata veľkosti 24×24 pixelov sa získa bikubickým znižovaním rozlíšenia so škálovacím faktorom $r = 4$. Hodnoty pixelov HR obrazov sa škálujú do intervalu $[-1, 1]$ a LR obrazov do intervalu $[0, 1]$ [5]. Autori ale neuvádzajú použitie augmentačných techník. Veľkosť a diverzita databázy ImageNet je považovaná za dostatočnú náhradu za explicitnú augmentáciu, na rozdiel od niektorých iných SR prác ako EDSR [13].

Referenčná implementácia použitá v tejto práci sa od pôvodného návrhu odlišuje v troch aspektoch: trénovacia množina je nahradená datasetom DIV2K [43] s 800 vysokorozlíšenými obrazmi, HR záplaty majú veľkosť 384×384 pixelov so zodpovedajúcimi LR záplatami 96×96 pixelov, a trénovanie zahŕňa augmentáciu formou náhodného orezania a horizontálneho prevrátenia.

3.4.4 Vyhodnotenie

Model sa v článku vyhodnocuje na troch štandardných benchmarkových datasetoch: Set5, Set14 a BSD100 pri škálovacom faktore $r = 4$. Metriky PSNR a SSIM sa počítajú na jasovom kanáli Y farebného priestoru YCbCr po orezaní 4-pixelového okraja, čo je štandardná metodológia umožňujúca porovnanie s predchádzajúcimi prácami. Perceptuálna kvalita sa hodnotí pomocou rozsiahleho MOS testu, v ktorom 26 hodnotiteľov priraďovalo skóre od 1 (nízka kvalita) do 5 (vysoká kvalita) rekonštruovaným obrazom [5].

3.5 Silné stránky a obmedzenia SRGAN

SRGAN predstavoval v čase publikácie zásadný posun v oblasti super-resolution a definoval návrhové princípy, na ktoré nadviazali všetky nasledujúce perceptuálne orientované metódy. Zároveň však trpí niekoľkými konkrétnymi obmedzeniami, ktorých pochopenie priamo motivuje úpravy navrhované v tejto práci.

3.5.1 Silné stránky

Hlavným prínosom SRGAN je preukázanie, že adversariálne trénovanie v kombinácii s perceptuálnou stratovou funkciou dokáže produkovať SR obrazy s výrazne vyššou vní-

manou kvalitou ako metódy optimalizované voči MSE. Rozsiahle testovanie potvrdilo, že výstupy SRGAN dosiahli v MOS testovaní skóre bližšie k referenčným HR obrazom ako ktorákoľvek porovnávaná metóda, a to aj napriek nižším hodnotám PSNR [5]. Tento výsledok empiricky potvrdil, že pixelová vernosť a perceptuálna kvalita sú odlišné a čiastočne protichodné ciele, čo zásadne ovplyvnilo smerovanie následného výskumu.

Druhou silnou stránkou je modulárna architektúra, ktorá jasne oddeľuje extrakciu príznakov, reziduálne spracovanie a upsampling do samostatných funkčných celkov. Táto štruktúra uľahčuje cielenú modifikáciu jednotlivých komponentov bez nutnosti prevrátenia celého návrhu. Táto vlastnosť je kľúčová pre experimentálnu prácu. Predtrénovací postup na báze MSE ďalej zabezpečuje, že adversariálne trénovanie začína z rozumného inicializačného bodu, čo výrazne stabilizuje konvergenciu.

3.5.2 Obmedzenia

Napriek týmto prínosom má SRGAN niekoľko dobre zdokumentovaných obmedzení, ktoré identifikovali už spomínané práce ESRGAN a Real-ESRGAN.

Artefakty dávkovej normalizácie

Najvýznamnejším štrukturálnym problémom je použitie dávkovej normalizácie [31] v reziduálnych blokoch generátora. Dávková normalizácia normalizuje aktivácie podľa štatistík celej trénovacej dávky, čo vnáša do procesu štatistickú závislosť medzi jednotlivými obrazmi v dávke. Pri inferenciách sa používajú štatistiky odhadnuté počas trénovania, ktoré nemusia presne zodpovedať štatistikám testovacích obrazov. Tento nesúlad sa prejavuje ako neprirozené jasové posuny, periodické vzory a ďalšie vizuálne artefakty v rekonštruovanom obraze, najmä pri spracovaní obrazov s výrazne odlišnými štatistickými vlastnosťami ako trénovacia množina [7].

Obmedzené receptívne pole reziduálnych blokov

Reziduálne bloky SRGAN používajú konvolúcie 3×3, ktoré v rámci jedného bloku pokrývajú iba úzke lokálne okolie. Hoci hĺbka siete teoreticky umožňuje zachytiť dlhodobé priestorové závislosti ich postupným skladaním, efektívne receptívne pole zostáva obmedzené v porovnaní s modernými architektúrami. Pri rekonštrukcii opakujúcich sa textúr, pravidelných štruktúr alebo veľkých hladkých plôch, kde je kontext z väčšieho

okolia kľúčový, môže toto obmedzenie viesť k nekonzistentnému alebo artefaktami zaťaženému výstupu.

Nestabilita adversariálneho trénovania

Adversariálne trénovanie je vo všeobecnosti citlivé na vyváženie kapacity generátora a diskriminátora a na voľbu hyperparametrov. Pri SRGAN sa táto nestabilita prejavuje predovšetkým v podobe halucinácií textúr kde generátor občas produkuje vizuálne presvedčivé, no fakticky nesprávne detaily, ktoré v skutočnom HR obraze nie sú prítomné. Hoci koeficient $\lambda = 10^{-3}$ v stratovej funkcii tento efekt čiastočne tlmí dominanciou perceptuálnej obsahovej straty, úplne ho neodstraňuje [5].

Závislosť od bikubickej degradácie

Ako bolo diskutované v sekcii 2.4.2, SRGAN predpokladá, že LR obraz vznikol bikubickým znižovaním rozlíšenia. Táto zjednodušená degradácia nezodpovedá komplexnosti reálnych degradácií, čo vedie k výraznému poklesu kvality pri aplikácii modelu na reálne obrazy obsahujúce kombináciu šumu, kompresných artefaktov a neznámeho rozmazania.

Kapitola 4

Návrh a implementácia upravenej metódy

Táto kapitola opisuje návrh a implementáciu modelu ConvNeXt-SRGAN, ktorý vznikol cielenými úpravami architektúry generátora pôvodného SRGAN. Popisuje motiváciu navrhovanej zmeny, štruktúru upravenej architektúry a implementačné detaily vrátane použitého frameworku, prípravy dát a nastavenia parametrov.

4.1 Cieľ navrhovanej úpravy

Analýza pôvodného SRGAN odhalila dva štrukturálne nedostatky generátora, ktoré priamo obmedzujú kvalitu rekonštrukcie. Prvým je použitie dávkovej normalizácie v reziduálnych blokoch, ktorá vnáša závislosť aktivácií od štatistík celej trénovacej dávky a pri inferenciách spôsobuje nežiaduce jasové artefakty [7]. Druhým je obmedzené receptívne pole štandardných konvolúcií 3×3, ktoré komplikuje zachytenie širšieho priestorového kontextu pri rekonštrukcii opakujúcich sa textúr a pravidelných štruktúr.

Cieľom tejto práce je overiť, či nahradenie reziduálnych blokov generátora modernými ConvNeXt blokmi [33] pri zachovaní ostatných komponentov SRGAN vedie k zlepšeniu perceptuálnej kvality rekonštruovaných obrazov. ConvNeXt je moderná architektúra, ktorá umožňuje používať väčšie kernely, poskytuje stabilitu tréningu, menší počet parametrov a vytvára lepšiu reprezentáciu údajov v porovnaní s tradičnými konvolučnými sieťami. ConvNeXt bloky priamo adresujú oba identifikované nedostatky generátora SRGAN: depthwise konvolúcia 7×7 zväčšuje receptívne pole bez proporcionálneho nárastu počtu parametrov a vrstvová normalizácia odstraňuje

závislosť od štatistík dávky, ktorá je zdrojom normalizačných artefaktov. Ostatné komponenty modelu, teda diskriminátor a trénovací postup zostávajú zámerne nezmenené, aby bolo možné izolovať príspevok samotnej zmeny generátora.

4.2 Navrhnutá architektúra ConvNeXt-SRGAN

V tejto sekcii je predstavená celková bloková štruktúra generátora a tok dát od vstupu po výstup, následne je podrobne opísaný ConvNeXt blok a jeho adaptácia pre úlohu super-resolution. Neskôr v sekcii si popíšeme diskriminátor a porovnanie navrhnutej architektúry s pôvodným SRGAN.

4.2.1 Bloková štruktúra generátora

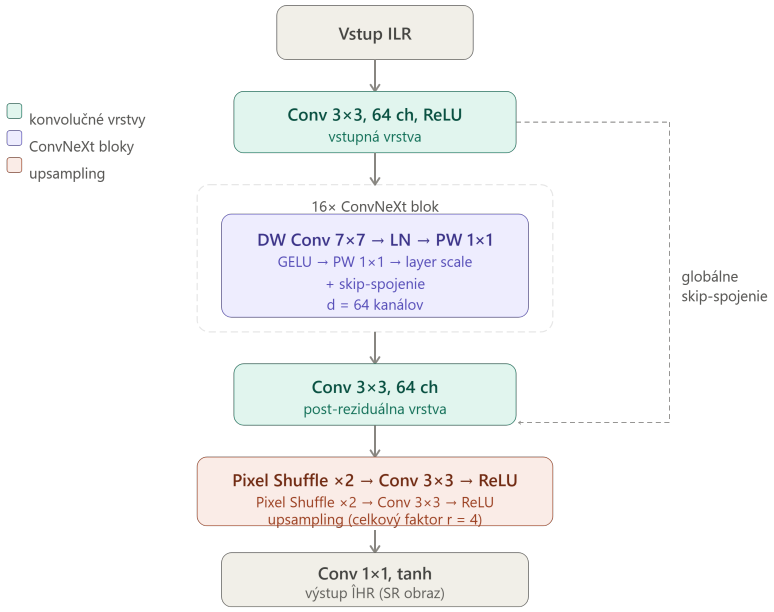
Generátor ConvNeXt-SRGAN preberá celkovú schému toku dát z pôvodného SRGAN, pričom jadro tvorené reziduálnymi blokmi nahrádza blokmi ConvNeXt. Sieť pozostáva zo štyroch funkčných celkov.

1. **Vstupná konvolučná vrstva:** Vstupný LR obraz $I_{LR} \in \mathbb{R}^{H \times W \times 3}$ prechádza konvolúciou 3×3 so 64 výstupnými kanálmi a aktivačnou funkciou ReLU. Výstup tejto vrstvy s rozmermi $H \times W \times 64$ sa ukladá pre neskoršie globálne skip-spojenie.

2. **Blok ConvNeXt vrstiev:** Za vstupnou vrstvou nasleduje $B = 16$ ConvNeXt blokov zapojených sériovo. Na rozdiel od pôvodných reziduálnych blokov SRGAN tieto bloky neobsahujú dávkovú normalizáciu.

3. **Post-reziduálna časť:** Za posledným ConvNeXt blokom nasleduje konvolúcia 3×3 so 64 kanálmi bez aktivačnej funkcie. Jej výstup sa sčíta s výstupom vstupnej konvolučnej vrstvy prostredníctvom globálneho skip-spojenia, čím sa zabezpečuje priamy tok nízkofrekvenčnej informácie zo vstupu do časti upsamplingu.

4. **Upsampling a výstupná vrstva:** Priestorové zväčšenie na cieľové rozlíšenie sa realizuje dvoma sub-pixelovými konvolučnými vrstvami (pixel shuffle) [42], z ktorých každá zvyšuje priestorové rozmery dvojnásobne, čím sa dosiahne celkový škálovací faktor $r = 4$. Každá upsamplingová vrstva obsahuje konvolúciu 3×3 generujúcu $r^2 \cdot C$ príznakových máp, po ktorej nasleduje operácia pixel



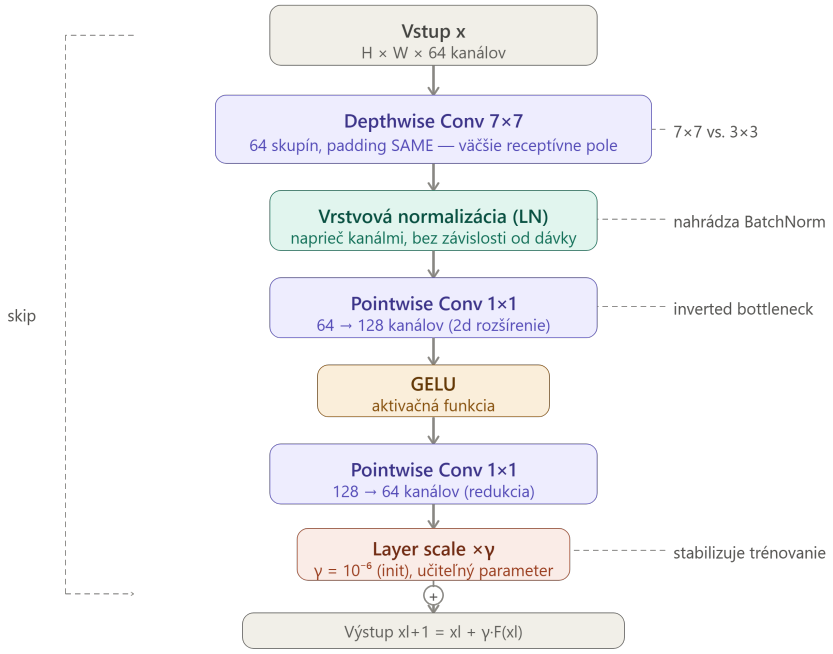
Obr. 4.9: Bloková schéma generátora ConvNeXt-SRGAN.

shuffle a aktivačná funkcia ReLU. Poslednou vrstvou je konvolúcia 1×1 s aktivačnou funkciou tanh, ktorá produkuje výsledný SR obraz $\hat{I}_{HR} \in \mathbb{R}^{rH \times rW \times 3}$ s hodnotami pixelov v intervale $[-1, 1]$.

4.2.2 ConvNeXt blok a jeho adaptácia pre SR

Pôvodný ConvNeXt blok bol navrhnutý pre klasifikačné úlohy a jeho priame použitie pre super-resolution si vyžaduje niekoľko adaptácií pre našu prácu. Implementovaný blok pozostáva z nasledujúcej postupnosti operácií:

1. **depthwise konvolúcia** 7×7 s d kanálmi (krok $s = 1$, padding `SAME`), realizovaná ako skupinová konvolúcia s počtom skupín rovným počtu kanálov ($n\_\text{group} = d$),
2. **Vrstvová normalizácia** aplikovaná pozdĺž kanálovej dimenzie,
3. **bodová konvolúcia** (*pointwise*) 1×1 rozširujúca počet kanálov na $4d$ (inverted bottleneck),
4. **aktivačná funkcia** GELU,
5. **bodová konvolúcia** 1×1 redukujúca počet kanálov späť na d,



Obr. 4.10: Vnútorná štruktúra ConvNeXt bloku v generátore.

6. **škálovanie** výstupu pomocou trénovateľného skalárom γ (layer scale) inicializovaným na hodnotu 10^{-6},

7. **reziduálne skip-spojenie** sčítavajúce výstup s pôvodným vstupom bloku.

V porovnaní s pôvodnými reziduálnymi blokmi SRGAN sa ConvNeXt blok líši v troch kľúčových aspektoch. Depthwise konvolúcia 7×7 namiesto štandardnej 3×3 zväčšuje receptívne pole každého bloku bez zvýšenia počtu parametrov v priestorovej konvolúcii. Vrstvová normalizácia nahrádza dávkovú normalizáciu a odstraňuje závislosť od štatistík dávky, čo zabraňuje vzniku normalizačných artefaktov pri inferenciách [7]. Aktivačná funkcia GELU nahrádza PReLU a poskytuje hladkejší gradientový signál pre záporné vstupy v porovnaní s funkciami rodiny ReLU [30]. Počet kanálov je $d = 64$ pre všetky ConvNeXt bloky, čo zodpovedá šírke pôvodného generátora SRGAN.

4.2.3 Architektúra diskriminátora

Diskriminátor ConvNeXt-SRGAN preberá bez zmeny architektúru diskriminátora pôvodného SRGAN. Cieľom práce je prispieť k zmene generátora, teda ostatné komponenty systému zostávajú nezmenené. Akákoľvek súčasná zmena diskriminátora by znemožnila pripisovať rozdiely vo výsledkoch konkrétnej úprave.

Diskriminátor naďalej pozostáva z ôsmich konvolučných vrstiev s dávkovou normalizáciou a aktivačnou funkciou LeakyReLU, po ktorých nasledujú dve plne prepojené vrstvy a sigmoid aktivácia produkujúca skalárnu pravdepodobnosť. Jeho úloha v systéme zostáva nezmenená. Počas adversariálneho trénovania poskytuje generátoru gradientový signál tlačiaci rekonštruované obrazy smerom k distribúcii prirodzených HR fotografií [5].

V kontexte diskriminátora dávková normalizácia nespôsobuje vizuálne artefakty v rekonštruovanom obraze, keďže diskriminátor sa nezúčastňuje priamo na syntéze výstupu, ale iba na generovaní gradientového signálu počas trénovania [4].

4.2.4 Porovnanie s pôvodným generátorom SRGAN

Navrhovaná úprava sa sústredí výlučne na nahradenie reziduálnych blokov generátora, pričom vstupná vrstva, globálne skip-spojenie, upsamplingová časť a výstupná vrstva zostávajú nezmenené. Tabuľka 4.1 zachytáva kľúčové rozdiely medzi pôvodným reziduálnym blokom SRGAN a implementovaným ConvNeXt blokom na úrovni jednotlivých komponentov.

Vlastnosť	**Reziduálny blok SRGAN**	**ConvNeXt blok (upravený pre SRGAN)**
Priestorová konvolúcia	3×3, štandardná	7×7, depthwise
Počet konvolúcií v bloku	2	3 (1 depthwise + 2 pointwise)
Rozšírenie kanálov	žiadne ($64 \to 64$)	$2\times$ ($64 \to 128 \to 64$) alebo $4\times$ ($64 \to 256 \to 64$)
Normalizácia	dávková normalizácia (BatchNorm)	vrstvová normalizácia (LayerNorm)
Aktivačná funkcia	PReLU	GELU
Škálovanie vrstvy	nie	áno ($\gamma = 10^{-6}$)
Receptívne pole bloku	5×5	7×7

Tabuľka 4.1: Porovnanie reziduálneho bloku SRGAN a ConvNeXt bloku.

Najvýznamnejším štrukturálnym rozdielom je nahradenie dvoch štandardných kon-

volúcií 3×3. Namiesto nich model používa depthwise konvolúciou 7×7 s dvoma bodovými konvolúciami 1×1. Táto zmena má dvojaký efekt. Depthwise konvolúcia spracúva každý kanál príznakových máp nezávisle, čo výrazne znižuje počet parametrov v priestorovej konvolúcii napriek väčšiemu jadru. Bodové konvolúcie následne modelujú závislosti naprieč kanálmi prostredníctvom inverted bottleneck štruktúry, ktorá najprv rozšíri počet kanálov dvojnásobne a vzápätí ho redukuje späť.

Z hľadiska počtu parametrov má depthwise konvolúcia 7×7 s d kanálmi celkovo

$$7 \times 7 \times d = 49d \tag{4.1}$$

parametrov, zatiaľ čo štandardná konvolúcia 3×3 s d vstupnými aj výstupnými kanálmi má

$$3 \times 3 \times d \times d = 9d^2 \tag{4.2}$$

parametrov. Pri $d = 64$ to zodpovedá

$$49 \cdot 64 = 3\,136, \qquad 9 \cdot 64^2 = 36\,864. \tag{4.3}$$

Depthwise konvolúcia má teda v tomto prípade takmer $12\times$ menej parametrov pri $2{,}3\times$ väčšom jadre. Napriek inverted bottleneck rozšíreniu má ConvNeXt blok pri použitej šírke $d = 64$ menej parametrov ako reziduálny blok SRGAN s dvoma plnými konvolúciami 3×3. Pri redukovanom rozšírení $2d$ použitom v hlavnom modeli ide približne o 0,27 násobok počtu parametrov reziduálneho bloku, pri rozšírení $4d$ približne o 0,49. Depthwise konvolúcia teda umožňuje väčšie receptívne pole bez nárastu počtu parametrov nad úroveň pôvodného bloku.

Zavedenie layer scale s hodnotou $\gamma = 10^{-6}$ pri inicializácii zabezpečuje, že na začiatku trénovania každý blok efektívne funguje ako identické zobrazenie. Tým sa stabilizuje trénovanie hlbokej siete a umožňuje sa postupné aktivovanie kapacity jednotlivých blokov počas optimalizácie [33].

4.3 Implementačné detaily

Predstavený je použitý framework a hardvér vrátane vlastnej implementácie vrstvovej normalizácie, ktorá nebola v použitom frameworku natívne dostupná, a následne postup prípravy trénovacích a validačných dát.

4.3.1 Framework a hardvér

Implementácia ConvNeXt-SRGAN je postavená na frameworku TensorLayerX (TLX) s backendom TensorFlow [44]. TensorLayerX je multibackendová knižnica hlbokého učenia, ktorá umožňuje zdieľanie kódu naprieč rôznymi výpočtovými backendmi bez nutnosti prepísania modelového kódu. Trénovanie prebiehalo na grafickom procesore NVIDIA A100-PCIE-40GB. Grafická karta disponuje 40 GB pamäte typu HBM2 s vysokou pamäťovou priepustnosťou, čo je dôležité najmä pri trénovaní neurónových sietí s obrazovými dátami a väčšími dávkami. GPU podporuje aj Tensor Cores tretej generácie a výpočty vo formátoch FP32, TF32 a FP16, ktoré môžu výrazne urýchliť trénovanie modelov hlbokého učenia [45].

Vlastná implementácia vrstvovej normalizácie

Štandardná implementácia vrstvovej normalizácie dostupná v TensorLayerX je navrhnutá pre sekvenčné modely a operuje pozdĺž poslednej dimenzie tenzora. Pre konvolučné siete pracujúce vo formáte `channels_first`, kde kanálová dimenzia nasleduje po dimenzii dávky, táto implementácia nie je priamo použiteľná.

Z tohto dôvodu bola v rámci tejto práce implementovaná vlastná trieda `LayerNorm2d`, ktorá aplikuje vrstvovú normalizáciu pozdĺž kanálovej dimenzie dvojrozmerných príznakových máp. Pre vstupný tenzor $\mathbf{x} \in \mathbb{R}^{N \times C \times H \times W}$, kde N je veľkosť dávky, C počet kanálov a $H \times W$ priestorové rozmery, sa normalizovaný výstup $\hat{\mathbf{x}}$ vypočíta ako

$$\hat{x}_{n,c,h,w} = \frac{x_{n,c,h,w} - \mu_{n,h,w}}{\sqrt{\sigma^2_{n,h,w} + \varepsilon}} \cdot \gamma_c + \beta_c, \tag{4.4}$$

kde

- $\mu_{n,h,w} = \frac{1}{C} \sum_{c=1}^{C} x_{n,c,h,w}$ je priemer aktivácií naprieč kanálmi pre n-tý príklad v dávke a danú priestorovú pozíciu (h, w),
- $\sigma^2_{n,h,w} = \frac{1}{C} \sum_{c=1}^{C} (x_{n,c,h,w} - \mu_{n,h,w})^2$ je zodpovedajúci rozptyl,
- $\varepsilon = 10^{-6}$ je malá konštanta zabraňujúca deleniu nulou,
- γ_c a β_c sú učiteľné skalárne parametre pre každý kanál c, inicializované na hodnoty 1 a 0.

Normalizácia sa teda vykonáva nezávisle pre každú priestorovú pozíciu a každý príklad v dávke, naprieč kanálovou dimenziou, čím sa odstraňuje závislosť od štatistík

ostatných príkladov v dávke aj od štatistík ostatných priestorových pozícií. Parametre γ_c a β_c sú tvarované na $(1, C, 1, 1)$ pred aplikáciou, aby bolo možné ich správne broadcastovanie pozdĺž priestorových dimenzií. Táto implementácia je funkčne ekvivalentná vrstvovej normalizácii používanej v pôvodnom ConvNeXt [33], avšak prispôsobená pre formát `channels_first` v rámci TensorLayerX.

4.3.2 Príprava dát

Ako trénovacia množina bol použitý dataset DIV2K [43], ktorý obsahuje 800 vysokorozlíšených RGB obrazov pokrývajúcich rozmanitú škálu prirodzených scén.

Trénovacie dvojice LR a HR sa generujú nasledovne. Z každého HR obrazu sa náhodne oreže záplata veľkosti 384×384 pixelov, na ktorú sa aplikuje náhodné horizontálne prevrátenie ako forma augmentácie. Zodpovedajúca LR záplata veľkosti 96×96 pixelov sa získa bikubickým znižovaním rozlíšenia so škálovacím faktorom $r = 4$. Hodnoty pixelov oboch záplat sa normalizujú do intervalu $[-1, 1]$ podľa vzťahu

$$x_{\text{norm}} = \frac{x - 127{,}5}{127{,}5}, \tag{4.5}$$

kde $x \in [0, 255]$ je pôvodná hodnota pixelu. Obrazy sú počas celého spracovania udržiavané vo formáte `channels_first` v súlade s konfiguráciou frameworku TensorLayerX.

4.4 Nastavenie experimentov

Táto sekcia opisuje konfiguráciu trénovacích experimentov. Popisuje obidve fázy trénovania, predtrénovanie generátora a adversariálne trénovanie. Uvádza všetky kľúčové hyperparametre, voľbu optimalizátora a kompozíciu stratových funkcií použitých počas trénovania.

4.4.1 Fázy trénovania

Trénovanie prebieha v dvoch fázach analogicky k pôvodnému postupu SRGAN [5]:

- **Predtrénovanie generátora:** V prvej fáze sa trénuje iba generátor bez diskriminátora minimalizáciou pixelovej MSE straty (1.13). Táto fáza trvá niekoľko epoch a jej cieľom je naučiť generátor základné zobrazenie z LR do HR priestoru pred začatím nestabilnejšieho adversariálneho trénovania.

- **Adversariálne trénovanie:** V druhej fáze sa trénuje celý GAN systém striedavou aktualizáciou generátora a diskriminátora po dobu 2000 epôch. Stratová funkcia generátora je zložená s perceptuálnou zložkou extrahovanou z vrstvy `pool3` (namiesto pool4) siete VGG19 [34] predtrénovanej na ImageNet [25]. V každej epoche sa rýchlosť učenia zníži desaťnásobne po uplynutí prvých 1000 epôch.

4.4.2 Hyperparametre, optimalizátor a stratové funkcie

Oba modely generátor aj diskriminátor sa trénujú optimalizátorom SGD s momentom [22] s koeficientom momentu $\mu = 0{,}9$. Počiatočná rýchlosť učenia je $\eta = 0{,}05$ so schémou postupného znižovania (*step decay*) s faktorom 0,1 každých 1000 epôch. Trénovanie hlavného modelu prebieha s veľkosťou dávky 8 (batch size).

Hyperparameter	**Hodnota**
Škálovací faktor r	4
Veľkosť HR záplaty	384×384 px
Veľkosť LR záplaty	96×96 px
Veľkosť dávky	8
Počet ConvNeXt blokov	16
Počet kanálov d	64
Rozšírenie bottleneck	$2d$
Optimalizátor	SGD s momentom ($\mu = 0{,}9$)
Počiatočná rýchlosť učenia	0,05
Step decay faktor	0,1 / 1000 epôch
Epochy predtrénovania	100
Epochy adversariálneho trénovania	2000
Váha VGG straty	$2 \cdot 10^{-6}$
Váha adversariálnej straty	10^{-3}
Perceptuálna vrstva VGG19	`pool3`

Tabuľka 4.2: Kľúčové hyperparametre trénovania ConvNeXt-SRGAN.

Stratová funkcia generátora počas adversariálneho trénovania kombinuje tri zložky:

$$L^{SR} = L_{\text{MSE}} + 2 \cdot 10^{-6}\, L_{\text{VGG}} + 10^{-3}\, L_{\text{adv}}, \tag{4.6}$$

kde L_{MSE} je stredná kvadratická chyba medzi rekonštruovaným a referenčným HR obrazom, L_{VGG} je perceptuálna strata vypočítaná na príznakových mapách vrstvy `pool3` siete VGG19 a L_{adv} je adversariálna strata diskriminátora. Na rozdiel od pôvodného modelu SRGAN, ktorý využíva perceptuálnu stratu z hlbšej vrstvy siete VGG19, bola v tejto práci použitá vrstva `pool3`. Váhové koeficienty $2 \cdot 10^{-6}$ a 10^{-3} sú z originálnej implementácie SRGAN. Stratová funkcia diskriminátora je binárna krížová entropia.

Súhrn kľúčových hyperparametrov jednej z implementácií je uvedený v tabuľke 4.2.

4.5 Experimentálne varianty

Počas vývoja modelu bolo otestovaných niekoľko konfigurácií s cieľom nájsť vyváženie medzi kapacitou modelu, nárokmi na pamäť grafického procesora a stabilitou trénovania. Experimenty sa sústredili na dve oblasti: veľkosť bottleneck rozšírenia v ConvNeXt bloku a kombináciu veľkosti dávky s počtom epôch predtrénovania.

4.5.1 Variant s redukovaným bottleneck rozšírením

Pôvodný ConvNeXt blok [33] používa inverted bottleneck so štvornásobným rozšírením kanálov ($4d$), čo pri $d = 64$ zodpovedá 256 výstupným kanálom prvej bodovej konvolúcie. Pri trénovaní s väčšou veľkosťou dávky alebo pri obmedzenej pamäti GPU môže táto konfigurácia narážať na pamäťové limity, keďže príznakové mapy rozšírené na $4d$ kanálov musia byť udržiavané v pamäti pre všetky bloky počas spätného šírenia chyby.

Z tohto dôvodu bol otestovaný variant s redukovaným rozšírením $2d$, kde prvá bodová konvolúcia generuje iba 128 kanálov namiesto 256. Táto zmena znižuje pamäťové nároky strednej časti každého bloku na polovicu, čo pri 16 blokoch predstavuje nezanedbateľnú úsporu. Zároveň však redukuje reprezentačnú kapacitu bloku v kanálovej dimenzii, čo môže negatívne ovplyvniť schopnosť siete modelovať zložité závislosti medzi príznakmi.

4.5.2 Varianty veľkosti dávky a predtrénovania

Okrem architektúry bloku boli systematicky testované kombinácie veľkosti dávky a počtu epôch predtrénovania generátora. Cieľom bolo určiť, ako tieto hyperparametre ovplyvňujú stabilitu adversariálneho trénovania a kvalitu inicializácie generátora pred jeho zapojením do GAN systému. Testované konfigurácie boli:

- Batch: 8, Epochy predtrénovania: 100, Bottleneck: 2d
- Batch: 16, Epochy predtrénovania: 100, Bottleneck: 2d
- Batch: 16, Epochy predtrénovania: 300, Bottleneck: 2d
- Batch: 16, Epochy predtrénovania: 300, Bottleneck: 4d

Konfigurácia s veľkosťou dávky 8 umožňuje trénovanie s nižšími pamäťovými nárokmi, no stochastický gradientový zostup pracuje s menej reprezentatívnym odhadom

gradientu v každom kroku, čo môže predĺžiť konvergenciu. Naopak, veľkosť dávky 16 poskytuje stabilnejší gradientový signál, kladie však vyššie nároky na pamäť GPU, najmä v kombinácii s bottleneck rozšírením $4d$.

Počet epôch predtrénovania ovplyvňuje kvalitu inicializačného bodu pre adversariálne trénovanie. Pri príliš malom počte epôch (100) generátor nemusí dosiahnuť dostatočnú pixelovú vernosť a adversariálne trénovanie štartuje z nestabilného bodu. Príliš dlhé predtrénovanie (300 epôch) naopak môže viesť k tomu, že generátor sa príliš prispôsobí optimalizácii voči MSE a následná perceptuálna strata má ťažšie presadiť generovanie jemných textúr. Konfigurácia s 100 epochami predtrénovania a veľkosťou dávky 8 bola vyhodnotená ako referenčná na základe stability trénovania a dosiahnutých výsledkov.

4.5.3 Varianty počtu ConvNeXt blokov

Popri konfigurácii bottleneck rozšírenia a hyperparametrov trénovania bola skúmaná aj závislosť kvality rekonštrukcie od hĺbky generátora, konkrétne od počtu ConvNeXt blokov B (B bude označované ako počet blokov). Referenčná konfigurácia používa $B = 16$ blokov analogicky k pôvodnému SRGAN [5], ktorý taktiež obsahuje 16 reziduálnych blokov. Boli otestované dva varianty: plytší model s $B = 12$ blokmi a pôvodná veľkosť s $B = 16$ blokmi:

- **Variant** $B = 12$ znižuje výpočtovú náročnosť a pamäťové nároky generátora pri zachovaní základnej schopnosti rekonštrukcie. Menší počet blokov zároveň skracuje celkovú hĺbku siete, čo môže obmedziť efektívne receptívne pole a schopnosť zachytiť dlhodobé priestorové závislosti. Tento variant je použitý v kombinácii s: Batch: 16, Epochy predtrénovania: 300, Bottleneck: 4d.
- **Variant** $B = 16$ naopak ponecháva počet blokov rovnaký, teda 16 blokov ako v SRGAN. Potenciálne umožňuje bohatšiu reprezentáciu príznakov, no prináša vyššie nároky na pamäť a riziko pomalšej konvergencie. Tento variant je použitý v kombinácii so všetkými ostatnými modelmi.

Počet blokov B	**Batch size**	**Epochy predtrénovania**	**Bottleneck**	**Poznámka**
12	16	300	4d	kompromis medzi šírkou a hĺbkou siete
16	16	300	4d	zlyhal pri trénovaní
16	16	300	2d	väčšie predtrénovanie
16	16	100	2d	väčšia dávka
16	8	100	2d	nižšia dávka, nižšie pamäťové nároky

Tabuľka 4.3: Varianty konfigurácie ConvNeXt-SRGAN.

Počas adversariálneho trénovania modelu s $B = 16$ ConvNeXt blokmi, rozšírením bottleneck $4d$ a optimalizátorom SGD s momentom došlo ku kolapsu diskriminátora kde hodnota d_loss klesla na 0,000 približne v epoche 856 a zostala na tejto úrovni až do konca trénovania. Tento jav naznačuje, že generátor príliš rýchlo dominoval nad diskriminátorom, čo znemožnilo ďalší zmysluplný adversariálny tréning a výsledný model produkoval degenerované výstupy. Pre porovnanie, identická konfigurácia s redukovaným rozšírením $2d$ pri rovnakom počte blokov a optimalizátore nevykazovala tento problém a trénovanie prebehlo stabilne. Táto nestabilita môže byť spôsobená vyššou kapacitou modelu v kombinácii s agresívnou počiatočnou rýchlosťou učenia $\eta = 0{,}05$. Model s väčším bottleneck rozšírením $4d$ má výrazne viac parametrov, čo môže viesť k rýchlejšej dominancii generátora nad diskriminátorom v počiatočných fázach adversariálneho trénovania.

Kapitola 5

Výsledky a vyhodnotenie

Táto kapitola prezentuje výsledky experimentov s modelom ConvNeXt-SRGAN a porovnáva ich s referenčnými metódami. Vyhodnotenie zahŕňa kvantitatívne metriky PSNR a SSIM, vizuálne porovnanie rekonštruovaných obrazov a analýzu výpočtovej náročnosti jednotlivých variantov.

5.1 Kvantitatívne výsledky

Všetky modely boli vyhodnotené na validačnej množine datasetu DIV2K [43], ktorá obsahuje 100 vysokorozlíšených obrazov pokrývajúcich rozmanité prirodzené scény. Vstupné LR obrazy boli získané bikubickým znižovaním rozlíšenia so škálovacím faktorom $r = 4$. Metriky PSNR a SSIM boli vypočítané na celých RGB obrazoch v rozsahu hodnôt pixelov $[0, 255]$.

Hlavným navrhovaným modelom tejto práce je variant ConvNeXt-SRGAN s redukovaným bottleneck rozšírením $2d$, $B = 16$ ConvNeXt blokmi, veľkosťou dávky 8, 100 epochami predtrénovania a 2000 epochami trénovania. Tento variant je v nasledujúcom texte označovaný ako hlavný model. Ostatné konfigurácie slúžia ako porovnávacie body, ktoré ilustrujú vplyv jednotlivých hyperparametrov na výsledné metriky. Hlavný model dosahuje najpresvedčivejšiu vnímanú kvalitu, hoci v pixelových metrikách zaostáva za ostatnými variantmi. Tento rozpor medzi kvantitatívnymi metrikami a vnímanou kvalitou je ústrednou témou vyhodnotenia.

Metóda	**PSNR (dB)**	**Δ PSNR**	**SSIM**	**Δ SSIM**
Bikubická interpolácia	25,52	-	0,7584	-
SRGAN (Referenčný)	26,48	+0,96	0,7705	+0,012
ConvNeXt-SRGAN ($4d$, $B = 12$, predtrén. 300 ep., batch=16)	24,90	−0,62	0,7020	−0,056
ConvNeXt-SRGAN ($2d$, $B = 16$, predtrén. 300 ep., batch=16)	24,80	−0,71	0,7037	−0,054
ConvNeXt-SRGAN ($2d$, $B = 16$, predtrén. 100 ep., batch=16)	24,47	−1,04	0,6801	−0,078
ConvNeXt-SRGAN ($2d$, $B = 16$, predtrén. 100 ep., batch=8)	24,29	−1,23	0,6686	−0,090

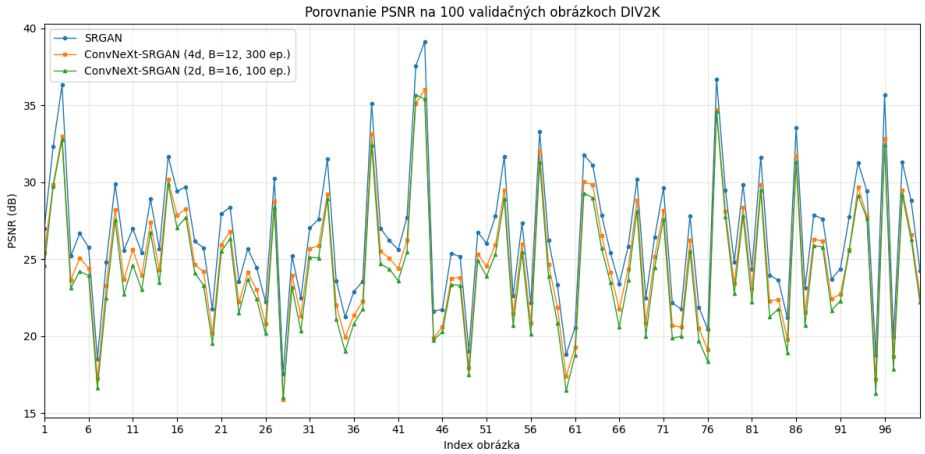
Tabuľka 5.1: Porovnanie metód na validačnej množine DIV2K (100 obrazov, škálovací faktor $r = 4$). Delta Δ vyjadruje rozdiel oproti bikubickej interpolácii. Hlavný model je zvýraznený.

5.1.1 Porovnanie hodnôt PSNR

Hodnoty PSNR pre všetky porovnávané metódy sú uvedené v tabuľke 5.1. Referenčná implementácia SRGAN dosiahla priemernú hodnotu PSNR lepšiu oproti bikubickej interpolácii a to o +0,96, zároveň dominovala nad verziami ConvNeXt-SRGAN. Tento výsledok je v súlade s očakávaniami, keďže SRGAN je inicializovaný predtrénovaním voči MSE, ktoré zabezpečuje vysokú pixelovú vernosť.

Všetky varianty ConvNeXt-SRGAN dosiahli hodnoty PSNR nižšie ako bikubická interpolácia. Hlavný model dosiahol $\Delta - 1{,}23$ dB oproti bikubickej, zatiaľ čo porovnávacie varianty mali lepšie výsledky. Pokles PSNR oproti bikubickej interpolácii je charakteristickým a očakávaným javom pre GAN-based metódy optimalizované voči perceptuálnej kvalite a priamo súvisí s kompromisom medzi perceptuálnou kvalitou a distorziou formalizovaným v [6]. Generátor zámerne uprednostňuje produkciu vierohodných vysokofrekvenčných detailov pred minimalizáciou priemernej pixelovej odchýlky. Nižšia hodnota PSNR teda v tomto kontexte neznamená horší výstup, ale

posun po osi kompromisu smerom k perceptuálnej vernosti. Na obrázku 5.11 môžeme vidieť porovnanie hlavného modelu a modelu v 4*dim, teda model s väčším bottleneck rozšírením 4d.

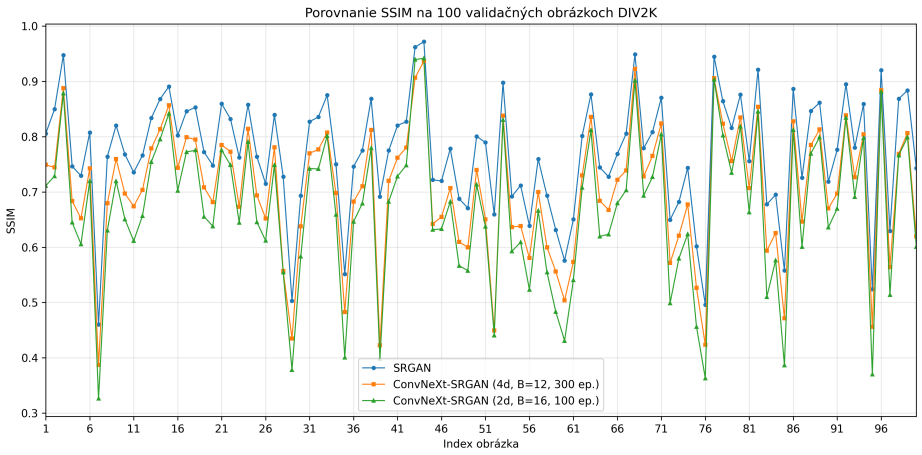


Obr. 5.11: Porovnanie hodnôt PSNR jednotlivých modelov na 100 validačných obrázkoch datasetu DIV2K pri škálovacom faktore $r = 4$.

5.1.2 Porovnanie hodnôt SSIM

V metrike SSIM dosiahol referenčný SRGAN hodnotu väčšiu o $\Delta + 0{,}012$, čo je mierne zlepšenie oproti bikubickej interpolácii . Hlavný model ConvNeXt-SRGAN dosiahol SSIM 0,6686 najnižšiu v porovnaní s inými modelmi. Rozdiel medzi ConvNeXt variantmi a referenčným SRGAN je v SSIM výraznejší ako v PSNR, čo naznačuje, že rekonštruované obrazy ConvNeXt-SRGAN vykazujú väčšie štrukturálne odchýlky od referenčných HR obrazov. SSIM je citlivejší na lokálne štrukturálne rozdiely ako PSNR, a preto lepšie zachytáva prípady, keď generátor produkuje textúry odlišné od referenčných, hoci môže byť miestami vizuálne presvedčivé.

Porovnanie ConvNeXt variantov ukazuje, že variant s väčším počtom epôch predtrénovania a väčším bottleneck rozšírením dosahuje vyššie hodnoty oboch pixelových metrík. Dlhšia MSE fáza zabezpečuje silnejší inicializačný bod voči pixelovej vernosti, čo sa prejaví vyšším PSNR aj SSIM. Tieto metriky však samy o sebe nehovoria o perceptuálnej kvalite výstupu, ktorá je predmetom nasledujúcej sekcie. Práve tu sa ukazuje, prečo bol ako hlavný model zvolený variant s nižšími pixelovými metrikami.



Obr. 5.12: Porovnanie hodnôt SSIM jednotlivých modelov na 100 validačných obrázkoch datasetu DIV2K pri škálovacom faktore $r = 4$.

5.1.3 Perceptuálna metrika LPIPS

Vizuálne pozorovania z predchádzajúcich podsekcií boli overené metrikou LPIPS [19], ktorá meria vzdialenosť v priestore príznakov predtrénovanej siete AlexNet; nižšia hodnota znamená vyššiu perceptuálnu podobnosť s referenčným obrazom. Variant LPIPS založený na sieti AlexNet bol zvolený ako štandardná kostra používaná vo väčšine súčasných prác z oblasti super-resolution (napr. ESRGAN [7], Real-ESRGAN), čo umožňuje priamu porovnateľnosť nameraných hodnôt s literatúrou.

Výsledky sú uvedené v tabuľke 5.2. Bikubická interpolácia dosiahla najväčšiu hodnotu LPIPS, a je výrazne horšie ako obe GAN-based metódy. SRGAN dosiahol LPIPS 0,1687 a hlavný model ConvNeXt-SRGAN hodnotu 0,1741. Rozdiel medzi oboma GAN metódami je menší ako štandardná odchýlka oboch meraní (SRGAN: 0,0806, ConvNeXt-SRGAN: 0,0709), takže z hľadiska perceptuálnej kvality meranej metrikou LPIPS sú obe metódy štatisticky porovnateľné.

Tento výsledok je kľúčový pre interpretáciu celkového vyhodnotenia: hlavný model ConvNeXt-SRGAN dosahuje nižšie hodnoty PSNR a SSIM ako SRGAN, no perceptuálnu kvalitu na úrovni porovnateľnej s SRGAN pri výrazne nižšej výpočtovej náročnosti. Pokles pixelových metrík teda skutočne zodpovedá posunu po osi kompromisu medzi distorziou a perceptuálnou kvalitou [6], nie zhoršeniu vnímanej kvality obrazu.

Metóda	LPIPS ↓
Bikubická interpolácia	0,4249
SRGAN	0,1687
ConvNeXt-SRGAN	0,1741

Tabuľka 5.2: Porovnanie metód na validačnej množine DIV2K (100 obrazov, škálovací faktor $r = 4$). Pri metrike LPIPS nižšia hodnota znamená vyššiu perceptuálnu podobnosť.

5.2 Výpočtová náročnosť

Trénovanie všetkých modelov prebiehalo na grafickom procesore NVIDIA A100-PCIE-40GB. Rozdiely vo výpočtovej náročnosti sú priamo pripísateľné zmene blokov generátora, keďže ostatné komponenty systému zostali nezmenené.

ConvNeXt blok nahrádza dve štandardné konvolúcie 3×3 trojicou operácií: depthwise konvolúciou 7×7 a dvoma bodovými konvolúciami 1×1. Pri $d = 64$ kanáloch má depthwise konvolúcia 7×7 celkovo 3 200 parametrov (s biasom), oproti 36 864 parametrom jednej štandardnej konvolúcie 3×3, teda takmer $12\times$ menej v priestorovej časti.

GMAC (giga-MAC, 10^9 MAC operácií) je bežne používaná jednotka na porovnávanie výpočtovej náročnosti konvolučných architektúr nezávisle od konkrétneho hardvéru či implementácie, napríklad pri návrhu efektívnych sietí ako MobileNet [46] alebo EfficientNet [47].

Hodnoty GMACs boli vypočítané analyticky podľa štandardného vzorca pre počet MAC operácií konvolučnej vrstvy (počet výstupných priestorových pozícií krát veľkosť jadra krát počet vstupných a výstupných kanálov, s ohľadom na prípadné skupinové, t.j. depthwise, spracovanie), pri vstupnom rozlíšení 96×96 pixelov zodpovedajúcom tréningovej LR záplate. Hodnoty zahŕňajú iba konvolučné vrstvy jadra generátora, čiže depthwise a bodové konvolúcie pre ConvNeXt blok, resp. 3×3 konvolúcie pre reziduálny blok SRGAN. Príspevok normalizačných vrstiev a aktivačných funkcií je v porovnaní s konvolučnými operáciami zanedbateľný, a preto sa v tomto kontexte bežne nezarátava.

Hlavný model ConvNeXt-SRGAN má na blok 19 968 parametrov (s biasom), čo zodpovedá 0,27 násobku počtu parametrov reziduálneho bloku SRGAN (74 112). Z hľa-

diska výpočtových operácií dosahuje jadro generátora ConvNeXt-SRGAN 2,878 GMACs oproti 10,872 GMACs jadra SRGAN, teda tiež približne 0,26 násobok. Porovnanie je zhrnuté v tabuľke 5.3.

Blok	**Parametre**	**GMACs (jadro)**
Reziduálny blok SRGAN	74 112	10,872
ConvNeXt blok	19 968	2,878

Tabuľka 5.3: Porovnanie výpočtovej náročnosti.

Depthwise konvolúcia 7×7 teda poskytuje väčšie receptívne pole pri výrazne nižšej výpočtovej náročnosti. Hlavný model ConvNeXt-SRGAN je nielen perceptuálne z časti lepší, ale aj výpočtovo úspornejší než pôvodný SRGAN na úrovni jadra generátora.

5.3 Vizuálne porovnanie

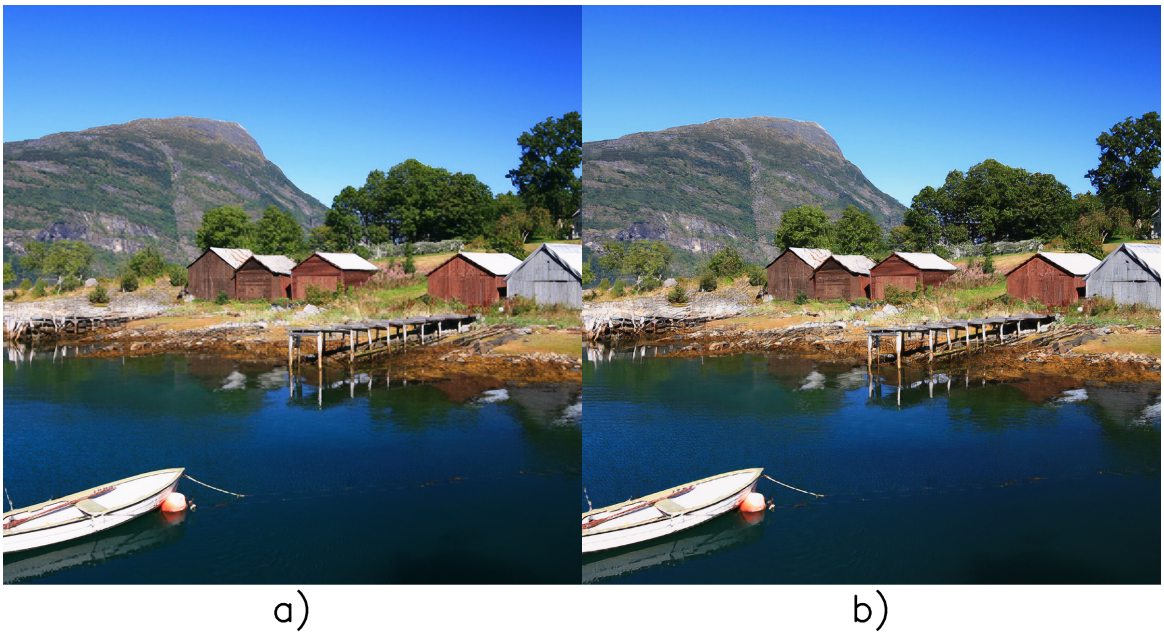
Keďže kvantitatívne metriky PSNR a SSIM hodnotia predovšetkým pixelovú a štrukturálnu zhodu s referenčným obrazom, nedokážu plne vystihnúť vnímanú kvalitu rekonštrukcie. Z tohto dôvodu je vizuálne porovnanie kľúčovou súčasťou vyhodnotenia perceptuálne orientovaných metód. V tejto sekcii sú porovnané výstupy hlavného modelu s referenčným modelom SRGAN na vybraných obrazoch z validačnej množiny DIV2K, reprezentujúcich rôzne typy scén a úrovne obtiažnosti rekonštrukcie.

5.3.1 Porovnanie na vybraných testovacích obrazoch

Na obrázku 5.13 je porovnaná rekonštrukcia scény s drevenými domami, ktorá obsahuje pravidelné štruktúry a aj nepravidelné prírodné textúry. Pri výstupe hlavného modelu ConvNeXt-SRGAN pôsobia hrany a prechody ostrejšie a lepšie ohraničené než pri SRGAN, ktorý tieto detaily mierne rozmazáva. Väčšie receptívne pole ConvNeXt blokov umožňuje sieti pri rekonštrukcii každého pixelu zohľadniť širší priestorový kontext, čo sa prejavuje konzistentnejšou rekonštrukciou opakujúcich sa štruktúr.



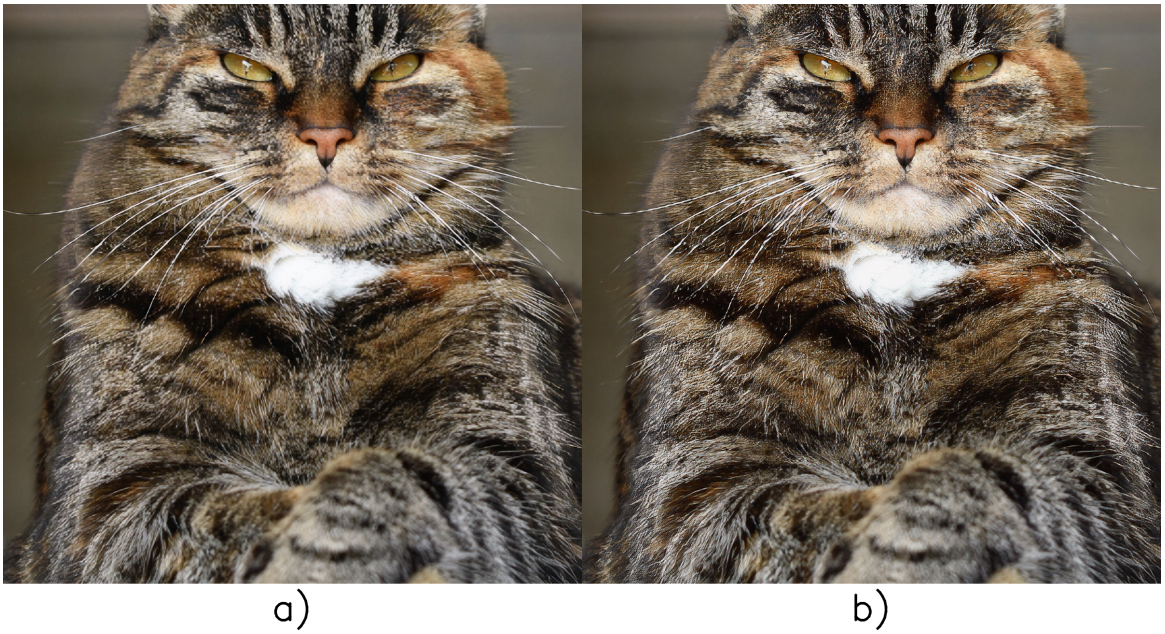
Obr. 5.13: Porovnanie metód na obraze č. 7 z validačnej množiny DIV2K. Zľava: SRGAN, ConvNeXt-SRGAN, referenčný HR obraz.



Obr. 5.14: Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na scéne s horským pozadím.

Obrázok 5.14 zachytáva širšiu scénu predošlého obrázka s horským pozadím, vegetáciou a vodnou hladinou. Rozdiely medzi metódami sú tu jemnejšie, no pri detailnom pohľade na stromy a textúru svahu produkuje hlavný model zreteľnejšie členené detaily. V oblastiach s jemnými, vysokofrekvenčnými textúrami (lístie, tráva) sa prejavuje tendencia hlavného modelu generovať ostrejšie a kontrastnejšie štruktúry, čo zodpovedá poklesu pixelových metrík pozorovanému v sekcii 5.3. Generované textúry nemusia presne zodpovedať referenčným pixelom, no pôsobia vierohodnejšie.

Najvýraznejší rozdiel medzi metódami sa prejavuje pri rekonštrukcii jemných textúr, ako je srsť. Na obrázku 5.15 hlavný model ConvNeXt-SRGAN rekonštruuje textúru srsti jemnejšie, zatiaľ čo výstup SRGAN pôsobí na týchto miestach hladšie a mierne rozmazane. Tento rozdiel je dôsledkom dvoch faktorov. Po prvé, väčšie receptívne pole ConvNeXt blokov umožňuje lepšie modelovať smerovú konzistenciu vlákien srsti. Po druhé, vrstvová normalizácia namiesto dávkovej odstraňuje normalizačné artefakty, ktoré pri SRGAN môžu potláčať jemné vysokofrekvenčné detaily. Výsledná textúra srsti pôsobí realistickejšie, hoci jej presná pixelová podoba sa odchyľuje od referenčného obrazu, čo opäť vysvetľuje nižšie hodnoty PSNR a SSIM.



Obr. 5.15: Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 68.

Obrázok 5.16 zachytáva prírodnú scénu so zamrznutým vodopádom, skalami a vodnou hladinou, teda kombináciu ostrých hrán (ľadové útvary) a jemných textúr (skalný povrch, voda). Aj tu hlavný model produkuje hrany ľadu a prechody medzi skalami s vyššou ostrosťou a vizuálne presvedčivejšími detailmi.



a) b)

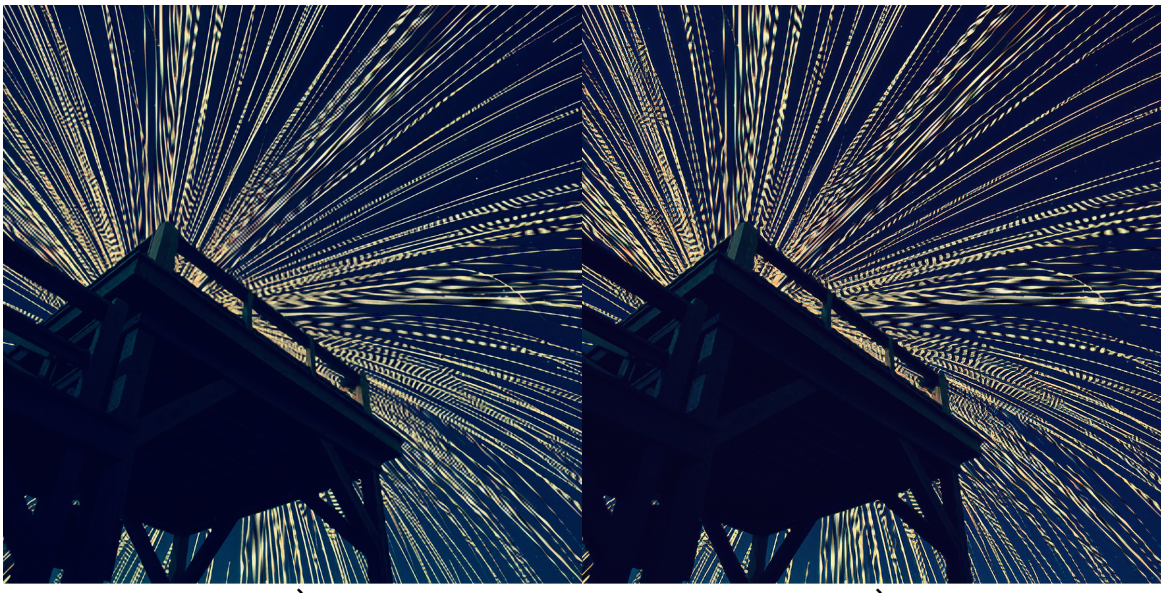
Obr. 5.16: Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 34.

Hlavný model však nie je presvedčivejší vo všetkých prípadoch, ako ukazujú obrázky 5.17 a 5.18. V týchto scénach širší kontext nepriniesol výrazné zlepšenie rekonštrukcie.



a) b)

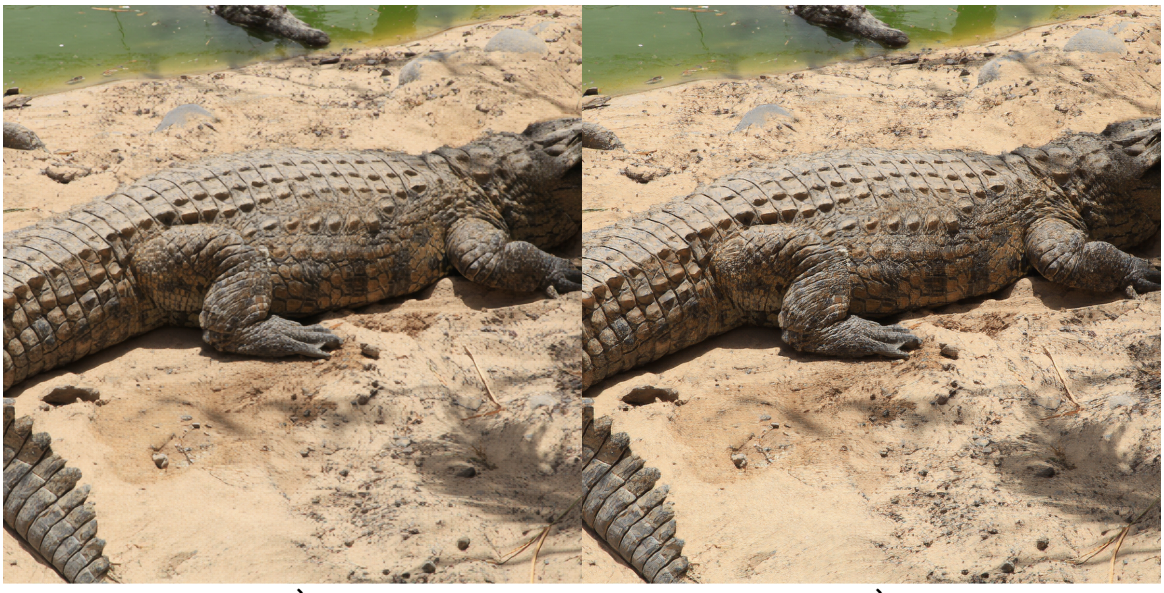
Obr. 5.17: Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 13.



Obr. 5.18: Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 27.



Obr. 5.19: Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 57.



Obr. 5.20: Porovnanie a) SRGAN a b) ConvNeXt-SRGAN na obrázku č. 58

5.4 Zhrnutie vyhodnotenia

Kvantitatívne výsledky ukazujú, že referenčná implementácia SRGAN prekonáva bikubickú interpoláciu v oboch sledovaných pixelových metrikách, čím potvrdzuje funkčnosť adversariálneho trénovania s perceptuálnou stratovou funkciou. Hlavný model ConvNeXt-SRGAN dosahuje nižšie hodnoty PSNR aj SSIM ako SRGAN baseline a bikubická interpolácia. Tento výsledok však nie je prejavom horšej kvality rekonštrukcie, ale snahou o kompromis medzi perceptuálnou kvalitou a pixelovou vernosťou.

Vizuálne porovnanie naprieč rôznymi typmi scén (prírodné textúry a srsť) ukazuje, že hlavný model niekedy produkuje realistickejšie textúry než SRGAN. Ukázalo sa to hlavne na opakujúcich sa vzoroch. Tieto perceptuálne rozdiely nedokážu metriky PSNR a SSIM zachytiť, keďže hodnotia predovšetkým pixelovú a štrukturálnu zhodu. Práve preto bol ako hlavný model práce zvolený variant s nižšími pixelovými metrikami, ale presvedčivejšou vnímanou kvalitou.

Z hľadiska výpočtovej náročnosti je hlavný model úspornejší než pôvodný SRGAN na úrovni jadra generátora, keďže ConvNeXt blok s redukovaným bottleneck rozšírením $2d$ obsahuje menej parametrov aj vykonáva menej operácií než reziduálny blok SRGAN. Hlavný príspevok navrhovanej úpravy teda spočíva v dosiahnutí kompromisu medzi perceptuálnou kvalitou, kvantitatívnymi výsledkami pri nižšej výpočtovej

náročnosti generátora.

Väčšie konvolučné jadro použité v ConvNeXt bloku umožňuje modelu pracovať so širším priestorovým kontextom. V porovnaní s menšími jadrami typu 3×3 môže jadro 7×7 zachytiť informáciu z väčšieho okolia daného pixelu, čo môže byť výhodné najmä pri rekonštrukcii prirodzených textúr, ako sú srsť, tráva, lístie alebo iné nepravidelné obrazové štruktúry. Pri takýchto oblastiach môže širší kontext pomôcť modelu lepšie odhadnúť opakujúce sa alebo štatisticky podobné vzory.

Na druhej strane, väčšie receptívne pole nemusí byť výhodné vo všetkých prípadoch. Pri veľmi jemných hranách, tenkých líniách alebo malých lokálnych detailoch môže byť dôležitá presná informácia z okolia pixelu. Ak model pri rekonštrukcii príliš zohľadňuje širší kontext, môže dôjsť k menej presnej rekonštrukcii drobných štruktúr alebo k ich čiastočnému vyhladeniu. Z tohto dôvodu možno použitie ConvNeXt blokov chápať ako kompromis medzi schopnosťou zachytiť širší kontext obrazu a potrebou zachovať presné lokálne detaily.

Porovnávacie varianty zároveň naznačujú, že ConvNeXt bloky môžu byť citlivé na nastavenie trénovania, čo otvára priestor pre ďalšie ladenie hyperparametrov v budúcej práci.

Kapitola 6

Diskusia

Táto kapitola interpretuje dosiahnuté výsledky v širšom kontexte, rozoberá silné a slabé stránky navrhnutého modelu ConvNeXt-SRGAN a načrtáva možné smery ďalšej práce.

6.1 Interpretácia výsledkov

Experimenty potvrdili, že nahradenie reziduálnych blokov generátora SRGAN blokmi ConvNeXt vedie k zmene charakteru rekonštrukcie. Hlavný model ConvNeXt-SRGAN dosahuje nižšie hodnoty PSNR a SSIM ako pôvodný SRGAN, no vizuálne porovnanie naprieč rôznymi typmi scén ukazuje miestami ostrejšie hrany a realistickejšie textúry. Tento výsledok je v súlade s teoretickým rámcom kompromisu medzi perceptuálnou kvalitou a distorziou [6]. Zlepšenie vnímanej kvality býva sprevádzané poklesom pixelovej vernosti meranej metrikami PSNR a SSIM.

Zároveň sa potvrdilo, že ConvNeXt blok s redukovaným bottleneck rozšírením $2d$ je výpočtovo úspornejší než reziduálny blok SRGAN, a to napriek väčšiemu receptívnemu poľu danému depthwise konvolúciou 7×7. Navrhnutá úprava teda nezvyšuje výpočtovú náročnosť generátora, čo je z praktického hľadiska výhodné.

6.2 Obmedzenia práce

Dosiahnuté výsledky je potrebné interpretovať s ohľadom na niekoľko obmedzení. Perceptuálne hodnotenie sa opiera predovšetkým o vizuálne porovnanie na obmedzenom počte obrazov a o metriku LPIPS, no neobsahuje rozsiahle používateľské hodnotenie (MOS), aké použili autori pôvodného SRGAN. Takéto hodnotenie by poskytlo

robustnejší dôkaz o perceptuálnej prevahe navrhnutého modelu.

Porovnávané varianty sa líšia vo viacerých hyperparametroch súčasne (bottleneck rozšírenie, počet blokov, počet epôch predtrénovania, veľkosť dávky), čo sťažuje izoláciu vplyvu jednotlivých faktorov. Systematicky by sme mali meniť vždy iba jeden parameter, čo by umožnilo presnejšie určiť, ktorý komponent najviac prispieva k výslednej kvalite.

Počas trénovania niektorých konfigurácií (najmä variantu $4d$ s optimalizátorom SGD a agresívnou rýchlosťou učenia) došlo k nestabilite adversariálneho trénovania až ku kolapsu diskriminátora. To naznačuje, že ConvNeXt bloky môžu vyžadovať odlišné nastavenie trénovania než klasické reziduálne bloky, napríklad nižšiu počiatočnú rýchlosť učenia alebo dlhšie predtrénovanie generátora.

6.3 Možné smery ďalšej práce

Navrhnutá architektúra otvára niekoľko smerov pre ďalší výskum.

Prirodzeným rozšírením je aplikácia ConvNeXt blokov na novšie a výkonnejšie architektúry, než je pôvodný SRGAN. ESRGAN [7] aj Real-ESRGAN [12] predstavujú novšie GAN-based metódy, ktoré už vo svojich generátoroch odstránili dávkovú normalizáciu a nahradili reziduálne bloky hustejšie prepojenými blokmi RRDB, čím dosahujú vyššiu perceptuálnu kvalitu než SRGAN. Integrácia ConvNeXt blokov do takejto architektúry by mohla skúmať, či väčšie receptívne pole depthwise konvolúcie prináša zlepšenie aj nad rámec týchto modernejších blokov. V prípade Real-ESRGAN by navyše bolo možné vyhodnotiť navrhnutú úpravu v kontexte slepej super-resolution s realistickým degradačným modelom, čím by sa prekonalo obmedzenie súčasnej práce SRGAN viazanej na bikubickú degradáciu.

Ďalším smerom je dôkladnejšie ladenie trénovacieho postupu pre ConvNeXt bloky, zahŕňajúce systematické hľadanie vhodnej rýchlosti učenia, dĺžky predtrénovania a stratégie vyvažovania generátora a diskriminátora. Vzhľadom na pozorovanú nestabilitu pri väčšom bottleneck rozšírení by sa dala preskúmať aj alternatívna inicializácia alebo regularizácia diskriminátora.

Napokon by rozsiahlejšie kvantitatívne vyhodnotenie na viacerých štandardných testovacích množinách, ako Set5, Set14, BSD100 a Urban100, spolu s používateľským hodnotením umožnilo objektívnejšie posúdiť prínos navrhnutého modelu v porovnaní s existujúcimi metódami super-resolution.

Model	**Rok**	**Hlavný prínos**	**Výhody / Nevýhody**
SRCNN [27]	2014	Prvá CNN metóda učiaca sa mapovanie medzi LR a HR obrazmi.	**Výhody:** jednoduchá architektúra, lepšie výsledky než klasické interpolačné metódy. **Nevýhody:** plytká sieť, malé receptívne pole, hladšie a menej detailné výstupy.
SRGAN [5]	2017	Použitie GAN a perceptuálnej straty pre fotorealistickejšiu super-resolution.	**Výhody:** vyššia vnímaná kvalita než pri metódach optimalizovaných voči MSE. **Nevýhody:** možné artefakty, nestabilnejšie trénovanie a závislosť od bikubickej degradácie.
ESRGAN [7]	2018	RRDB bloky, odstránenie dávkovej normalizácie a relativistický diskriminátor.	**Výhody:** ostrejšie detaily, menej artefaktov než SRGAN, lepšia perceptuálna kvalita. **Nevýhody:** vyššia výpočtová náročnosť a zložitejšia architektúra.
Real-ESRGAN [12]	2021	Syntetický degradačný model pre reálne a neznáme degradácie obrazu.	**Výhody:** lepšia použiteľnosť na reálne degradovaných obrazoch. **Nevýhody:** zložitejší tréning a riziko vytvárania detailov, ktoré v pôvodnom obraze nemusia byť prítomné.
SwinIR [38]	2021	Využitie Swin Transformer architektúry pri obnove obrazu.	**Výhody:** dobré modelovanie širšieho kontextu a vysoké hodnoty kvantitatívnych metrík. **Nevýhody:** vyššia pamäťová a výpočtová náročnosť.
ConvNeXt RN [39]	2023	Použitie ConvNeXt reziduálnych blokov v ľahkej SR architektúre.	**Výhody:** nízky počet parametrov a dobrá výpočtová efektívnosť. **Nevýhody:** metóda nie je zameraná na adversariálne trénovanie a perceptuálnu kvalitu.
NeXtSRGAN [41]	2025	Integrácia ConvNeXt prvkov do diskriminátora SRGAN.	**Výhody:** modernizácia diskriminátora môže zlepšiť realistickosť výstupov. **Nevýhody:** generátor zostáva založený na tradičnej architektúre.
ConvNeXt-SRGAN	2026	Náhrada reziduálnych blokov generátora SRGAN adaptovanými ConvNeXt blokmi.	**Výhody:** väčšie receptívne pole, použitie Layer Normalization a nižšia výpočtová náročnosť jadra generátora. **Nevýhody:** nižšie hodnoty PSNR/SSIM a citlivosť na nastavenie trénovania.

Tabuľka 6.1: Porovnanie navrhnutej metódy s vybranými publikovanými prácami v oblasti super-resolution.

Záver

Cieľom tejto práce bolo spracovať prehľad techník na zvýšenie rozlíšenia a rekonštrukciu obrazu, navrhnúť novú metódu založenú na generatívnych neurónových sieťach, implementovať ju a porovnať jej výsledky na vybranej dátovej sade.

V teoretickej časti práca poskytla prehľad základných princípov zvyšovania rozlíšenia obrazu, od reprezentácie digitálneho obrazu a modelu degradácie, cez klasické interpolačné metódy, až po metódy hlbokého učenia. Podrobne bol opísaný vývoj konvolučných architektúr pre super-resolution (SRCNN, FSRCNN, VDSR, SRResNet, EDSR) aj generatívnych prístupov (SRGAN, ESRGAN, Real-ESRGAN) a súčasných trendov vrátane Vision Transformerov a modernizovaných konvolučných sietí typu ConvNeXt. Samostatná kapitola sa venovala detailnej analýze architektúry a trénovacieho postupu modelu SRGAN, ktorý tvorí základ navrhovanej úpravy.

Praktickým príspevkom práce je model ConvNeXt-SRGAN, ktorý vznikol cielenou úpravou generátora pôvodného SRGAN, konkrétne nahradením reziduálnych blokov modernými ConvNeXt blokmi. Obmedzené receptívne pole štandardných konvolúcií 3×3 a artefakty spôsobené dávkovou normalizáciou priamo adresujú dva nedostatky pôvodného generátora. Depthwise konvolúcia 7×7 zväčšuje receptívne pole a vrstvová normalizácia odstraňuje závislosť od štatistík trénovacej dávky. Ostatné komponenty systému (diskriminátor, stratová funkcia a trénovací postup) zostali zámerne nezmenené, aby bolo možné izolovať príspevok samotnej úpravy generátora.

Navrhnutý model bol implementovaný vo frameworku TensorLayerX a trénovaný na datasete DIV2K. Vyhodnotenie ukázalo, že hlavný model ConvNeXt-SRGAN dosahuje nižšie hodnoty pixelových metrík PSNR a SSIM ako pôvodný SRGAN, no vizuálne porovnanie naprieč rôznymi typmi scén odhalilo miestami ostrejšie hrany a realistickejšie textúry. Tento rozpor medzi kvantitatívnymi metrikami a vnímanou kvalitou zodpovedá kompromisu medzi vizuálnou kvalitou výsledného obrazu a hodnotami číselných metrík, ktorý je charakteristický pre metódy super-resolution. Analýza výpočtovej náročnosti navyše ukázala, že navrhnutý model je na úrovni jadra

generátora úspornejší než pôvodný SRGAN.

Hlavným prínosom je systematické preskúmanie integrácie ConvNeXt blokov priamo do generátora SRGAN s adversariálnym trénovaním na prirodzených obrazoch, čo v čase začatia práce nebolo v literatúre preskúmané. Dosiahnuté výsledky naznačujú, že modernizácia stavebných blokov generátora je perspektívnym smerom, a zároveň otvárajú priestor pre ďalšiu prácu, najmä aplikáciu ConvNeXt blokov na novšie architektúry typu ESRGAN a Real-ESRGAN, dôkladnejšie ladenie trénovacieho postupu a rozsiahlejšie perceptuálne hodnotenie.

Zoznam použitej literatúry

[1] Park, Sung Cheol, Min Kyu Park a Moon Gi Kang, 2003. Super-resolution image reconstruction: a technical overview. In: *IEEE Signal Processing Magazine*. Vol. 20, no. 3, s. 21–36. doi: 10.1109/MSP.2003.1203207.

[2] Shih, Frank Y., 2010. *Image Processing and Pattern Recognition: Fundamentals and Techniques*. Hoboken, New Jersey: John Wiley & Sons a IEEE Press. ISBN 978-0-470-40461-4.

[3] Szeliski, Richard, 2022. *Computer Vision: Algorithms and Applications*. Cham: Springer. ISBN 978-3-030-34371-2.

[4] Goodfellow, Ian et al, 2014. Generative Adversarial Nets. In: *Advances in Neural Information Processing Systems*: . Curran Associates, Inc.. Vol. 27,

[5] Ledig, Christian et al, 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, . S. 105–114. doi: 10.1109/CVPR.2017.19.

[6] Blau, Yochai a Tomer Michaeli, 2018. The Perception-Distortion Tradeoff. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, . S. 6228–6237. doi: 10.1109/CVPR.2018.00652.

[7] Wang, Xintao et al, 2018. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: *Proceedings of the European Conference on Computer Vision Workshops*, . S. 63–79. doi: 10.1007/978-3-030-11021-5_5.

[8] Žára, Jiří, Bedřich Beneš, Jiří Sochor a Petr Felkel, 2004. *Moderní počítačová grafika*. Brno: Computer Press. ISBN 80-251-0454-0.

[9] Gonzalez, Rafael C. a Richard E. Woods, 2018. *Digital Image Processing*. Harlow: Pearson Education. ISBN 978-1-292-22304-9.

[10] Pratt, William K., 2007. *Digital Image Processing: PIKS Scientific Inside*. Hoboken, New Jersey: John Wiley & Sons. ISBN 978-0-471-76777-0.

[11] Chen, Honggang et al, 2022. Real-world single image super-resolution: A brief review. In: *Information Fusion*. Vol. 79, s. 124–145. ISSN 1566-2535. doi: 10.1016/j.inffus.2021.09.005.

[12] Wang, Xintao, Liangbin Xie, Chao Dong a Ying Shan, 2021. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. In: *ICCV Workshops*, . S. 1905–1914. doi: 10.1109/ICCVW54120.2021.00217.

[13] Lim, Bee, Sanghyun Son, Heewon Kim, Seungjun Nah a Kyoung Mu Lee, 2017. Enhanced Deep Residual Networks for Single Image Super-Resolution. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, . S. 1132–1140. doi: 10.1109/CVPRW.2017.151.

[14] Rasool, M. J. Aashik, Shabir Ahmad, Sevara Mardieva, Sumaiya Akter a Taeg Keun Whangbo, 2025. A Comprehensive Survey on Real-Time Image Super-Resolution for IoT and Delay-Sensitive Applications. In: *Applied Sciences*. Vol. 15, no. 1, s. 274. doi: 10.3390/app15010274.

[15] Dong, Chao, Chen Change Loy a Xiaoou Tang, 2016. Accelerating the Super-Resolution Convolutional Neural Network. In: *Computer Vision – ECCV 2016*. Cham: Springer International Publishing. Vol. 9906, s. 391–407. ISBN 978-3-319-46475-6. doi: 10.1007/978-3-319-46475-6_25.

[16] Kim, Jiwon, Jung Kwon Lee a Kyoung Mu Lee, 2016. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, . S. 1646–1654. doi: 10.1109/CVPR.2016.182.

[17] Bevilacqua, Marco, Aline Roumy, Christine Guillemot a Marie Line Alberi-Morel, 2012. Low-Complexity Single-Image Super-Resolution Based on Non-negative Neighbor Embedding. In: *Proceedings of the British Machine Vision Conference*: . BMVA Press, s. 135.1–135.10. doi: 10.5244/C.26.135.

[18] Wang, Zhou, Alan C. Bovik, Hamid R. Sheikh a Eero P. Simoncelli, 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. In: *IEEE Transactions on Image Processing*. Vol. 13, no. 4, s. 600–612. doi: 10.1109/TIP.2003.819861.

[19] Zhang, Richard, Phillip Isola, Alexei A. Efros, Eli Shechtman a Oliver Wang, 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, . S. 586–595. doi: 10.1109/CVPR.2018.00068.

[20] LeCun, Yann, Yoshua Bengio a Geoffrey Hinton, 2015. Deep Learning. In: *Nature*. Vol. 521, no. 7553, s. 436–444. doi: 10.1038/nature14539.

[21] Goodfellow, Ian, Yoshua Bengio a Aaron Courville, 2016. *Deep Learning*. Cambridge, Massachusetts: MIT Press. ISBN 978-0-262-03561-3.

[22] Rumelhart, David E. Geoffrey E. Hinton a Ronald J. Williams, 1986. Learning Representations by Back-Propagating Errors. In: *Nature*. Vol. 323, s. 533–536. doi: 10.1038/323533a0.

[23] Kingma, Diederik P. a Jimmy Ba, 2014. Adam: A Method for Stochastic Optimization. In: *arXiv preprint arXiv:1412.6980*.

[24] Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever a Ruslan Salakhutdinov, 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In: *Journal of Machine Learning Research*. Vol. 15, s. 1929–1958.

[25] Krizhevsky, Alex, Ilya Sutskever a Geoffrey E. Hinton, 2012. ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances in Neural Information Processing Systems*: . Curran Associates, Inc.. Vol. 25,

[26] He, Kaiming, Xiangyu Zhang, Shaoqing Ren a Jian Sun, 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In: *Proceedings of the IEEE International Conference on Computer Vision*, . S. 1026–1034. doi: 10.1109/ICCV.2015.123.

[27] Dong, Chao, Chen Change Loy, Kaiming He a Xiaoou Tang, 2014. Learning a Deep Convolutional Network for Image Super-Resolution. In: *Computer Vision – ECCV 2014*. Cham: Springer International Publishing. Vol. 8692, s. 184–199. ISBN 978-3-319-10593-2. doi: 10.1007/978-3-319-10593-2_13.

[28] Maas, Andrew L. Awni Y. Hannun a Andrew Y. Ng, 2013. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In: *Proceedings of the 30th International Conference on Machine Learning*.

[29] Nair, Vinod a Geoffrey E. Hinton, 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In: *Proceedings of the 27th International Conference on Machine Learning*, . S. 807–814.

[30] Hendrycks, Dan a Kevin Gimpel, 2016. Gaussian Error Linear Units (GELUs). In: *arXiv preprint arXiv:1606.08415*.

[31] Ioffe, Sergey a Christian Szegedy, 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37, s. 448–456.

[32] Ba, Jimmy Lei, Jamie Ryan Kiros a Geoffrey E. Hinton, 2016. Layer Normalization. In: *arXiv preprint arXiv:1607.06450*.

[33] Liu, Zhuang, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell a Saining Xie, 2022. A ConvNet for the 2020s. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, . S. 11976–11986. doi: 10.1109/CVPR52688.2022.01167.

[34] Simonyan, Karen a Andrew Zisserman, 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *International Conference on Learning Representations*.

[35] Johnson, Justin, Alexandre Alahi a Li Fei-Fei, 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In: *Computer Vision – ECCV 2016*. Cham: Springer International Publishing. Vol. 9906, s. 694–711. doi: 10.1007/978-3-319-46475-6_43.

[36] Dosovitskiy, Alexey et al, 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: *International Conference on Learning Representations*.

[37] Liu, Ze et al, 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, . S. 10012–10022. doi: 10.1109/ICCV48922.2021.00986.

[38] Liang, Jingyun, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool a Radu Timofte, 2021. SwinIR: Image Restoration Using Swin Transformer. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, . S. 1833–1844. doi: 10.1109/ICCVW54120.2021.00210.

[39] Zhang, Yong, Haomou Bai, Yaxing Bing a Xiao Liang, 2023. Lightweight Image Super-Resolution with ConvNeXt Residual Network. In: *Neural Processing Letters*. Vol. 55, no. 7, s. 9545–9561. doi: 10.1007/s11063-023-11213-4.

[40] Ma, Zhuang, Jianwei Ding, Shuai Li a Bin Zou, 2023. Super-resolution of underwater sonar image based on generative adversarial network. In: *IECON 2023 – 49th Annual Conference of the IEEE Industrial Electronics Society*, . S. 1–6. doi: 10.1109/IECON51785.2023.10312168.

[41] Park, Sung-Wook, Se-Hoon Jung a Chun-Bo Sim, 2025. NeXtSRGAN: enhancing super-resolution GAN with ConvNeXt discriminator for superior realism. In: *The Visual Computer*. Vol. 41, s. 7141–7167. doi: 10.1007/s00371-024-03797-2.

[42] Shi, Wenzhe et al, 2016. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, . S. 1874–1883. doi: 10.1109/CVPR.2016.207.

[43] Agustsson, Eirikur a Radu Timofte, 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, . S. 126–135. doi: 10.1109/CVPRW.2017.150.

[44] Lai, Cheng, Jiarong Han a Hao Dong, 2021. TensorLayer 3.0: A Deep Learning Library Compatible With Multiple Backends. In: *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, . S. 1–3.

[45] NVIDIA Corporation, 2020. *NVIDIA A100 Tensor Core GPU Architecture*. Dostupné na: https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/nvidia-ampere-architecture-whitepaper.pdf.

[46] Howard, Andrew G. et al, 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In: *arXiv preprint arXiv:1704.04861*.

[47] Tan, Mingxing a Quoc Le, 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: *Proceedings of the 36th International Conference on Machine Learning*, . S. 6105–6114.