



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**ZVÝŠENIE ROZLIŠENIA OBRAZU POMOCOU HLBO-  
KÝCH NEURÓNOVÝCH SIETÍ**

IMAGE SUPER-RESOLUTION USING DEEP LEARNING

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. TOMÁŠ MOJŽIŠ**

**VEDOUĆÍ PRÁCE**

SUPERVISOR

**Ing. MICHAL ŠPANĚL, Ph.D.**

BRNO 2022

## Zadání diplomové práce



Student: **Mojžiš Tomáš, Bc.**  
Program: Informační technologie a umělá inteligence  
Specializace: Inteligentní systémy  
Název: **Zvýšení rozlišení obrazu pomocí hlubokých neuronových sítí  
Image Super-Resolution Using Deep Learning**  
Kategorie: Zpracování obrazu  
Zadání:

1. Seznamte se s problematikou hlubokých neuronových sítí a jejich učení.
2. Zorientujte se v současných super-resolution technikách využívajících hlubokých neuronových sítí pro umělé zvýšení rozlišení pořízeného obrazu.
3. Vytvořte datovou sadu pro vlastní experimenty.
4. Vyberte vhodné metody a navrhnete architekturu neuronové sítě pro danou úlohu.
5. Experimentujte s vaší implementací a případně navrhnete vlastní modifikace metod.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručný plakát prezentující vaši práci, její cíle a výsledky.

### Literatura:

- Wang *et al.*, "Deep Learning for Image Super-resolution: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021 (<https://www.computer.org/csdl/journal/tp/2021/10/09044873/1iqPppDb0SQ>).
- Yang *et al.*, "Deep Learning for Single Image Super-Resolution: A Brief Review", *IEEE Transactions on Multimedia*, 2019 (<https://dl.acm.org/doi/10.1109/TMM.2019.2919431>).

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních tří bodů zadání.

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

Vedoucí práce: **Španěl Michal, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 18. května 2022

Datum schválení: 2. listopadu 2021

## Abstrakt

Cielom tejto práce je vytvoriť hlbokú neurónovú sieť schopnú zvýšiť rozlíšenie obrazov získaných elektrónovými mikroskopmi. Jedná sa o úlohu zväčšenia rozlíšenia na základe jedného obrázku. Skladá sa z dvoch častí - nájdenie vhodných dát a vytvorenie dátovej sady pre danú úlohu a návrh architektúry neurónovej siete, ktorá je schopná riešiť úlohu zvýšenia rozlíšenia obrazu.

V práci vznikli dve dátové sady pozostávajúce z obrázkov pochádzajúcich z elektrónových mikroskopov. Dátové sady majú rôzny prístup k augmentácii dáta. Je vďaka nim možné natrénovať neurónové siete pre úlohu zvýšenia rozlíšenia obrazu. Pri riešení boli otestované dve architektúry typu U-Net a jedna typu GAN. Rozlíšenie bolo zvyšované dvakrát a štyrikrát.

Najlepšie umelo zväčšené obrázky z testovaných architektúr vytvára sieť Real-ESRGAN. V hodnotách metrík nie je lepšia ako testovaná interpolačná metóda, obrázky však často pôsobia kvalitnejšie, najmä pri zvýšení rozlíšenia štyrikrát.

Vďaka tejto práci bola vytvorená dátová sada, na ktorej je možné trénovať ďalšie architektúry a zlepšovať tak kvalitu výsledných obrázkov. Neurónové siete z tejto práce je možné použiť pri získavaní kvalitnejších dát z elektrónového mikroskopu pri nízkom rozlíšení.

## Abstract

The aim of this thesis is to create a deep neural net capable of super-resolution on images acquired by electron microscopes. The thesis consists of two parts - finding appropriate data and creating a dataset for the super-resolution task and designing a neural net architecture capable of solving the super-resolution task.

Within the thesis, two datasets comprised of images acquired by electron microscopes were created. The datasets differ in the approach to data augmentation. They allow to train a neural network which fulfills the super-resolution task. To solve this task, two U-Net based and one GAN based architecture were trained. The resolution of images was upscalled by a factor of two and four.

The best artificially upscalled images were created by neural network Real-ESRGAN. The values of metrics were not higher than the tested interpolation method, but the images seem more visually pleasing especially when they were upscalled four times.

Thanks to this thesis, two datasets were created allowing to train other possible neural network architectures to improve the quality of the artificially upscalled images. The neural networks trained in this thesis can be utilized in the process of acquiring higher quality data from low resolution electron microscope images.

## Kľúčové slová

hlboká neurónová sieť, konvolučná neurónová sieť, hlboké učenie, počítačové videnie, zvýšenie rozlíšenia obrazu, elektrónová mikroskopia, gan, unet, dátová sada

## Keywords

deep neural network, convolutional neural network, deep learning, computer vision, super-resolution, electron microscopy, gan, unet, dataset

## Citácia

MOJŽIŠ, Tomáš. *Zvýšenie rozlíšenia obrazu pomocou hlbokých neurónových sietí*. Brno, 2022. Diplomová práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Španěl, Ph.D.

# Zvýšenie rozlíšenia obrazu pomocou hlbokých neurónových sietí

## Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Michala Španěla, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....  
Tomáš Mojžiš  
15. mája 2022

## Podakovanie

Chcel by som poďakovať vedúcemu diplomovej práce, doktorovi Michalovi Španělovi za poskytnuté informácie a materiály. Ďalej by som chcel poďakovať firme TESCAN Brno, sro za poskytnutie náhľadu do práce s elektrónovými mikroskopmi. Tiež chcem poďakovať firme TESCAN 3DIM, sro za navrhnutie zadania, sprostredkovanie komunikácie s firmou TESCAN Brno, sro a za odborné konzultácie. Výpočtové zdroje boli poskytnuté projektom „e-Infrastruktura CZ“ (e-INFRA CZ LM2018140) podporovaná Ministerstvom školství, mládeže a tělovýchovy České Republiky.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Elektrónový mikroskop</b>	<b>3</b>
2.1	Komponenty mikroskopu . . . . .	3
2.2	Základné typy elektrónových mikroskopov . . . . .	5
2.3	Akvizícia obrazu pomocou SEM . . . . .	9
<b>3</b>	<b>Zvýšenie rozlíšenia obrazu</b>	<b>12</b>
3.1	Definícia problému . . . . .	13
3.2	Interpolácia . . . . .	13
3.3	Metriky . . . . .	14
<b>4</b>	<b>Hlboké neurónové siete pre zvýšenie rozlíšenia obrazu</b>	<b>17</b>
4.1	Špeciálne prvky neurónových sietí používané pri zvýšení rozlíšenia obrazu . . . . .	17
4.2	Typy architektúr neurónových sietí pre zvýšenie rozlíšenia obrazu . . . . .	20
4.3	Chybové funkcie . . . . .	22
4.4	Konkrétne architektúry schopné zvýšenia rozlíšenia obrazu . . . . .	23
<b>5</b>	<b>Návrh riešenia pre zvýšenie rozlíšenia obrazu z elektrónového mikroskopu</b>	<b>32</b>
5.1	Obrázky z elektrónových mikroskopov . . . . .	32
5.2	Architektúry . . . . .	35
5.3	Použité chybové funkcie . . . . .	36
<b>6</b>	<b>Implementácia</b>	<b>39</b>
<b>7</b>	<b>Experimenty</b>	<b>46</b>
7.1	U-Net a RUNet pri zvýšení rozlíšenia dvakrát . . . . .	47
7.2	RUNet a Real-ESRGAN pri zvýšení rozlíšenia dvakrát . . . . .	49
7.3	RUNet a Real-ESRGAN pri zvýšení rozlíšenia štyrikrát . . . . .	53
7.4	Real-ESRGAN pri dvoch rôznych veľkostiach tréningových obrázkov . . . . .	56
7.5	Real-ESRGAN na reálnych dátach . . . . .	59
<b>8</b>	<b>Záver</b>	<b>62</b>
	<b>Literatúra</b>	<b>63</b>
<b>A</b>	<b>Obsah pamäťového média</b>	<b>67</b>

# Kapitola 1

## Úvod

Zvýšenie rozlíšenia obrazu je známa a často používaná technika. Či už ide o približovanie obrázkov na telefóne alebo inom zariadení, alebo zvýšenie rýchlosti renderovania obrazu. Zvýšenie rozlíšenia sa tiež dá využiť pri predspracovaní obrazových dát pre iné úlohy strojového učenia ako je segmentácia obrazu, detekcia objektov a podobne. Problém tejto úlohy spočíva v tom, že hľadaná informácia sa priamo v obraze nenachádza, treba ju odvodiť na základe toho, čo poznáme.

Pri práci s elektrónovým mikroskopom je možné nastaviť veľké množstvo parametrov snímania. Ovplyvňujú kvalitu získaného obrázku, ale aj dĺžku procesu snímania. Ak sú nastavené parametre tak, aby mal vzniknutý obrázok nízke rozlíšenie, je možné použiť zvýšenie rozlíšenia obrazu pre urýchlenie analýzy danej snímky. To platí v prípade, že zvýšenie rozlíšenia obrazu prebehne rýchlejšie ako snímame obrázku s vyšším rozlíšením. Tiež je ale možné, že pri tomto procese vzniknú nechcené artefakty, alebo sa stratia maličké, ale dôležité detaily.

Cielom práce je navrhnúť a natrénovať neurónovú sieť, schopnú zvýšiť rozlíšenie jedného vstupného obrázku. V rámci tejto práce vznikli dve dátové sady, ktoré obsahujú snímky z elektrónového mikroskopu z rovnakého zdroja. Dátové sady však rôzne prístupujú k vytváraniu a následnej augmentácii obrázkov. S dátovými sadami boli uskutočnené experimenty s dvoma rôznymi typmi neurónových sietí U-Net a GAN. Sieť typu GAN je pri zvýšení rozlíšenia štyrikrát schopná generovať obrázky, ktoré pôsobia lepšie ako obrázky zväčšené pomocou bikubickej interpolácie. Pri drobných detailoch však môžu vzniknúť nechcené artefakty.

Kapitola 2 sa venuje problematike elektrónových mikroskopov. Je tu popísané z čoho sa skladajú, sú tu popísané základné typy elektrónových mikroskopov a princíp ich fungovania. Tiež sa tu nachádza časť, ktorá sa venuje opisu charakteru takýchto obrázkov.

V kapitole 3 sú definované klasické metódy a metriky pre úlohu zvýšenia rozlíšenia obrazu.

V poslednej teoretickej kapitole 4 sú zhrnuté rôzne typy architektúr neurónových sietí, vrstiev a chybových funkcií používaných pre zvýšenie rozlíšenia obrazu.

V kapitole 5 je návrh riešenia daného problému. Je tu popísaný proces hľadania použiteľných dát a následného vytvorenia dátovej sady. Tiež sú tu spomenuté rôzne architektúry neurónových sietí použité v implementačnej časti práce.

Posledná kapitola obsahuje niekoľko experimentov s navrhnutými dátovými sadami a architektúrami neurónových sietí a porovnanie ich výsledkov. Pri experimentoch sa porovnáva schopnosť neurónových sietí zvýšiť rozlíšenie obrazu dvakrát a štyrikrát. Experimenty sa tiež zaoberajú aj odstraňovaním šumu pri zvýšení rozlíšenia.

## Kapitola 2

# Elektrónový mikroskop

Informácie v tejto kapitole pochádzajú z [4] a [9]. Elektrónová mikroskopia je technológia určená pre získanie extrémne jemných detailov alebo informácie o ultraštruktúre najmä biologických vzoriek. Pre ľudí priniesla úplne novú perspektívu, keď sa biologická ultraštruktúra ukázala ako dynamické a architektonicky komplexné usporiadanie makromolekúl. Vďaka elektrónovej mikroskopii boli vizualizované zložité komponenty bunky s ich biochemickou aktivitou. Taktiež umožnila pozorovať a ďalej klasifikovať jednotlivé vírusy. Objasnila sa štrukturálna organizácia DNA v chromozómoch a elektrónové mikroskopy poskytli náhľad na vývoj rôznych chorôb.

Existujú dva základné typy elektrónových mikroskopov. Rastrovací elektrónový mikroskop (ďalej SEM - *scanning electron microscope*) slúži na vizualizáciu povrchu vzoriek. Naopak transmisný elektrónový mikroskop (ďalej TEM - *transmission electron microscope*) je určený pre získanie informácií o interných aj externých vlastnostiach extrémne tenkej vzorky. SEM získava informácie pomocou elektrónových lúčov, ktoré sú postupne bod po bode vysielané na povrch vzorky. Výsledný obrázok tak pôsobí ako trojrozmerný. V TEM sú elektróny vysielané cez vzorku a preto vzniká dvojrozmerný obraz internej štruktúry vzorky. Napriek tomu, že tieto princípy sú pomerne odlišné, oba používajú akcelerované elektróny a elektromagnetické šošovky. Pri oboch typoch elektrónových mikroskopov je vyžadované aby boli elektróny vysielané cez vákuum, inak by sa kvôli ich malej hmotnosti mohli zrážať s molekulami vzduchu, čo by znížilo kvalitu získaného obrazu. Obrázky bývajú zaznamenané na fotografický film, alebo digitálne.

V nasledujúcich podkapitolách budú bližšie popísané časti, základné typy elektrónových mikroskopov a princíp ich fungovania.

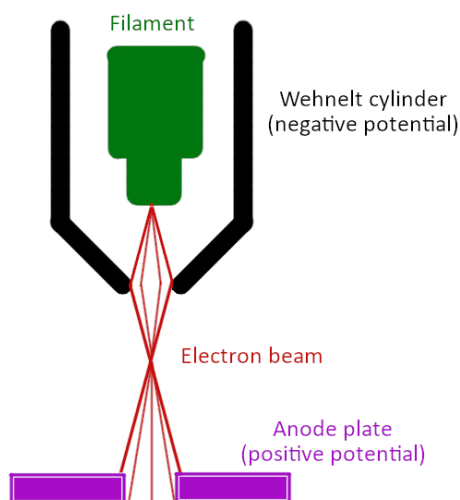
### 2.1 Komponenty mikroskopu

Jeden z prvých elektrónových mikroskopov bol zostrojený už v roku 1931. Jednalo sa o transmisný typ elektrónového mikroskopu. Princípiálne je toto zariadenie pomerne jednoduché. Využíva niekoľko znalostí z oblasti fyziky. V nasledujúcich častiach sú popísané jeho komponenty.

#### Zdroj elektrónov

Pre generovanie elektrónov je potrebný silný a stabilný zdroj napätia. Elektróny sa generujú v urýchľovacej trubici (nazývanej *electron gun*). Funguje to na princípe ako klasická žiarovka - filament z prvku s požadovanými vlastnosťami (napr. tungsten) sa zahrieva pomocou

jednosmerného prúdu, čo spôsobí vypudenie valenčných elektrónov z filamentu. Elektróny sú akcelerované elektrickým poľom paralelným s optickou osou. Toto elektrické pole je vytvorené aplikovaním potenciálového rozdielu medzi katódou (napr. tungstenový filament) v obale (napr. Wehneltov cylinder) a anódou - okrúhla kovová doštička s dierou uprostred (vertikálne pod katódou). Princíp je znázornený na obrázku 2.1. Veľa z akcelerovaných elektrónov narazí do doštičky a len zhruba 1% elektrónov úspešne prejde cez dieru. Napätie lúča pri TEM je obvykle 1% emitovaného prúdu z katódy. Elektróny sú potom usmernené pomocou sústavy elektromagnetických šošoviek.



Obr. 2.1: Princíp zdroja elektrónov.

Okrem generovania elektrónov je možné použiť iný zdroj a generovať silnejší lúč, pomocou ktorého je možné fyzicky zasiahnuť do vzorky (narezať ju alebo vytvoriť rôzne značky) pre účely prípravy vzoriek pre mikroskop typu TEM.

### Elektromagnetické šošovky

Šošovky usmerňujú elektróny na princípe vytvárania elektromagnetického poľa. V elektrónových mikroskopoch sa používa niekoľko typov elektromagnetických šošoviek.

**Kondenzné šošovky** (angl. *condenser lenses*) sa nachádzajú pod zdrojom elektrónov. Účel sústavy kondenzných šošoviek je regulovať šírku elektrónového lúča.

**Objektové šošovky** (angl. *objective lenses*) upravujú finálnu šírku elektrónového lúča a zaostrujú ho na potrebný bod.

### Fluorescenčná obrazovka

Ide o obrazovku používanú pri TEM pokrytú fluorescenčným materiálom. Po zasiahnutí obrazovky elektrónom sa vyžaruje svetlo. Na túto obrazovku je premietaný výsledný obraz. Táto obrazovka je zachytávaná kamerou, ktorá umožňuje uložiť vytvorený obraz.



## Detektor

Pri SEM je potrebné detegovať sekundárne a späťne rozptýlené elektróny, ktoré vznikajú po kolízii elektrónového lúča s preparátom (viac v časti 2.2). Detektor vytvára elektromagnetické pole, ktoré priťahuje elektróny. Rôzne typy elektrónov majú značne odlišnú energiu, čo umožní nastaviť silu magnetického poľa a zachytávať tak vybraný typ elektrónov.

## Držiak preparátu

Ide o mechanický prvok, ktorého účel je držať preparát. Držiak je možné ovládať z vonku mikroskopu. Pohyb je možný vo všetkých osiach vrátane rotácie. Presnosť pohybu musí byť v jednotkách mikrometrov. Okrem pohybu majú tieto držiaky aj ďalšie funkcie odvíjajúce sa od odvetvia, v ktorom je mikroskop použitý. Biologické držiaky umožňujú napríklad schladiť alebo úplne zmraziť preparát. V niektorých odvetviach je užitočné zohrievanie preparátu, kde je možné skúmať vplyv teploty na jeho vnútornú štruktúru.

## Kamery

Okrem kamery, ktorá zachytáva fluorescenčnú obrazovku sa v elektrónových mikroskopoch nachádza aj kamera na sledovanie stavu preparátu. Často sa tu nachádza niekoľko kamier - pre sledovanie z rôznych uhlov.

## Vákuový systém

Ako bolo spomenuté vyššie, vo vnútri elektrónového mikroskopu sa musí nachádzať vákuum. Okrem toho, že by sa elektróny mohli zrážať s inými časticami v systéme, je dôležité aby pri zahrievaní filamentu nebol prítomný kyslík. Často sa jedná o systém viacerých púmp, ktoré dokážu vytvoriť silné vákuum.

## Riadiaci počítač

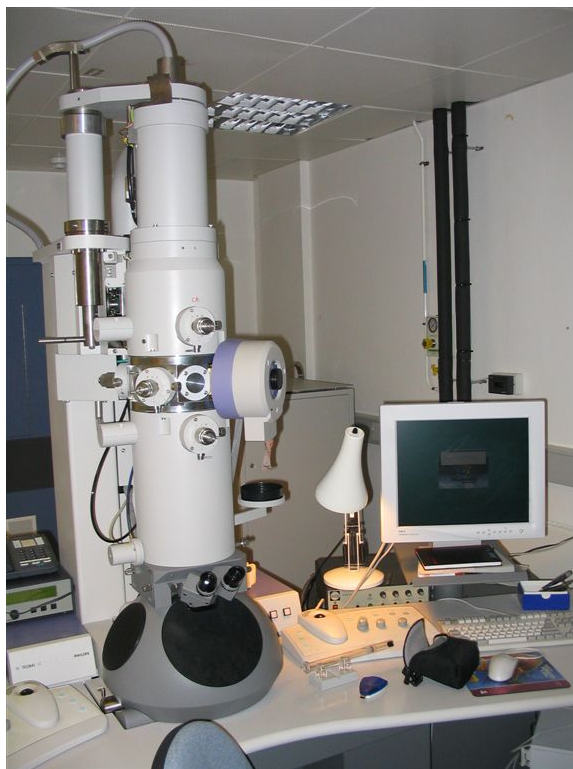
Úlohou počítača je ovládať celý proces snímania a kontrolovať stav jednotlivých komponent pri snímaní. Obvykle sú riadiace počítače dodávané ako súčasť mikroskopu. Elektrónové mikroskopy sú drahé zariadenia, preto je potrebné počítač zabezpečiť pri pripojení do siete. Ak je počítač pripojený do internetu, je možné mikroskop ovládať aj diaľkovo.

## 2.2 Základné typy elektrónových mikroskopov

Existuje niekoľko typov elektrónových mikroskopov. Dva základné typy sú TEM a SEM. Tieto dva typy mikroskopov sú principiálne odlišné. Princípy je však možné kombinovať, čoho príkladom je mikroskop typu STEM. V nasledujúcich častiach sú tieto tri typy mikroskopov bližšie popísané.

### TEM

Konvenčný transmisný elektrónový mikroskop generuje lúč elektrónov zahrievaním obvykle tungstenového filamentu. Elektróny majú často rýchlosť rovnajúcej sa zhruba polovičnej rýchlosti svetla. S touto rýchlosťou vstúpia do magnetického poľa prvej a druhej šošovky, ktoré sústredia elektróny na preparát.



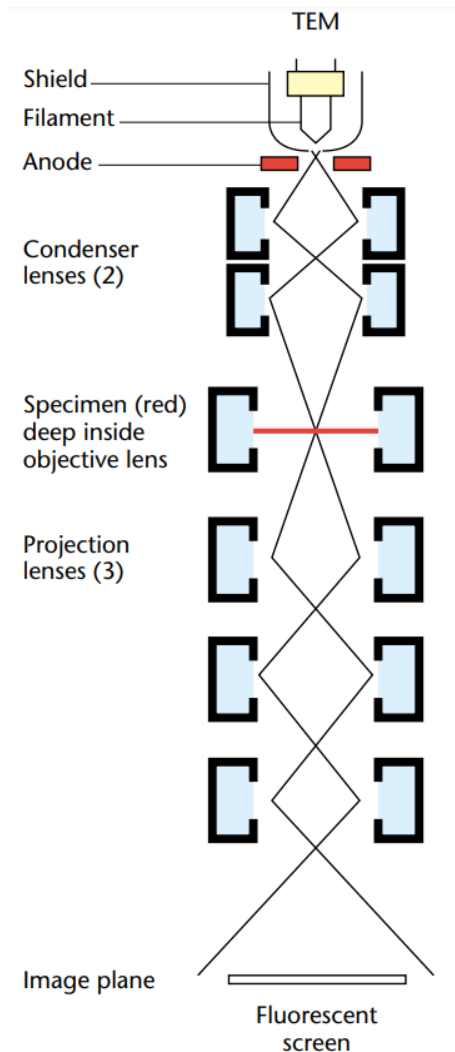
Obr. 2.2: Fotografia mikroskopu typu TEM. Obrázok prevzatý z [35].

Kondenzné a objektové šošovky generujú magnetické pole závislé na množstve prúdu ktorý nimi prúdi. Upravením množstva prúdu sa mení ohnisková vzdialenosť šošoviek, čím sú elektróny sústredené na konkrétne miesto študovaného preparátu.

Po zasiahnutí preparátu elektrónmi sa elektróny odrazia pod rôznymi uhlami. Ak je preparát koncentrovanejší elektróny sa odrazia tak, že sa dostanú mimo optickej osi mikroskopu. Tieto oblasti sú na výslednom obraze tmavé. V oblastiach, kde je preparát menej koncentrovaný sa elektróny rozptýlia pod malým uhlom, čo vo výsledku spôsobí, že sú svetlejšie vo výslednom obraze.

Keď sa elektrónový lúč vynorí z preparátu je zachytený objektovou šošovkou. Tento obraz obsahuje informácie o štruktúre preparátu. Informácie sú potom zväčšené systémom projekčných šošoviek. Elektróny sú potom zobrazené na fluorescenčnej obrazovke. Čím väčší počet elektrónov, tým svetlejší obraz. Obraz je možné aj fotograficky zachytiť vystavením fotografického filmu priamo lúču elektrónov. Ten reaguje na elektróny podobne ako na fotóny. Fotografický film je potom zachytený digitálnou kamerou. Tento princíp je znázornený na obrázku 2.3.

Z mikroskopov typu TEM vznikajú obrázky s veľkým priblížením a vysokým rozlíšením s množstvom detailov. Príprava preparátov je však časovo aj technologicky náročná. Preparáty musia mať hrúbku do 100 nanometrov. Polovodičové rezy sú vyrábané pomocou zostreňujúceho lúču iónov. Biologické vzorky sú chemicky zafixované, dehydrované a vložené do polymérovej živice pre ich zastabilizovanie, aby ich bolo možné deliť na ultra tenké časti.



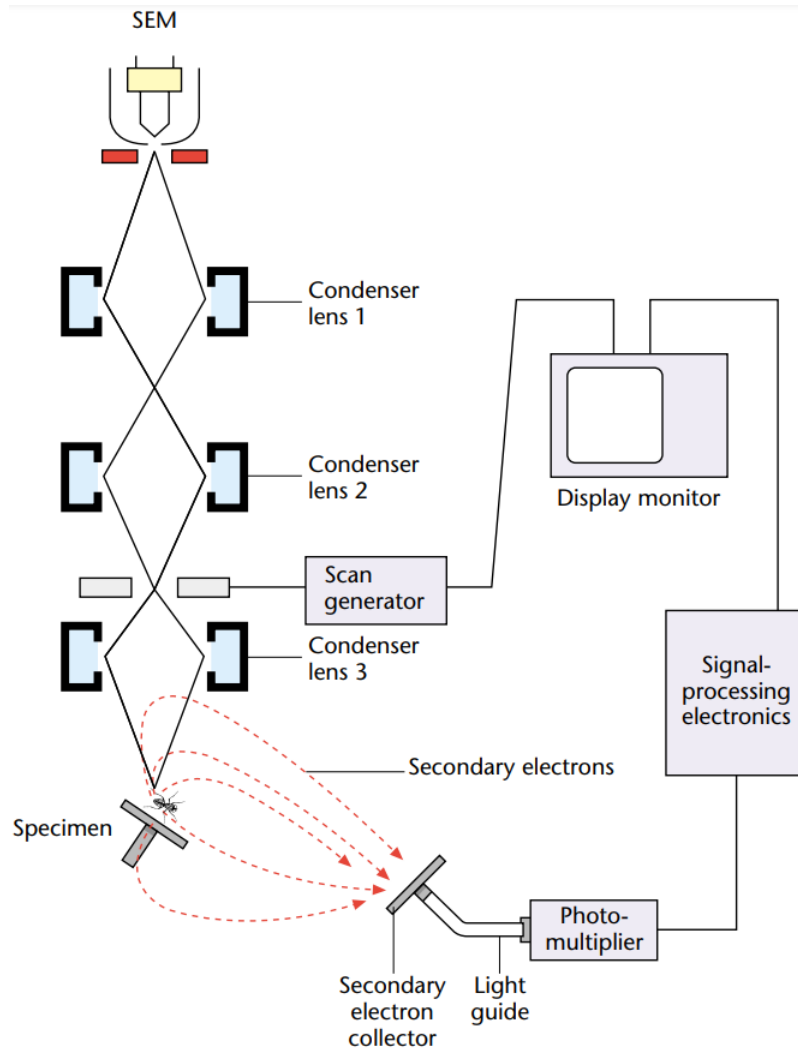
Obr. 2.3: Schéma popisujúca základné komponenty a princíp TEM. Obrázok prevzatý z [4].

## SEM

Jednou z najväčších nevýhod TEM je, že pokiaľ nie je preparát dostatočne tenký, materiál elektróny úplne absorbuje alebo priveľmi odrazí, čo spôsobí stratu informácie. Táto skutočnosť viedla k iniciatíve skonštruovať elektrónové mikroskopy, pomocou ktorých by bolo možné skúmať aj hrubé preparáty. Takéto mikroskopy musia fungovať na princípe klasického mikroskopu, ale poskytovať omnoho vyššiu možnosť priblíženia.

Zdroj elektrónov a šošovky SEM sú takmer identické s tými, ktoré sa používajú v TEM. Pri SEM šošovky zaostrí lúč elektrónov na veľmi malú časť preparátu. Každá zo šošoviek znižuje lúč elektrónov. Keď elektrónový lúč narazí na preparát, spôsobí vypudenie nízkoenergetických, sekundárnych elektrónov z povrchu preparátu. Sekundárne elektróny nesú informácie o povrchu vzorky a sú používané na vygenerovanie obrazu. Schéma sa nachádza na obrázku 2.4.

Narozdiel od TEM, ktorý ožiari celý preparát jedným veľkým bodom, SEM postupne skenuje povrch preparátu po bodoch vychyľovaním elektrónového lúča. Tento princíp je podobný ako v CRT televízoroch. V každom bode, kam dopadne lúč elektrónov, je vygenero-



Obr. 2.4: Schéma popisujúca základné komponenty a princíp SEM. Obrázok prevzatý z [4].

vaných niekoľko sekundárnych elektrónov. Množstvo je závislé hlavne na topografii a uhle vstupu elektrónového lúča do preparátu. Sekundárne elektróny sú priťahované k detektoru sekundárnych elektrónov, ktorý má vysoké kladné napätie - okolo 12 000 V. Na detektore je umiestený fosforeskujúci film, ktorý pri zachytení elektrónov vytvára výboj svetla. Tento výboj svetla cestuje do fotónového násobiča, kde sú fotóny konvertované na foto-elektróny a slabý signál je zosilnený. Oblasti, ktoré generujú veľké množstvo sekundárnych elektrónov sa na výslednom obraze zobrazia ako jasné body. Niekoľko odtieňov šedej farby dodávajú dojem trojrozmerného obrazu podobne ako čierne biele fotografie.

Pri SEM sa nesledujú však len sekundárne elektróny, ale aj späť rozptýlené elektróny (angl. *back-scattered electrons* - BSE). Jedná sa o elektróny, ktoré nesú väčšiu energiu ako sekundárne elektróny. Oba druhy elektrónov majú významný vplyv na výzor finálneho obrazu.

Pri vizualizácii sú teda použité dva elektrónové lúče. Jeden skenuje povrch preparátu a druhý rastruje obraz na obrazovke. Pre každý bod na obrazovke existuje korešpondujúci bod na povrchu vzorky. Veľkosť obrazovky je fixná, ale veľkosť skenovanej oblasti je v ru-

kách operátora. Zaostrovanie pri SEM spočíva v nastavovaní veľkosti bodu usmerňovacími šošovkami v závislosti s úrovňou priblíženia.

Obraz z TEM mikroskopov má obvykle vyššie rozlíšenie ako pri SEM, ale je omnoho zložitejšie pripraviť tenký preparát.

SEM boli vyvinuté pár rokov po TEM, ale trvalo dlhšie ich prispôbiť ako praktickú pomôcku pri vedeckom výskume. Dnes ich počet prevyšuje TEM a sú využívané v mnohých oblastiach - medicínsky a materiálny výskum, polovodičový priemysel a forenzno-vedeckých laboratóriách.

## STEM

Skenovacia technika SEM je taktiež použiteľná pri skúmaní tenkej vzorky. Ide teda o spojenie princípov oboch predchádzajúcich mikroskopov. V dnešnej dobe väčšina TEM obsahuje skenovacie doplnky, čím je možné zabezpečiť duálny mód (TEM/STEM).

Pri STEM je elektrónový lúč zaostrý na body o veľkosti menšej ako 1 nanometer a vzorka je skenovaná bod po bode. Výhodou tohto princípu je, že je možné získať dodatočné informácie zo vzorky. Ide napríklad o informácie o generovaní sekundárnych elektrónov, strate energie elektrónov alebo o charakteristiku röntgenového žiarenia. Oproti SEM poskytujú tieto mikroskopy oveľa vyššie priestorové rozlíšenie.

## 2.3 Akvizícia obrazu pomocou SEM

Operátor SEM je schopný nastaviť množstvo parametrov snímania ako napríklad akceleračné napätie, vzdialenosť preparátu od šošovky alebo množstvo prúdu pretekajúceho cez šošovky, čím sa mení hrúbka elektrónového lúča.

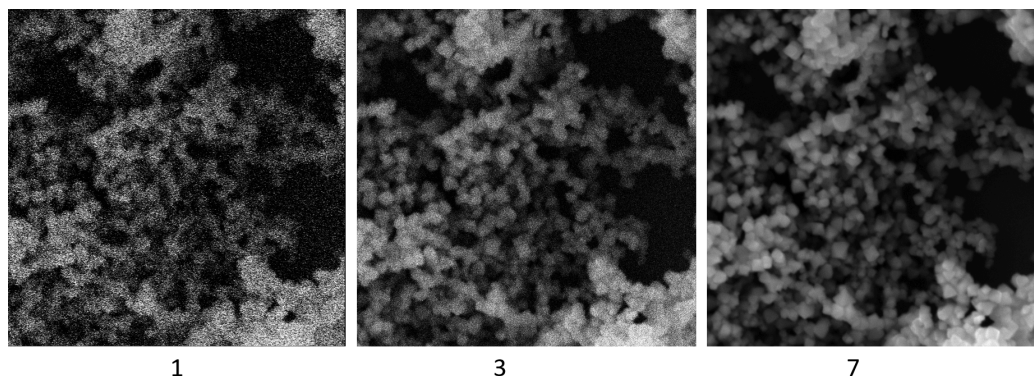
Rýchlosť akcelerácie elektrónov udáva kinetickú energiu primárnych elektrónov, čím je možné nastaviť penetračnú hĺbku. Sekundárne elektróny sú však generované vo veľmi malej hĺbke pod povrchom vzorky, preto by obraz zhotovený pomocou sekundárnych elektrónov nemal závisieť od nastavenia tohto parametru. Avšak spätne rozptýlené elektróny môžu spôsobiť generovanie sekundárnych elektrónov vo väčšej hĺbke (nie však väčšej, ako je hĺbka penetrácie primárnych elektrónov). Tým je možné získať informácie spod povrchu vzorky.

### Charakter získaných obrázkov z SEM

Táto časť obsahuje informácie z demonštrácie, ktorú poskytla firma TESCAN Brno s.r.o. Kvalita obrázkov závisí od parametrov snímania. Mikroskop je možné nastaviť tak, aby konečný obrázok bol veľmi kvalitne zaostrý a obsahoval minimum šumu. Akvizícia takéhoto obrázku je však časovo náročnejšia - je zložité nastaviť všetky parametre správne a aj samotný proces získania obrazu môže byť zdĺhavejší. Kvalita obrazu tiež závisí od účelu. Jeden účel je získavanie rýchleho obrazu v náhľadovom okne operátora a druhý je získavanie finálnej snímky, ktorá má omnoho vyššiu kvalitu, čomu odpovedá aj čas snímania. V závislosti od nastavenia môže celý tento proces trvať desiatky sekúnd až niekoľko desiatok minút. Toto môže spôsobiť problémy napríklad pri biologických vzorkách, ktoré sú síce zafixované v substráte, ale nie vždy to úplne zastaví všetky biologické procesy a vzorka tak môže pomaly meniť svoj vzhľad.

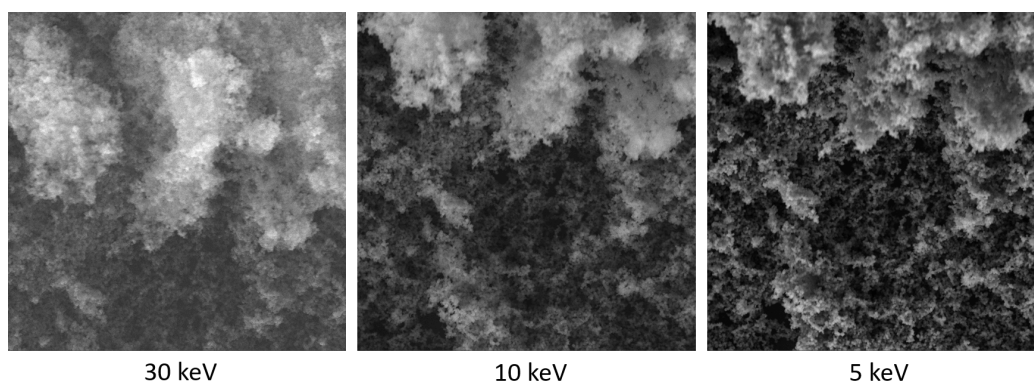
Z princípu snímania obrazu často vzniká poissonov šum. Množstvo šumu sa dá operátorovi redukovať napríklad spriemerovaním niekoľkých snímok po sebe. Spomaľuje to však proces snímania. Pri niektorých vzorkách to ani nie je možné, pretože dlhodobým vystavo-

vaním vzorky lúču elektrónov by mohlo dôjsť k poškodeniu vzorky. Porovnanie získaných snímok kde je použité spriemerovanie hodnôt sa nachádza na obrázku 2.5.



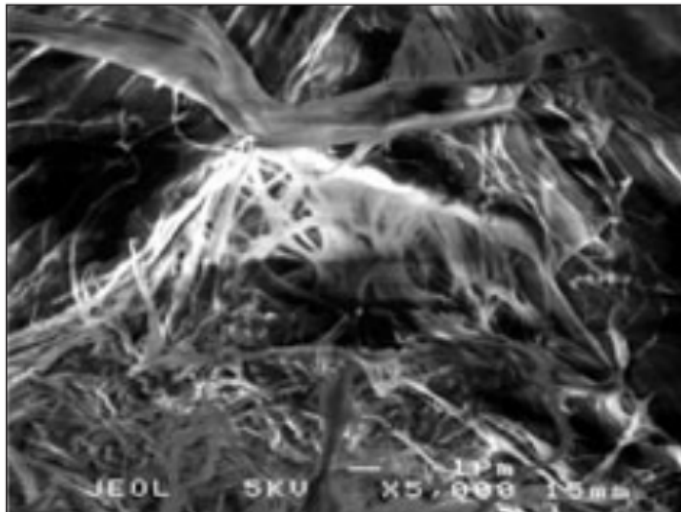
Obr. 2.5: Porovnanie snímok kedy je použité spriemerovanie niekoľkých snímok. Počet spriemerovaných snímok je 1, 3 a 7 (v tomto poradí). Šum je omnoho výraznejší bez použitia spriemerovania. Obrázky poskytla firma TESCAN Brno s.r.o.

Pre získanie kvalitného a ostrého obrazu je potrebné, aby na vzorku narazil dostatočný počet elektrónov. Počet elektrónov zvýšime rozšírením elektrónového lúča. Avšak širší elektrónový lúč spôsobuje že 1 bod na snímke je väčší a celkovo sa tak zníži rozlíšenie výslednej snímky. Ak by sme zvýšili rýchlosť akcelerácie elektrónov tak by ich za rovnaký časový okamih dopadlo viac, avšak s ich väčšou energiou by sa mohli dostať hlbšie pod povrch vzorky, čo nie je vždy ideálne. Niektoré vzorky by sa mohli takýmto nastavením ľahko poškodiť, alebo úplne zničiť. Porovnanie snímok s rôznym nastavením energie elektrónov sa nachádza na obrázku 2.6.



Obr. 2.6: Porovnanie snímok kedy je použitá iná energia elektrónov. Energia elektrónov je 30 keV, 10 keV a 5 keV (v tomto poradí). Obrázky poskytla firma TESCAN Brno s.r.o.

Ďalším problémom je, ak je vzorka schopná meniť svoj elektrický potenciál a dopadajúce elektróny ju nabíjajú. Zmena potenciálu ovplyvňuje sekundárne elektróny s malým nábojom, čo má negatívny dopad na ich detekciu. Vo výsledku obrázok vypadá tmavšie alebo svetlejšie, závisí či má vzorka kladný alebo záporný potenciál. Na obrázku 2.7 je snímka vzorky, ktorý mení svoj potenciál. Kvôli zmene potenciálu niektoré časti obrázku pôsobia príliš tmavo (elektrický potenciál produkovaný nabíjaním vzorky narušil trajektórie sekundárnych elektrónov a tie neboli zachytené detektorom) alebo príliš svetlo.



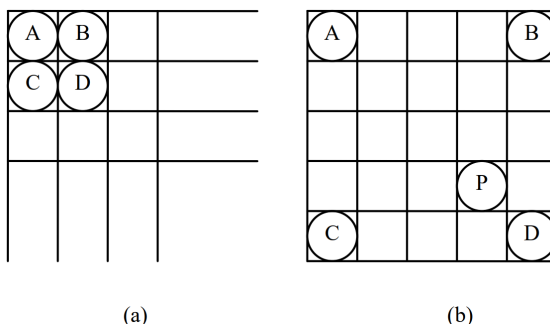
Obr. 2.7: Nežiadaný jav, kedy je elektrický potenciál vzorky zvýšený dopadajúcimi elektrónmi. Tmavé časti v obrázku sú spôsobené narušením trajektórie sekundárnych elektrónov elektrickým potenciálom vytvoreným nabitou vzorkou. Obrázok prevzatý z [17].

Je potrebné aby bol operátor schopný nájsť správny pomer medzi rozličnými nastaveniami akvizície. Jedine tak je schopný vytvoriť kvalitnú snímku, ktorá je ostrá, málo zašumená a sú v nej viditeľné potrebné detaily.

## Kapitola 3

# Zvýšenie rozlíšenia obrazu

Častou metódou zvýšenia kvality obrazu je zvýšenie jeho rozlíšenia (angl. *super-resolution* - SR). Cieľom tohto procesu je zobrať vstupný obrázok (jeden alebo viacero) s nízkym rozlíšením (angl. *low-resolution* - LR) a vytvoriť obrázok s vysokým rozlíšením (angl. *high-resolution* - HR). Náročnosť tohto procesu sa odvíja od požadovanej miery zvýšenia rozlíšenia. Super-resolution sa dá rozdeliť na dve hlavné kategórie. Prvou je SISR (*single image super-resolution*), teda zvýšením rozlíšenia len jedného obrázku, ktorou sa zaoberá táto práca. Naopak úlohou MISR (*multiple image super-resolution*) je vytvoriť obrázok s vyšším rozlíšením za použitia viacerých obrázkov s nízkym rozlíšením. Obvykle sú výsledky MISR kvalitnejšie ako SISR, pretože MISR má viac obrázkov rovnakého subjektu, tým pádom má viac informácií. Pri zvýšení rozlíšenia obrazu je potrebné nájsť hodnoty pre novovzniknuté pixely, ktoré sa pôvodnom obrázku nenachádzali, viď. obrázok 3.1.



Obr. 3.1: Hodnoty a poloha pixelov A, B, C, D v pôvodnom obrázku sú vľavo (a). Pri zväčšení rozlíšenia 4x (alebo 400% priblížení), obrázok vpravo (b), vznikne množstvo nových pixelov medzi pôvodnými A, B, C, D, ktorým je potrebné priradiť hodnotu. Obrázok prevzatý z [11].

Zvýšiť rozlíšenie obrazu sa dá viacerými spôsobmi. Niektoré klasické, neadaptívne spôsoby sú popísané v časti 3.2. V dnešnej dobe sa často pre túto úlohu používajú neurónové siete, tento proces je popísaný až v nasledujúcej kapitole 4.

Pre zmeranie kvality procesu zvýšenia rozlíšenia obrazu sa používajú metriky bližšie popísané v časti 3.3.



### 3.1 Definícia problému

Informácie v tejto podkapitole pochádzajú z [33].

Všeobecne sa dá LR obrázok  $I_x$  definovať ako výsledok nasledujúcej degradácie:

$$I_x = \mathcal{D}(I_y; \delta), \quad (3.1)$$

kde  $\mathcal{D}$  je mapovacia funkcia degradácie,  $I_y$  je korešpondujúci HR obrázok a  $\delta$  sú parametre degradačného procesu (mierka zmeny veľkosti alebo šum). Vo väčšine prípadov sú proces degradácie a jeho parametre neznáme a sú dodané len LR obrázky. V tomto prípade, známom ako slepé zvýšenie rozlíšenia obrazu, je potrebné odhadnúť HR odhad  $\hat{I}_y$  pôvodného HR obrázku  $I_y$  z LR obrázku  $I_x$ :

$$\hat{I}_y = \mathcal{F}(I_x; \theta), \quad (3.2)$$

kde  $\mathcal{F}$  je model pre zvýšenie rozlíšenia obrazu a  $\theta$  sú jeho parametre.

Napriek tomu, že degradačný proces je neznámy a môže byť ovplyvnený viacerými faktormi (kompresné artefakty, anizotropická degradácia, sensorový šum), výskumníci sa snažia modelovať mapovaciu funkciu degradácie. Na degradáciu sa dá pozeráť ako na jedinú operáciu podvzorkovania:

$$\mathcal{D}(I_y; \delta) = (I_y) \downarrow_s, \{s\} \subset \delta, \quad (3.3)$$

kde  $\downarrow_s$  je operácia podvzorkovania s mierkou  $s$ . Väčšina dátových sád pre SR je vybudovaných na tomto vzore. Najčastejšie používanou metódou podvzorkovania je bikubická interpolácia s anti-aliasingom (viac v časti 3.2). Avšak niektoré práce [38] modelujú degradáciu ako kombináciu niekoľkých operácií:

$$\mathcal{D}(I_y; \delta) = (I_y \otimes \kappa) \downarrow_s + n_\varsigma, \{\kappa, s, \varsigma\} \subset \delta, \quad (3.4)$$

kde  $(I_y \otimes \kappa)$  značí konvolúciu rozmazávacieho jadra (angl. *blur kernel*)  $\kappa$  a pôvodného HR obrázku  $I_y$  a  $n_\varsigma$  je aditívny biely Gaussovský šum so štandardnou odchýlkou  $\varsigma$ . V porovnaní s jednoduchšou definíciou 3.3 je táto definícia bližšia prípadom z reálneho sveta a taktiež viac vhodná pre SR [38].

Objektívnou úlohou zvýšenia rozlíšenia obrazu je:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\hat{I}_y, I_y) + \lambda \Phi(\theta), \quad (3.5)$$

kde  $\mathcal{L}(\hat{I}_y, I_y)$  reprezentuje chybovú funkciu medzi vygenerovaným HR obrázkom  $\hat{I}_y$  a pôvodným HR obrázkom  $I_y$ .  $\lambda \Phi(\theta)$  je výraz pre reguláciu výrazu. Najznámejšou chybovou funkciou je MSE (viď. 3.6), lepšie modely využívajú kombináciu viacerých chybových funkcií (viď. 4.3).

### 3.2 Interpolácia

Informácie v tejto podkapitole pochádzajú z [11]. Pri zvýšení rozlíšenia obrazu potrebujeme preniesť pôvodný obrázok do novej siete pixelov. Ak je výsledná sieť väčšia, musíme pridať hodnoty novým pixlom, ktoré v pôvodnej mriežke neboli. Interpolácia funguje na princípe

získania hodnoty neznámych bodov na základe známych bodov v ich okolí. Výsledky interpolácie sa môžu výrazne líšiť na základe zvoleného algoritmu. V každom prípade však pri znížení rozlíšenia dôjde k strate informácie. Preto je pri zvýšení rozlíšenia potrebné čo najviac vhodnejšie túto stratenú informáciu nahradiť, aby bol výsledok vizuálne prívetivý pre oko diváka. Najbežnejšie metódy interpolácie sú znázornené na obrázku 3.2 a sú bližšie popísané v nasledujúcich podkapitolách.

### Metóda najbližších susedov

Táto metóda je najjednoduchšia. Nahradí každý chýbajúci pixel hodnotou najbližšieho suseda. Neberie ohľad na hodnoty ostatných pixelov. Takýmto spôsobom vznikajú obrázky, ktoré zanechávajú ostré detaily, ale sú kostrbaté, aj napriek tomu, že pôvodný obrázok bol hladký.

### Bilineárna interpolácia

Bilineárna interpolácia pri dvojdimenzionálnych obrázkoch vykonáva lineárnu interpoláciu v jednom a potom v druhom smere. Na základe hodnoty 4 bodov lineárne interpoluje hodnotu nového, zatiaľ neznáameho bodu. Výsledok je váhovaný priemer hodnôt pôvodných bodov na základe vzdialenosti od nového bodu.

### Bikubická interpolácia

Bikubická interpolácia je podobná bilineárnej. Avšak berie sa ohľad na väčšie okolie nového bodu. Z pôvodných  $2 \times 2$  pri bilineárnej interpolácii, bikubická interpolácia berie do výpočtu až 16 okolitých bodov (okolie  $4 \times 4$ ). Následne je rovnakým spôsobom ako pri bilineárnej interpolácii spočítaný váhovaný priemer všetkých bodov a výsledná hodnota je priradená novému bodu. Výstupný obrázok je často hladší, vizuálne prijateľnejší a obsahuje menej artefaktov ako pri predchádzajúcich metódach, ale výpočet je náročnejší. V dnešnej dobe je však optimálnou strednou cestou medzi rýchlosťou výpočtu a kvalitou výsledku.

## 3.3 Metriky

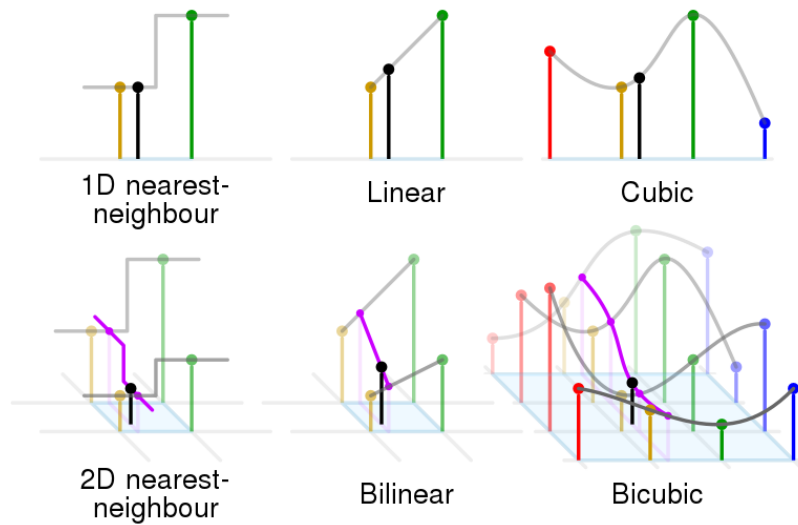
Pre porovnanie kvality metód zvyšovania rozlíšenia obrazu je potrebné nájsť vhodné metriky. Je zložité nájsť správnu metriku. Metriky často počítajú funkciu, ktorou ohodnotia odchýlku pôvodného obrazu so zväčšeným. Takáto číselná reprezentácia nemusí korešpondovať s tým, ako rozdiel vníma človek.

Podľa [15] sa metriky často delia na subjektívne a objektívne. Subjektívne sú založené na ľudskom úsudku a nemajú konkrétne referenčné kritéria. Objektívne metódy sú založené na porovnaní použitím numerického ohodnotenia.

### PSNR

Podľa [15] je *Peak-Signal-to-Noise Ratio* jednou z najpoužívanejších metrík pre hodnotenie kvality obrazových dát. Výpočet je pomerne jednoduchý. Je založený na chybe MSE (*mean squared error*) pôvodných a upravených obrazových dát. Tá je definovaná ako [15]:

$$MSE(x, y) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - y_{ij})^2, \quad (3.6)$$



Obr. 3.2: Porovnanie základných spôsobov interpolácie jednorozmerných a dvojrozmerných dát pri zmene rozlíšenia. Čierne body znázorňujú interpolovaný bod a červené, zelené, žlté a modré body znázorňujú okolité vzorky, na základe ktorých sa získa nová hodnota. Obrázok prevzatý z [36].

kde  $x_{ij}$  je hodnota pixlu v riadku  $i$  a stĺpci  $j$  referenčného obrázku,  $y_{ij}$  je hodnota pixlu v riadku  $i$  a stĺpci  $j$  vo výslednom obrázku,  $MN$  je počet riadkov a počet stĺpcov obrázku.

PSNR sa potom dá spočítať ako:

$$PSNR(x, y) = 10 \cdot \log_{10} \left( \frac{S^2}{MSE(x, y)} \right), \quad (3.7)$$

kde  $S$  je maximálna možná hodnota pixlu na obrázku, a  $MSE(x, y)$  je *mean squared error* pôvodného a výstupného obrázku. Maximálna hodnota pixlu pre 8 bitový obrázok v odtieni šedej je 255. Obrázky je možné normalizovať do intervalu  $\langle 0, 1 \rangle$ .

PSNR má opačný smer ako MSE. Ak sa MSE blíži k nule (minimálna chyba) tak PSNR sa blíži k nekonečnu. Nevýhodou je však, že PSNR hodnotí obrázky po jednotlivých pixloch. Preto môže mať lepšiu PSNR obrázok, ktorý je viac rozmazaný - viď. na obrázku 3.3.

## SSIM

*Structural Similarity* (SSIM) je ďalšia známa metrika pre meranie podobnosti dvoch obrázkov. Bola predstavená v práci Wang et al. [34]. Podľa autorov výsledná hodnota zohľadňuje vnímanie ľudským okom. Narozdiel od klasických sumačných metód (ako napr. vyššie spomínané MSE) je navrhnutá tak, že zahrňuje tri zložky - jas, kontrast a štruktúru. SSIM medzi dvojicou obrázkov  $x$  a  $y$ :

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma, \quad (3.8)$$

kde  $\alpha > 0, \beta > 0, \gamma > 0$  sú parametre, ktorými sa dá nastaviť relatívna váha pre jednotlivé zložky. Zložka jas  $l$  sa vypočíta zo stredných hodnôt porovnávaných obrázkov ako:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad (3.9)$$

kde  $\mu_x$  a  $\mu_y$  sú stredné hodnoty obrázkov. Zložka kontrastu  $c$  sa dá spočítať pomocou štandardných odchýlok:

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (3.10)$$

kde  $\sigma_x$  a  $\sigma_y$  sú štandardné odchýlky obrázkov. Poslednú zložku štruktúry  $s$  je možné spočítať pomocou korelácie:

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \quad (3.11)$$

kde  $\sigma_{xy}$  je korelácia medzi obrázkami a  $\sigma_x$  a  $\sigma_y$  sú štandardné odchýlky. Koeficienty  $C_i$  sú malé konštanty, ktoré zabezpečujú stabilné riešenie v prípade, že sa menovateľ zlomku blíži k nule. Ak zoberieme ako vstup obrázku s jedným kanálom (v odtieňoch šedej) a nastavíme konštanty  $\alpha = \beta = \gamma = 1$  a  $C_3 = C_2/2$  dostaneme zjednodušenú rovnicu, ktorá sa bežne používa ako metrika SSIM:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (3.12)$$

kde sú znaky definované rovnako ako v rovniach vyššie. Výsledok SSIM naberá hodnoty v intervale  $(0, 1)$ , pričom 0 znamená že medzi dvojicou obrázkov nie je žiadna korelácia a naopak 1 znamená že obrázky sú identické.

Podľa [15] je medzi hodnotami PSNR a SSIM jednoduché analytické spojenie, ktoré podľa experimentov v tejto práci platí pre bežné degradácie obrázkov ako sú pridanie Gaussovského šumu, Gaussovské rozostrenie a kompresie jpeg a jpeg2000. PSNR je citlivejšia na Gaussovský šum ako SSIM, naopak SSIM je citlivejšia na kompresiu jpeg. Pri Gaussovskom rozostrení a kompresii jpeg2000 je citlivosť oboch parametrov podobná. Neexistuje však žiadne všeobecné pravidlo, kedy zvoliť PSNR alebo SSIM. Závisí to od konkrétneho obrázka.



Obr. 3.3: Inkonzistencia hodnôt PSNR a SSIM voči perceptuálnej kvalite obrázka. Vľavo je obrázok, ktorého rozlíšenie bolo zvýšené pomocou jednoduchšej interpolácie algoritmom najbližšieho suseda (spomenutý v časti 3.2). V strede je obrázok, ktorý je výstupom neurónovej siete SRResNet [23], ktorej cieľom bolo dosiahnuť čo najlepšie hodnoty PSNR a SSIM. Nakoniec vpravo je výstup zo siete SRGAN [23], ktorá mala za úlohu maximalizovať perceptuálnu kvalitu obrazu. Hodnoty PSNR a SSIM sú vyššie pri interpolácii metódou najbližšieho suseda ako pri sieti SRGAN, ale vizuálny dojem hovorí presný opak. Obrázok prevzatý z [3].

## Kapitola 4

# Hlboké neurónové siete pre zvýšenie rozlíšenia obrazu

Hlboké neurónové siete sa v dnešnej dobe používajú na celé spektrum úloh. Základnou stavebnou jednotkou je neurón. Je inšpirovaný biologickým neurónom. V tomto prípade ide o matematický model, ktorý má niekoľko vstupov, telo a jeden výstup. Zo vstupov spočíta váhovanú sumu, výsledok vloží na vstup aktivačnej funkcie. Výstup z aktivačnej funkcie je aj výstup neurónu. Samotný neurón vie riešiť len lineárne separovateľné úlohy. Ak však usporiadame neuróny do neurónovej siete, je schopný riešiť aj zložitejšie problémy. Učenie neurónovej siete prebieha pomocou algoritmu spätného šírenia chyby, kedy sa spočíta akej chyby sa každý neurón dopustil a jeho váhy sú upravené v takom smere, aby sa chyba minimalizovala.

Pre úlohu spracovania obrazu sa často používajú konvolučné neurónové siete. Sú to neurónové siete, ktoré sa skladajú z konvolučných vrstiev. V konvolučnej vrstve sa počíta konvolúcia filtra so vstupným obrazom. Rozmery a počet filtrov sú hlavnými parametrami konvolučných vrstiev.

Táto kapitola sa venuje rôznym častiam a architektúram neurónových sietí pre zvýšenie rozlíšenia obrazu. V časti 4.1 sú predstavené upravené vrstvy neurónových sietí pre danú úlohu. Potom sú v časti 4.2 popísané rôzne spôsoby organizácie vrstiev neurónových sietí pre zvýšenie rozlíšenia obrazu. Ďalej je v časti 4.3 spomenutá chybová funkcia, často využívaná pri zvýšení rozlíšenia v obraze. Nakoniec je v časti 4.4 predstavených niekoľko základných architektúr pre danú úlohu.

### 4.1 Špeciálne prvky neurónových sietí používané pri zvýšení rozlíšenia obrazu

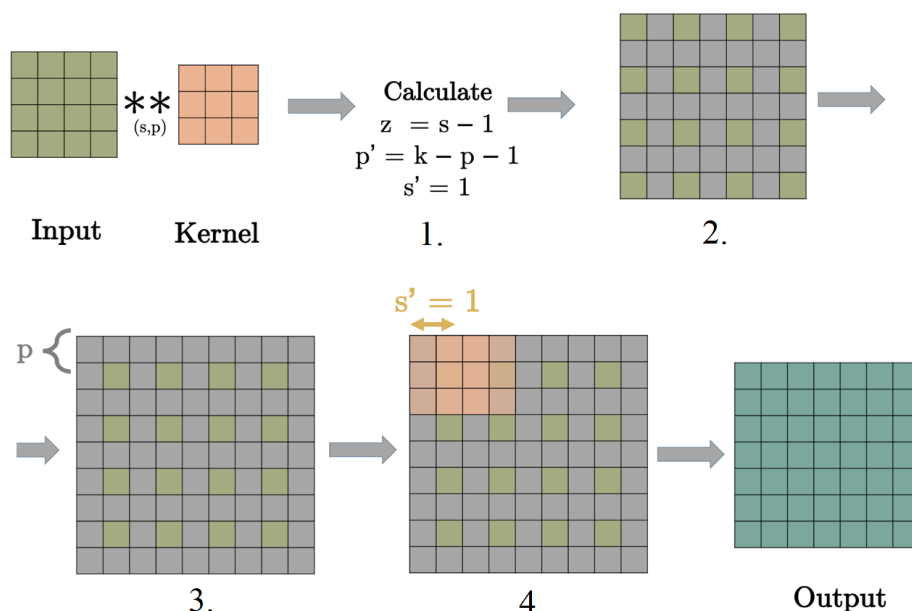
Oproti klasickým vrstvám použitým v konvolučných neurónových sieťach je pre úlohu zvýšenia rozlíšenia obrazu využit špeciálne vrstvy, ktoré umožnia zväčšiť rozlíšenie príznakových máp. Výhodou týchto vrstiev oproti klasickým interpolačným metódam je, že majú učiteľné parametre.

#### Transponovaná konvolučná vrstva

Podľa [37] sa pre účely zníženia šírky a výšky vstupu v konvolučných neurónových sieťach používa klasická konvolučná vrstva s parametrami *padding* a *stride*. Pre zvýšenie rozlíšenia

je prirodzene potrebna aj operacia opačna. Takáto vrstva sa nazýva transponovaná konvolučná vrstva alebo (nie úplne správne <sup>1</sup>) dekonvolučná vrstva. Takouto operáciou je možné obnoviť pôvodný tvar príznakovej mapy pred konvolúciou. Výsledkom nie sú však pôvodné hodnoty.

Transponovaná konvolúcia má rovnaké parametre ako klasická konvolučná vrstva. Ich nastavením je možné upraviť šírku a výšku výstupu podobne ako pri klasickej konvolučnej vrstve. Navyše sa tu nachádza zvolený interpolačný operátor (napr. metóda najbližšieho suseda, alebo len vloženie hodnoty 0). Pôvodná príznaková mapa sa zväčší pomocou interpolačného operátora (medzi pôvodné stĺpce a riadky sa doplnia nové hodnoty) a okolo novej príznakovej mapy sa vytvorí požadovaný *padding*. Konvolučné jadro s parametrom *stride* rovným 1 potom postupne vykonáva klasickú konvolúciu. Výsledkom je nová príznaková mapa, ktorá má väčšie rozlíšenie (šírku a výšku) ako pôvodná.



Obr. 4.1: Jednotlivé kroky transponovanej konvolúcie. V 1. kroku sa spočítajú potrebné parametre. V kroku číslo 2 sa pridávajú nuly (alebo iné čísla podľa zvolenej interpolačnej metódy) medzi riadky a stĺpce. V 3. kroku sa pridá  $p'$  núl (alebo iných čísel podľa zvolenej interpolačnej metódy) okolo obrázku. V poslednom sa jadro konvolúcie pohybuje postupne po 1 pixly cez celý obrázok. Výstup tejto konvolúcie je výstupom danej vrstvy. Obrázok prevzatý z [1] a upravený.

### Sub-pixel konvolučná vrstva

Podľa [37] je transponovaná konvolučná vrstva ponúka pomerne dobré riešenie. Má však svoje nevýhody - hlavnou z nich je doplnenie novo vytvorených hodnôt interpolačnou metódou najbližšieho suseda. To spôsobuje že pri nadvzorkovaní (hlavne 4 a viac násobnom) sú vzory opakované v každom smere. Tým pádom sú redundantné.

S cieľom obísť tento problém *Shi et al.* predstavili sub-pixel konvolučnú vrstvu [27] v architektúre známej ako ESPCN. Učiteľné parametre tejto vrstvy sú váhy konvolučných

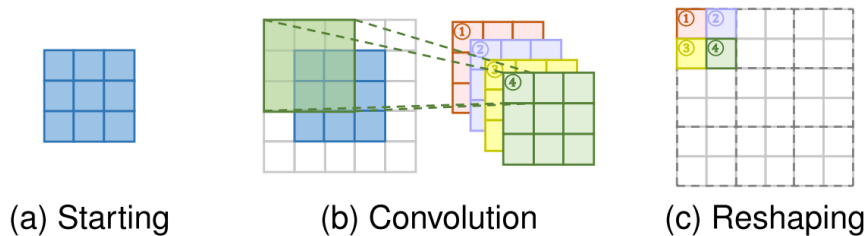
<sup>1</sup>je to niekedy používaný pojem, ale matematicky je to opačná operácia konvolúcie, čo nie je to isté ako transponovaná konvolúcia [7]

jadier. Rozlíšenie mapy príznakov robí tak, že pomocou konvolúcie vytvorí niekoľko ďalších vrstiev, čím získa viac bodov. Následne tieto body špecificky preukladá, aby vznikla mapa príznakov s vyšším rozlíšením (šírka a výška).

Ak je vstupom mapa príznakov s rozmermi  $h \times w \times c$  a mierka zvýšenia rozlíšenia je  $s^2$ , výstupom konvolúcie bude príznaková mapa o veľkosti  $h \times w \times s^2c$ . Výsledok konvolúcie sa preukladá tak, aby nakoniec vznikla mapa príznakov s rozmermi  $sh \times sw \times c$  (viď. na obrázku 4.2) [33].

Vzhľadom na to, že sa pri konvolúcii zvýši počet kanálov, je malá veľkosť konvolučného jadra postačujúca. V porovnaní s transponovanou konvolučnou vrstvou je sub-pixel konvolučná vrstva efektívnejšia [37].

Podľa [33] táto vrstva zachytáva viac kontextuálnej informácie, čo pomáha pri vytváraní lepších detailov v obrázkoch. Avšak vzhľadom na rozdelenie blokov pre jednotlivé konvolúcie môžu vzniknúť artefakty v blízkosti hraníc jednotlivých blokov.



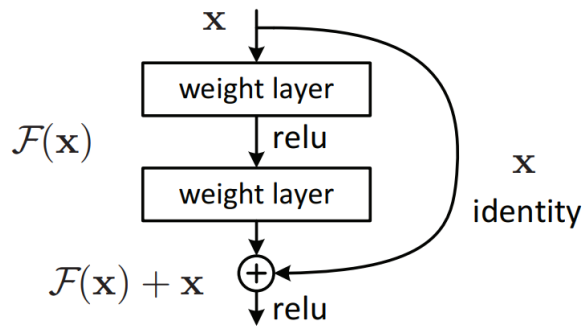
Obr. 4.2: Sub-pixel konvolučná vrstva. Modrý štvorec predstavuje vstup. Farebné štvorce v obrázku b predstavujú rôzne konvolučné jadrá. V časti c je znázornené preskladanie výsledkov všetkých konvolúcií, aby bol výstupný obrázok väčší, v tomto prípade dvojnásobne väčší. Obrázok prevzatý z [33].

## Reziduálne bloky

Zvyšovaním hĺbky konvolučných neurónových sietí sa zvyšuje aj náročnosť ich tréovania. V práci [14] boli predstavené reziduálne bloky, ktoré uľahčujú tréovanie hlbších neurónových sietí. Reziduálny blok obsahuje niekoľko klasických vrstiev. Na vstupe týchto vrstiev sa nachádza vektor  $x$ . Po spracovaní vektoru vrstvami dostaneme  $\mathcal{F}(x)$ . Reziduálny blok spočíva v pričítaní vstupu  $x$  k výstupu  $\mathcal{F}(x)$  a následnou aplikáciou aktivačnej funkcie. V závislosti od vrstiev použitých v reziduálnom bloku je niekedy potrebné upraviť dimenzie vstupného vektoru  $x$  pred sčítaním, tak aby sa rovnali dimenziám vektoru  $\mathcal{F}(x)$ . Princíp je znázornený na obrázku 4.3.

Siete skladajúce sa z reziduálnych blokov autori nazvali Reziduálne siete, tiež známe ako *ResNets*. V práci [14] boli porovnané s vtedajšími state of the art metódami pre klasifikáciu a detekciu objektov. Reziduálne siete boli schopné výrazne znížiť chybu klasifikácie a zlepšiť presnosť detekcie s porovnateľným počtom parametrov, pričom si tieto siete zachovali porovnateľný počet parametrov ako konkurenčné neurónové siete.

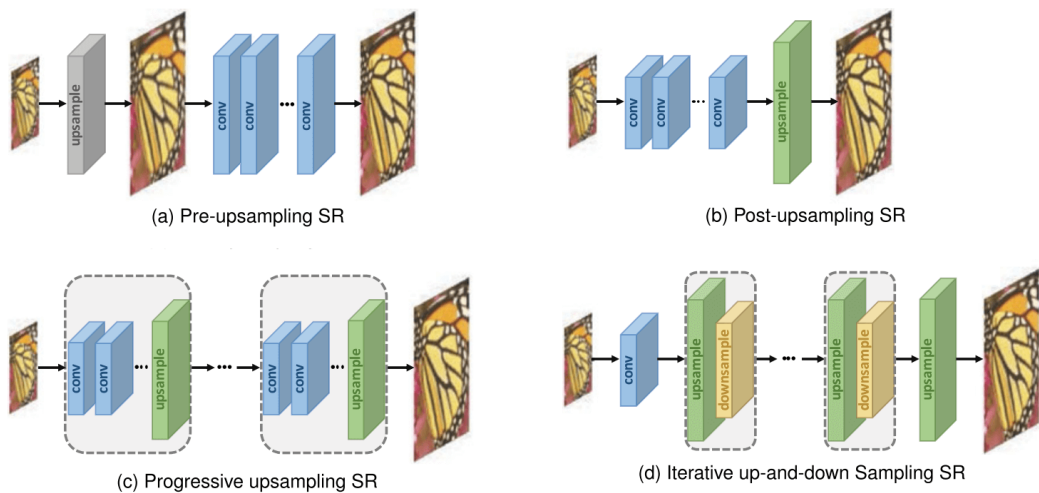
Reziduálne bloky sa často nachádzajú aj v architektúrach, ktoré sa používajú na zvýšenie rozlíšenia obrazu.



Obr. 4.3: Princíp reziduálneho bloku. Vstup bloku je pripočítaný k výstupu bloku. V tomto prípade má vstupný vektor  $x$  rovnaké rozmery a výstupný vektor  $\mathcal{F}(x)$ . Obrázok prevzatý z [14].

## 4.2 Typy architektúr neurónových sietí pre zvýšenie rozlíšenia obrazu

Informácie v tejto podkapitole pochádzajú z [33]. Problém zvýšenia rozlíšenia obrazu je problém, ktorého riešenie nie je univerzálne a závisí výrazne na vstupných dátach. Kľúčovou časťou problému je určiť spôsob ako nadzorkovať (angl. *upsampling*) obraz (generovať HR výstup pomocou LR vstupu). Architektúry existujúcich modelov sa často líšia, avšak všeobecne sa dajú rozdeliť do štyroch typov, ktoré sa líšia v operátore nadzorkovania a pozícii tohto operátora. Všetky štyri typy architektúr sú znázornené na obrázku 4.4 a sú bližšie popísané v tejto podkapitole.



Obr. 4.4: Rôzne typy architektúr pre zvýšenie rozlíšenia obrazu založených na hlbokých neurónových sieťach. Šedé bloky reprezentujú interpolačné metódy zvýšenia rozlíšenia obrazu (bez učiteľných parametrov). Zelené, žlté a modré bloky majú učiteľné parametre a ide o nadzorkovacie, podzorkovacie a konvolučné vrstvy v tomto poradí. Obrázok prevzatý z [33].



## Pre-Upsampling

Vzhľadom na náročnosť úlohy mapovania z nižšie dimenzionálneho priestoru do vyššie dimenzionálneho priestoru je najpriamočiarejším postupom využiť klasické algoritmy pre zvýšenie rozlíšenia obrazu a následne vylepšiť takto získaný obrázok pomocou neurónových sietí. *Dong et al.* [5, 6] využili tento typ (na obrázku 4.4a) a prišli so sieťou SRCNN. Tá má za úlohu naučiť sa namapovať interpolovaný LR obrázok na HR obrázok.

Konkrétne je rozlíšenie LR obrázkov zvýšené pomocou interpolačných metód (napr. bikubicou interpoláciou, viď. v časti 3.2) a na výstupný obraz je aplikovaná hlboká konvolučná neurónová sieť, ktorej úlohou je rekonštruovať vysoko kvalitné detaily.

Keďže najťažšia úloha zvýšenia rozlíšenia je už splnená, cieľom konvolučnej neurónovej siete je upraviť neuhladené obrázky, čo markantne znižuje náročnosť učenia. Ďalšou výhodou je, že ako vstup tejto neurónovej siete môžu byť interpolované obrázky s ľubovoľným rozlíšením a mierkou zvýšenia rozlíšenia (angl. *scaling factor*), pričom výsledky sú porovnateľné s modelmi neurónových sietí, ktoré vedú zvýšiť rozlíšenie len s konštantnou mierkou [21].

Aj vďaka tomu sa stali takéto neurónové siete populárnymi. Jednoduchý dizajn však prináša aj svoje nevýhody. Medzi tie patrí napríklad možné zosilnenie šumu a rozmazanie použitím jednoduchých interpolačných metód. Taktiež väčšina výpočtových operácií sa vykonáva vo vysoko dimenzionálnom priestore, čo spôsobuje nárast výpočtových a priestorových nárokov oproti iným typom architektúr.

## Post-Upsampling

Za účelom zvýšenia efektívnosti výpočtu a využitím potenciálu technológie hlbokého učenia pre automatické zvýšenie rozlíšenia obrazu, je možné nahradiť interpolačné metódy plne učiteľnými vrstvami integrovanými na konci modelu (viď. na obrázku 4.4b). LR obrázky vstupujú do hlbokej konvolučnej neurónovej siete bez zvýšenia ich rozlíšenia. V posledných vrstvách sa nadchádza plne učiteľná vrstva, ktorá je schopná nadzorkovať obraz.

Extrakcia príznakov je výpočtovo náročný proces. V tomto prípade sa extrakcia deje v nízko dimenzionálnom priestore a rozlíšenie sa zvýši až na konci. Vďaka tomu je výpočet menej náročný časovo aj priestorovo. Tento typ architektúry sa tiež používa často [23, 24, 29, 27]. Modely sa ďalej líšia v type učiteľných vrstiev na konci, v štruktúre konvolučných neurónových sietí a metódach učenia.

## Progresívne nadzorkovanie

Modely typu Post-Upsampling priniesli zníženie výpočtových nárokov, avšak majú svoje nevýhody. Keďže zvýšenie rozlíšenia obrazu sa deje v jednom kroku, každá miera zvýšenia rozlíšenia obrazu (napr. 4, 8) vyžaduje individuálny tréning. V dnešnej dobe sú potrebné modely, ktoré by boli schopné zväčšiť vstupný obraz vo viacerých mierkach (tzv. multi-scale SR). Vzhľadom k týmto potrebám bol vyvinutý typ architektúry s progresívnym nadzorkovaním (na obrázku 4.4c).

Modely tohto typu sú založené na kaskáde konvolučných neurónových sietí a progresívne rekonštruujú obrázky s vyšším rozlíšením. V každom kroku je zvýšené rozlíšenie a obrázok je vylepšený konvolučnou neurónovou sieťou.

Modely z tejto kategórie sú napríklad MS-LapSRN [22] alebo Progressive SR (ProSR) [32]. Dosahujú relatívne kvalitných výsledkov.

Vďaka rozdeleniu náročnej úlohy na menšie jednoduchšie časti, sú tieto modely ľahšie učiteľné aj pre úlohy, kde ide o vyššiu mierku zvýšenia rozlíšenia (4x, 8x). Tieto modely majú aj svoje problémy - komplikovanejší dizajn architektúry pre viacero krokov zvýšenia rozlíšenia, nižšia tréningová stabilita a celkovo komplikovanejší proces tréningovania, kde sú často potrebné pokročilejšie stratégie.

## Iteratívne nadvzorkovanie a podvzorkovanie

Pre lepšie zachytenie vzájomného vzťahu LR a HR obrázkov, bola do úloh zvyšovania rozlíšenia obrazu zakomponovaná operácia spätnej projekcie (angl. *back-projection*). Tento typ architektúry (viď na obrázku 4.4d) funguje na princípe aplikovania vylepšenia obrazu pomocou spätnej projekcie - spočítaním chyby rekonštrukcie obrazu a použitím tejto informácie pre zlepšenie výsledku.

Napríklad *Haris et al.* [12] využívajú iteratívne nadvzorkovacie a podvzorkovacie vrstvy siete DBPN, v ktorej sú tieto vrstvy prepojené striedavo a výsledný HR obrázok sa skladá z týchto prechodných výsledkov.

Architektúry tohto typu sú schopné lepšie vydolovať hlboké vzťahy medzi LR a HR obrázkami a vytvárať tak kvalitnejšie výstupy. Avšak dizajnové kritériá modulov spätnej projekcie sú stále nejasné. Ide o pomerne nový mechanizmus hlbokého učenia pre zvýšenie rozlíšenia obrazu, ktorý má dobrý potenciál, ale potrebuje ďalšie skúmanie.

## 4.3 Chybové funkcie

V časti 3.3 sú spomenuté metriky, ktorými je možné ohodnotiť kvalitu výsledného obrázku. Ako bolo spomenuté, tieto metriky presne neodpovedajú vnímaniu obrázkov ľudským vizuálnym systémom. Preto je vhodné využiť inú chybovú funkciu na tréningovanie neurónových sietí.

### Príznakové (feature-wise) chybové funkcie

Úlohou perceptuálnej chybovej funkcie je vynútiť prirodzené a pre ľudské vnímanie atraktívne výsledky. V dnešnej dobe sú veľmi často používané kvôli ich kvalitným výsledkom. Výstupy zo sietí tréningovaných týmito chybovými funkciami majú často nižšie hodnoty PSNR a SSIM, ale vizuálne pôsobia veľmi kvalitne [23]. Chybové funkcie tohoto typu sú tiež známe pod inými názvami ako *Perceptual loss* [18] alebo *VGG loss* [23]. Fungujú na základe L2 vzdialenosti medzi aktiváciami skrytých vrstiev siete tréningovanej na úlohu klasifikácie (napríklad sieť VGG [28]).

Namiesto porovnania pixelov obrázkov teda porovnávajú príznakové vektory výsledného obrázku s pôvodným HR obrázkom. Chybovú funkciu je možné definovať ako [23]:

$$l_{VGG/i,j}^{SR} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2, \quad (4.1)$$

kde  $\phi_{i,j}$  je príznakový vektor získaný z  $j$ -tej konvolúcie (po aktivačnej funkcii) pred  $i$ -tou maxpooling vrstvou siete VGG19. Ide teda o euklidovskú vzdialenosť príznakových vektorov rekonštruovaného obrázku  $G_{\theta_G}(I^{LR})_{x,y}$  a referenčného HR obrázku  $I^{HR}$ .  $W_{i,j}$  a  $H_{i,j}$  predstavujú rozmery príznakových vektorov siete VGG.

Ako bolo ukázané v práci [31], pri zvýšení rozlíšenia obrazu je lepšie využiť príznakovú mapu z  $j$ -tej konvolúcie pred aktivačnou funkciou. V prípade VGG ide o aktivačné funkcie

typu ReLU, kedy táto funkcia odstráni všetky hodnoty menšie ako nula, čo spôsobí stratu informácie.

V práci [30] autori použili rozšírenú verziu tejto chybovej funkcie. Rozšírenie spočíva v použití výstupov z viacerých skrytých vrstiev použitej neurónovej siete. Každé z vrstiev je možné nastaviť váhu. Z každej vybranej vrstvy je získaný pár príznakových vektorov (príznakový vektor pre HR obrázok a pre výstup z neurónovej siete). Pre tieto páry sa počíta euklidovská vzdialenosť. Vzdialenosti sú vynásobené váhou danej vrstvy a sčítané. To reprezentuje konečný výsledok chybovej funkcie.

## 4.4 Konkrétne architektúry schopné zvýšenia rozlíšenia obrazu

V tejto podkapitole sú popísané skúmané architektúry neurónových sietí schopné plniť úlohu zvýšenia rozlíšenia obrazu.

### Konvolučná neurónová sieť U-Net

Ide o typ architektúry neurónových sietí inšpirovaný autoenkodérmi. Tie sa skladajú z dvoch spojených častí - enkodéru a dekodéru. Úlohou enkodéru je vytvoriť reprezentáciu celého vstupu, ktorá je kompaktnejšia a zahŕňa hlavné črty vstupu. Dekodér potom transformuje túto internú reprezentáciu na výstup.

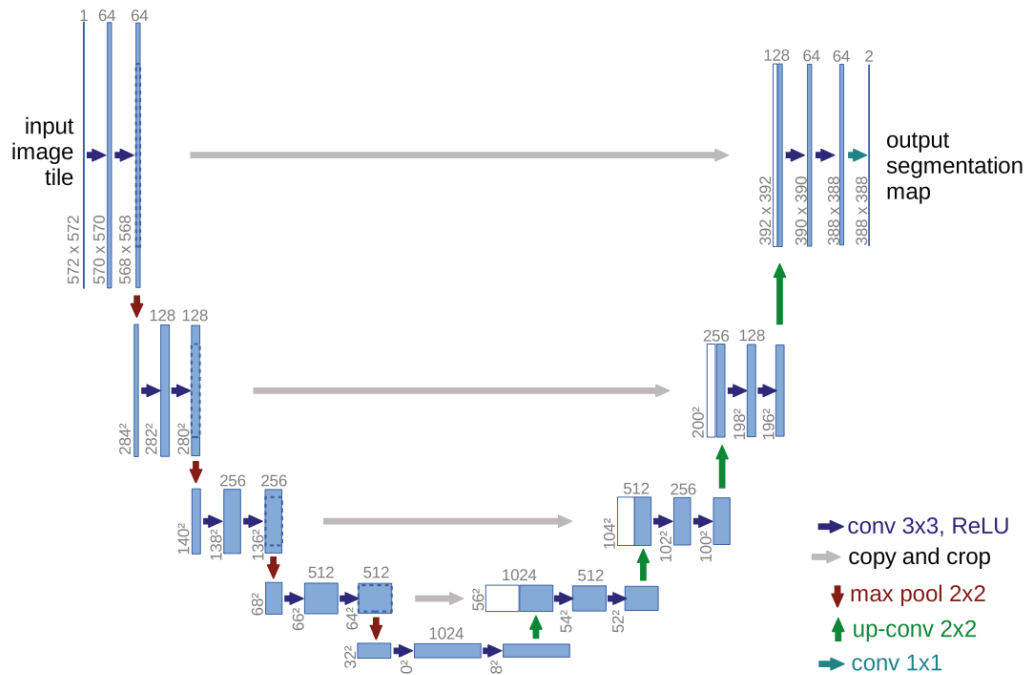
Architektúra typu U-Net, bola prvýkrát predstavená na účel segmentácie biomedických obrazových dát [26]. Autori v práci riešia problém, že väčšina konvolučných neurónových sietí rieši problém klasifikácie tak, že výstupom neurónovej siete je informácia o tom do akej triedy obrázok zaradiť. Pre mnoho typov dát je však užitočnejšia informácia o tom, kde konkrétne sa daný objekt nachádza. Zjednodušene, klasifikovať každý pixel na obrázku. Takéto úlohy sa väčšinou riešili pomocou posuvného okna [26], ale mali značné nevýhody.

Sieť, ako je vidno na obrázku 4.5, sa skladá z ľavej, zmenšujúcej časti (angl. *contractive path*) a pravej, expanznej časti (angl. *expansive path*). Ľavá časť odpovedá typickej architektúre konvulčnej neurónovej siete. Skladá sa z dvoch konvulčných vrstiev s filtrom o veľkosti 3x3, nasledovaných aktivačnou funkciou ReLU a maxpooling vrstvou o veľkosti 2x2 s krokom 2 zmenšenie rozlíšenia. Pri každom zmenšení rozlíšenia sa zdvojnásobí počet kanálov. Kroky v pravej časti sa skladajú z konvulčnej vrstvy, ktorá zníži počet kanálov o polovicu. Za touto konvulčnou vrstvou sa nachádza upsampling vrstva, ktorá zvýši šírku a výšku príznakovkej mapy dvojnásobne. Výstup z upsampling vrstvy sa konkatenuje s korešpondujúcou (s rovnakými rozmermi) príznakovou mapou z ľavej časti. Potom nasledujú znova dve 3x3 konvulcie a ReLU. Toto sa opakuje až po poslednú vrstvu, ktorá sa skladá z konvulčnej vrstvy s filtrom o veľkosti 1x1. Tá má za úlohu namapovať daný príznakový vektor na požadovaný počet tried. Konvulčné vrstvy v tejto architektúre nepoužívajú padding, čo v praxi znamená, že výsledný obrázok je menší ako vstupný.

Túto architektúru je teoreticky možné využiť aj na úlohu zvýšenia rozlíšenia obrazu. Bližší popis tejto úpravy sa nachádza v časti 5.2.

### Konvolučná neurónová sieť RUNet

Architektúra U-Net nebola pôvodne navrhnutá pre úlohu zvýšenia rozlíšenia obrazu. Preto bolo vhodné nájsť architektúru podobného typu, ktorá bola navrhnutá na tento účel. Architektúra v práci *Hu et al.* [16] s názvom RUNet patrí tiež do kategórie sietí typu U-Net.



Obr. 4.5: Architektúra siete U-Net pre segmentáciu biomedických dát. V tomto prípade je na vstupe obrázok s rozlíšením 572x572, ktorý je postupne transformovaný na mapu príznakových vektorov o veľkosti 32x32 a následne znova zväčšený na rozlíšenie 388x388. Každý modrý obdĺžnik znázorňuje multi-kanálový príznakový vektor, počet kanálov sa nachádza nad obdĺžnikom. Jeho výška a šírka sa nachádza naľavo od obdĺžnika. Obrázok prevzatý z [26].

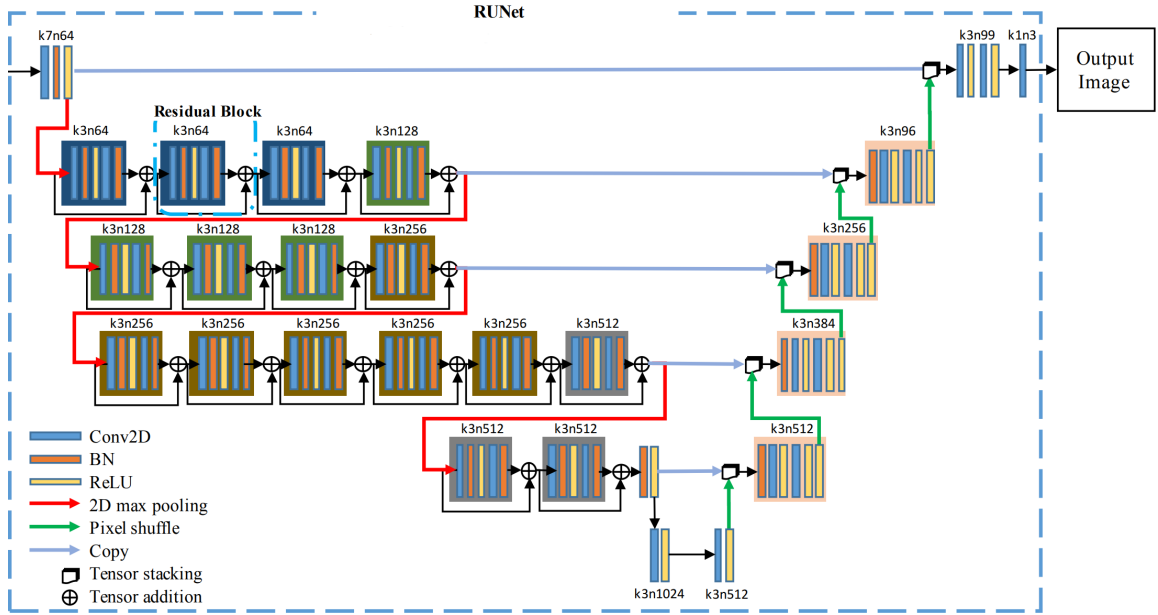
Oproti pôvodnej architektúre U-Net spomenutej vyššie, využíva navyše reziduálne bloky s batch normalizačnými vrstvami a pri zvýšení rozlíšenia používa sub-pixel konvolučnú vrstvu namiesto transponovanej konvolučnej vrstvy ako v prípade U-Net architektúry vyššie. Architektúra konvolučnej siete je znázornená na obrázku 4.6.

Každý krok ľavej časti obsahuje niekoľko reziduálnych blokov (princíp reziduálnych blokov je popísaný v časti 4.1), kde sa vstup z predchádzajúceho bloku propaguje aj na vstup aktuálneho bloku pomocou sčítania vektorov. To umožňuje sieti učiť sa napodobňovať komplexnejšie štruktúry [16]. Pre efektívne zvýšenie rozlíšenia sú v pravej časti použité sub-pixel konvolučné vrstvy.

Neurónová sieť je v pôvodnej práci učená pomocou príznakovej chybovej funkcie popísanej v časti 4.3. Robustnosť siete zaručuje spôsob tréningu, kde sa pri tréningu vstupnému obrázku zmenší rozlíšenie, rozmaže sa pomocou gaussovského filtra ako forma augmentácie a následne je jeho rozlíšenie zvýšené na pôvodnú veľkosť. Takýto obrázok je vylepšený pomocou siete RUNet a porovnaný s pôvodným obrázkom s vysokým rozlíšením.

## Generative Adversarial Network

*Generative Adversarial Networks* (GAN) boli prvýkrát predstavené v práci *Generative Adversarial Nets* [10]. Sú založené na princípe súperenia dvoch neurónových sietí, ktoré proti sebe súperia a tak iteratívne vylepšujú svoju presnosť. Prvá sieť sa nazýva *generátor* a má za úlohu aproximovať rozloženie tréningových dát. Druhá sieť je *diskriminátor* a jeho úloha



Obr. 4.6: Architektúra RUNet. Modré bločky sú klasické konvolučné vrstvy, oranžové sú batch normalizačné vrstvy, žlté reprezentujú aktivačnú funkciu ReLU. Červené šípky reprezentujú podvzorkovanie max pooling vrstvou, zelené šípky reprezentujú sub-pixel konvolučné vrstvy a modré šípky značia kopírovanie výstupu danej vrstvy, kde je výstup potom skombinovaný s výstupom z inej vrstvy. Popisy blokov v tvare  $kz_1nz_2$  reprezentujú konvolučnú vrstvu s veľkosťou jadra  $z_1 \times z_1$  a  $z_2$  kanálmi. Obrázok prevzatý z [16].

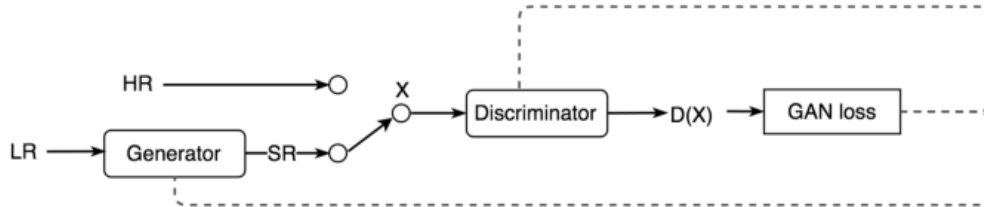
spočíva v predikcii pravdepodobnosti, že dáta na jeho vstupne pochádzajú z pôvodnej dátovej sady a nejedná sa o výstup generátora. Siete typu GAN sa často používajú na úlohu zvýšenia rozlíšenia obrazu. V ďalšom odstavci je popísaný všeobecný princíp týchto sietí a ďalej budú popísané konkrétne architektúry používané pre zvýšenie rozlíšenia obrazu.

V pôvodnom článku [10] generátor funguje tak, že vytvára falošné dáta, ktoré by mali vypadať ako dáta v tréningovej sade. Generátor transformuje náhodný šum na dáta, z ktorých sa iteratívne stanú dáta podobné tréningovej sade. Diskriminátor má na rozdiel od generátora prístup k tréningovým dátam. Na vstupe má buď vzorku z tréningových dát, alebo výstup generátora. Trénuje sa klasickými spôsobmi tréningovania s učiteľom, kde má za úlohu priradiť správnu triedu pre každý vstup. Tréningovanie oboch sietí je cieľom vylepšiť generátor do takej miery, že diskriminátor nebude schopný rozumne rozoznať medzi reálnymi a vygenerovanými dátami. Diskriminátor a generátor „hrajú“ nasledovnú minimax hru s ohodnocovacou funkciou  $V(G, D)$  [10]:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))], \quad (4.2)$$

kde pre natréningovanie rozloženia generátora  $p_g$  založeného na dátach  $x$  definujeme náhodné rozloženie premennej  $z$  ako  $p_z(z)$  a mapovaciu funkciu do priestoru tréningových dát  $G(z; \theta_G)$ , kde  $G$  reprezentuje generátor s parametrami  $\theta_G$ . Diskriminátor je znázornený ako funkcia  $D(x; \theta_D)$ , kde  $\theta_D$  sú jeho parametre. Výstupom tejto funkcie je jediný skalár.  $D(x)$  určuje pravdepodobnosť, že  $x$  pochádza z tréningových dát a nie z  $p_g$ . Diskriminátor je tréningovaný aby maximalizoval svoju schopnosť dátam priradiť správnu triedu. Zároveň je generátor tréningovaný aby minimalizoval  $\log(1 - D(G(z)))$ .

Pri použití sietí typu GAN pre **zvýšenie rozlíšenia obrazu** nie je vstupom generátora náhodný šum, ale obrázkov so zníženým rozlíšením. Proces je zjednodušene znázornený na obrázku 4.7. Generátor sa tento obrázok snaží transformovať tak, aby vypadal ako obrázok, ktorého rozlíšenie nebolo vôbec zmenšené. Konkrétne architektúry sietí GAN sú popísané nižšie.



Obr. 4.7: Schéma znázorňujúca architektúru GAN pre úlohu zvýšenia rozlíšenia obrazu. Obrázok prevzatý z [20].

## SRGAN

Podľa autorov práce je architektúra SRGAN (*SR* - super-resolution, *GAN* - Generative Adversarial Networks) schopná ako prvá realisticky zvyšovať rozlíšenie obrázkov štyrikrát [23]. Dosiahla to vďaka spojeniu *adversarial* a *content* chybových funkcií do jednej príznakovej (perceptuálnej) chybovej funkcie a použitím architektúry typu GAN.

V rámci práce autori upravili problém minimax hry z rovnice 4.2 aby odpovedal úlohe zvýšenia rozlíšenia obrazu [23]:

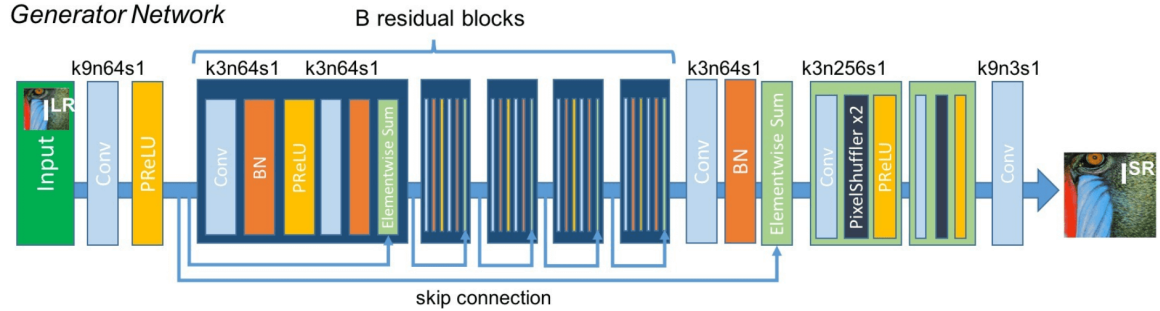
$$\min_{\theta_G} \max_{\theta_D} V(D, G),$$

$$V(D, G) = \mathbb{E}_{I^{HR} \sim p_{train}(I^{HR})} [\log D_{\theta_D}(I^{HR})] + \mathbb{E}_{I^{LR} \sim p_G(I^{LR})} [\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))]], \quad (4.3)$$

kde  $D$  je diskriminátor a  $G$  je generátor s naučenými váhami  $\theta_D$  a  $\theta_G$ .  $I^{HR}$  a  $I^{LR}$  značia obrázok s vysokým rozlíšením a odpovedajúci obrázok s nízkym rozlíšením. Myšlienkou tejto rovnice je natrénovať generatívny model  $G$  s cieľom oklamať diskriminátor  $D$ , ktorý je na úlohu rozoznať reálny obrázok s vysokým rozlíšením od obrázku, ktorého rozlíšenie bolo zvýšené umelo. Tento spôsob trénovania umožňuje generátoru naučiť sa zvyšovať rozlíšenie obrázkov tak, že výsledok je veľmi podobný reálnemu obrázku.

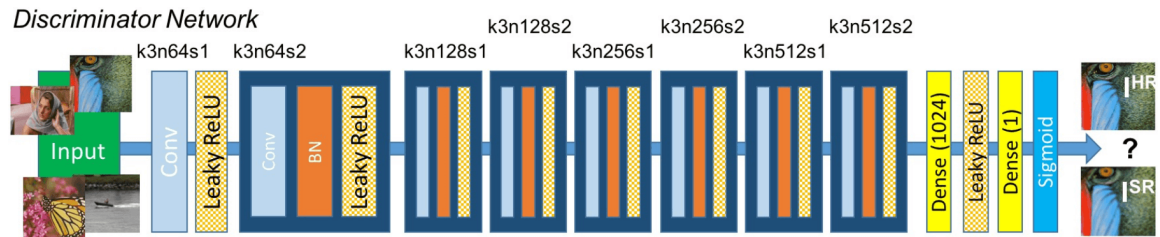
Jadro architektúry generátora  $G$  pozostáva z  $B$  identických reziduálnych blokov, pred ktorými je konvolučná vrstva so 64 veľkými konvolučnými jadrami o veľkosti  $3 \times 3$ . V každom reziduálnom bloku sa nachádza konvolučná vrstva so 64 konvolučnými jadrami o veľkosti  $3 \times 3$ . Za konvolučnou vrstvou nasleduje batch-normalizačná vrstva a aktivačná funkcia ParametricReLU [13]. Za touto aktivačnou funkciou sa opakuje rovnaká konvolučná a batch-normalizačná vrstva. K výstupu poslednej vrstvy sa pripočíta vstup celého reziduálneho bloku. Po  $B$  reziduálnych blokoch je použitá rovnaká konvolučná vrstva ako vo vnútri blokov, za ktorou je opäť batch-normalizačná vrstva. Výstup z tejto vrstvy je sčítaný so vstupom do prvého reziduálneho bloku. Nasleduje časť siete, ktorá zväčšuje výšku a šírku vzniknutých vysoko dimenzionálnych vektorov a zároveň znižuje počet kanálov (dimenzií). Ide o sub-pixel konvolučnú vrstvu [27] (popísanú v časti 4.1) s 256 konvolučnými

jadrami  $3 \times 3$  s parametrom zvýšenia mierky 2. Táto vrstva má znova aktivačnú funkciu ParametricReLU. Tento „zväčšovací“ blok sa v architektúre nachádza dvakrát, čo zvýši rozlíšenie vstupného obrazu celkovo štyrikrát. Na konci siete je posledná konvolučná vrstva s veľkými konvolučnými jadrami  $9 \times 9$ , ktorá vytvorí potrebný počet kanálov (3 pre farebný RGB obrázok a 1 pre čiernobiely obrázok). Architektúra generátora sa nachádza na obrázku 4.8.



Obr. 4.8: Architektúra generátora z práce SRGAN. Nad jednotlivými konvolučnými vrstvami (*Conv*) sa nachádza veľkosť konvolučných jadier ( $k$ ), ich počet ( $n$ ) a parameter stride ( $s$ ). Obrázok prevzatý z [23].

Na rozoznanie reálneho obrázku od umelo zväčšeného je použitá architektúra diskriminátora  $D$ , ktorá je trénovaná aby riešila problém maximalizácie z rovnice 4.3. Nachádza sa tu celkovo osem konvolučných vrstiev s jadrami o veľkosti  $3 \times 3$ , kde sa ich počet dvojnásobne zvýši každú druhú vrstvu. Počet konvolučných jadier začína na 64 a končí na 512. Architektúra je principiálne podobná sieťam VGG [28]. Za každou, okrem prvej, konvolučnej vrstvy je znova použitá batch-normalizačná vrstva a aktivačná funkcia LeakyReLU ( $\alpha = 0.2$ ). Každá druhá konvolučná vrstva má nastavený parameter  $stride = 2$ , čo postupne znižuje rozlíšenie obrázku a počet konvolučných jadier zároveň zvyšuje počet kanálov. Po ôsmich konvolučných vrstvách má príznakový vektor 512 kanálov a putuje do plne prepojenej vrstvy s 1024 jednotkami a aktivačnou funkciou LeakyReLU. Posledná vrstva je plne prepojená vrstva s 1 jednotkou a aktivačnou funkciou sigmoid. To zaručuje, že výstup siete je pravdepodobnosť triedy, do ktorej trénovacia vzorka patrí - reálny obrázok alebo umelo zväčšený obrázok. Architektúra je znázornená na obrázku 4.9.



Obr. 4.9: Architektúra diskriminátora z práce SRGAN. Nad jednotlivými konvolučnými vrstvami (*Conv*) sa nachádza veľkosť konvolučných jadier ( $k$ ), ich počet ( $n$ ) a parameter stride ( $s$ ). Pri plne prepojených vrstvách (*Dense*) je v zátvorke počet jednotiek danej vrstvy. Obrázok prevzatý z [23].

Definícia chybovej funkcie sa skladá z dvoch častí. Je definovaná ako [23]:

$$l^{SR} = l_X^{SR} + 10^{-3}l_{Gen}^{SR}, \quad (4.4)$$

kde prvá časť odpovedá *content loss*. V tejto časti značí  $X$  typ tejto loss. Buď ide o klasickú euklidovskú vzdialenosť (MSE) alebo o príznakovú chybovú funkciu založenú na VGG popísanú v časti 4.3. Pri tréňovaní sa využívajú obidve varianty. Proces tréňovania je popísaný ďalej. Druhá časť značí *adversarial loss*. Tá učí generátor vytvárať realistickejšie obrázky, aby bol schopný oklamať diskriminátor. Je založená na pravdepodobnosti diskriminátora  $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ , že generátorom zväčšený obrázok  $G_{\theta_G}(I^{LR})$  je reálny obrázok s vysokým rozlíšením. Je definovaná ako [23]:

$$l_{Gen}^{SR} = \sum_{N=1}^N -\log D_{\theta_D}(G_{\theta_G}(I^{LR})), \quad (4.5)$$

kde ide o druhú časť z rovnice minimax problému 4.3, upravenú z dôvodu zlepšenia výpočtu gradientov.

Chybová funkcia diskriminátora je klasická binárna krížová entropia, kde obrázok vytvorený generátorom má triedu 0 a reálny obrázok s vysokým rozlíšením patrí do triedy 1.

Proces tréňovania je v práci [23] rozdelený do dvoch častí. Najskôr bol generátor predtréňovaný s klasickou MSE chybovou funkciou aby sa autori vyhli problému lokálneho optima. V druhej časti zobrali váhy predtréňovaného generátora, MSE vymenili za kombináciu príznakovej chybovej funkcie a *adversarial loss* (rovnica 4.4). V tejto časti sa už tréňoval aj diskriminátor. Použitá architektúra generátoru má počet blokov nastavených na  $B = 16$ . Ako vstup používali výrezy z podvzorkovaných obrázkov s veľkosťou  $96 \times 96$  px.

Výsledky práce ukázali, že táto architektúra je schopná zvyšovať rozlíšenie obrazu lepšie ako iné metódy používajúce chybové funkcie založené na priamom porovnaní obrázkov (napr. MSE). Podľa testovania, kedy výsledné obrázky hodnotili ľudia, bola najlepšia sieť SRGAN-VGG54, kde VGG54 značí, že pri príznakovej chybovej funkcii zobrali výstup 4. konvolučnej vrstvy (po aktivačnej funkcii) pred 5. max-pooling vrstvou. Po dokončení druhého kroku tréňovania boli hodnoty metrík PSNR a SSIM nižšie ako po prvom kroku tréňovania s chybovou funkciou MSE, ale kvalita zväčšených obrázkov bola vyššia.

## ESRGAN

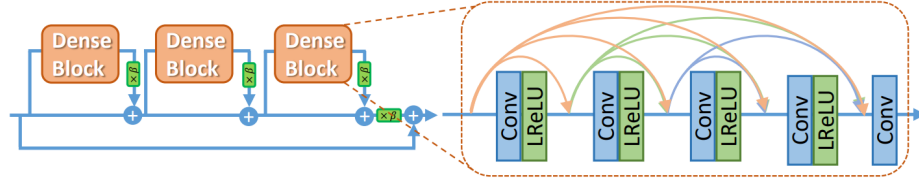
Práca s architektúrou ESRGAN [31] (*Enhanced SRGAN*) vznikla ako vylepšenie práce SRGAN. Priniesla niekoľko zmien v architektúre generátora, diskriminátora a aj v chybovej funkcii.

V **architektúre generátora** boli vykonané dve veľké zmeny. Prvou z nich je odstránenie všetkých batch-normalizačných vrstiev podľa práce EDSR [24]. Táto modifikácia pomohla k odstráneniu artefaktov v prípade, že dátová sada má vysoký rozptyl hodnôt. Odstránenie týchto vrstiev tiež pomohlo k lepšej generalizácii siete a k zníženiu výpočtových a pamäťových nárokov. Druhou zmenou je nahradenie reziduálnych blokov použitých v architektúre SRGAN, novými *Residual-in-Residual Dense* blokmi (ďalej RRDB). Tieto bloky kombinujú viacúrovňové reziduálne siete a plne prepojené spojenia. Sú znázornené na obrázku 4.10. Oproti pôvodným reziduálnym blokom v práci SRGAN [23] tieto bloky neobsahujú batch-normalizačné vrstvy a oproti pôvodným 2 konvolučným vrstvám sa ich tu nachádza celkovo 5. Pre lepšie tréňovanie je v RRDB blokoch použité škálovanie reziduálov, kedy sa pred sčítaním vstupu a výstupu z bloku, výstup vynásobí konštantou medzi 0 a 1. Vďaka tejto operácii je tréňovanie stabilnejšie. Pridaním nových blokov vznikla hlbšia a komplexnejšia architektúra s viacerými prepojeniami, čo zvyšuje kapacitu siete.

Okrem zmien v architektúre generátora, bol vylepšený aj **diskriminátor**. Vylepšenie je založené na práci *Relativistic GAN* [19]. Oproti štandardnému diskriminátoru z práce SR-



## Residual in Residual Dense Block (RRDB)



Obr. 4.10: Nový RRDB blok. Výstup z bloku je vynásobený parametrom reziduálneho škálovania  $\beta$ . Obrázok prevzatý z [31].

GAN, ktorý odhaduje pravdepodobnosť, že obrázok je reálny s vysokým rozlíšením, sa relativistický diskriminátor snaží odhadnúť pravdepodobnosť, že reálny obrázok  $x_r$  je relatívne viac reálnejší ako umelo zväčšený obrázok  $x_f$ . Pôvodný diskriminátor  $D(x)$  vyhodnocuje na reálnom obrázku  $x_r$  a umelo zväčšenom obrázku  $x_f$  nasledovne:

$$\begin{aligned} D(x_r) &= \sigma(C(x_r)) \rightarrow 1, \\ D(x_f) &= \sigma(C(x_f)) \rightarrow 0, \end{aligned} \quad (4.6)$$

kde  $\sigma$  je posledná sigmoid aktivačná funkcia a  $C(x)$  je výstup diskriminátora pred touto aktivačnou funkciou. Ideálny výstup pre reálny obrázok  $x_r$  je rovný 1 a pre umelo zväčšený obrázok  $x_f$  je ideálny výstup rovný 0. Relativistický diskriminátor  $D_{Ra}(x_1, x_2)$  na tom istom páre obrázkov vyhodnocuje nasledovne [31]:

$$\begin{aligned} D_{Ra}(x_r, x_f) &= \sigma(C(x_r) - \mathbb{E}[C(x_f)]) \rightarrow 1, \\ D_{Ra}(x_f, x_r) &= \sigma(C(x_f) - \mathbb{E}[C(x_r)]) \rightarrow 0, \end{aligned} \quad (4.7)$$

kde majú symboly rovnaký význam ako v rovnici vyššie. V práci je použitý konkrétne *Relativistic average Discriminator RaD* [19], označený ako  $D_{Ra}$ . Operácia  $\mathbb{E}[\cdot]$  vyjadruje priemernú hodnotu všetkých tréningových dát v danom mini-batchi.

Chybová funkcia diskriminátora  $D$  sa teda dá definovať ako [31]:

$$L_D^{Ra} = -\mathbb{E}_{x_r}[\log(D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(1 - D_{Ra}(x_f, x_r))], \quad (4.8)$$

a *adversarial* časť chybovej funkcie generátora  $G$  má symetrickú podobu [31]:

$$L_G^{Ra} = -\mathbb{E}_{x_r}[\log(1 - D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(D_{Ra}(x_f, x_r))], \quad (4.9)$$

kde  $x_f = G(x_i)$  a  $x_i$  je vstupný obrázok s nízkym rozlíšením. Táto chybová funkcia oproti pôvodnej potrebuje oba vstupy  $x_r$  a  $x_f$ . To umožňuje generátoru ťažiť z gradientov aj generovaných dát aj reálnych dát. V práci SRGAN generátor ťažil len z gradientov generovaných dát. V práci ESRGAN bolo ukázané, že použitie týchto chybových funkcií napomáha k vytváraniu ostrejších hrán a detailnejších textúr.

Ďalším vylepšením prešla **príznaková chybová funkcia**. Táto časť chybovej funkcie v práci SRGAN obsahovala príznakové vektory extrahované z  $j$ -tej konvolučnej vrstvy po aktivačnej funkcii. V práci ESRGAN sledovali percento aktivovaných neurónov vrstvy  $VGG19_{5,4}$  (architektúra VGG19, 4. konvolučná vrstva pred 5. max-pooling vrstvou) na obrázkoch. Na referenčnom obrázku bolo toto percento rovné 11.17%. Zistili že mapa aktivovaných neurónov je hustejšia pred aktivačnou funkciou a teda obsahuje viac informácií. Celková chybová funkcia generátora je v práci ESRGAN definovaná nasledovne [31]:

$$L_G = L_{percep} + \lambda L_G^{Ra} + \eta L_1, \quad (4.10)$$

kde  $L_{percep}$  je príznaková chybová funkcia (vrstva VGG19<sub>5,4</sub>),  $L_G^{Ra}$  je *adversarial* chybová funkcia z rovnice 4.9 a  $L_1$  je manhattanská vzdialenosť (MAE) medzi reálnym obrázkom a umelo zväčšeným obrázkom.  $\lambda$  a  $\eta$  sú váhové parametre jednotlivých zložiek.

Proces tréovania je podobný ako pri práci SRGAN. Najskôr je generátor pred-tréovaný s  $L_1$  vzdialenosťou. Potom sa k tréningu pridá aj diskriminátor a príznaková chybová funkcia. Pri hlbšej architektúre bolo použitých celkovo 23 RRDB blokov. Kvalita zväčšených obrázkov sa oproti SRGAN zlepšila, hlavný rozdiel je vidieť vo vysoko frekvenčných textúrach.

## Real-ESRGAN

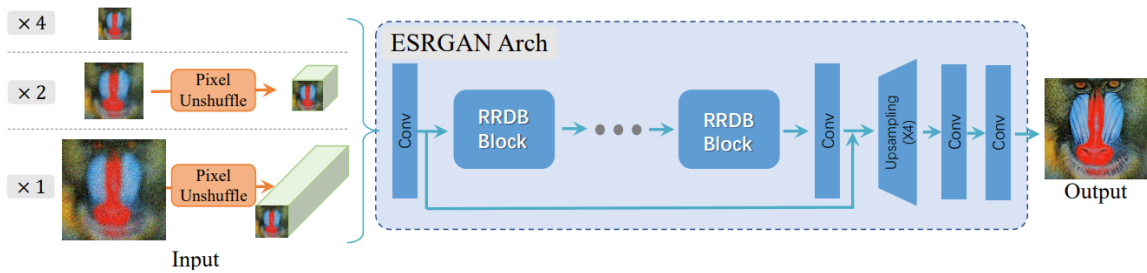
Práca Real-ESRGAN [30] priniesla ďalšie vylepšenia oproti ESRGAN. Týkajú sa najmä problému, že tréovacia dátová sada je umelo vytvorená. Tento problém spôsobuje to, že natréované architektúry nie sú schopné kvalitne zvýšiť rozlíšenie reálnych obrázkov, ktoré často trpia omnoho komplexnejšou degradáciou. Zmeny sa týkajú procesu augmentácie tréovacích dát, kam bol pridaný proces vyššieho rádu degradácie. Ďalšou zmenou je predstavenie novej architektúry diskriminátora. Nová architektúra je založená na konvolučných neurónových sieťach typu U-Net. Vďaka tomu diskriminátor dokáže ohodnotiť „reálnosť“ konkrétneho bodu vo vstupnom obrázku.

Väčšina dátových sád určených pre zvýšenie rozlíšenia obrazu často znižuje rozlíšenie obrázkov pomocou ideálneho bikubického filtra. Reálne obrázky však trpia často komplexnejšou degradáciou, ktorá zahŕňa napríklad rozmazanie pri snímaní, senzorický šum, artefakty vznikajúce pri zaostrovaní alebo JPEG kompresia. Preto autori v práci Real-ESRGAN [30] predstavili proces druhého rádu degradácie. V tomto procese ide o použitie rôznych typov degradácie v dvoch kolách. V práci sú použité nasledovné degradačné procesy:

- rozostrenie (blur),
  - pomocou gaussovského filtra,
  - pomocou 2D *sinc* filtra,
- zmena rozlíšenia,
  - bikubickou metódou,
  - bilineárnou metódou,
  - metódou najbližšieho suseda,
- pridanie šumu,
  - gaussovský šum,
  - poissonov šum,
- JPEG kompresia.

Tento proces sa opakuje dvakrát po sebe s rôznymi parametrami, ktoré sú vybrané z náhodných rozsahov. Veľkosť tréovacieho obrázku je potom upravená na potrebnú veľkosť. V práci autori hovoria o vyššom ráde degradácie, kedy sa tento proces môže opakovať celkovo  $n$ -krát. Poukázali však na fakt, že ideálne by sa degradačné operácie mali vyberať náhodne, avšak nie je to potrebné a je dôležité aby bol tento degradačný model použitý aspoň dvakrát.

Zložitejšia degradácia trénovacích dát zvyšuje aj obtiažnosť trénovania sietí. Diskriminátor potrebuje silnejšiu schopnosť rozoznávať reálne obrázky od umelo zväčšených. Preto v práci predstavili architektúru diskriminátora založenú na architektúre typu U-Net. Oproti rozoznaniu reálneho obrázku od umelého na základe globálneho štýlu, je tento diskriminátor schopný produkovať gradientnú spätnú väzbu aj pre lokálne oblasti v obrázku. Diskriminátor sa skladá z 10 konvolučných vrstiev kedy prvá konvolučná vrstva má 64 konvolučných jadier o veľkosti  $3 \times 3$ . Úloha 3 nasledovných konvolučných vrstiev je znížiť výšku a šírku príznakového vektora a zväčšiť jeho počet kanálov podobne ako pri sieťach VGG - vrstvy majú nastavený parameter  $stride = 2$  a každou vrstvou sa počet konvolučných jadier zdvojnásobuje. Konvolučné jadrá majú veľkosť  $4 \times 4$ . Ďalšie 3 konvolučné vrstvy znižujú počet kanálov a zároveň sa pomocou bilineárnej interpolácie zvyšuje šírka a výška príznakového vektora. Konvolučné jadrá majú v tomto prípade veľkosť  $3 \times 3$ . Ku každej zväčšenej príznakovkej mape sa pripočíta odpovedajúca príznaková mapa z ľavej strany siete. Na konci sa nachádzajú zvyšné 3 konvolučné vrstvy, kde 2 majú 64 konvolučných jadier o veľkosti  $3 \times 3$  a posledná vrstva obsahuje konvolučné jadrá s rovnakou veľkosťou ale ich počet je nastavený podľa potrebného počtu kanálov výstupného obrázku (3 pre RGB a 1 pre čiernobiele obrázky). Všetky konvolučné vrstvy používajú aktivačnú funkciu LeakyReLU ( $\alpha = 0.2$ ). Trénovací proces je zložitejší a menej stabilný. Autorom sa ho podarilo stabilizovať použitím regularizácie pomocou spektrálnej normalizácie [25] pre každú konvolučnú vrstvu.



Obr. 4.11: Úprava architektúry generátora z práce ESRGAN, ktorá umožňuje zvýšenie rozlíšenia vo viacerých mierkach. Obrázok prevzatý z [30].

V rámci práce upravili aj vstupnú vrstvu generátora z práce ESRGAN, kde pomocou opačnej operácie preskladania kanálov ako v sub-pixel konvolučnej vrstve (popísanej v časti 4.1), umožnili použiť generátor pre zväčšenie rozlíšenia obrazu 2-krát a 1-krát (dá sa použiť na odstránenie šumu) oproti pôvodným štyrikrát. Vďaka preskladaniu kanálov sa zmenší výška a šírka vstupného obrázku a zvýši sa počet jeho kanálov. Operácia je znázornená na obrázku 4.11.

## Kapitola 5

# Návrh riešenia pre zvýšenie rozlíšenia obrazu z elektrónového mikroskopu

Cieľom tejto práce je vytvoriť a natrénovať hlbokú neurónovú sieť, ktorá bude schopná zvýšiť rozlíšenie obrázku z elektrónových mikroskopov na základe jedného vstupného obrázku. Natrénovaná sieť by mala uľahčiť prácu operátorovi elektrónového mikroskopu počas navigácie v náhľadovom okne pri hľadaní časti vzorky, ktorý chce nasnímať vo väčšej kvalite. Neurónovú sieť je možné použiť aj pre zlepšenie kvality finálnej snímky, ale prináša to nevýhodu - sieť si môže „vymyslieť“ nejaký vzor, ktorý reálne na vzorke nie je.

Pre riešenie úlohy je potrebné nájsť alebo vytvoriť dátovú sadu pre tréning. Ďalej je potrebné navrhnuť novú, alebo použiť existujúcu (prípadne aj vylepšiť) architektúru neurónovej siete. V nasledujúcich častiach je popísaný proces získania dát a vytvorenia dátovej sady. Potom sú v časti 5.2 popísané použité architektúry a nakoniec sú spomenuté všetky použité chybové funkcie.

### 5.1 Obrázky z elektrónových mikroskopov

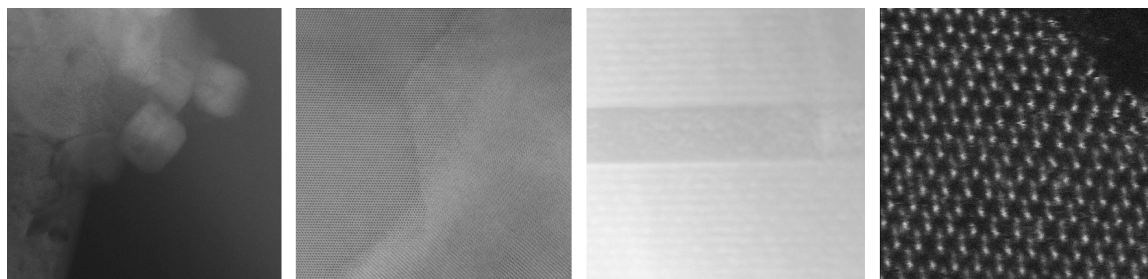
Pri hľadaní na internete nebola nájdená žiadna verejná dátová sada pre potrebnú úlohu. Nájdené dátové sady pre zvýšenie rozlíšenia obrazu boli väčšinou klasické dátové sady s farebnými obrázkami. Mali však nasledovnú štruktúru:

- priečinkov s pôvodnými HR obrázkami (všetky s rovnakým rozlíšením)
- niekoľko priečinkov, kde v každom z nich boli obrázky z priečinka HR podvzorkované s rôznou mierkou, napr. 2x, 4x, ...

Bolo teda potrebné nájsť kvalitný zdroj snímok z elektrónového mikroskopu, ktoré sú dobre zaostrené a minimálne zašumené. Obrázky by tiež mali mať dostatočné rozlíšenie, aby dávalo zmysel ich zmenšiť aspoň štyrikrát (pre úlohu zvýšenia rozlíšenia štyrikrát). Množstvo obrázkov by malo byť čo najväčšie, aby bola natrénovaná neurónová sieť schopná generalizovať aj na iných dátach.

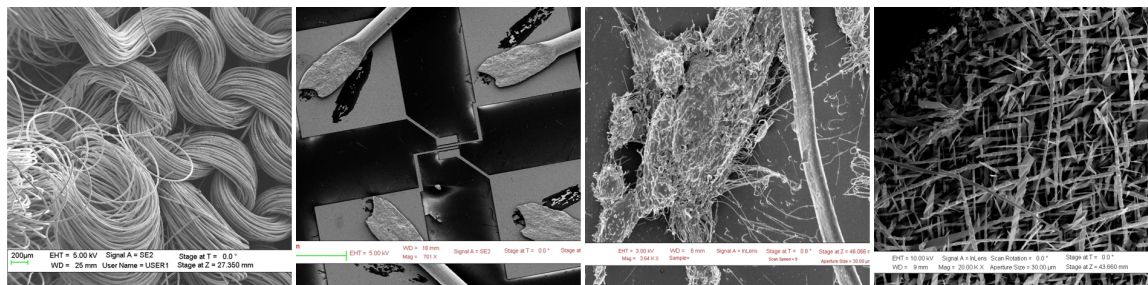
Dátová sada *Univerzity Warwick* [8] obsahuje množstvo obrázkov rozdelených podľa typu mikroskopu (SEM, TEM, STEM). V každej kategórii je viac ako 15 000 obrázkov.

Avšak v popise dátovej sady je písané, že obrázky sú získavané pomocou rôznych mikroskopov a personálu. Obrázky sú tak často v rôznych rozlíšeníach. Hlavnou nevýhodou je, že obsahujú obrázky, ktoré vytvoril slabšie zaškolený personál. Veľká časť obrázkov je zle zaostrená, obsahuje množstvo šumu a na časti obrázkov sa nachádza len maličká časť alebo vôbec žiadny objekt. Táto dátová sada by však mohla mať potenciál po prefiltrovaní zlých obrázkov. Ukážky niekoľkých snímok je na obrázku 5.1



Obr. 5.1: Obrázky z dátovej sady *Univerzity Warwick* [8]. Väčšina obrázkov neobsahuje žiadny konkrétny objekt alebo viditeľný a zaostrený štrukturovaný povrch.

Ďalšia dátová sada patrí *Výskumnému Inštitútu CNR-IOM* [2]. Táto dátová sada pozostáva z obrázkov zostrojených pomocou SEM. Výhodou je, že je rozdelená do viacerých kategórií, podľa toho čo sa na obrázkoch nachádza. Medzi kategórie s najväčším množstvom obrázkov patrí napríklad biologické dáta, nanodrôty, častice a mikro-elektromechanické zariadenia a elektródy. Táto dátová sada je vhodnejší kandidát na danú úlohu. Väčšina obrázkov obsahuje zaostrené objekty s malým množstvom šumu. Preto bola vybraná pre ďalší postup. Príklady snímok z rôznych kategórií sú na obrázku 5.2.



Obr. 5.2: Obrázky z dátovej sady *Výskumného Inštitútu CNR-IOM* [2]. Obrázky pochádzajú z kategórií (v tomto poradí): vlákna, mikro-elektromechanické zariadenia a elektródy, biologické dáta, nanodrôty.

## Základná dátová sada z existujúcich dát

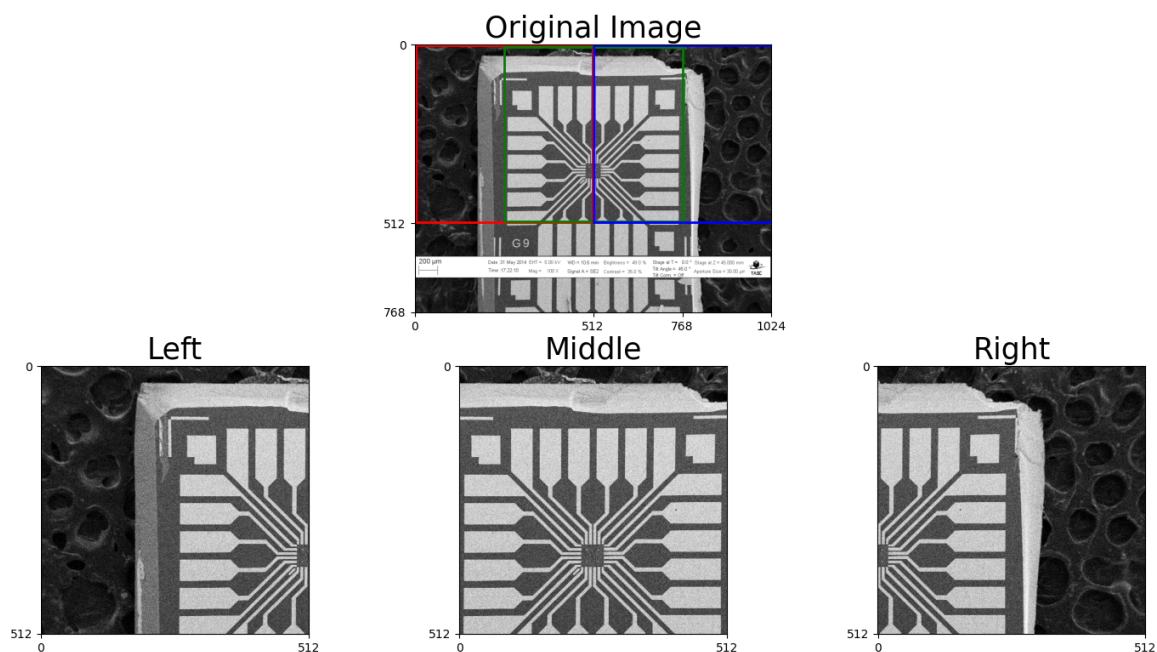
Rôzne kategórie obrázkov vo vybranej dátovej sade majú rozdielny charakter. V rámci tejto práce bola vybraná jedna konkrétna kategória obrázkov. Najpočetnejšia bola kategória mikro-elektromechanické zariadenia a elektródy. Obsahuje 4 590 obrázkov s rozlíšením minimálne  $1024 \times 768$  px. Niekoľko obrázkov má vyššie rozlíšenie, ale jedná sa o jednotky percent z celkového počtu. Nevýhodou je, že obrázky (v každej kategórii) obsahujú meta-dáta o každom obrázku priamo v obrázku ako je možné vidieť na obrázku 5.3.

Obrázky boli orezané tak, aby sa metadáta odstránili. Orezaný výsledok bol rozdelený na 3 štvorcové obrázky s rozlíšením  $512 \times 512$  px, ktoré sa prekrývajú. Týmto spôsobom sa počet obrázkov v dátovej sade zväčšil trojnásobne, celkovo ich je **13770**. Stredný obrázok je redundantný (obidve jeho polovice sa nachádzajú buď v ľavom, alebo v pravom obrázku), ale často je hlavný objekt snímania práve v strede obrázku. Objekt v strede často obsahuje väčšie množstvo detailov a bola by škoda ho rozdeliť.

Získané obrázky boli zmenšené v mierkach **2x**, **4x** a **8x**, pre neskoršie účely tréningu neurónových sietí. To zaručuje možnosť tréningu vo viacerých mierkach zväčšenia s rôznymi rozlíšeniami. Dát je aj tak pomerne málo, preto je vhodné použiť augmentáciu dát:

- pridávanie poissonovho šumu (tento šum vzniká pri získavaní obrázkov z elektrónového mikroskopu),
- zmena celkového jasú,
- zmena kontrastu
- horizontálne a vertikálne otáčanie obrázkov,

Týmto spôsobom sa umelo zväčší dátová sada a natrénovaná neurónová sieť by sa nemala pretréňovať a mala by lepšie generalizovať.



Obr. 5.3: Rozdelenie obrázka z dátovej sady [2] pre vytvorenie vlastnej dátovej sady. Obrázok hore je pôvodný obrázok. Obsahuje farebné štvorce, ktoré reprezentujú výsledné 3 prekrývajúce sa výrezy. Dole sú zobrazené jednotlivé výrezy použité vo vytvorenej dátovej sade ako 3 rôzne obrázky. Touto operáciou sa odstráni aj spodná nežiadúca časť obrázka, obsahujúca metadáta o danom obrázku.

## Dátová sada s pokročilou degradáciou obrázkov

Dodatočne bolo načítanie dátovej sady upravené, aby odpovedalo degradačnému modelu testovacích obrázkov z práce Real-ESRGAN [30] (popísanému v časti 4.4). Táto verzia načítavania dátovej sady oreže spodnú časť obrázka obsahujúcu informácie o snímaní rovnako ako predchádzajúca verzia. Potom sú aplikované rovnaké augmentačné metódy, navyše je pridaná dvoj-úrovňová degradácia pri vytváraní obrázka s nízkym rozlíšením. Použité degradačné procesy sú:

- rozostrenie (blur),
  - pomocou gaussovského filtra,
- zmena rozlíšenia,
  - bikubickou metódou,
  - bilineárnou metódou,
  - metódou najbližšieho suseda,
- pridanie šumu,
  - poissonov šum,
- JPEG kompresia,

ktoré sú inšpirované prácou Real-ESRGAN, pričom je vylúčené použitie *sinc* filtra a pridávanie gaussovského šumu. Je zachovaná aj JPEG kompresia, pretože obrázky z dátovej sady sú dostupné len vo formáte JPEG.

Obrázok s nízkym rozlíšením je v tomto prípade 2x alebo 4x menší ako pôvodný obrázok s vysokým rozlíšením. Je možné ho zmenšiť ešte celkovo dvakrát, aby miera zmeny veľkosti odpovedala predchádzajúcej základnej sade. Miera degradačných procesov je nastaviteľná intervalom, z ktorého sa vyberie náhodné číslo pre daný typ degradácie. Obrázok nie je v tomto prípade rozdelený na 3 časti, ale vždy je z neho pri tréningu vybraný náhodný výrez o požadovanej veľkosti, čo zaručuje ďalšie umelé zväčšenie dátovej sady.

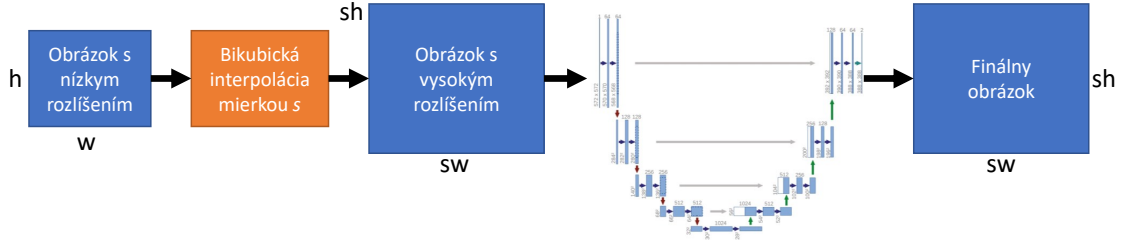
## 5.2 Architektúry

V práci bolo vybraných niekoľko architektúr neurónových sietí schopných plniť úlohu zvýšenia rozlíšenia obrazu. Pri architektúrach boli v niektorých prípadoch vykonané drobné zmeny. Architektúry boli tréňované pomocou dvoch dátových sád popísaných vyššie v časti 5.1 a 5.1, kde sa menila aj miera zväčšenia rozlíšenia a aj rozlíšenie vstupných obrázkov. Siete boli tiež tréňované pomocou rôznych kombinácií chybových funkcií a bola porovnaná ich schopnosť zvýšiť rozlíšenie obrazu voči zvýšeniu rozlíšenia pomocou bikubickej interpolácie, ktorá sa bežne používa.

### Architektúra U-Net

Architektúra U-Net bola vybratá ako prvotné riešenie. Bližšie je popísaná v časti 4.4. Pre úlohu zvýšenia rozlíšenia bol oproti pôvodnej architektúre pridaný *padding* pri konvolučných vrstvách, aby bola veľkosť vstupu v rovnakom rozlíšení ako veľkosť výstupu, nie menšia

ako pri pôvodnej architektúre. Vstupný obrázok s nízkym rozlíšením je potrebné zväčšiť ľubovoľnou jednoduchou metódou (napr. bikubickou interpoláciou). Sieť následne „opraví“ chyby, ktoré vzniknú (napr. kostrbaté hrany). Tento proces je znázornený na obrázku 5.4.



Obr. 5.4: Princíp zvýšenia rozlíšenia obrázkov pri sieti U-Net. Písmená  $h$  a  $w$  reprezentujú výšku a šírku daného obrázku. Písmeno  $s$  je mierka zvýšenia rozlíšenia. Obrázok architektúry prevzatý z [26].

## Architektúra RUNet

Táto architektúra bola vybraná preto, že je principiálne podobná predchádzajúcej architektúre U-Net. Bola však vyvinutá priamo pre úlohu zvýšenia rozlíšenia ako bolo spomenuté v časti 4.4 pri popise tejto architektúry. V práci [16] však pracovali s farebnými obrázkami, preto bolo potrebné zmeniť odpovedajúce počty kanálov konvolučných vrstiev pri vstupe a výstupe. Zvýšenie rozlíšenia funguje na rovnakom princípe ako pri U-Net (obrázok 5.4). Pri práci s architektúrou bolo zistené, že v niektorých prípadoch dosť mení intenzitu čiernobielych obrázkov. Tento problém sa podarilo redukovat', nie však úplne odstrániť. Úpravy spočívali v odstraňovaní rôznych batch-normalizačných vrstiev, čo malo pozitívny efekt. Nakoniec bola odstránená len prvá batch-normalizačná vrstva v architektúre. O ktorú vrstvu ide znázorňuje obrázok 5.5.

## Architektúra Real-ESRGAN

V dnešnej dobe pri zvýšení rozlíšenia obrazu excelujú architektúry typu GAN. Preto bola vybratá architektúra Real-ESRGAN popísaná v časti 4.4. Jedná sa o vylepšenie prác popísaných v časti 4.4. Pri tejto architektúre nie je potrebné zvyšovať rozlíšenie obrázku vopred. Architektúra generátora ani diskriminátora nebola v tejto práci nijak modifikovaná.

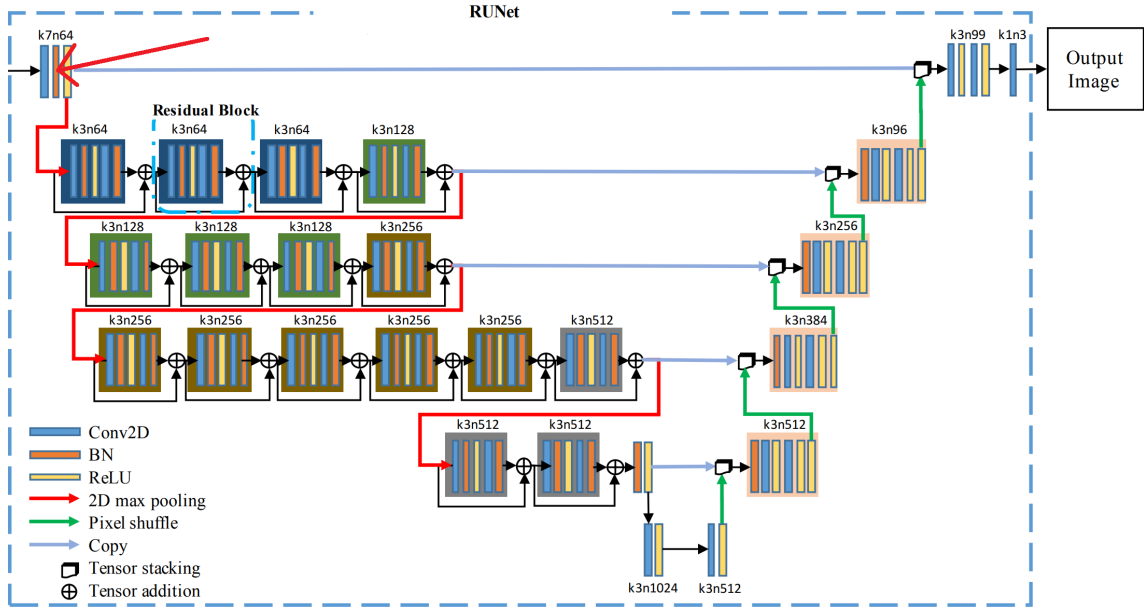
## 5.3 Použité chybové funkcie

V práci bolo použitých niekoľko chybových funkcií. Prvou z nich je chybová funkcia, ktorá je kombináciou metrík PSNR a SSIM. Jej cieľom je maximalizovať hodnoty týchto metrík. Vyzerá nasledovne:

$$\mathcal{L}_{met} = -1(\alpha \cdot PSNR + \beta \cdot SSIM), \quad (5.1)$$

kde  $PSNR$  a  $SSIM$  sú popísané v časti 3.3 a  $\alpha$  a  $\beta$  sú váhové hodnoty pre jednotlivé zložky. Celý výraz je záporný, pretože pri tréňovaní neurónových sietí sa chyba minimalizuje





Obr. 5.5: Šípka znázorňuje, ktorú batch-normalizačnú vrstvu bolo potrebné odstrániť aby sa redukoval problém zmeny intenzity výstupným obrázkov. Obrázok architektúry prevzatý z [16] a upravený.

a obe zložky majú charakter opačný - čím sú vyššie, tým by obrázok mal byť kvalitnejší. Táto chybová funkcia bola použitá pri oboch sieťach typu U-Net.

Ďalšou chybovou funkciou je absolútna odchýlka (tzv. L1 alebo manhattanská vzdialenosť). Chybová funkcia bola vybraná po vzore z práce ESRGAN [31], kde ju využili na predtrénovanie generátora. Jej predpis je nasledovný:

$$\mathcal{L}_{L_1} = \frac{1}{N} \sum_{i=1}^N |y_i - x_i|, \quad (5.2)$$

kde  $N$  reprezentuje počet pixelov v obrázku,  $y_i$  je hodnota pixelu pôvodného obrázku s vysokým rozlíšením a  $x_i$  je hodnota toho istého pixelu v umelo zväčšenom obrázku. Táto chybová funkcia bola použitá samostatne, ale aj v kombinácii s príznakovou chybovou funkciou popísanou ďalej.

V práci bola použitá aj príznaková chybová funkcia v kombinácii s  $\mathcal{L}_{L_1}$  popísanou vyššie. Táto kombinácia bola použitá tiež v práci ESRGAN [31] v kompletnej chybovej funkcii generátora (rovnica 4.10). Je definovaná nasledovne:

$$\mathcal{L}_{perc} = \sum_{x=1}^K \left[ w_x \cdot l_{VGG/i,j}^{SR} \right] + \eta \mathcal{L}_{L_1}, \quad (5.3)$$

kde  $l_{VGG/i,j}^{SR}$  je príznaková chybová funkcia (popísaná v podrobne v rovnici 4.1), výstup po  $j$ -tej konvolučnej vrstve **pred aktivačnou funkciou** pred  $i$ -tou maxpooling vrstvou predtrénovanej siete VGG (pre úlohu klasifikácie obrázkov). Ide o súčet výstupov z  $K$  vybraných vrstiev, kde je váha pre danú vrstvu  $w_x$ . Parameter  $\eta$  je váha L1 vzdialenosti (rovnica 5.2) medzi reálnym a umelo zväčšeným obrázkom.

Okrem chybových funkcií popísaných vyššie sú pri tréovaní použité chybové funkcie špecifické pre architektúry typu GAN. Ide o chybovú funkciu relativistického diskriminátora  $L_D^{Ra}$  (popísanú v rovnici 4.8). Chybová funkcia generátora je kombináciou  $\mathcal{L}_{perc}$  a *adversarial* časti chybovej funkcie generátora  $L_G^{Ra}$  (popísanej v rovnici 4.9), kde sa definícia celej tejto funkcie nachádza už v rovnici 4.10.

## Kapitola 6

# Implementácia

Pri implementácii riešenia bol hlavne použitý jazyk Python<sup>1</sup> vo verzii 3.8. Pre prácu s neurónovými sieťami bol využitý nástroj TensorFlow<sup>2</sup> vo verzii 2.5.0 a jeho nadstavba Keras<sup>3</sup>, ktoré výrazne uľahčujú a urýchľujú implementáciu a trénovanie neurónových sietí. Mimo to poskytujú aj množstvo rozličných funkcií pre prácu s dátami, pre profilovanie alebo hľadanie chýb v implementácii.

TensorFlow poskytuje aj nástroj TensorBoard<sup>4</sup>, ktorý slúži najmä na vizualizáciu procesu trénovania v podobe grafov pre chybové funkcie alebo metriky. Tiež je možné celý výpočet vrátane implementovaných modelov vizualizovať pomocou grafu, kde sa táto reprezentácia mimochodom používa aj na optimalizáciu celého výpočtu. Pri riešení úlohy zvýšenia rozlíšenia obrazu sa hodí aj možnosť vykresľovať predikcie modelov počas procesu učenia. Tento nástroj umožňuje aj toto. Tiež je možné ukladať si informácie v textovej podobe.

Verzie použitých nástrojov boli dôkladne vybrané tak, aby bolo možné vykonať proces trénovania na distribuovanej výpočtovej infraštruktúre Metacentrum<sup>5</sup>. Metacentrum poskytuje výpočtové stroje s grafickými kartami značky Nvidia<sup>6</sup>, ktoré je možné využiť na akceleráciu neurónových sietí. Toto umožnilo niekoľkonásobné zrýchlenie celého trénovacieho procesu. Nvidia poskytuje Docker<sup>7</sup> kontajnery obsahujúce nástroje pre strojové učenie. Tieto kontajnery je možné na Metacentre spustiť pomocou Singularity<sup>8</sup>, ktorý oproti Docker umožňuje vykonávať vysoko výkonné výpočty (angl. *high-performance computing* - HPC). Pri implementácii bol vybraný existujúci kontajner na Metacentre `TensorFlow\21.08-tf2-py3.SIF`, ktorý obsahoval potrebné verzie nástrojov.

V nasledujúcich podkapitolách budú popísané implementačné detaily rôznych častí od spôsobu načítania dátových súborov, cez architektúry, až po implementáciu trénovacieho procesu.

---

<sup>1</sup><https://www.python.org/>

<sup>2</sup><https://www.tensorflow.org/>

<sup>3</sup><https://keras.io/>

<sup>4</sup><https://www.tensorflow.org/tensorboard>

<sup>5</sup><https://metavo.metacentrum.cz/>

<sup>6</sup><https://www.nvidia.com/>

<sup>7</sup><https://www.docker.com/>

<sup>8</sup><https://apptainer.org/>

## Načítanie dátovej sady

Implementované boli dve verzie načítania dátovej sady spomenuté v kapitole 5. Všetky augmentácie sa dejú za behu programu, čo prináša viac náhodnosti do celého procesu tréningovania. Hlavným rozdielom medzi dátovými sadami je to, kedy sa obrázky zmenšujú. Inak je načítanie dát principiálne rovnaké. Základná dátová sada je obalená triedou `NFFA_SEM` a dátová sada s pokročilou degradáciou je obalená v triede `NFFA_SEM_v2`. Pri vytváraní oboch spôsobov načítania dátovej sady bolo využité profilovanie pomocou TensorFlow Profiler<sup>9</sup>, vďaka ktorému bolo možné zistiť (a potom zamedziť), či je proces tréningovania brzdený čakáním na načítanie dát.

Dátová sada ako objekt je typu `tf.data.Dataset`, čo umožňuje použiť metódu `map()`, ktorá na každú položku v dátovej sade zavolá poskytnutú funkciu. Výhodou tohto postupu je, že TensorFlow vie automaticky odhadnúť počet jadier procesora potrebných pre optimálne načítanie dát. Objekt `tf.data.Dataset` je vytvorený zo zoznamu ciest k obrázkom pomocou `from_tensor_slices()`. Potom je pomocou `map()` nad každou cestou obrázka zavolaná funkcia, ktorá obrázok načíta a prenормalizuje do potrebného intervalu.

Dátová sada je rozdelená na tréningovú, validačnú a testovaciu. Pre každú z týchto častí je vytvorený iterátor, ktorý vykoná potrebné augmentácie a vytvorí tzv. dávky (ďalej angl. *batch*). Pre tréningovanie je potrebné načítať dáta čo najrýchlejšie, čomu pomáha metóda `tf.data.Dataset.prefetch()`, ktorá obrázky načíta a transformuje ich do potrebného tvaru dopredu. To ako veľmi dopredu je obrázky potrebné načítať vie TensorFlow odhadnúť automaticky. Vďaka metóde `tf.data.Dataset.shuffle()` je poradie obrázkov pri každej iterácii zamiešané.

### Základná dátová sada

Pri základnej dátovej sade je potrebné obrázky zmenšiť ešte pred celým procesom tréningovania. K tomu slúži skript `create_dataset.py`, ktorý nájde všetky obrázky s formátom JPEG v danej zložke a zmenší ich v mierkach 2x, 4x a 8x. Pre každú z týchto mierok sa vytvorí vlastná zložka (s názvom podľa mierky zmenšenia), plus sa pôvodné obrázky vložia do separátnej zložky `HR`. Okrem toho každý obrázok rozdelí na tri časti, čo je znázornené na obrázku 5.3. Takto transformovaná dátová sada je načítaná triedou `NFFA_SEM`.

Pri vytváraní iterátora sa otvorí dvojica obrázkov, kde jeden odpovedá obrázku s vysokým rozlíšením a druhý obrázku s nízkym rozlíšením. Obrázky sú v pároch na základe rovnakého mena súboru. Každý z nich sa však nachádza v inej zložke, napr. `HR` (512×512 px) a `2x` (256×256 px). Z ktorých zložiek je potrebné čítať sa nastaví parametrom `train_ratio` a `gt_ratio` pri inštanciování triedy tejto dátovej sady.

Na každú dvojicu obrázkov sú pri tréningovej časti dátovej sady aplikované spomínané augmentácie. Pre úpravu obrázkov pri augmentácii sú použité operácie v module `tf.image`. Pri generovaní poissonovho šumu do obrázkov je možné nastaviť pravdepodobnosť vygenerovania a intenzitu šumu. Generovanie poissonovho šumu bolo implementované pomocou TensorFlow operácií na základe implementácie v balíčku `scikit`.

### Dátová sada s pokročilou degradáciou

V tomto prípade je kompletná úprava vstupných obrázkov, vrátane zmenšovania, robená za behu, čo zjednodušuje prácu s dátovou sadou oproti predchádzajúcemu prípadu, kedy

<sup>9</sup>[https://www.tensorflow.org/tensorboard/tensorboard\\_profiling\\_keras](https://www.tensorflow.org/tensorboard/tensorboard_profiling_keras)

bolo potrebné najskôr predspracovať dátovú sadu. Deje sa tu aj samotné orezanie spodnej časti obrázkov, ktorá obsahuje informácie o parametroch akvizície.

Z disku sa načítava teda priamo obrázok z nájdenej dátovej sady *Výskumnému Inštitútu CNR-IOM* [2]. Pri trénovacej časti je z každého orezaného obrázka vyrezaný náhodný výrez, na ktorý sú aplikované rovnaké augmentácie ako v predchádzajúcom prípade. Potom je obrázok duplikovaný a na jeho kópiu sú aplikované degradačné metódy, ktorých parametre sú popísané v slovníku `degr_params`. Všetky degradačné operácie sú implementované pomocou TensorFlow. Kópia je následne zmenšená na požadovanú veľkosť. Takto vznikne trénovací pár. Pri validačnej a testovacej časti dátovej sady sa z pôvodného obrázka vyreže výrez v potrebnom rozlíšení zo stredu obrázka a potom sa duplikuje a zmenší, aby bolo zaistené, že bude pri každej iterácii rovnaký.

## Architektúry

Každá architektúra je implementovaná pomocou vlastnej triedy. Keras poskytuje niekoľko možností ako poskladať architektúru neurónovej siete. Okrem vrstiev Keras obsahuje možnosť vytvoriť vlastné stavebné bloky (niekoľko vrstiev po sebe), z ktorých je možné poskladať aj zložitejšie architektúry. Takto vytvorené modely, t.j. dedením z triedy `tf.keras.Model`, poskytujú funkcionálnu, ktorá uľahčuje prácu ako napríklad metóda `.fit()`, vďaka ktorej sa model dá natrénovať bez nutnosti implementácie vlastného trénovacieho cyklu.

### Základný U-Net

Implementácia architektúry siete U-Net bola prevzatá z repozitára Joela Akereta<sup>10</sup>. Model bol však upravený aby ho bolo možné použiť pre úlohu zvýšenia rozlíšenia. Ako je načrtnuté na obrázku 5.4, je potrebné zakomponovať operáciu zvýšenia rozlíšenia pomocou bikubickej interpolácie. Ďalšou úpravou bolo nastavenie parametra `padding` na hodnotu `'same'` pre konvolučné vrstvy, aby sa rozmery vstupného a výstupného obrázka rovnali. Vďaka tomu bolo možné z bloku `CropConcatBlock`, ktorý mal za úlohu spájať ľavú časť U-Net s pravou, odstrániť operáciu orezávania príznakových máp z pravej časti tak, aby sa dimenziami zhodovali s tými v ľavej časti. Poslednou úpravou bola zmena inicializácie váh na `'glorot_uniform'`<sup>11</sup>, ktorá sa pre danú úlohu a dáta ukázala ako najvhodnejšia, a to pri všetkých architektúrach.

Model sa skladá z troch typov vlastných blokov, kde `ConvBlock` reprezentuje jednu úroveň na ľavej strane, `UpconvBlock` zas na pravej. Posledným blokom je `CropConcatBlock`, ktorý ma za úlohu operáciu konkatenácie príznakových vektorov z ľavej a pravej strany. Celý model je vytvorený pomocou Funkcionálneho API<sup>12</sup>, kedy sa vytvorí graf danej siete pozostávajúci z vrstiev.

### RUNet

Pre túto architektúru nebola nájdená žiadna oficiálna ani neoficiálna implementácia. Model bol teda implementovaný podľa obrázka 4.6. Spôsob vytvorenia modelu je princípálne rovnaký ako pri sieti U-Net. Obsahuje však päť typov blokov. Vstupný `InputBlock` vstup zväčší bikubicou interpoláciou, po ktorej nasleduje konvolučná vrstva. Každý reziduálny

<sup>10</sup><https://github.com/jakeret/unet>

<sup>11</sup>[https://www.tensorflow.org/api\\_docs/python/tf/keras/initializers/GlorotUniform](https://www.tensorflow.org/api_docs/python/tf/keras/initializers/GlorotUniform)

<sup>12</sup><https://www.tensorflow.org/guide/keras/functional>

blok ľavej strany je obalený v `DownSampleBlock`. V najhlbšej časti siete sa nachádza jeden `BottleNeckBlock`. Pravá strana pozostáva z `UpSampleBlock` medzi ktorými je tzv. pixel shuffle (sub-pixel konvolučná vrstva z kapitoly 4.1). Táto operácia je v TensorFlow reprezentovaná funkciou `tf.nn.depth_to_space()`<sup>13</sup>. Na konci sa nachádza jeden `OutputBlock`, ktorý zahrňuje dve predposledné konvolučné vrstvy.

Pre vytvorenie celého modelu je využitý Funkcionálne API rovnako ako pri U-Net. Pri práci bolo experimentované s odstraňovaním batch-normalizačných vrstiev, preto je pri inštanciovaní modelu možné poskytnúť parameter so zoznamom indexov blokov, v ktorých má byť táto vrstva odstránená.

## Real-ESRGAN

K tejto architektúre existuje repozitár s oficiálnou implementáciou<sup>14</sup> pomocou nástroja PyTorch<sup>15</sup>. Pri vlastnej implementácii pomocou TensorFlow bola implementácia v tomto repozitári braná ako referenčná.

Pri architektúre generátora je použitý `ResidualDenseBlock_5C` s piatimi konvolučnými vrstvami. Trojica týchto blokov tvorí jeden RRDB blok ako je znázornené na obrázku 4.10. Z potrebného počtu RRDB je vytvorený sekvenčný model pomocou `tf.keras.Sequential`. Tento sekvenčný model tvorí väčšinu modelu generátora. Po ňom nasleduje už len dvojica sub-pixel konvolučných vrstiev a dvojica klasických konvolučných vrstiev.

Architektúra diskriminátora sa skladá len zo základných konvolučných vrstiev `tf.keras.layers.Conv2D`, nebolo potrebné vytvárať vlastné stavebné bloky. Pri každej konvolučnej vrstve je však použitá spektrálna normalizácia [25]. Tá nie je implementovaná priamo v základe nástroja TensorFlow, ale možné použiť rozširujúci modul `tfa`<sup>16</sup>, v ktorom sa už nachádza `tfa.layers.SpectralNormalization`<sup>17</sup>.

## Bikubický model

Pre jednoduchú prácu pri porovnávaní výstupov bol vytvorený aj `BicubicModel`, ktorý obaľuje operáciu zvýšenia rozlíšenia obrazu pomocou bikubickej interpolácie. Vďaka tomu, že sa jedná o triedu `tf.keras.Model`, je možné zavolať metódu `.evaluate()`, ktorá ohodnotí konkrétny model pomocou poskytnutých metrík na poskytnutej testovacej časti dátovej sady.

## Chybové funkcie

Všetky chybové funkcie popísané v kapitole 5.3 sú implementované v súbore `models/losses.py`.

## Kombinácia metrík PSNR a SSIM

Chybovú funkciu kombinujúcu metriky PSNR a SSIM popísanú v rovnici 5.1 bolo nutné obaliť do vlastnej funkcie `weighted_loss_wrapper()`, kedy sa funkcii predajú parametre (váhy  $\alpha$  a  $\beta$ ) a vráti sa funkcia, kompatibilná s tréňovaním pomocou metódy `tf.keras.Model.fit()`.

<sup>13</sup>[https://www.tensorflow.org/api\\_docs/python/tf/nn/depth\\_to\\_space](https://www.tensorflow.org/api_docs/python/tf/nn/depth_to_space)

<sup>14</sup><https://github.com/xinntao/BasicSR>

<sup>15</sup><https://pytorch.org/>

<sup>16</sup>[https://www.tensorflow.org/addons/api\\_docs/python/tfa](https://www.tensorflow.org/addons/api_docs/python/tfa)

<sup>17</sup>[https://www.tensorflow.org/addons/api\\_docs/python/tfa/layers/SpectralNormalization](https://www.tensorflow.org/addons/api_docs/python/tfa/layers/SpectralNormalization)

## Príznaková chybová funkcia

V súbore sa nachádza aj trieda `VGGandPixelLoss`, ktorá obaluje príznakovú chybovú funkciu z rovnice 5.3. Pri inštanciovaní triedy je potrebné zadať váhu  $\eta$  a zoznam blokov, po ktorých sa má extrahovať príznakový vektor zo siete VGG. K tomu je nutné pridať aj zoznam váh pre jednotlivé vektory. Sieť VGG, ktorá je jadrom príznakovej chybovej funkcie pochádza z `tf.keras.applications.vgg19`, odkiaľ je možné automaticky stiahnuť natrénovaný model. Výstupy modelu sú definované zoznamom blokov. Konvolučné vrstvy modelu však obsahovali aktivačnú funkciu priamo vo vrstve. Bolo teda potrebné vytvoriť kópiu modelu a výstupy získať z miesta ešte pred aktivačnou funkciou. Stiahnutý model bol trébovaný na úlohu klasifikácie obrázkov. Obrázky mali rozsah  $\langle 0; 255 \rangle$ . Preto boli vstupné obrázky (v tejto práci sú obrázky normalizované do intervalu  $\langle 0; 1 \rangle$ ) preškálované do tohto intervalu. To spôsobilo problém v nepomere vzdialeností príznakových máp zo siete VGG a L1 vzdialenosti medzi obrázkami, ktoré boli v inom intervale. Preto je L1 vzdialenosť medzi obrázkami tiež počítaná na intervale  $\langle 0; 255 \rangle$ .

## Chybové funkcie špecifické pre GAN

Tieto chybové funkcie sú obalené v triede `RaGANLoss`, kde sa nachádzajú dve metódy: pre chybovú funkciu generátora a diskriminátora. Implementácia týchto chybových funkcií je založená na repozitári<sup>18</sup>, ktorý rozširuje TensorFlow o prácu so sieťami typu GAN. Ide o implementáciu krížovej entropie pomocou relativistického diskriminátora (popísané v rovniciach 4.8 a 4.9). Výstup generátora sa transformuje odčítaním priemernej hodnoty výstupu diskriminátora. Pri výstupe diskriminátora sa naopak odčíta priemerná hodnota výstupu generátora. Z takto transformovaných hodnôt sa spočíta krížová entropia, kedy pri generátore majú transformované výstupy generátora triedu 1 a transformované výstupy diskriminátora majú triedu 0. Pri diskriminátore sú tieto triedy opačné.

## Trébovanie modelov

Pri implementácii trébovacích skriptov bol kladený dôraz na to, aby bolo možné celý proces pohodlne sledovať. To je možné vďaka modulu `TensorBoard`. Každá architektúra má vlastný trébovací skript. Trébovanie sietí U-Net a RUNet je riešené pomocou metódy `tf.keras.Model.fit()`, kedy nie je potrebné implementovať trébovací cyklus. Pri architektúrach GAN je proces o niečo zložitejší. V prípade tejto práce je tiež potrebné architektúru generátora predtrébovať samostatne, preto sú skripty pre sieť Real-ESRGAN dva.

Všetky skripty obsahujú na začiatku Python slovník, kde je možné nastaviť všetky parametre trébovacieho procesu. Slovník s parametrami sa taktiež uloží k danému behu ako text. Okrem parametrov je tu možné nastaviť napríklad aj komentár, ktorý sa využije pri vytváraní priečinku s logmi a natrénovanými váhami modelu. Názov týchto zložiek vždy obsahuje dátum a čas, názov architektúry, niekoľko dôležitých parametrov trébovania a prípadne komentár. To umožňuje ľahké filtrovanie behov pomocou regulárnych výrazov v `TensorBoard`.

<sup>18</sup><https://github.com/tensorflow/gan>

## U-Net a RUnet

Medzi hlavné parametre pri týchto sieťach patrí výber spôsobu načítania dát a voľba chybovej funkcie. Na základe zvolených parametrov sa vytvorí model, inicializuje sa pomocou metódy `tf.keras.Model.compile()`, ktorej sa zadá metóda tréovania (vo forme `tf.keras.optimizers`<sup>19</sup>), chybová funkcia a potrebné metriky. Následne sa vytvoria tzv. *callback*<sup>20</sup> funkcie pre ukladanie váh so zadanou periódou, pre logovanie metrík a vykresľovanie predikcie po každej epoche pomocou TensorBoard. Takto inicializovaný model je možné tréovať pomocou `.fit()` po poskytnutí iterátora pre tréovacie dáta, počet epoch a niekoľkých ďalších parametrov.

Váhy siete sú uložené počas tréovania so zadanou periódou a sú uložené vždy po ukončení tréovania a evaluácie na testovacej časti dátovej sady. Bolo pozorované, že siete RUnet a U-Net v prípade použitia príznakovej chybovej funkcie, nie sú vždy schopné upravovať svoje váhy tak, aby vznikali zmysluplné obrázky. Preto bolo nutné implementovať aj možnosť predtréovania na inej chybovej funkcii, alebo s nastavením vysokej hodnoty parametra  $\eta$  v chybovej funkcii (rovnica 5.3), kedy L1 vzdialenosť dostatočne prevážila príznakovú chybovú funkciu. Počet epoch pre predtréovanie je voliteľný parameter, často bola však jedna epocha postačujúca.

## Real-ESRGAN

Pri sieti Real-ESRGAN nebolo potrebné vyberať medzi viacerými chybovými funkciami alebo spôsobmi načítania dát. Dáta sa vždy načítavali s pokročilou degradáciou. Chybová funkcia sa líšila len pri predtréovaní generátora, kedy išlo o L1 vzdialenosť medzi obrázkami. Pri spoločnom tréovaní generátora a diskriminátora šlo o chybové funkcie popísané v rovniciach 4.8 a 4.10.

Pri tréovaní je potrebné tréovať dva modely naraz. Bolo teda nutné implementovať tréovací cyklus ručne. Tým sa stratila výhoda *callback* funkcií, kedy bolo ich funkcionality potrebné zakomponovať do tréovacieho cyklu. Vytváranie modelov a dátovej sady je rovnaké ako pri sieťach typu U-Net. Ukladanie váh je však riešené pomocou `tf.train.CheckpointManager`<sup>21</sup>. To uľahčuje prácu s váhami a je možné napríklad držať len posledných 5 uložených váh, kedy sa staršie automaticky vymažú. Pre logovanie chybových funkcií, metrík a predikcií do TensorBoard je potrebné vytvoriť objekt schopný zapisovať pomocou `tf.summary.create_file_writer()`<sup>22</sup>.

Ďalšou vecou, ktorú bolo potrebné implementovať bolo vypisovanie informácií o tréovaní do príkazového riadku. Na to poskytuje TensorFlow nástroj `tf.keras.utils.Progbar`<sup>23</sup>, ktorý je použitý aj priamo v automatickej metóde tréovania `tf.keras.Model.fit()` a poskytuje prívetivú formu vypisovania potrebných metrík a informácií. Okrem čísla aktuálnej epochy, dávky a metrík sa tu nachádza aj odhadovaný čas do konca epochy.

Jedna iterácia tréovacieho cyklu pozostáva z tréovacieho kroku pre všetky dávky a validačného kroku. Tréovací krok zahrňuje predikciu, spočítanie chybových funkcií a metrík a úpravu váh siete pomocou vypočítaných gradientov z chybovej funkcie. Aktualizácia logovaných hodnôt prebieha periodicky po nastavenom počte dávok. Po vykonaní tréovacieho kroku na všetkých dávkach v aktuálnej epoche nasleduje validačný krok. Ten

<sup>19</sup>[https://www.tensorflow.org/api\\_docs/python/tf/keras/optimizers](https://www.tensorflow.org/api_docs/python/tf/keras/optimizers)

<sup>20</sup>[https://www.tensorflow.org/api\\_docs/python/tf/keras/callbacks/Callback](https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/Callback)

<sup>21</sup>[https://www.tensorflow.org/api\\_docs/python/tf/train/CheckpointManager](https://www.tensorflow.org/api_docs/python/tf/train/CheckpointManager)

<sup>22</sup>[https://www.tensorflow.org/api\\_docs/python/tf/summary/create\\_file\\_writer](https://www.tensorflow.org/api_docs/python/tf/summary/create_file_writer)

<sup>23</sup>[https://www.tensorflow.org/api\\_docs/python/tf/keras/utils/Progbar](https://www.tensorflow.org/api_docs/python/tf/keras/utils/Progbar)



spočíta hodnoty metrik pre validačnú časť dátovej sady a uloží ich. Potom sa vykoná predikcia na obrázku v testovacej časti dátovej sady a tiež sa uloží do TensorBoard. Po skončení tréningovania sa uložia potrebné váhy.



Obr. 6.1: Pokročilé logovanie pomocou TensorBoard. Ide o kartu zobrazujúcu hodnoty chybových funkcií a metrik. Vľavo je možné jednotlivé behy filtrovať pomocou regulárnych výrazov. Na grafoch s dvoma rôznymi farbami čiar ide o výsledok na tréningovej časti oproti validačnej časti dátovej sady v tomto prípade. Tiež je možné porovnávať rôzne behy medzi sebou.

## Kapitola 7

# Experimenty

Táto kapitola sa venuje vykonaným experimentom. Vzhľadom na to, že boli implementované dva rôzne spôsoby načítania dát, kde sa v oboch prípadoch vytvára testovacia časť dát za behu, bolo potrebné vytvoriť testovaciu dátovú sadu pre porovnanie výsledkov. Prvý experiment obsahuje porovnanie dvoch architektúr typu U-Net, kde boli obidve tréované na jednoduchej verzii načítania dát, preto aj vyhodnotenie prebehlo na testovacej časti vytvorenej za behu. V ostatných experimentoch boli pri tréovaní použité oba spôsoby načítania dát, preto bola vytvorená testovacia dátová sada, ktorá obsahuje celkom 142 ručne vybraných obrázkov zo 6 rôznych kategórií snímok pochádzajúcich z elektrónového mikroskopu. Ide o kategórie:

- biologické vzorky (26 snímok),
- vlákna (13 snímok),
- nanodrôty (30 snímok),
- častice (31 snímok),
- pórovité povrchy (13 snímok),
- mikro-elektromechanické zariadenia a elektródy (29 snímok).

Siete boli tréované len na kategórii mikro-elektromechanické zariadenia a elektródy. Testovacia dátová sada obsahuje aj preto snímky z iných kategórií, kde je často charakter obrázkov odlišný. Týmto spôsobom je možné zistiť, či sú siete schopné generalizovať aj na dátach, ktoré predtým nevideli. Obrázky v tejto sade nemajú pomer strán 1:1 a pre jednoduché zobrazenie je z nich vybraný výrez v tomto pomere.

Čo sa týka augmentácii, tak pri všetkých experimentoch boli robené za behu základné obrázkové augmentácie s nasledovným nastavením:

- náhodná zmena jasů v rozsahu  $\langle -0.2; 0.2 \rangle$ ,
- náhodná zmena kontrastu v rozsahu  $\langle 0; 0.3 \rangle$ ,
- otočenie vo horizontálnom smere s pravdepodobnosťou 50%,
- otočenie vo vertikálnom smere s pravdepodobnosťou 50%,

pre obidva spôsoby načítania dát.

## 7.1 U-Net a RUNet pri zvýšení rozlišení dvakrát

V tomto experimente boli porovnané architektúry U-Net a RUNet, kde boli tréované pomocou rozličných chybových funkcií a mali za úlohu zvýšiť rozlišenie obrázku dvakrát. Bola pri tom použitá základná dátová sada bez pokročilej degradácie.

Pri tréovaní boli použité dve chybové funkcie. Prvou je chybová funkcia  $\mathcal{L}_{met}$  kombinujúca metriky PSNR a SSIM popísaná v rovnici 5.1. Druhá je príznaková chybová funkcia v kombinácii s L1 vzdialenosťou medzi obrázkami  $\mathcal{L}_{perc}$  popísaná v rovnici 5.3.

### Nastavenie parametrov

Pri chybovej funkcii  $\mathcal{L}_{met}$  bol nastavený parameter  $\alpha = 1$  a  $\beta = 10$  aby sa váhované hodnoty metrík pohybovali v podobnom rozsahu. Parameter  $\eta$  pri chybovej funkcii  $\mathcal{L}_{perc}$  bol nastavený na  $\eta = 5$  pri sieti U-Net a  $\eta = 1$  pri sieti RUNet. Sieť U-Net produkovala pri nižšej hodnote  $\eta$  nežiadúce artefakty, preto bolo nutné zvýšiť váhu L1 vzdialenosti medzi obrázkami. V oboch prípadoch bola veľkosť jednej dávky (angl. *batch size*) nastavená na 16. V oboch prípadoch boli siete predtrénované na 2 iteráciách s parametrom  $\eta = 150$  aby boli schopné produkovať zmysluplné obrázky pri následnom použití príznakovkej chybovej funkcie. Celkovo prebehlo 15 iterácií tréovacieho cyklu. Bola použitá metóda gradientného zostupu Adam. Koeficient učenia bol nastavený na hodnotu  $10^{-4}$  a znižoval sa každou iteráciou koeficientom 0.95.

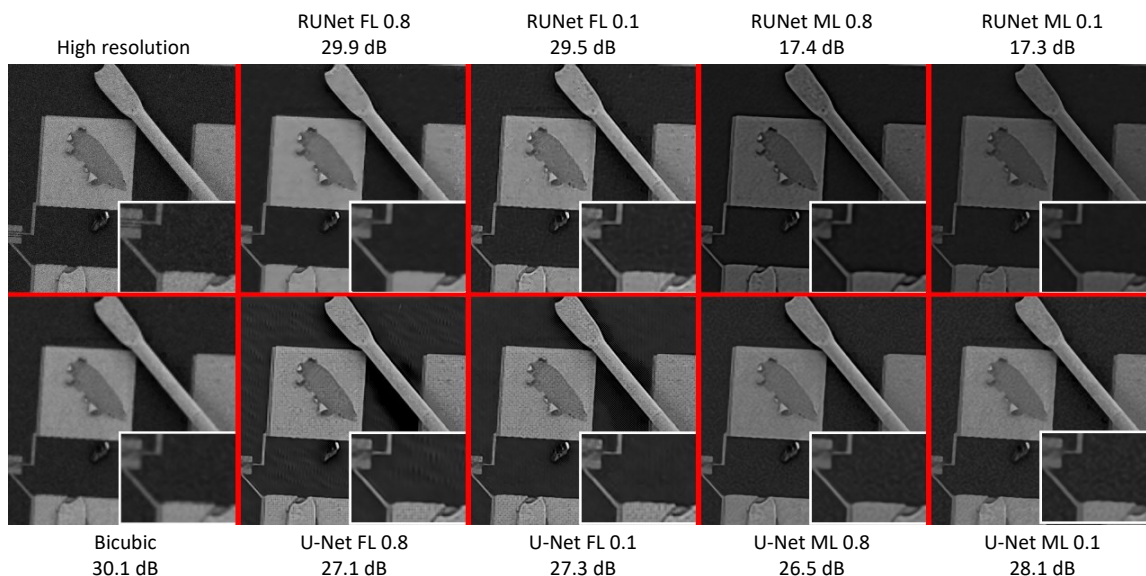
Obrázky pochádzali z kategórie mikro-elektromechanické zariadenia a elektródy. Do obrázkov bol pridávaný umelý poissonov šum v dvoch pomeroch. V prvom prípade bol šum umelo pridaný do 10% tréovacej dátovej sady a v druhom prípade do 80% tréovacej dátovej sady. Vstupné obrázky mali rozlišenie  $128 \times 128$  px a boli umelo zväčšované dvakrát na rozlišenie  $256 \times 256$  px.

### Výsledky

Vyhodnotenie pomocou metrík PSNR a SSIM na testovacej časti dátovej sady je možné vidieť v tabuľke 7.1. Celkovo sa natréovaným sieťam nepodarilo prekonať hodnoty metrík pri zvýšení rozlišenia pomocou bikubickej interpolácie. Tento výsledok bol očakávaný, keďže tieto metriky úplne neodpovedajú tomu, ako vníma kvalitu zvýšenia rozlišenia ľudské oko.

	PSNR	SSIM
U-Net ML 0.1	27.66 dB	<b>0.83</b>
U-Net ML 0.8	26.00 dB	<b>0.83</b>
U-Net FL 0.1	25.60 dB	0.69
U-Net FL 0.8	25.59 dB	0.71
RUNet ML 0.1	15.67 dB	0.71
RUNet ML 0.8	15.66 dB	0.72
RUNet FL 0.1	29.59 dB	0.81
RUNet FL 0.8	30.82 dB	0.80
Bicubic	<b>31.38 dB</b>	0.82

Tabuľka 7.1: Porovnanie výsledkov pre rôzne kombinácie architektúry (U-Net, RUNet), chybovej funkcie („FL“ - príznaková, „ML“ - kombinujúca metriky) a pravdepodobnosti pridania umelého šumu do tréovacej časti dátovej sady (0.1 a 0.8).



Obr. 7.1: Porovnanie predikcií sietí U-Net a RUNet s rôznou kombináciou chybových funkcií a rôznym množstvom pridania umelého šumu do trénovacej dátovej sady. Obrázky v prvom stĺpci odpovedajú pôvodnému obrázku vo vysokom rozlíšení (hore) a obrázku zväčšenému s pomocou bikubickej interpolácie (dole). Ostatné obrázky sú predikcie sietí, kedy je v popise pri danom obrázku odpovedajúca architektúra a nastavenie. Popis je v rovnakom tvare ako v tabuľke 7.1. Pod popisom sa nachádza hodnota metriky PSNR pre porovnanie. Obrázok je vhodné si priblížiť.

Sieť U-Net mala pri oboch chybových funkciách a pri oboch množstvách šumu hodnoty podobné. Predikcie na testovacej snímke je možné vidieť v spodnom riadku na obrázku 7.1. Druhý a tretí obrázok odpovedajú sieti U-Net s príznakovou chybovou funkciou. Predikcie pôsobia o niečo ostrejšie ako pri bikubickej interpolácii (prvý obrázok v spodnom riadku). V pozadí však v oboch prípadoch vznikajú artefakty vo forme vertikálnych čiar. Tiež sa zväčšila tmavá škvrna v strede obrázku. Aj pri pridání 80% šumu do trénovacích obrázkov sieť U-Net nijak výrazne šum neodstránila. V prípade použitia chybovej funkcie kombinujúcej metriky sú obrázky podobné bikubickej interpolácii, ale pôsobia o niečo ostrejšie. Nevznikli žiadne artefakty, ale so šumom si v tomto prípade sieť tiež neporadila.

Predikcie siete RUNet sa dosť líšia pri použití rôznych chybových funkcií. Pri použití chybovej funkcie kombinujúcej metriky sa vo všetkých obrázkoch menila intenzita jasu ako je možné vidieť v posledných dvoch obrázkoch v hornom riadku na obrázku 7.1. Hodnoty metrick PSNR a SSIM sú oveľa nižšie práve kvôli tomuto javu. Ako bolo spomínané (v časti 5.2) bola odstránená prvá batch-normalizačná vrstva. Pred odstránením tejto vrstvy bola zmena jasu ešte výraznejšia. Okrem zmeny jasu pôsobí obrázok o niečo ostrejšie. V rámci práce s touto sieťou boli vykonané experimenty, kedy boli zo siete odstránené aj iné kombinácie batch-normalizačných vrstiev. Problém sa podarilo najviac redukovať práve odstránením spomínanej prvej batch-normalizačnej vrstvy.

Pri použití príznakovej chybovej funkcie (druhý a tretí obrázok v hornom riadku) intenzita jasu ostala podobná pôvodnému obrázku. Pri trénovaní s pridáním 80% šumu do trénovacích obrázkov (druhý obrázok v hornom riadku) sa podarilo šum redukovať. Nevýhodou

však je, že sa pri redukovaní šumu v obrázku strácajú aj rôzne drobné detaily, ktoré sieť vyhodnotí ako šum.

Charakter zmien v predikciách na iných testovacích obrázkoch odpovedá charakteru zmien vo vybranom obrázku. Kvalitu zvýšenia rozlíšenia obrazu necháva autor na čitateľa. Pre ďalšie experimenty bola vybraná sieť RUNet z dôvodu, že podľa názoru autora vytvárala kvalitnejšie predikcie.

## 7.2 RUNet a Real-ESRGAN pri zvýšení rozlíšenia dvakrát

Úlohou tohto experimentu bolo porovnať sieť RUNet z predchádzajúceho experimentu, kde bola trévaná pomocou príznakovej chybovej funkcie  $\mathcal{L}_{perc}$ . Výsledky tejto siete vyzerali kvalitnejšie ako pri sieti U-Net. V tomto experimente boli jej výsledky porovnané s výsledkami architektúry Real-ESRGAN.

Sieť Real-ESRGAN bola trévaná na dátovej sade s pokročilou degradáciou popísanou v časti 6. Sieť RUNet bola taktiež najskôr natrévaná na tejto dátovej sade, ale výsledky boli horšie ako v prípade, kedy bola sieť trévaná na základnej dátovej sade bez pokročilej degradácie tréovacích obrázkov. Sieť RUNet je preto v tomto experimente trévaná, rovnako ako v predchádzajúcom experimente, na dátovej sade bez pokročilej degradácie.

### Nastavenie parametrov RUNet

Nastavenie parametrov je uvedené ešte raz, pre jednoduchšie odlíšenie od parametrov siete U-Net. Jedná sa o sieť RUNet z predchádzajúceho experimentu, trévanú na príznakovej chybovej funkcii  $\mathcal{L}_{perc}$  s nastavením parametra  $\eta = 150$  na 2 iterácie pri predtrévaní. Potom je parameter nastavený na  $\eta = 1$ , kedy L1 vzdialenosť predstavuje zhruba 5% celkovej chyby. Počet iterácií a metóda učenia s koeficientom učenia a veľkosť dávky je rovnaká ako vo všetkých prípadoch z predchádzajúceho experimentu.

Obrázky mali rovnaké rozlíšenie ako pri predchádzajúcom experimente, teda boli zväčšované z rozlíšenia  $128 \times 128$  px na rozlíšenie  $256 \times 256$  px. Šum bol pridávaný v 10% a 80% tréovacej dátovej sady.

### Nastavenie parametrov Real-ESRGAN

V celkovej chybovej funkcii generátora (rovnica 4.10) boli nastavené parametre  $\eta = 1$  a  $\lambda = 0.1$ . To zaručilo, že L1 vzdialenosť v príznakovej chybovej funkcii odpovedá zhruba 5% celkovej chyby a že *adversarial* časť chybovej funkcie generátora  $L_G^{Ra}$  odpovedá asi 30% celkovej chyby. Sieť generátora aj diskriminátora využívali rovnakú metódu gradientného zostupu s rovnakým nastavením koeficientu učenia ako sieť RUNet. Generátor bol predtrévaný na 30 iteráciách tréovacieho cyklu s chybovou funkciou, ktorá odpovedá L1 vzdialenosti medzi obrázkami. Následne bol na ďalších 20 iterácií pridaný aj diskriminátor a chybová funkcia bola nahradená plnou chybovou funkciou generátora z rovnice 4.10.

Obrázky v tréovacej dátovej sade mali rovnaké rozlíšenie a mieru šumu ako pri sieti RUNet. Ich kvalita bola však znížená dvojúrovňovým degradačným procesom, ktorého parametre sa nachádzajú nižšie. Tréovacie obrázky v pochádzali len z kategórie mikroelektromechanické zariadenia a elektródy ako v predchádzajúcom experimente.

## Nastavenie parametrov degradačného procesu dátovej sady

Sieť Real-ESRGAN bola trénovaná na tréningových dátach s dvojúrovňovým degradačným procesom (popísaným v časti 5.1) s nasledujúcimi parametrami:

- rozostrenie gaussovským filtrom - veľkosť hrany štvorcového filtra bola pri oboch úrovniach náhodne vybraná spomedzi hodnôt: 7, 9, 11, 13, 15, 17.
- Náhodná zmena veľkosti - metóda bola náhodne vybraná spomedzi hodnôt: *náhodný sused*, *bilinéárna* a *bikubická*. Obrázok buď bol náhodne zväčšený s pravdepodobnosťou 20% v mierke z intervalu  $\langle 1; 1.5 \rangle$ , alebo bol náhodne zmenšený s pravdepodobnosťou 70% v mierke z intervalu  $\langle 0.15; 1 \rangle$ , alebo bola ponechaná pôvodná veľkosť s pravdepodobnosťou 10%. V druhej úrovni boli pravdepodobnosti 30%, 40% a 30% a intervaly  $\langle 1; 1.2 \rangle$  a  $\langle 0.3; 1 \rangle$  v rovnakom poradí.
- Pridanie poissonovho šumu - pravdepodobnosť pridania šumu bola 0.1 alebo 0.8 v oboch úrovniach podľa experimentu a miera šumu bola upravená vynásobením s číslom z intervalu  $(0.05; 2.0)$  v prvej úrovni a  $\langle 0.05; 1.5 \rangle$  v druhej úrovni.
- JPEG kompresia - kvalita JPEG kompresie zmenená náhodnou hodnotou z intervalu  $\langle 70; 95 \rangle$  v oboch úrovniach.

## Výsledky

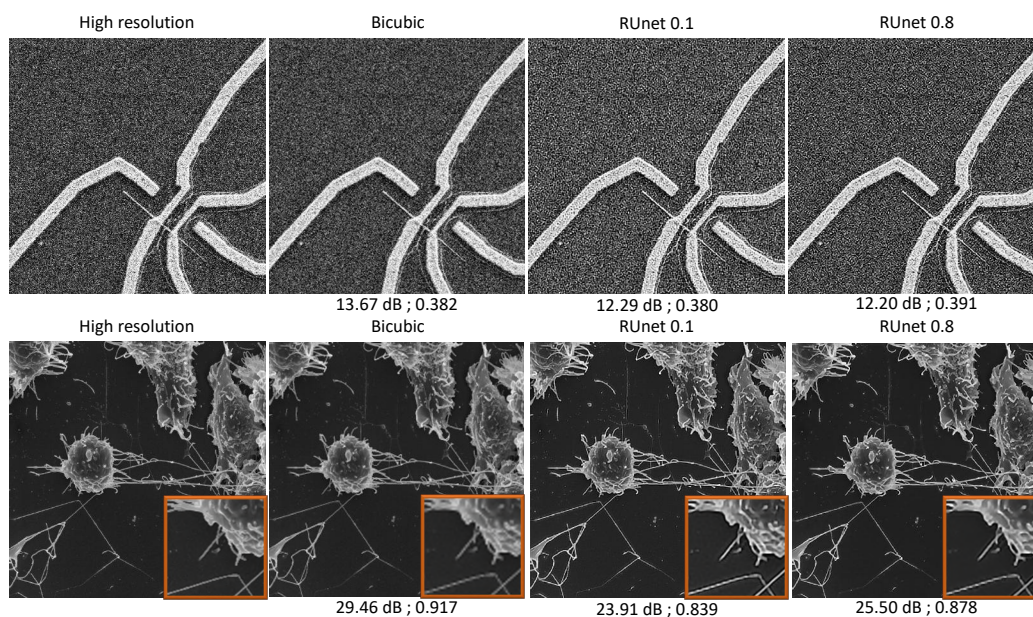
Siete sú vyhodnotené podľa metrík PSNR a SSIM v tabuľke 7.2. Tentokrát ide už o obrázky z testovacej dátovej sady skladajúcej sa zo 6 kategórií. Tabuľka obsahuje hodnoty metrík pre kategóriu mikro-elektromechanické zariadenia a elektródy ( $PSNR_m$  a  $SSIM_m$ ), na ktorej boli siete trénované a priemernú hodnotu metrík pre ostatné kategórie ( $PSNR_o$  a  $SSIM_o$ ). Pri zvýšení rozlíšenia dvakrát nie sú zmeny v obrázkoch veľmi viditeľné, ale pri priblížení sú už výraznejšie.

	$PSNR_m$	$PSNR_o$	$SSIM_m$	$SSIM_o$
RUNet 0.1	23.35 dB	25.02 dB	0.548	0.696
RUNet 0.8	24.30 dB	26.20 dB	0.574	0.725
R-ESRGAN 0.1	21.36 dB	22.93 dB	0.367	0.510
R-ESRGAN 0.8	23.42 dB	24.90 dB	0.435	0.600
Bicubic	<b>25.83 dB</b>	<b>28.78 dB</b>	<b>0.611</b>	<b>0.766</b>

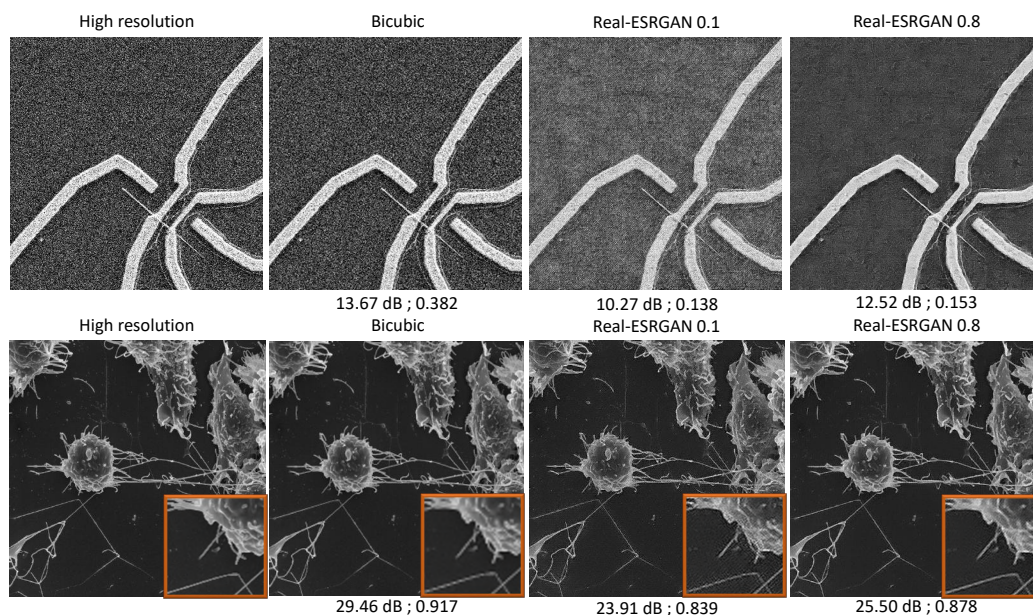
Tabuľka 7.2: Porovnanie metrík PSNR a SSIM na testovacej dátov sade pri zvýšení rozlíšenia dvakrát. Za názvom architektúry sa nachádza pravdepodobnosť generovania šumu v tréningovej sade. Nachádzajú sa tu aj hodnoty pre bikubickú interpoláciu. Dolný index  $m$  znamená že ide o kategóriu snímok na ktorej bola sieť aj trénovaná. Dolný index  $o$  znamená, že ide o priemer ostatných kategórií.

Obrázky vygenerované sieťou RUNet sa nachádzajú na obrázku 7.2. Obrázok 7.1 z prvého experimentu obsahoval obrázok, ktorý bol zväčšený z rozlíšenia  $128 \times 128$  px na  $256 \times 256$  px. Obrázky v testovacej dátov sade so 6 kategóriami obsahujú obrázky v rozlíšení  $288 \times 512$  px a zväčšujú sa na  $576 \times 1024$  px. Pri väčších obrázkoch si sieť RUNet, trénovaná s 80% pravdepodobnosťou šumu v tréningovej sade, nevedela so šumom poradiť. Generované obrázky majú zvýraznené hrany, zároveň sa zvýrazňuje aj šum (ako napríklad v riadku 1 na

obrázku 7.2). Hrany v predikciách pôsobia výraznejšie ako pri bikubickej interpolácii. Pri priblížení (viď. obrázok 7.4) sú však hrany podobne kostrbaté ako pri bikubickej interpolácii.

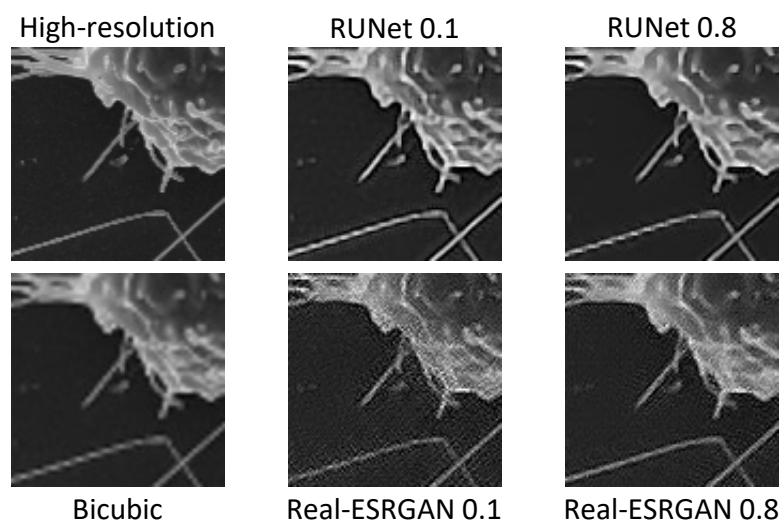


Obr. 7.2: Predikcie architektúry RUNet s 10% a 80% šumu v trénovacej dátovej sade pri zvýšení rozlíšenia dvakrát. Pod obrázkami sa nachádzajú hodnoty PSNR a SSIM pre porovnanie. Prvý obrázok pochádza z trénovacej kategórie, druhý pochádza z kategórie biologických dát. V druhom obrázku je v pravom dolnom rohu obrázku priblížený detail pre lepšiu viditeľnosť. Obrázky je vhodné si priblížiť.



Obr. 7.3: Predikcie rovnakých obrázkov architektúry Real-ESRGAN pri zvýšení rozlíšenia dvakrát. Popisy v obrázku majú identický význam ako v obrázku 7.2.

Obrázky zväčšené pomocou siete Real-ESRGAN sa nachádzajú na obrázku 7.3. Sieť si v oboch prípadoch poradila so šumom, viac však v prípade kedy bola trénovaná s 80% pravdepodobnosťou šumu v trénovacích dátach. V priblížených detailoch je vidno, že hrany sú menej kostrbaté ako pri RUNet. V prípade kedy bola sieť trénovaná s 10% šumu vznikajú okolo hrán nechcené šumové artefakty, ktoré je taktiež vidno na priblížených detailoch v obrázku 7.4. Pri zvýšení rozlíšenia dvakrát sa tieto artefakty nachádzajú vo veľkej časti testovacích obrázkov. Môže to byť spôsobené tým, že táto architektúra je vytvorená pre zvýšenie rozlíšenia štyrikrát. Pri zvýšení rozlíšenia dvakrát sa pri vstupe výška a šírka obrázkov zmenší a zväčší sa počet kanálov. Táto modifikácia je popísaná na obrázku 4.11 a môže byť dôvodom týchto artefaktov. Pri zvýšení rozlíšenia štyrikrát alebo pri trénovaní s väčšou pravdepodobnosťou šumu v trénovacích obrázkoch je tento nechcený jav menej viditeľný.



Obr. 7.4: Porovnanie zväčšených výrezov sietí RUNet a Real-ESRGAN pri zvýšení rozlíšenia dvakrát. Sieť RUNet lepšie zachováva detaily a zvýrazňuje hrany. Pri Real-ESRGAN v oblasti okolo hrán vzniká šum, čím hrany pôsobia menej kostrbaté. Táto sieť si však vie lepšie poradiť s prirodzeným šumom v obrázkoch ako je vidieť v hornom riadku na obrázku 7.3.



### 7.3 RUNet a Real-ESRGAN pri zvýšení rozlíšenia štyrikrát

Pri zvýšení rozlíšenia štyrikrát ide o náročnejšiu úlohu. Aj v pôvodnom článku Real-ESRGAN [30] riešili práve túto úlohu. V tomto experimente sú porovnané architektúru RUNet a Real-ESRGAN. Architektúra RUNet bola trénovaná pomocou príznakovej chybovej funkcie. Ako v predchádzajúcich prípadoch je v experimente menená pravdepodobnosť pridávania šumu do trénovacej časti dátovej sady. Pravdepodobnosti majú v tomto experimente hodnoty 30% a 80%. Nižšia pravdepodobnosť šumu 10% bola teda zvýšená na 30%, pretože sieť Real-ESRGAN produkovala pri hranách nechcené artefakty ako je možné vidieť na obrázku 7.4. Silu týchto artefaktov sa podarilo redukovať práve týmto zvýšením pravdepodobnosti pridaného šumu. Trénovacie obrázky v pochádzali len z kategórie mikroelektromechanické zariadenia a elektródy ako v predchádzajúcich prípadoch.

#### Nastavenie parametrov siete RUNet

Nastavenie parametrov siete RUNet bolo rovnaké ako pri predchádzajúcom experimente, zmenila sa len veľkosť obrázkov a miera ich zväčšenia.

Pôvodne bola pre tento experiment sieť RUNet trénovaná na zvýšení rozlíšenia z  $128 \times 128$  px na  $512 \times 512$  px. V tomto prípade však produkovala obrázky takmer identické s bikubicou interpoláciou pri oboch nastaveniach pravdepodobnosti šumu. Preto bola sieť nakoniec natrénovaná na zvýšení rozlíšenia z  $64 \times 64$  px na  $256 \times 256$  px.

#### Nastavenie parametrov siete Real-ESRGAN

Parametre tejto siete sú rovnaké ako v predchádzajúcom experimente. Rovnako sa zmenila len veľkosť obrázkov a pravdepodobnosť pridávania šumu.

V tomto prípade bola sieť trénovaná na zvýšení rozlíšenia z  $128 \times 128$  px na  $512 \times 512$  px aj pri predtrénovaní generátora a aj pri trénovaní generátora s diskriminátorom.

#### Nastavenie parametrov degradačného procesu dátovej sady

Degradačný proces trénovacích obrázkov mal nastavené rovnaké parametre ako pri predchádzajúcom experimente. Jediná zmena bola v pravdepodobnosti pridávania umelého poissonovho šumu do trénovacích obrázkov, kedy namiesto 10% v predchádzajúcom experimente to bolo v tomto prípade 30%.

#### Výsledky

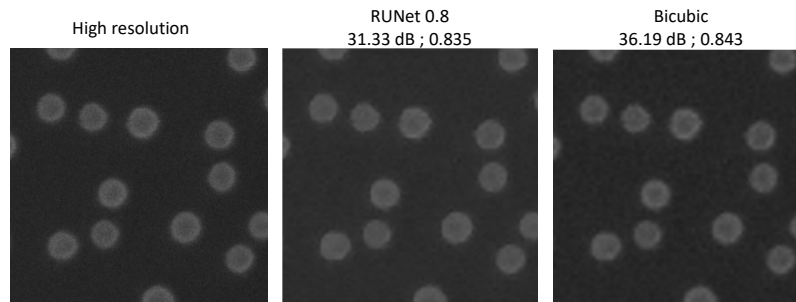
Porovnanie hodnôt metrík PSNR a SSIM na testovacej dátovej sade sa nachádza v tabuľke 7.3. Tabuľka má rovnaký formát ako pri predchádzajúcom experimente. Ani jednej sieti sa znova nepodarilo prekonať bikubicú interpoláciu v ohľade metrík.

Niekoľko predikcií siete RUNet sa nachádza na obrázku 7.6. Ak bol pridaný šum v 30% trénovacej dátovej sade sieť sa pokúša zvýrazniť hrany aby pôsobili ostrejšie. Hrany sú však stále kostrbaté a výsledok nepôsobí oproti bikubickej interpolácii ako zlepšenie. Pri 80% šume v trénovacej dátovej sade sieť nezvýrazňuje hrany ako v predchádzajúcom prípade. Hrany sú však rovnako kostrbaté ako pri bikubickej interpolácii. Jediným rozdielom oproti bikubickej interpolácii je mierna redukcia šumu. Hodnoty metrík PSNR a SSIM sú lepšie ako v prípade siete Real-ESRGAN. Predikcie siete RUNet však nie sú podľa autora

	$PSNR_m$	$PSNR_o$	$SSIM_m$	$SSIM_o$
RUNet 0.3	21.29 dB	22.89 dB	0.402	0.587
RUNet 0.8	22.41 dB	24.09 dB	0.403	0.585
R-ESRGAN 0.3	21.69 dB	23.36 dB	0.338	0.502
R-ESRGAN 0.8	21.33 dB	22.19 dB	0.321	0.425
Bicubic	<b>22.90 dB</b>	<b>24.97 dB</b>	<b>0.418</b>	<b>0.611</b>

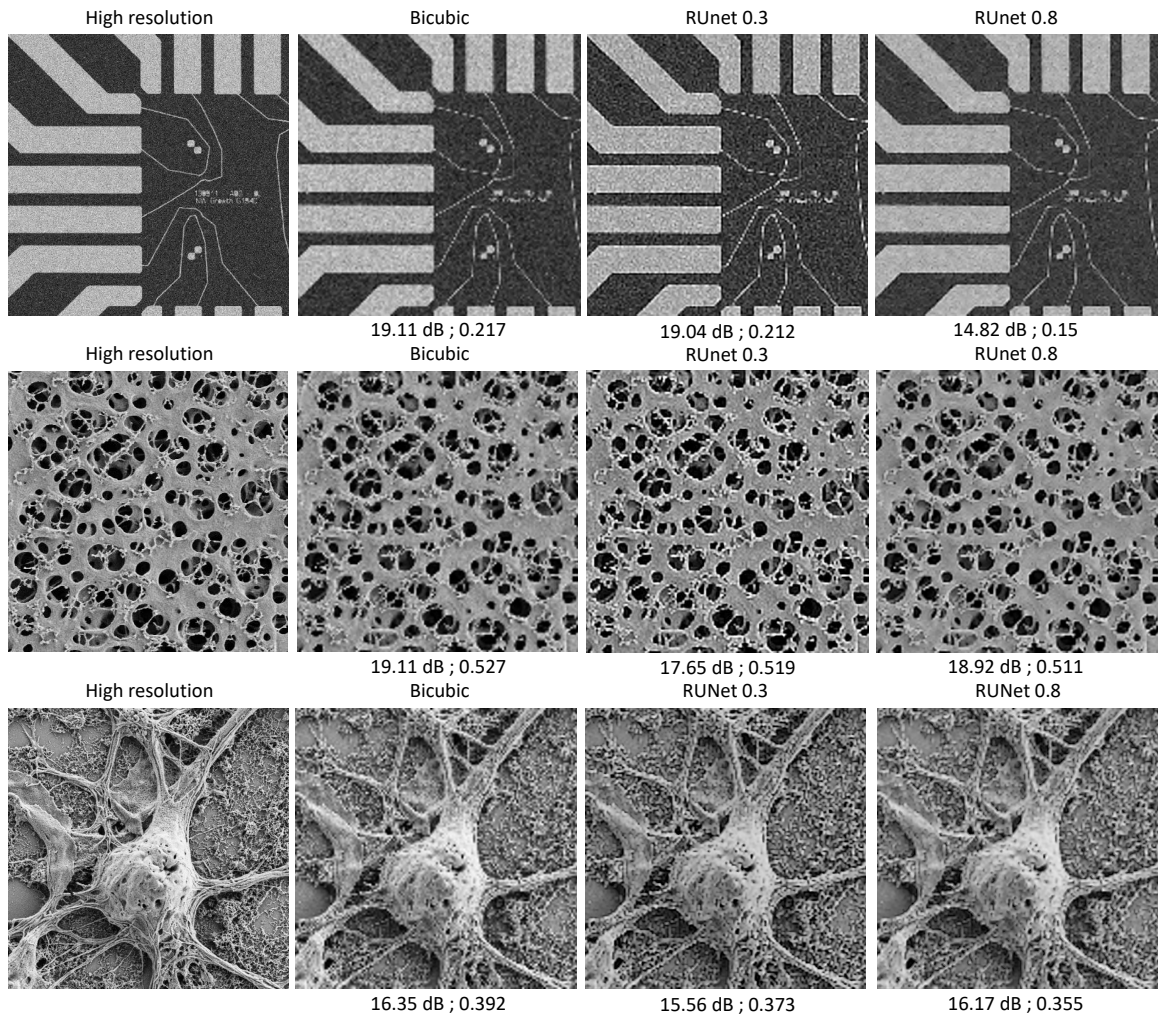
Tabuľka 7.3: Porovnanie metrík PSNR a SSIM na testovacej dátovej sade pri zvýšení rozlíšenia štyrikrát. Význam indexov je rovnaký ako v tabuľke 7.2 v predchádzajúcom experimente.

lepšie ako v prípade Real-ESRGAN a viac sa podobajú výsledkom bikubickej interpolácie. Veľké zlepšenie sa však pozorovať nedá. Malé zlepšenie voči bikubickej interpolácii pri sieti RUNet je možné sledovať ak má vstupný obrázok malé rozlíšenie (rádovo okolo  $100 \times 100$  px). Tento jav je možné vidieť na obrázku 7.5.

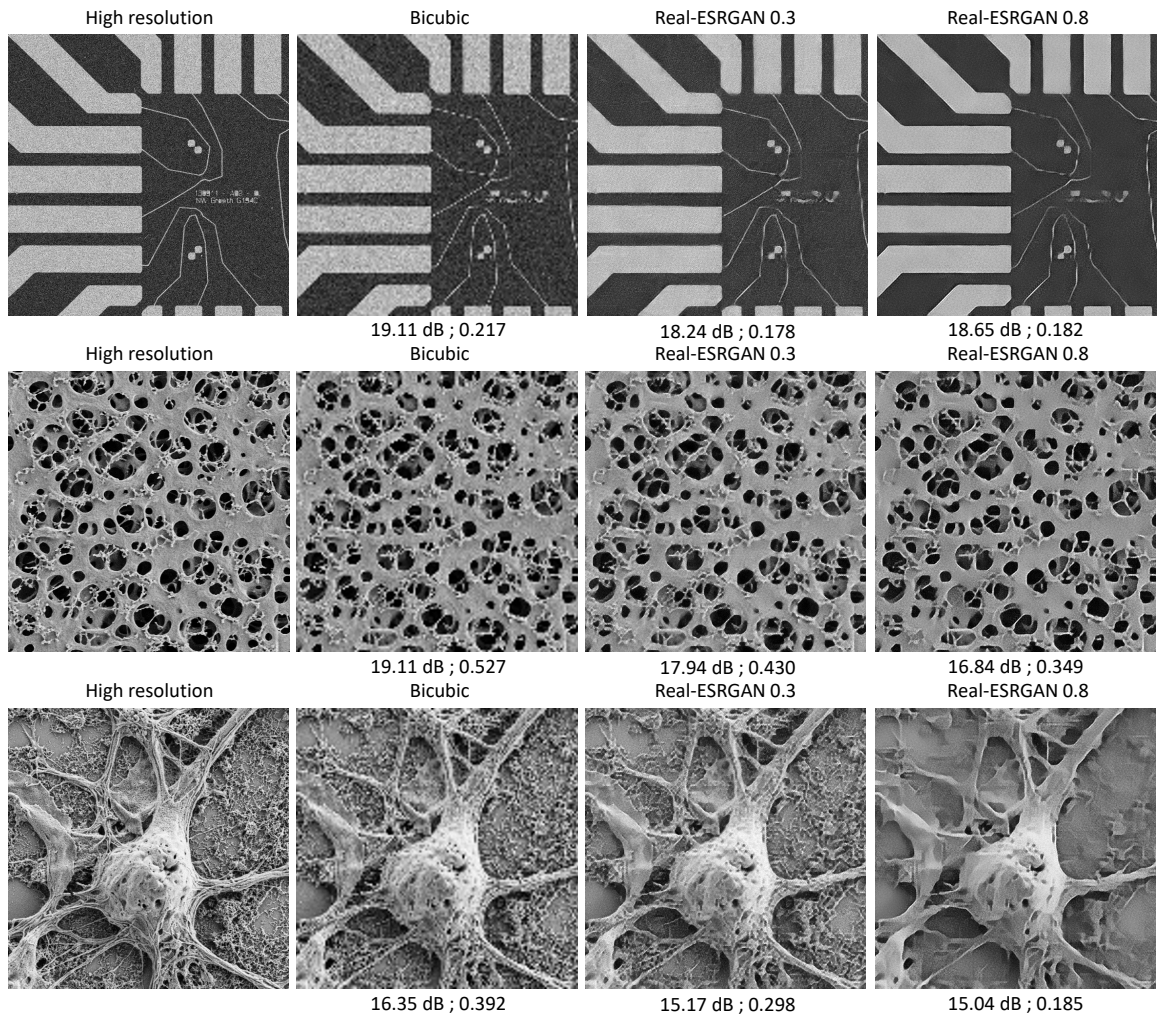


Obr. 7.5: Zlepšenie kvality voči bikubickej interpolácii na malom obrázku pomocou siete RUNet. Obrázok mal nízke rozlíšenie  $72 \times 72$  px a bol zväčšený 4-krát. Obrázok je z kategórie častice. Obrázok je vhodné si priblížiť.

Predikcie siete Real-ESRGAN sa nachádzajú na obrázku 7.7. Napriek nižším hodnotám metrík PSNR a SSIM pôsobia vizuálne atraktívnejšie ako obrázky získané pomocou bikubickej interpolácie alebo siete RUNet (podľa subjektívneho názoru autora). Pri prirodzene zašumených obrázkoch je sieť trénovaná na 80% zašumených obrázkoch schopná väčšiu časť šumu odstrániť. Nevýhodou však je, že drobné detaily vyhodnotí tiež ako šum a odstráni ich ako napríklad na obrázku 7.7 v prvom riadku. Pri pridávaní šumu do 30% trénovacej sady sieť odstráni šum v rozumnej miere a obrázky pôsobia omnoho ostrejšie ako bikubicná interpolácia.



Obr. 7.6: Výsledky sietí RUNet pri zvýšení rozlíšenia štyrikrát. Pod obrázkom sa nachádzajú metriky PSNR a SSIM. Prvý riadok obsahuje obrázky z rovnakej kategórie aká bola použitá pri tréňovaní. Druhý riadok je z kategórie pórovité povrchy a tretí je z kategórie biologických obrázkov. Obrázky je vhodné si priblížiť pre lepšiu viditeľnosť.



Obr. 7.7: Výsledky siete Real-ESRGAN pri zvýšení rozlíšenia štyrikrát na rovnakých Obrázky je vhodné si priblížiť pre lepšiu viditeľnosť.

## 7.4 Real-ESRGAN pri dvoch rôznych veľkostiach tréningových obrázkov

V tomto experimente je porovnaný dopad veľkosti tréningových obrázkov na finálnu predikciu pri rôznych rozlíšeniach. Je tu porovnaná sieť Real-ESRGAN z minulého experimentu, kedy bol do 30% tréningovej časti dátovej sady pridaný umelý poissonov šum. Sieť bola tréningovaná na obrázkoch s rozlíšením  $128 \times 128$  px a zväčšovala ich na  $512 \times 512$  px. Druhá sieť v tomto experimente je rovnaká, ale bola natréningovaná na obrázkoch s rozlíšením dvakrát menším, t.j.  $64 \times 64$  px a zväčšovala ich na  $256 \times 256$  px.

Nastavenie hodnôt parametrov je rovnaké ako v predchádzajúcom experimente. Šum sa pridáva len do 30% tréningovej dátovej sady a rozdiel je vo veľkosti tréningových obrázkov. Parametre degradačného procesu ostávajú tiež rovnaké.

## Výsledky

Vyhodnotenie prebehlo na testovacej dátovej sade, kde v jednom prípade siete umelo zvyšovali rozlíšenie obrázkov s rozlíšením  $72 \times 128$  px na rozlíšenie  $288 \times 512$  px a v druhom prípade z  $144 \times 256$  px na  $576 \times 1024$  px. Výsledné hodnoty metrík sú v tabuľkách 7.4 a 7.5. Metriky majú horný index *small* pre menšie rozlíšenie a *big* pre vyššie rozlíšenie. Rovnako sú v tomto experimente označené aj siete, horný index *small* značí, že sieť bola trébovaná na menších obrázkoch, index *big* znamená väčšie trébovacie obrázky.

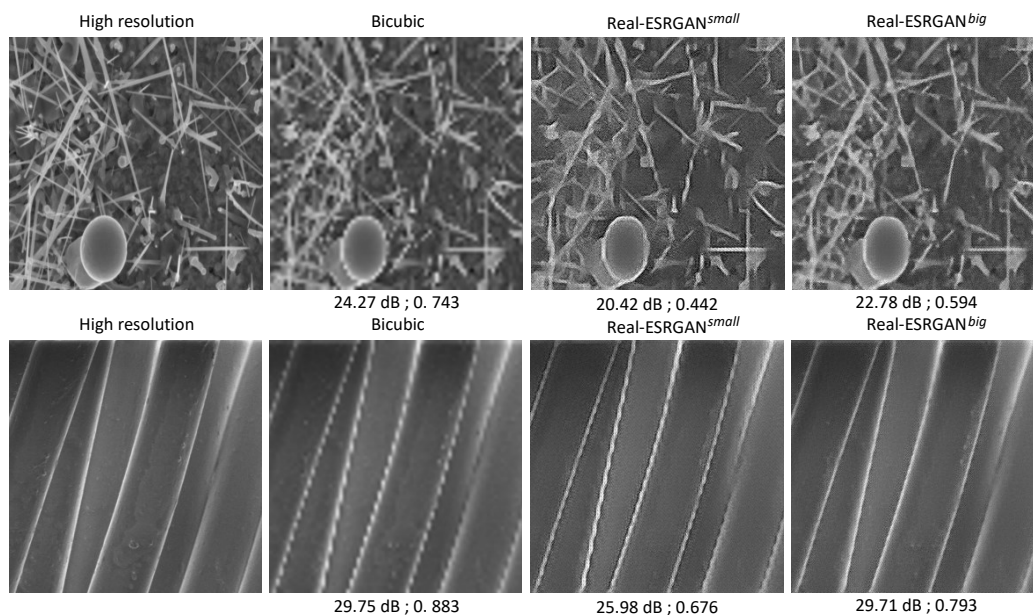
	$PSNR_m^{big}$	$PSNR_m^{small}$	$PSNR_o^{big}$	$PSNR_o^{small}$
R-ESRGAN <sup>big</sup>	21.69 dB	23.14 dB	23.08 dB	23.81 dB
R-ESRGAN <sup>small</sup>	20.71 dB	21.60 dB	21.42 dB	21.09 dB
Bicubic	<b>22.90 dB</b>	<b>25.15 dB</b>	<b>24.63 dB</b>	<b>24.67 dB</b>

Tabuľka 7.4: Hodnoty metriky PSNR. Horné indexy *small* a *big* znamenajú buď menšie alebo väčšie obrázky. Dolný index *m* a *o* má pri metrikách rovnaký význam ako v tabuľke 7.2.

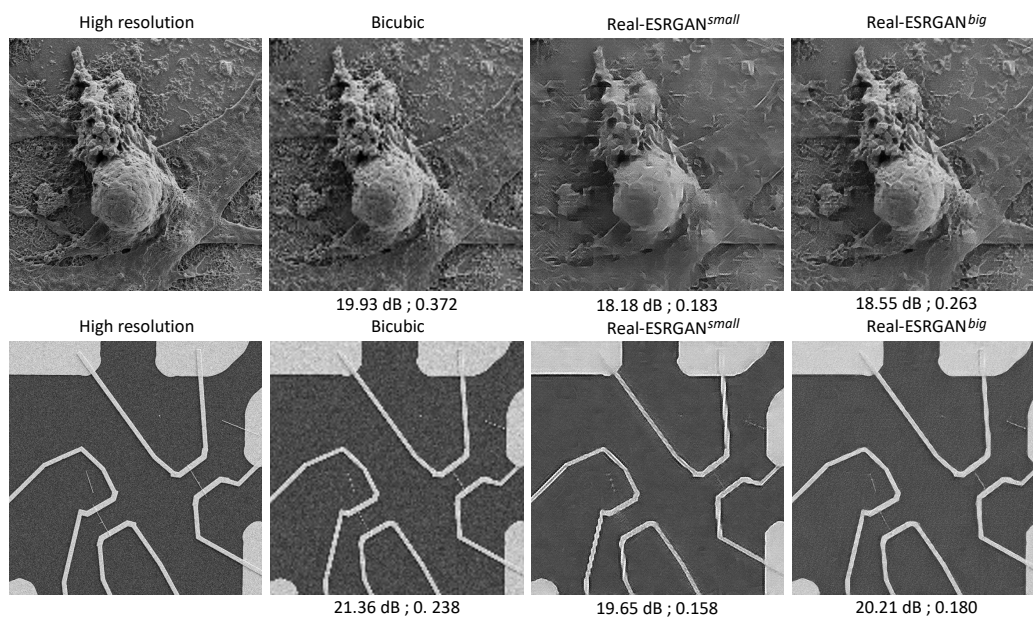
	$SSIM_m^{big}$	$SSIM_m^{small}$	$SSIM_o^{big}$	$SSIM_o^{small}$
R-ESRGAN <sup>big</sup>	0.338	0.511	0.475	0.535
R-ESRGAN <sup>small</sup>	0.290	0.427	0.384	0.422
Bicubic	<b>0.418</b>	<b>0.624</b>	<b>0.579</b>	<b>0.659</b>

Tabuľka 7.5: Hodnoty metriky SSIM. Význam symbolov je popísaný v predchádzajúcej tabuľke 7.4.

Znova žiadna sieť neprekonalala hodnoty metrík bikubickej interpolácie. Hodnoty metrík sú však vyššie pri použití väčších trébovacích obrázkov, to aj v prípade menších, aj v prípade väčších testovacích obrázkov. Aj v ohlade predikcií je sieť trébovaná na väčších obrázkoch lepšia. Hrany v obrázkoch sú menej kostrbaté a obrázok pôsobí ostrejšie. Sieť trébovaná na menších obrázkoch generuje v okolí hrán viditeľnejšie artefakty v podobe šumu. Predikcie menších testovacích snímok sú na obrázku 7.8. Predikcie väčších snímok sa nachádzajú na obrázku 7.9. Sieť trébovaná na väčších obrázkoch je schopná lepšie zvyšovať rozlíšenie aj pri menších, aj pri väčších obrázkoch. Pri menších obrázkoch je viac viditeľné aj zlepšenie voči bikubickej interpolácii.



Obr. 7.8: Predikcie sietí na menších testovacích obrázkoch. Obrázky boli zväčšené z rozlíšenia  $72 \times 128$  px na rozlíšenie  $288 \times 512$  px. V obrázkoch sa nachádzajú výrezy s veľkosťou  $288 \times 288$  px. Horný index pri názve siete hovorí o tom, či bola sieť trébovaná na menších, alebo na väčších trébovacích obrázkoch.



Obr. 7.9: Predikcie sietí na väčších testovacích obrázkoch. Obrázky boli zväčšené z rozlíšenia  $144 \times 256$  px na rozlíšenie  $576 \times 1024$  px. V obrázkoch sa nachádzajú výrezy s veľkosťou  $288 \times 288$  px. Horný indexy pri názve siete má rovnaký význam ako v predchádzajúcom obrázku 7.8.

## 7.5 Real-ESRGAN na reálnych dátach

V rámci posledného experimentu bola otestovaná sieť Real-ESRGAN na reálnych dátach, ktoré poskytla firma TESCANA Brno s.r.o. Dátová sada bola poskytnutá príliš neskoro na to, aby bola zakomponovaná do tréningového procesu neurónových sietí. Použitím reálnej dátovej sady pri tréningu by nebolo nutné obrázky umelo podvzorkovať, čím by sa sieť mohla naučiť generovať reálnejšie predikcie. Obrázky v tejto dátovej sade obsahujú v niektorých prípadoch aj mierny posun. Tento posun medzi rovnakými obrázkami by bolo vhodné odstrániť.

Celkovo obsahuje jednu snímku s tromi rôznymi nastaveniami šírky záberu (angl. *FOV*). Snímky majú teda veľmi podobný charakter. Pre každú šírku záberu je finálna snímka vytvorená v troch rozlíšeniach -  $512 \times 512$  px,  $1024 \times 1024$  px a  $2048 \times 2048$  px. Pri snímaní bolo nastavených aj sedem rôznych hodnôt pre „dwell time“, čo je parameter, ktorý nastavuje čas snímania elektrónovým lúčom pre každý pixel. Rôzne nastavenie tohto parametra spôsobuje rôznu mieru zašumenia výsledných obrázkov.

V tomto experimente bola neurónová sieť vyhodnotená na jednej snímke (jedno nastavenie FOV) so štyrmi intenzitami šumu. Ide o tri obrázky s najväčšou mierou šumu (v obrázku 7.10 sú očíslované 4, 5 a 6) a obrázok, ktorý obsahoval najmenšiu mieru šumu (v rovnakom obrázku ho značí číslo 0). Z pôvodnej snímky v rozlíšení  $512 \times 512$  px bol vytvorený výrez zo stredu o veľkosti  $128 \times 128$  px. Tento výrez bol umelo zväčšený pomocou siete Real-ESRGAN z predchádzajúceho experimentu, kedy bola tréningovaná pre zvýšenie rozlíšenia štyrikrát na obrázkoch s rozlíšením  $128 \times 128$  px. Miera šumu v tréningových dátach bola 30%. Výrez teda reprezentuje obrázok s prirodzene nízkym rozlíšením a je možné vytvoriť rovnaký výrez s štyrikrát väčším rozlíšením zo snímky, ktorá má štyrikrát väčšie rozlíšenie. Porovnanie predikcií siete s reálnymi obrázkami s vysokým rozlíšením sa nachádza v obrázku 7.10. Pre porovnanie sa tu nachádzajú aj obrázky zväčšené pomocou bikubickej interpolácie. V oranžovom štvorci sa nachádza malý zväčšený výrez pre lepšiu viditeľnosť detailov.

### Výsledky

Vstupné obrázky s najväčšou mierou šumu robia neurónovej sieti problém. Je to spôsobené tým, že šum je v tomto prípade iný od šumu, ktorý bol umelo pridaný do dátovej sady. Šum má zaujímavý „riadkový“ charakter, nie len bodový. Problém je aj v tom, že každý mikroskop môže produkovať rôzny charakter šumu. Predikcia pre najviac zašumený obrázok nepôsobí omnoho lepšie ako použitie bikubickej interpolácie. Do budúcnosti by to mohlo byť vylepšené zakomponovaním obrázkov do tréningovej dátovej sady, alebo lepšou simuláciou šumu pri tréningu.

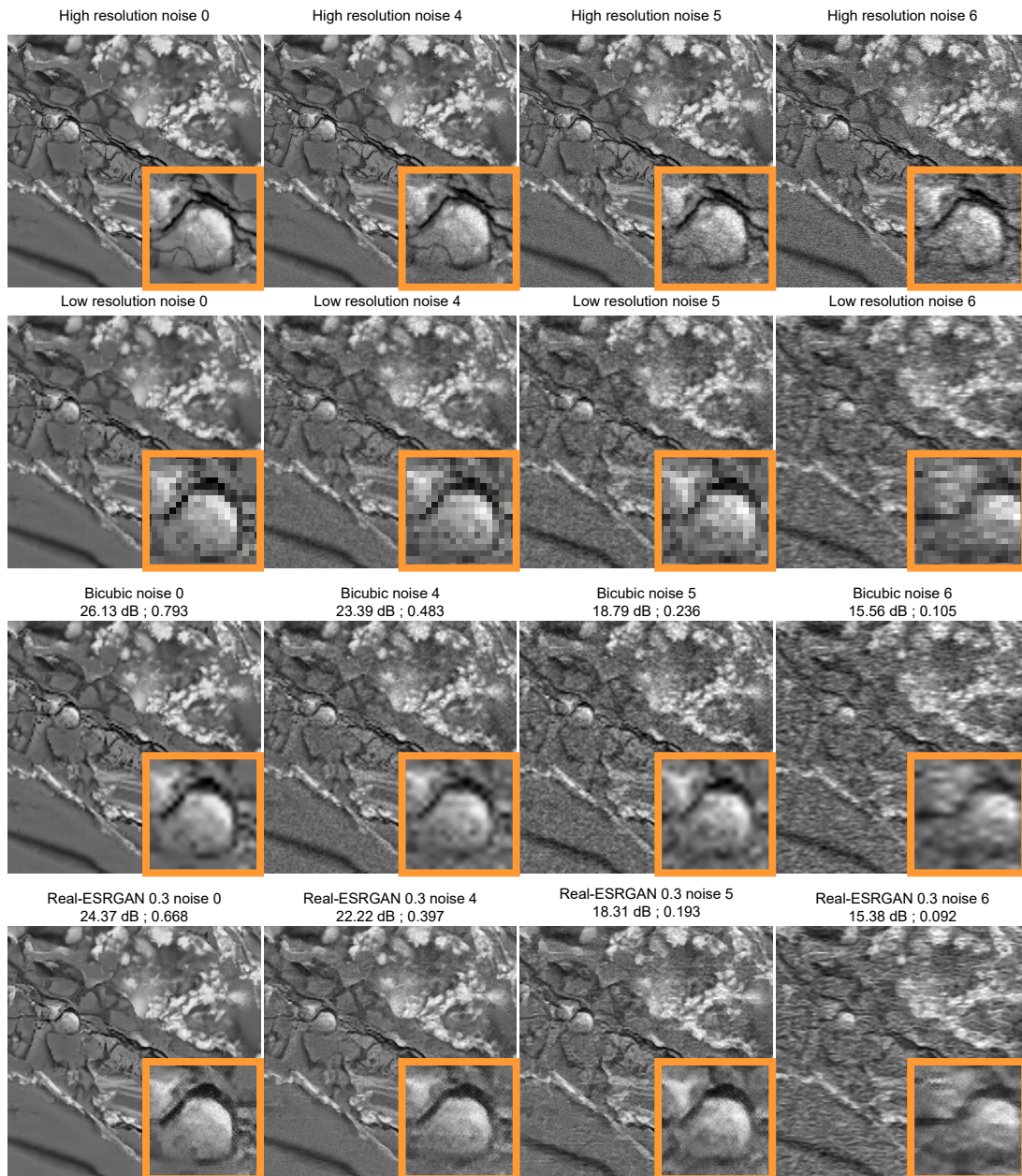
Pri menšej miere šumu (na 7.10 tretí stĺpec s označením 5) sieť stále vytvára artefakty spôsobené šumom vo vstupnom obrázku. Detail v pravom dolnom rohu však pôsobí ostrejšie ako v prípade bikubickej interpolácie.

Pri najmenšej miere šumu (označenie 4) sa v predikcii siete nachádza výrazne menšie množstvo šumu ako pri obrázku zväčšenom pomocou bikubickej interpolácie. Hrany pôsobia tiež ostrejšie.

V obrázkoch, ktoré neobsahujú šum (označenie 0) sú hrany pri predikcii ostrejšie a menej kostrbaté ako pri bikubickej interpolácii. Stále však okolo týchto hrán vzniká drobný šum, ktorý môže byť spôsobený tým, že sieť bola tréningovaná na dátach, ktoré majú odlišný

charakter od dát v tejto dátovej sade.

Čo sa týka hodnôt metrick PSNR a SSIM, neurónová sieť v žiadnom prípade v obrázku 7.10 neprekonalá bikubickú interpoláciu, avšak ako bolo ukázané, neznamená to, že by predikcie vyzerali horšie. Ohodnotenie subjektívnej kvality predikcii necháva autor na čitateľa.



Obr. 7.10: Porovnanie siete Real-ESRGAN s bikubickou interpoláciou na reálnych dátach. V stĺpcoch sa nachádzajú štyri rôzne intenzity šumu v obrázkoch. V prvom riadku sa nachádzajú obrázky vo vysokom rozlíšení, v druhom sú vstupné obrázky s nízkym rozlíšením, v treťom riadku sú výsledky pre bikubickú interpoláciu a v poslednom štvrtom riadku sú predikcie siete Real-ESRGAN.



## Zhodnotenie výsledkov experimentov

V prvom experimente boli porovnané siete U-Net a RUNet pri použití rozličných chybových funkcií. Tento experiment ukázal, že pri použití príznakovej chybovej funkcie sú siete v niektorých prípadoch schopné vylepšovať umelo zväčšené obrázky oproti zväčšeniu pomocou bikubickej interpolácie. Rozdiely však nie sú pri zväčšení rozlíšenia dvakrát veľmi veľké. Sieť RUNet bola schopná odstrániť aj malé množstvo prirodzeného šumu v testovacom obrázku.

V druhom experimente boli porovnané architektúry RUNet a Real-ESRGAN pri zvýšení rozlíšenia dvakrát. Tu sa ukázalo že RUNet sa snaží hrany zvýrazniť, čo v konečnom dôsledku pôsobí, že je obrázok ostrejší. Pri priblížení sú však hrany stále kostrbaté. Architektúra Real-ESRGAN zas okolo hrán vytvára nechcený šum. Pri tréovaní s 80% pravdepodobnosťou pridávania umelého šumu do tréovacích obrázkov je nechcený šum menej výrazný a sieť si bola schopná poradiť aj s obrázkom, ktorý obsahoval dosť veľké množstvo prirodzeného šumu.

V treťom experimente boli tie isté siete porovnané na zvýšení rozlíšenia štyrikrát. Obrázky generované architektúrou RUNet boli v tomto prípade veľmi podobné obrázkom zväčšeným pomocou bikubickej interpolácie, došlo najmä k zvýrazneniu hrán. Sieť Real-ESRGAN bola schopná umelo zväčšovať obrázky, ktoré pôsobia ostrejšie ako obrázky zväčšené bikubicou interpoláciou. Hrany sú menej kostrbaté. Nevýhodou je však, že ak sú vstupné obrázky malé, sieť odstráni drobné detaily, ktoré pôsobia ako šum.

V štvrtom experimente bolo ukázané že väčšie rozlíšenie tréovacích obrázkov má na tréovanie siete Real-ESRGAN pri zvýšení rozlíšenia štyrikrát pozitívny vplyv.

V poslednom experimente bola sieť Real-ESRGAN otestovaná na reálnej dátovej sade. Ukázalo sa, že pri menšej miere šumu vo vstupnom obrázku je schopná tento šum odstrániť a vyprodukovať obrázok, ktorý vyzerá kvalitnejšie ako obrázok zväčšený pomocou bikubickej interpolácie. Nedokázala si však poradiť s veľkou mierou šumu, ktorý má iný charakter ako šum v tréovacej dátovej sade.

# Kapitola 8

## Záver

Cieľom práce bolo nájsť a natréňovať architektúru neurónovej siete, ktorá by bola schopná zvýšiť rozlíšenie snímok pochádzajúcich z elektrónových mikroskopov. Pri riešení sa bolo nutné zoznámiť s problematikou elektrónových mikroskopov a snímok, ktoré zhotovujú. Takisto sa bolo potrebné zoznámiť s procesom tréňovania neurónových sietí, najmä pre úlohu zvýšenia rozlíšenia obrazu. Pre úlohu bolo dôležité nájsť vhodné obrázky a vytvoriť dátovú sadu. Na dátovej sade sa podarilo natréňovať vybrané architektúry neurónových sietí a porovnať ich výsledky v rôznych podmienkach.

Pri zvýšení rozlíšenia dvakrát je architektúra Real-ESRGAN schopná odstrániť aj pomerne vysoké množstvo šumu. Okolo jemných hrán však produkuje nechcené artefakty v podobe šumu. Ak teda obrázok obsahuje mnoho detailov výsledok nepôsobí lepšie ako bikubická interpolácia. Pri zvýšení rozlíšenia štyrikrát najlepšie výsledky produkovala architektúra Real-ESRGAN. Ukázalo sa, že architektúra je schopná zvýšiť rozlíšenie obrazu a odstrániť nechcený šum, ktorý vzniká prirodzene pri snímaní elektrónovým mikroskopom. To bolo možné aj vďaka vytvorenej dátovej sade, kde sú tréňovacie obrázky vystavené dvojúrovňovému degradačnému procesu. Schopnosť siete zvyšovať rozlíšenie obrázkov bola ukázaná na testovacej dátovej sade, ktorá obsahovala snímky z iných kategórií ako tréňovacia dátová sada. Nevýhodou však je, že sieť môže odstrániť drobné detaily, ktoré vyhodnotí ako šum, alebo môže v niektorých prípadoch produkovať nechcené artefakty. Kvalitu umelo zväčšených obrázkov je možné hodnotiť subjektívne, kde je možné obrázky porovnať s obrázkami zväčšenými pomocou bikubickej interpolácie.

V budúcnosti by bolo umelo zväčšené snímky možné ohodnotiť dotazníkom, kedy by účastníci hodnotili kvalitu zvýšenia rozlíšenia. To však nie je možné použiť priamo pri tréňovaní, ale len pri hodnotení. Vďaka dátovým sadám, ktoré vznikli je možné experimentovať s ďalšími architektúrami neurónových sietí. Najväčším problémom je však chýbajúca možnosť objektívne ohodnotiť kvalitu získaných obrázkov.

Počas práce vznikla dátová sada priamo pre účel zvýšenia rozlíšenia snímok z elektrónových mikroskopov, ktorú poskytla firma TESCAN Brno sro. Dátová sada bola však poskytnutá neskoro, preto nebola zahrnutá do tréňovacieho procesu v tejto práci. V práci bolo však aspoň otestované, že vytvorená neurónová sieť dokáže zvýšiť rozlíšenie obrázkov z tejto dátovej sady ak neobsahujú veľké množstvo šumu. Do budúca však existujú plány, že sa táto dátová sada použije pri tréňovaní architektúr neurónových sietí z tejto práce. Natréňované neurónové siete by bolo možné zakomponovať do procesu akvizície obrázkov elektrónovým mikroskopom a uľahčiť tak prácu operátorom.

# Literatúra

- [1] AQEEL ANWAR. *What is Transposed Convolutional Layer?* [online]. Marec 2020 [cit. 2021-01-09]. Dostupné z: <https://towardsdatascience.com/what-is-transposed-convolutional-layer-40e5e6e31c11>.
- [2] AVERSA, R., MODARRES, M. H., COZZINI, S. a CIANCIO, R. *NFFA-EUROPE - Majority SEM Dataset*. NFFA-EUROPE Project, 2018. DOI: 10.23728/B2SHARE.E344A8AFEF08463A855ADA08AADB352. Dostupné z: <https://b2share.eudat.eu/records/e344a8afef08463a855ada08aadb352>.
- [3] BLAU, Y., MECHREZ, R., TIMOFTE, R., MICHAELI, T. a ZELNIK MANOR, L. The 2018 PIRM Challenge on Perceptual Image Super-Resolution. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. September 2018.
- [4] BOZZOLA, J. J. a RUSSELL, L. D. *Electron microscopy: principles and techniques for biologists*. Jones & Bartlett Learning, 1999. ISBN 978-0-763-70192-5.
- [5] DONG, C., LOY, C. C., HE, K. a TANG, X. Learning a Deep Convolutional Network for Image Super-Resolution. In: *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014, s. 184–199. ISBN 978-3-319-10593-2.
- [6] DONG, C., LOY, C. C., HE, K. a TANG, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2016, zv. 38, č. 2, s. 295–307. DOI: 10.1109/TPAMI.2015.2439281.
- [7] DUMOULIN, V. a VISIN, F. A guide to convolution arithmetic for deep learning. *ArXiv e-prints*. marec 2016.
- [8] EDE, J. M. Warwick electron microscopy datasets. *Machine Learning: Science and Technology*. IOP Publishing. sep 2020, zv. 1, č. 4, s. 045003. DOI: 10.1088/2632-2153/ab9c3c. Dostupné z: <https://doi.org/10.1088/2632-2153/ab9c3c>.
- [9] EGERTON, R. F. et al. *Physical principles of electron microscopy*. Springer, 2005. ISBN 978-3-319-39877-8.
- [10] GOODFELLOW, I., POUGET ABADIE, J., MIRZA, M., XU, B., WARDE FARLEY, D. et al. Generative adversarial nets. *Advances in neural information processing systems*. 2014, zv. 27.
- [11] HAN, D. Comparison of commonly used image interpolation methods. In: Atlantis Press. *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*. 2013, sv. 10, s. 1556–1559. ISBN 978-90-78677-61-1.

- [12] HARIS, M., SHAKHNAROVICH, G. a UKITA, N. Deep Back-Projection Networks for Super-Resolution. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, s. 1664–1673. DOI: 10.1109/CVPR.2018.00179.
- [13] HE, K., ZHANG, X., REN, S. a SUN, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, s. 1026–1034. DOI: 10.1109/ICCV.2015.123.
- [14] HE, K., ZHANG, X., REN, S. a SUN, J. Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, s. 770–778. DOI: 10.1109/CVPR.2016.90.
- [15] HORE, A. a ZIOU, D. Image quality metrics: PSNR vs. SSIM. In: IEEE. *2010 20th international conference on pattern recognition*. 2010, s. 2366–2369.
- [16] HU, X., NAIEL, M. A., WONG, A., LAMM, M. a FIEGUTH, P. RUNet: A Robust UNet Architecture for Image Super-Resolution. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, s. 505–507. DOI: 10.1109/CVPRW.2019.00073.
- [17] JEOL LTD.. *Scanning Electron Microscope A To Z* [online]. [cit. 2022-04-12]. Dostupné z: [https://www.jeol.co.jp/en/applications/pdf/sm/sem\\_atoz\\_all.pdf](https://www.jeol.co.jp/en/applications/pdf/sm/sem_atoz_all.pdf).
- [18] JOHNSON, J., ALAHI, A. a FEI FEI, L. Perceptual losses for real-time style transfer and super-resolution. In: Springer. *European conference on computer vision*. 2016, s. 694–711. ISBN 978-3-319-46475-6.
- [19] JOLICOEUR-MARTINEAU, A. The relativistic discriminator: a key element missing from standard GAN. *CoRR*. 2018, abs/1807.00734. Dostupné z: <http://arxiv.org/abs/1807.00734>.
- [20] JONATHAN HUI. *GAN — Super Resolution GAN (SRGAN)* [online]. Jún 2018 [cit. 2021-01-09]. Dostupné z: <https://jonathan-hui.medium.com/gan-super-resolution-gan-srgan-b471da7270ec>.
- [21] KIM, J., LEE, J. K. a LEE, K. M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, s. 1646–1654.
- [22] LAI, W.-S., HUANG, J.-B., AHUJA, N. a YANG, M.-H. Fast and Accurate Image Super-Resolution with Deep Laplacian Pyramid Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2019, zv. 41, č. 11, s. 2599–2613. DOI: 10.1109/TPAMI.2018.2865304.
- [23] LEDIG, C., THEIS, L., HUSZÁR, F., CABALLERO, J., CUNNINGHAM, A. et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, s. 105–114. DOI: 10.1109/CVPR.2017.19.
- [24] LIM, B., SON, S., KIM, H., NAH, S. a LEE, K. Enhanced Deep Residual Networks for Single Image Super-Resolution. In: *2017 IEEE Conference on Computer Vision and*

- Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, Jul 2017, s. 1132–1140. DOI: 10.1109/CVPRW.2017.151. ISSN 2160-7516.
- [25] MIYATO, T., KATAOKA, T., KOYAMA, M. a YOSHIDA, Y. *Spectral Normalization for Generative Adversarial Networks*. arXiv, 2018. DOI: 10.48550/ARXIV.1802.05957.
- [26] RONNEBERGER, O., FISCHER, P. a BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: Springer. *International Conference on Medical image computing and computer-assisted intervention*. 2015, s. 234–241. DOI: 10.1007/978-3-319-24574-4\_28. ISBN 978-3-319-24574-4.
- [27] SHI, W., CABALLERO, J., HUSZAR, F., TOTZ, J., AITKEN, A. P. et al. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016. DOI: 10.1109/CVPR.2016.207.
- [28] SIMONYAN, K. a ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *ArXiv preprint arXiv:1409.1556*. 2014.
- [29] TONG, T., LI, G., LIU, X. a GAO, Q. Image Super-Resolution Using Dense Skip Connections. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct 2017, s. 4809–4817. DOI: 10.1109/ICCV.2017.514. ISSN 2380-7504.
- [30] WANG, X., XIE, L., DONG, C. a SHAN, Y. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. In: IEEE Computer Society. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2021, s. 1905–1914. DOI: 10.1109/ICCVW54120.2021.00217.
- [31] WANG, X., YU, K., WU, S., GU, J., LIU, Y. et al. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: *ECCV Workshops (5)*. 2018, s. 63–79. DOI: 10.1007/978-3-030-11021-5\_5.
- [32] WANG, Y., PERAZZI, F., MCWILLIAMS, B., SORKINE HORNUNG, A., SORKINE HORNUNG, O. et al. A fully progressive approach to single-image super-resolution. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, s. 864–873.
- [33] WANG, Z., CHEN, J. a HOI, S. C. H. Deep Learning for Image Super-Resolution: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021, zv. 43, č. 10, s. 3365–3387. DOI: 10.1109/TPAMI.2020.2982166.
- [34] WANG, Z., BOVIK, A. C., SHEIKH, H. R. a SIMONCELLI, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*. IEEE. 2004, zv. 13, č. 4, s. 600–612.
- [35] WIKIPEDIA. *Electron microscope* [online]. December 2021 [cit. 2022-01-07]. Dostupné z: [https://en.wikipedia.org/wiki/Electron\\_microscope](https://en.wikipedia.org/wiki/Electron_microscope).
- [36] WIKIPEDIA. *Interpolation* [online]. Január 2022 [cit. 2022-01-07]. Dostupné z: <https://en.wikipedia.org/wiki/Interpolation>.

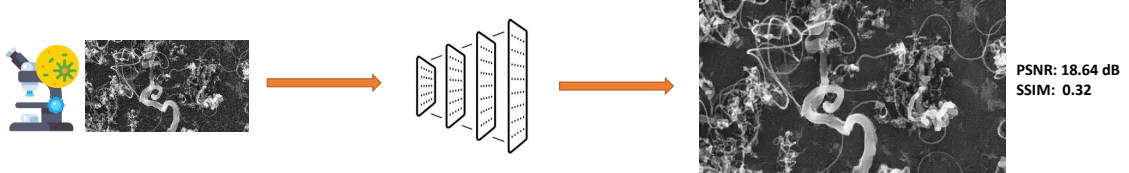
- [37] YANG, W., ZHANG, X., TIAN, Y., WANG, W., XUE, J.-H. et al. Deep Learning for Single Image Super-Resolution: A Brief Review. *IEEE Transactions on Multimedia*. 2019, zv. 21, č. 12, s. 3106–3121. DOI: 10.1109/TMM.2019.2919431.
- [38] ZHANG, K., ZUO, W. a ZHANG, L. Learning a Single Convolutional Super-Resolution Network for Multiple Degradations. In: *Proceedings - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018*. United States: IEEE Computer Society, December 2018, s. 3262–3271. DOI: 10.1109/CVPR.2018.00344.

# Príloha A

## Obsah pamäťového média

V rámci práce bol vytvorený plagát s veľkosťou strany A1. Náhľad plagátu sa nachádza na ďalšej strane (obrázok [A.1](#)). Plagát sa tiež nachádza na priloženom pamäťovom médiu. Pamäťové médium má nasledujúcu štruktúru:

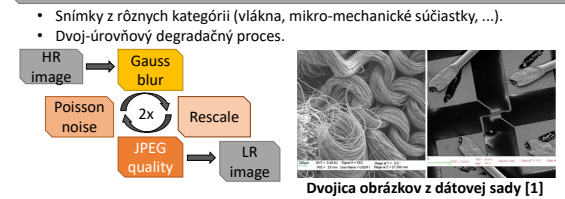
- `poster.pdf` – PDF súbor s plagátom,
- `text.pdf` a `text_print.pdf` – text technickej správy,
- adresár `/tex` – zdrojové texty a súbory technickej správy v jazyku  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ,
- adresár `/src` – zdrojové kódy k práci, nachádza sa tu súbor `README.md`, ktorý bližšie popisuje štruktúru tohto adresára, návod na stiahnutie tréningovej dátovej sady a návod na vytvorenie predikcie pomocou natrénovanej siete Real-ESRGAN 0.3, ktorej váhy sa tiež nachádzajú na pamäťovom médiu,
- nachádza sa tu aj ručne vytvorená testovacia dátová sada.



### Problémy

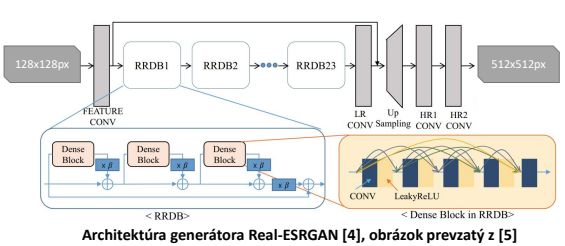
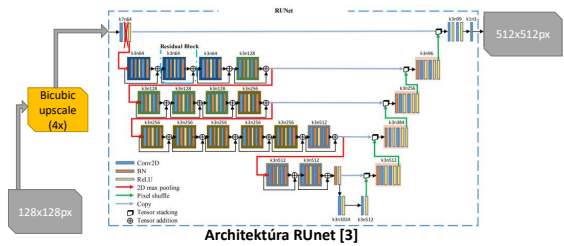
- Dlhý proces akvizície elektrónovým mikroskopom, najmä pri vysokom rozlíšení.
- Dlhodobejšie ožarovanie vzorku spôsobuje jeho nabíjanie, alebo ju môže znehodnotiť.
- Niektoré vzorky nejde úplne zafixovať a pri snímaní sa môžu hýbať.
- Prírodné vznikajúci šum v snímkach z elektrónového mikroskopu.
- Prírodný šum v obrázkoch má často rozličný charakter.
- Metriky používané pre zvýšenie rozlíšenia obrazu často zlyhávajú.
- Umelé podzvorkovanie obrázkov nemusí odpovedať procesu snímania v nižšom rozlíšení.

### Dátová sada



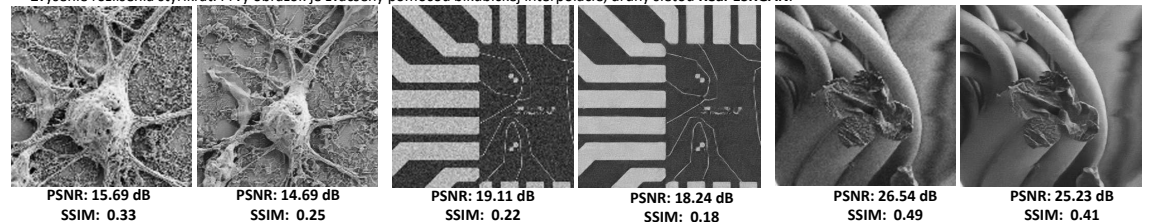
### Použité architektúry

- Prvotné riešenie pomocou architektúry U-Net[2] navrhutej pre úlohu segmentácie.
- Porovnanie s ďalšími architektúrami, ktoré boli navrhnuté pre zvýšenie rozlíšenia obrazu.

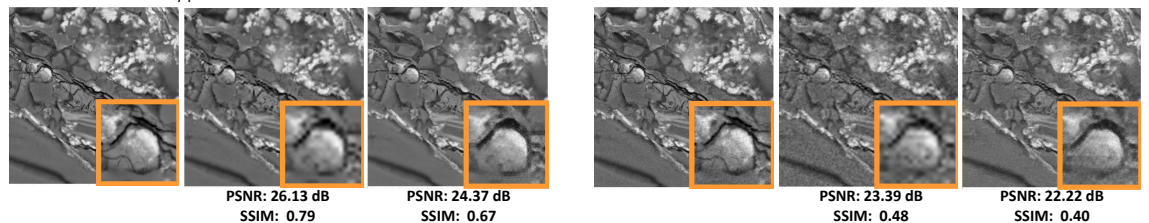


### Experimenty

- Experimenty s pridávaním rôzneho množstva šumu do tréningovej časti dátovej sady – šum v 80% a šum v 30% dátovej sady s architektúrami UNet a ESRGAN.
- Zvýšenie rozlíšenia štyrikrát. Prvý obrázok je zväčšený pomocou bikubickej interpolácie, druhý sieťou Real-ESRGAN.



- Experimenty na reálnej dátovej sade od firmy TESCAN Brno s.r.o.
- Prvý obrázok je s vysokým rozlíšením, druhý je pôvodný malý obrázok zväčšený pomocou bikubickej interpolácie a posledný je predikcia siete Real-ESRGAN.
- Pre dve rôzne intenzity prírodného šumu v obrázkoch.



[1] Rossella Aversa, Mohammad Hadi Modarres, Stefano Cozzini, and Regina Ciaccio. Nffa-europe - majority sem dataset, 2018.  
 [2] Ronneberger, O., Fischer, P. a Brox, T., -net: Convolutional networks for biomedical image segmentation. 2015.  
 [3] Hu, X., Naiel, M. A., Wong, A., Lamm, M. a Fieguth, P. UNet: A Robust UNet Architecture for Image Super-Resolution. 2019.  
 [4] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y. et al. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. 2018.  
 [5] J. Mun and J. Kim. Universal super-resolution for face and non-face regions via a facial feature network. 2020.

Obr. A.1: Vytvorený plagát.