BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

COMIC IMAGES SUPER-RESOLUTION USING DEEP LEARNING

ZVÝŠENÍ ROZLIŠENÍ KOMIKSOVÝCH OBRÁZKŮ POMOCÍ HLUBOKÝCH NEURONOVÝCH SÍTÍ

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **PETER ZDRAVECKÝ**
AUTOR PRÁCE

SUPERVISOR **Ing. MICHAL ŠPANĚL, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2022

Department of Computer Graphics and Multimedia (DCGM) Academic year 2021/2022

Bachelor's Thesis Specification

24494

Student: **Zdravecký Peter**
Programme: Information Technology
Title: **Comic Images Super-Resolution Using Deep Learning**
Category: Image Processing

Assignment:

1. Study deep neural networks and their learning.
2. Get acquainted with current super-resolution techniques using deep neural networks to artificially increase resolution of images.
3. Create a dataset of comic images for your own experiments.
4. Select appropriate methods and design a neural network architecture for the task of comic image super-resolution.
5. Experiment with your implementation and, if necessary, design your own method modifications.
6. Compare the achieved results and discuss the possibilities of future development.
7. Create a poster, or short video, presenting your work, its goals and results.

Recommended literature:

- 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).

Requirements for the first semester:

- Completion of the first three items of the assignment.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Španěl Michal, Ing., Ph.D.**
Head of Department: Černocký Jan, doc. Dr. Ing.
Beginning of work: November 1, 2021
Submission deadline: May 11, 2022
Approval date: April 11, 2022

Abstract

This paper demonstrates a super-resolution method for improving the resolution and quality of comic images by using deep learning. The challenging part of the task was to keep the quality of the text parts and drawings simultaneously, without significant deformation of any part. Two deep neural networks were used to achieve satisfying results. U-Net network and its modification called Robust U-Net. The chosen loss functions to train these networks were the Mean Squared Error and Perceptual loss. The work contains experiments on U-Net and modified RUNet networks with a combination of each loss function. Additional experiments looked at how the number of used blocks from the VGG16 loss network affects the Perceptual loss function. Experiments have shown that a Robust U-Net network using a Perceptual loss with three extracted blocks got the best results.

Abstrakt

Táto práca demonštruje metódu super rozlíšenia na zlepšenie kvality komiksových obrázkov pomocou hlbokého učenia. Náročnou časťou tejto úlohy bolo súčasne zachovať kvalitu textových a kreslených častí, bez výraznej deformácie ktorejkoľvek časti z nich. Na dosiahnutie uspokojivých výsledkov boli skúmané dve hlboké neurónové siete. Sieť U-Net a modifikácia s názvom Robustný U-Net (RUNet). Zvolené stratové funkcie na trénovanie týchto sietí boli stredná kvadratická chyba a perceptuálna strata. Práca obsahuje experimenty na týchto sieťach v kombinácii s každou stratovou funkciou. Ďalšie experimenty sa zamerali na vplyv počtu použitých blokov zo stratovej siete VGG16 na funkciu perceptuálnej straty. Experimenty ukázali, že sieť RUNet využívajúca perceptuálnu stratu s tromi extrahovanými blokmi dosiahla najlepšie výsledky.

Keywords

single image super-resoltuion, deep learning, convolutional neural networks, comic images, U-Net, RUNet

Kľúčové slová

zvýšenie rozlíšenia obrazu, hlboké učenie, konvolučné neuronové siete, komiksové obrázky, U-Net, RUNet

Reference

ZDRAVECKÝ, Peter. *Comic Images Super-Resolution Using Deep Learning.* Brno, 2022. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Michal Španěl, Ph.D.

Rozšírený abstrakt

Úvod

Zväčšovanie rozlíšenia obrázkov a zlepšovanie ich kvality sa často spája s procesmi, ako spracovanie lekárskych snímok alebo detekcia poznávacích značiek. Hoci klasické metódy zväčšovania rozlíšenia obrazu (napr. bilineárna interpolácia) sú rýchle a ľahko implementovateľné, metódy hlbokého učenia využívajúce hlboké neurónové siete ukázali v posledných rokoch potenciál a vynikajúce výsledky.

Cieľom práce je navrhnúť optimálne riešenie pre zväčšenie rozlíšenia a zlepšenie kvality (tzv. super rozlíšenie) komiksových obrázkov. Náročnosť spočíva v kompozícii vybraných obrázkov. Štruktúra komiksových obrázkov je tvorená z kreslených a textových častí. Táto kompozícia sťažuje učenie neurónovej siete, keďže je dôležité zaručiť kvalitu kreslených aj textových častí. Preto je dôležité nielen zväčšenie a vylepšenie kvality obrázkov, ale zároveň aj zachovanie kvality textu a kresieb súčasne. To znamená vyhnúť sa výraznému skresleniu jednej časti kompozície ná úkor tej druhej. Práca prezentuje porovnanie dvoch konvolučných neurónových sietí, U-Net a RUNet, s využitím funkcií strednej kvadratickej odchýlky a perceptuálnej straty s cieľom nájsť najlepšie fungujúce riešenie pre tento problém. Bola vytvorená webová aplikácia, ktorá poslúži na užívateľsky prívetivé zväčšovanie komiksových obrázkov pomocou výslednej architektúry, ktorá dosiahla najlepšie výsledky.

Popis navrhnutého riešenia

Veľmi dôležitým aspektom každej úlohy hlbokého učenia je mať vhodnú dátovú sadu. Bolo nutné zostaviť dátovú sadu tvorenú z komiksových obrázkov. Keďže cieľom je zabezpečiť zvyšovanie kvality textu a súčasne aj kreslených časti, tak dátová sada musí spĺňať určité kritéria:

1. Kompozícia jednotlivých obrázkov musí byť vyvážená. Teda pomer obsahu textových a kreslených častí na jednej komiksovej stránke by mal byť 1 k 1.

2. Obrázky musia byť vo vysokej kvalite – dôležité kvôli podvzorkovaniu pre účely trénovania.

3. Na obrázkoch sa nesmú nachádzať akékoľvek chyby a artefakty.

Jednotlivé body bolo potrebné dodržiavať pri tvorení dátovej sady. Každý obrázok bol vybraný ručne pre zabezpečenie kvality dátovej sady. Bolo potrebné zostaviť až 2 dátové sady obrázkov – jednu na trénovanie a druhú na validáciu. Výsledná sada pre trénovanie obsahovala okolo 2000 obrázkov a validačná sada približne 200 obrázkov.

Pre hľadaní riešenia boli skúmané 2 architektúry. Prvou bola architektúra U-Net a druhou Robustný U-Net (RUNet), čo je modifikácia U-Net prispôsobená na zväčšovanie obrázkov. Spoločne s týmito architektúrami na ich trénovanie boli použité dve stratové funkcie – stredná kvadratická chyba a perceptuálna strata. Vyššie spomenutá dátová sada bola tvorená len z obrázkov vo vysokom rozlíšení. Aby sa mohla neurónová sieť naučiť zväčšovať konkrétny typ obrázkov, bolo nutné si zaobstarať rovnaké obrázky aj v nízkej kvalite. Na to bolo použité umelé podvzorkovanie obrázkov pomocou degradačného modelu, ktorý bol navrhnutý v práci [18]. Použitý degradačný model fungoval na princípe zmenšenia obrázku pomocou vybraného faktoru. Po zmenšení bolo na obrázok aplikované gaussovo rozmazanie a potom sa pridal šum. Následne bol obrázok zväčšený do pôvodnej veľkosti. Takto vytvorené obrázky v nízkej kvalite sa používali na tréning jednotlivých architektúr. Webová aplikácia bola navrhnutá tak, aby využívala výslednú neurónovú sieť na zväčšovanie komiksových obrázkov. Užívateľ by mal pomocou aplikácie možnosť vylepšiť vlastné obrázky, a to bez akýchkoľvek ďalších inštalácií, len prostredníctvom prehliadača.

Experimenty

Experimenty overovali schopnosti vybraných architektúr v spojení s vybranými stratovými funkciami pri zväčšovaní rozlíšenia obrázkov. Prvé testy spočívali v porovnaní architektúr U-Net a RUNet. Architektúra U-Net používala stratovú funkciu priemernej kvadratickej chyby a architektúra RUNet využívala perceptuálnu stratu. Už v týchto experimentoch boli spozorované lepšie výsledky, ako pri klasických interpolačných metódach. Avšak objavilo sa zvláštne chovanie architektúry RUNet. Bolo možné pozorovať zvláštne odtiene pri okrajoch vyfarbených plôch. Preto bola navrhnutá modifikácia RUNet architektúry, ktorá problém odstránila a zároveň dosiahla lepšie metriky, ako základné zostavenie. Ďalšie testy boli zamerané na použité stratové funkcie. Testovala sa výkonnosť architektúr pri použití jednotlivých funkcií. Výsledky ukázali, že pre úlohu zväčšovania rozlíšenia obrazu je lepšia funkcia perceptuálnej straty. Preto sa následné experimenty zamerali na túto funkciu. Percetuálna strata využíva na počítanie pomocnú neurónovú sieť, ktorá prevádza obrázky do “feature space” – priestor, v ktorom sa pohybujú skúmané vlastnosti/črty. Pomocná sieť sa skladá z 5 blokov obsahujúcich konvolučné vrstvy, ktoré postupne prevádzajú obrázky do feature space. Experiment sa snažil nájsť optimálny počet použitých blokov z rozsahu od 1 do 5 s cieľom zachovania najlepšej kvality textu a kresby. Najúspešnejšia bola architektúra RUNet pri použití perceptuálnej straty s 3 použitými blokmi z pomocnej siete.

Zhrnutie výsledkov a budúca práca

Celkovo experimenty ukázali prijateľné výsledky. Obe skúmané architektúry (U-Net aj RUNet) dosiahli lepšie výsledky, ako bilineárna interpolácia. Zároveň ale bola navrhnutá modifikácia RUNet architektúry, keďže pri farebne vyplnených oblastiach boli spozorované zvláštne odtiene v blízkosti tmavých obrysov postáv. Najlepšia kvalita textu a obrázkov bola dosiahnutá pri použití modifikovanej RUNet architektúry s 3 použitými blokmi zo stratovej siete. V rámci budúceho skúmania alebo potenciálneho rozšírenia práce by bolo zaujímavé sa zamerať na Generatívne adversariálne siete (GAN)[13]. Ich výhodou je využitie dvoch sietí, ktoré sa navzájom trénujú na vytváranie vylepšených, vysokokvalitných detailov obrazu. Podstatné vylepšenie by sa dalo dosiahnuť hlbším zameraním sa na textové časti komiksových obrázkov. V dosiahnutých výsledkoch je stále viditeľné určité skreslenie textu. Preto by bolo vhodné preskúmať oblasť textovo orientovaných techník super rozlíšenia a skombinovať ich s technikami použitými v tejto práci. Napríklad použitie inception blokov [34] môže pomôcť naučiť sa rekonštruovať textové časti a detaily tvorené nízkym počtom pixelov.

Comic Images Super-Resolution Using Deep Learning

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Michal Španěl, Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Peter Zdravecký
May 2, 2022

Acknowledgements

Many thanks to my work supervisor, without whom it would not have been possible to create this work. Thank you, Ing. Michal Španěl, Ph.D., for your guidance, advice, and time.

Contents

Chapter 1

Introduction

Super-resolution is often associated with image-related tasks, such as medical image processing or license plate detection. Although classic upscaling methods (e.g. bilinear interpolation) are fast and easy to implement, deep learning methods using deep neural networks have shown potential and excellent results in the last few years.

This thesis aims to develop the optimal solution for the super-resolution of the comic images. The challenging part of the task lies in the composition of the chosen images. The structure formed by the drawn parts and text parts makes this task more difficult for the neural network to learn. Therefore, the main goal is to simultaneously upscale and enhance the image with the preservation of quality of the text and drawings. That means avoiding significant distortion of one part of the composition on behalf of the other. A comparison of two convolutional neural networks, U-Net and RUNet, with the usage of Mean Square Error and Perceptual loss functions, is presented to find a working solution for the problem. Additionally, the web application is presented, which will be used for comic upscaling using the final network with the best results.

The conducted experiments have shown the capability of the chosen architectures and loss functions to produce acceptable results while also outperforming the bilinear interpolation method. The best results were obtained using the RUNet network with Perceptual loss function, with three extracted blocks from the VGG16 loss network used to calculate the loss.

Chapter 2 talks about the extended motivation, comic books, and the process of super-resolution with image enhancement such as denoising or deblurring. Then the basic concept of image scaling with classic methods is introduced and described in Chapter 3, along with image quality metrics. Subsequently, Chapter 4 summarizes the state-of-the-art, starting with learning-based upscaling and basic approaches for super-resolution. This chapter also talks about convolutional neural networks based on U-Net and ends with generative adversarial networks. Chapter 5 closely describes the proposal for the solution of the task. Chapter 6 then explains the implementation details. Lastly, the conducted experiments are described in Chapter 7.

Chapter 2

Super-resolution and Denoising of Comic-like Images

Deep neural networks are nowadays becoming one of the most popular and researched fields in computer engineering. They are used in many areas, such as automatic speech recognition, image restoration, medical image analysis, segmentation, and more. One of the possible applications for deep learning is super-resolution, also known as image upscaling from low-resolution images. Furthermore, super-resolution not only upscales the image but also enhances the quality by eliminating minor imperfections and defects, such as noise, blurriness, and various artifacts.

Recently, deep learning methods have succeeded in image upscaling and denoising. It is therefore possible to observe the widespread use of the deep learning methods in real life. An exciting example of super-resolution usage is in game development and graphics cards. The game scenes are rendered in lower resolution and then upscaled using deep learning methods. Such an approach leads to better performance (higher rates of frames per second), which means a better player experience without losing the image quality. The way modern graphics cards are manufactured also makes this process very efficient.

When working with images and image quality metrics, it is vital to consider how a person looking at it sees the image. The quality of the image, when perceived by the human eye, lies in the ability to observe it and get an impression. Attention, eye movement, and the emotion it evokes in a person are also notable factors when perceiving image quality [42]. These features are crucial in modern art, book paintings, film effects, etc.

Comic books are becoming increasingly popular as one of the more modern art forms. They were initially meant mainly for children, as the many images found within comic books were more appealing to them than classic literature. However, nowadays, comic book fans can be found in all age categories. One of the main contributions to the increase in comic book popularity is the recent rise of cinematic adaptations of comic books. Drawings, which can be found in the pages of comic books, are an essential part of the comic book reading experience. Drawings often depict characters in colorful scenes that contain many small details. It is why they must be of high quality to create interest and emotions in the reader.

The challenging part of comic books, which makes them different from other art forms, is the composition. Typical literature consists of pages of text with occasional illustrations. The illustrations accompany the story, give additional information that can not be described only by text, or help to immerse the reader in the atmosphere. However, the structure of comic books is different. Pages are commonly made up of multiple blocks representing different moments in the story. Each block of a comic book page could be compared to a shot from a film scene. In a film scene, the dialogues and narrations are expressed verbally by characters. However, in comic books, they are drawn inside text bubbles which are a part of the picture or image itself. The text bubbles contain either a description of the current event depicted in the picture or the characters' speech and thoughts. The unusual combination of text bubbles and colorful drawings makes comic books stand out and makes working with them a challenging task for deep neural networks.

Over time, the number of comic books has increased. The first comics were printed on paper, and only later, with the invention of better technologies, did comics begin to be created in digital form. People who have kept comic books in paper form can transfer them to digital form. Nevertheless, the quality of the scanned pages is not always good. Scanned and digitalized images can often be in low-resolution (caused by compressions, etc.) and contain noise and artifacts. Due to noise amplification and other shortcomings, basic interpolation methods used for image upscaling are not ideal. That is where the image super-resolution comes in handy. The expected results of the super-resolution are shown in Figure 2.1.

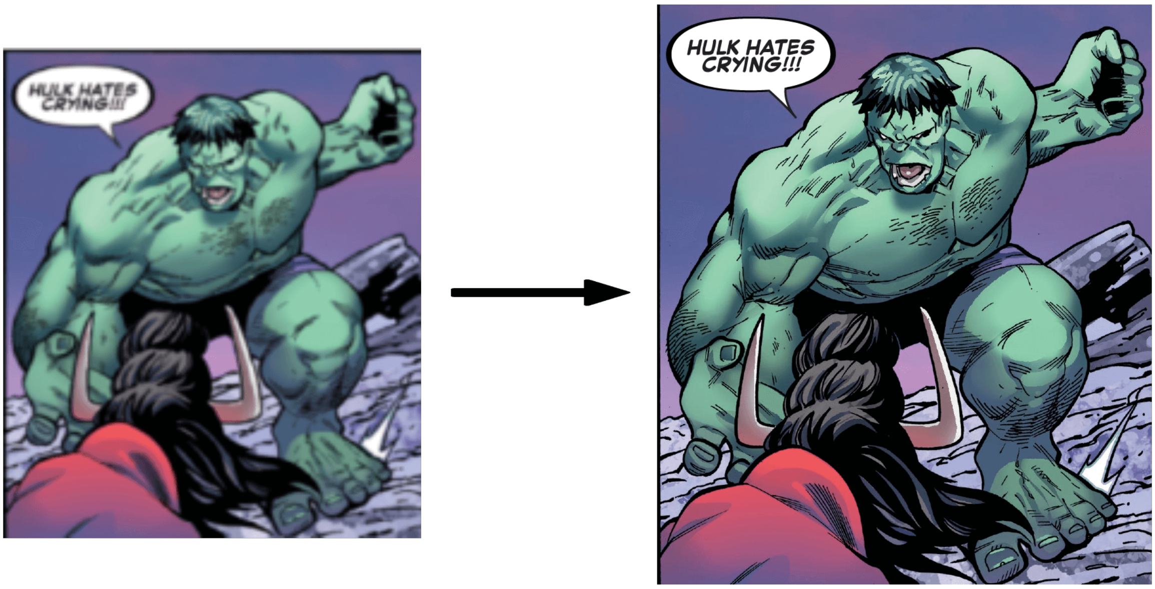


Figure 2.1: Example of the expected output from a super-resolution process.

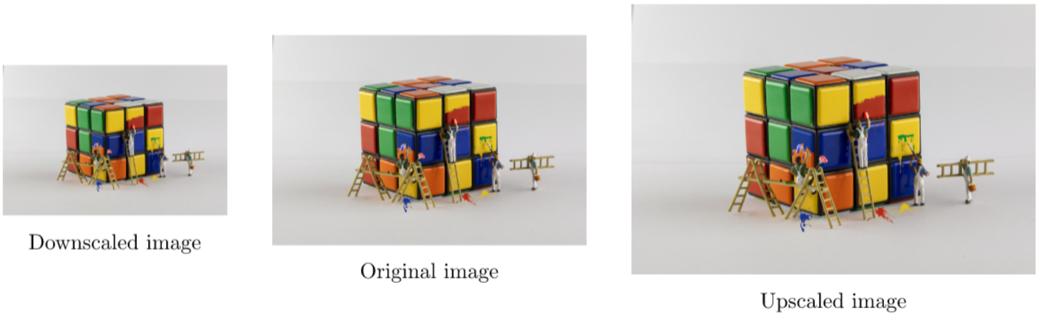
Chapter 3

Image Scaling

This chapter describes the basics of image scaling. The definition of the problem is clarified in the beginning. Afterwards, the scaling methods used to solve this problem are explained and classified. Finally, the chapter talks about metrics used to quantify and compare image quality.

3.1 Definition of the Problem

"Image scaling refers to representing and resizing an original image with the use of a higher or lower number of pixels" [3]. It is a frequently used method when working with images. One of the uses of image scaling is to stretch the image to fit the display. Image scaling also has applications in the scientific field, like astronomy or art. Many advanced techniques have been developed to handle image scaling tasks, which are described in more detail in the next chapter 3.2.



Downscaled image

Original image

Upscaled image

Figure 3.1: Visualization of image scaling.

Before diving into scaling methods, basic terms used in image scaling terminology need to be introduced. The image interpolation process from higher to lower resolution refers to *down-scaling* or down-sampling of the original content. The opposite process that brings the original content from lower resolution to higher is called *upscaling*, or up-sampling. HR stands for high-resolution image I^{HR}, and LR stands for low-resolution image I^{LR}. Low-resolution images are typically in the range of hundreds of pixels on both axes.

3.2 Scaling Methods

Image scaling methods can be divided into two groups:

- Interpolation-based methods,
- Learning-based upsampling.

The interpolation-based methods (e.g. Nearest-neighbor, bilinear interpolation and bicubic interpolation) are the most used generic methods. However, they have some shortcomings, such as computational complexity when upscaling to very HR images or image quality connected with noise, blur, etc. Learning-based methods try to eliminate these issues using new techniques (e.g. Transposed Convolution Layer or Sub-Pixel Layer).

3.2.1 Interploation-based Methods

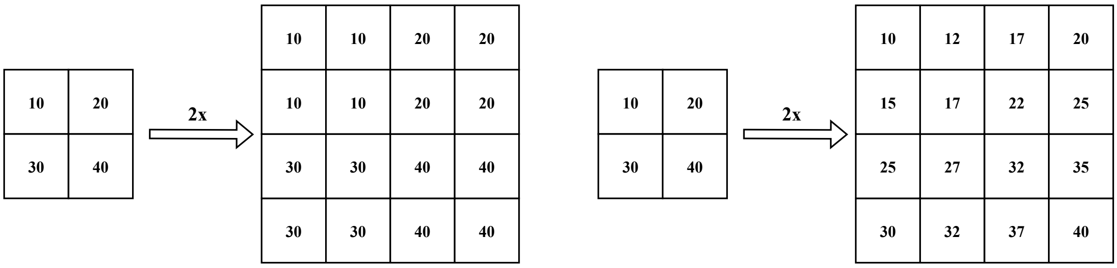
To deal with the problem of super-resolution, the first and most basic methods developed were the interpolation-based methods. Their implementation is straightforward and easy to understand.

Nearest-neighbor Interpolation

The nearest-neighbor is a simple algorithm that uses the nearest pixel values to fill newly created pixels. The advantages of the nearest-neighbor interpolation method are speed and simplicity, whereas the output quality is worse.

Bilinear Interpolation

The bilinear interpolation algorithm applies linear interpolation to both of the axes of the image, first to one axis and then to the other. Linear interpolation can be achieved by constructing new data within the range of known data points. The new interpolated point is then equal to the weighted average of the 4 nearest pixels located in diagonal directions from a given pixel (figure 3.2b). Comparison made in the paper [30] showed that bilinear interpolation has the same computational time and complexity as the nearest-neighbor algorithm, but bilinear interpolation reached the higher quality of the interpolated image.



(a) Nearest-neighbor Interpolation

(b) Bilinear Interpolation

Figure 3.2: Visualization of algorithm outputs – 2×2 input matrix is upscaled by factor 2 to the final 4×4 matrix.

Bicubic Interpolation

As described in [40, 21], the bicubic interpolation performs cubic interpolation on each of the two axes. In contrast to bilinear interpolation, which only takes 4 (2 × 2) pixels into account, bicubic interpolation considers 16 (4 × 4) pixels. Bicubic interpolation achieves smoother results with fewer artifacts, but contributes to slower computation time.

3.2.2 Learning-based Upsampling

Interpolation-based upsampling methods use information directly from a given input without using any additional information that could help. Therefore, these methods come with unwanted outcomes such as blurriness or noise amplification. The upcoming trend comes to replace interpolation-based methods with Learning-based upsampling methods. [40]

Learning-based methods are new in the field of super-resolution. They contain trainable parameters that can be adjusted to a particular task and offer better performance than interpolation-based methods. A more detailed description of the learning-based methods can be found in Chapter 4.1.

3.3 Image Quality Evaluation Metrics

It is important to track image quality in the super-resolution process. Various methods (subjective and objective) have been developed for this purpose. One of them is human evaluation (or judgment), which classifies as a subjective method. Objective methods are based on numerical criteria comparisons. Image quality tracking in this paper uses some of the objective methods.

3.3.1 Mean Squared Error (MSE)

One of the image quality evaluation methods is the Mean Square Error method. The MSE represents the cumulative squared error between the input (in our case, output from the super-resolution model) and the original image (ground truth[1]). Lower MSE value is more desirable. MSE is computed as:

$$MSE = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} [I(i,j) - \hat{I}(i,j)]^2 \tag{3.1}$$

As the input, images are given with the same resolution $H \times W$. I stands for ground truth image, and $\hat{I}$ stands for input image (predicted output from a model). Squared error is calculated over each pixel and then summed up. The summed squared error is divided by images resolution $H \times W$ to get the MSE.

3.3.2 Peak Signal-to-noise Ratio (PSNR)

Peak signal-to-noise ratio metric, described in [32], stands for the ratio between the maximum power of a signal to the maximum power of the noise signal. PSNR records peak signal power between images. The PSNR measurement unit is decibels (dB). A higher value of PSNR is more desirable. For the original image I and predicted image $\hat{I}$, PSNR is defined as:

[1] In machine learning, ground truth refers to checking the results of machine learning against reality.

$$\begin{aligned} PSNR &= 20\log_{10}\frac{L^2}{HW}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}[I(i,j)-\hat{I}(i,j)]^2 \\ &= 20\log_{10}\frac{L^2}{MSE} \end{aligned} \tag{3.2}$$

PSNR uses MSE equation to calculate peak signal power. L is the dynamic range of the image pixels values ($L = 2^B - 1$, B stands for bits per sample). PSNR is used to compare images with different dynamic ranges. Otherwise, it does not contain any more new information than MSE.

3.3.3 Structural Similarity Index (SSIM)

Structural similarity index is considered to have strong correlations to the way the human visual system perceives quality. The method uses three factors to model any image distortion: loss of correlation, luminance distortion, and contrast distortion. SSIM calculates the similarity between image structures instead of an absolute error like MSE and PSNR do. [41, 17]

The SSIM is calculated on several windows of an image. The formula of SSIM is defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{3.3}$$

where μ_x an μ_y are average values of x and y. σ_x^2 stands for variance of x, and σ_y^2 is variance of y. σ_{xy} denote as covariance of x and y. Variables $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$, where L stands for dynamic range of the image pixel values, also mentioned in PSNR. Lastly, for constants k_1 and k_2, values $k_1 = 0.01, k_2 = 0.03$ are used as default.

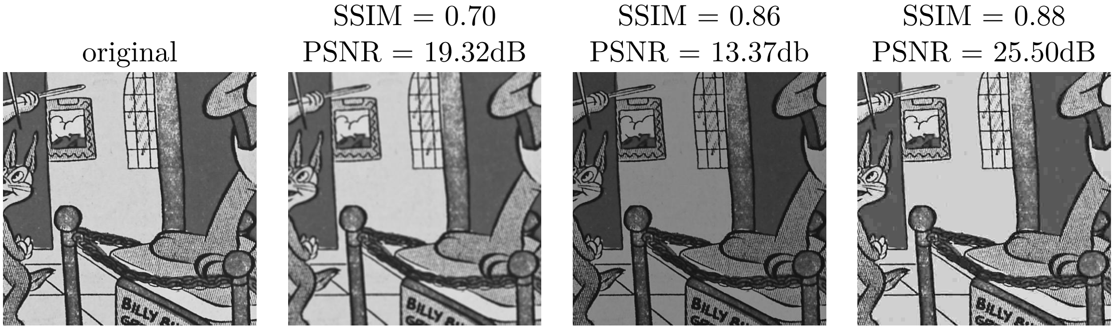


Figure 3.3: An overview of applied effects from left: blurriness, brightness adjustment, JPEG compression, and their effect on the image observed by the SSIM and PSNR metrics.

Chapter 4

State-of-the-Art Methods for Super-resolution using Deep Neural Networks

The super-resolution task is a challenging ill-posed problem of reconstructing a high-resolution image from the low-resolution one. This process should not be confused with similar techniques, such as interpolation or image restoration. Unlike super-resolution, interpolation-based methods upscale the image but do not restore the details. Techniques such as deblurring or sharpening produce the result of improved quality but the same size. In super-resolution, the details are improved, and also the output size is increased. SR has applications in many fields, such as medical image processing [20, 45, 11], satellite and aerial imaging [44, 46, 28], improvement of text images [36, 4], and reading of numbers on the plate [29, 37].

This thesis focuses on single image super-resolution (SISR) [12] without further examination of techniques for restoring an image from multiple images [8]. The topic of SR was studied for the first time by Tsai in his work [38] in 1984. Over time, several algorithms have been proposed for SISR, which can be categorized into a few types: prediction based methods [22, 6] that comes with filtering approaches (e.g. linear or bicubic), edge-based methods [9], statistical methods [24] and patch-based methods [2, 10].

Recent advancements in deep learning methods have allowed convolution neural networks (CNNs) modified for super-resolution tasks to develop and demonstrate excellent results in state-of-the-art performance. For example, UnetSR [27], Dense U-Net [26], or RUNet [18]. The development has progressed to the use of generative adversarial networks (e.g. SRGAN [25]). The main differences between deep learning methods are in network architectures [23, 1], different use of loss functions [19, 27], and different types of learning types [16, 39].

This chapter describes state-of-the-art methods used for super-resolution tasks. Firstly, methods of learning-based upscaling are discussed followed by the basic super-resolution approaches used in various architectures. Furthermore, the chapter talks about the convolutional neural networks, mainly the U-Net, and the modifications for the super-resolution. And lastly, the generative adversarial networks are briefly introduced.

4.1 Learning-based Upsampling

Although traditional upsampling methods are well known and easy to implement, learning-based upsampling methods are getting more popular and effective in the deep learning field. These methods (e.g. sub-pixel layer, transposed convolution, or unpooling upsampling) are used to create fully connected convolutional networks. That means the end-to-end mapping of a low-resolution image to high-resolution are also included in the learning process.

Transposed Convolution Layer

Transposed convolution – also known as fractionally strided convolution or deconvolution (however, deconvolution can be mathematically defined as the inverse of a convolution, which is different from a transposed convolution, so this term should be avoided in the context of transposed convolution) attempts to perform the opposite transformation as normal convolution[1]. The method uses feature maps to predict inputs similar to convolution outputs. By inserting zeros and performing convolution, it increases the image resolution. [7] One of the disadvantages of the transposed convolution is that it first needs to learn the optimal kernel weights, and only then it is suitable for the application.

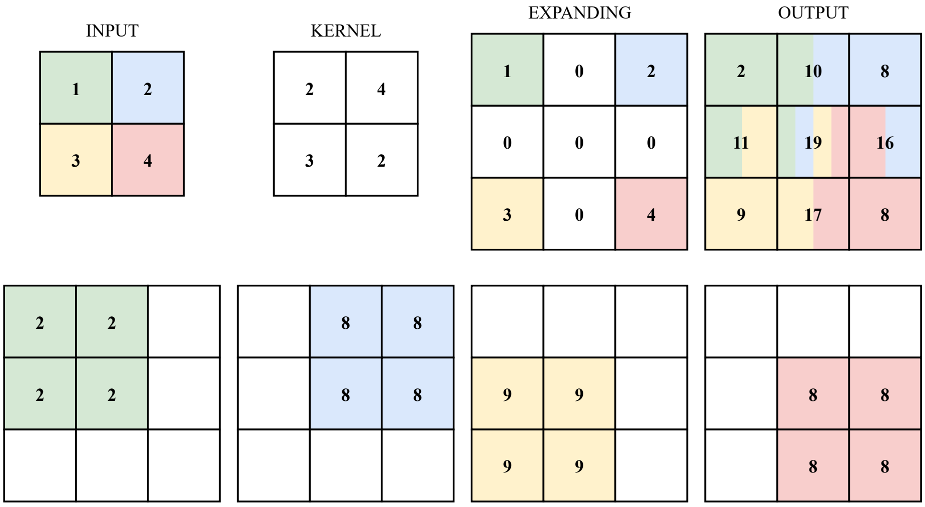


Figure 4.1: Figure shows an example of how an algorithm works. A convolutionon kernel with size 2 × 2 is used, with stride 1 and padding 0. Input is expanded to a 3 × 3 matrix and filled empty space with zeros. Afterward, convolution is applied. The result is an upscaled input matrix.

Sub-Pixel Layer

The sub-pixel layer performs upscaling by generating channels by a convolutional layer and applies a sub-pixel layer (Pixel shuffle) to reshape them. Firstly, a convolutional layer is used to obtain an image with the r^2 number of channels (r stands for upscale factor). Each pixel in r^2 channels corresponds to a sub-pixel block of size $r \times r$ in I^{HR}. Then, a sub-pixel layer is applied to the input shape $H \times W \times Cr^2$, to obtain an upscaled image with the final shape $rH \times rW \times C$. [43] The process is illustrated in figure 4.2.

[1] https://mathworld.wolfram.com/Convolution.html

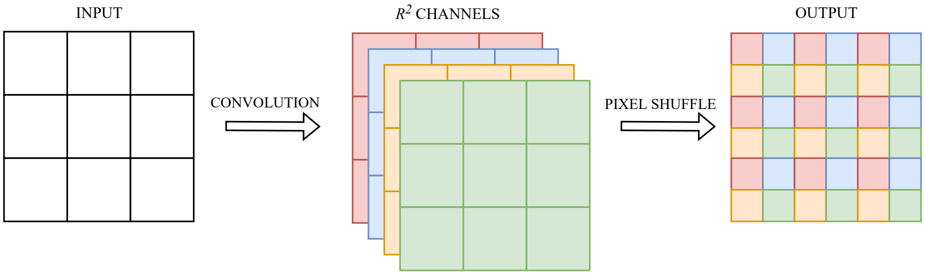


Figure 4.2: The input of 3×3 pixels, with upscale factor, $r = 2$ is used to represent the process of upscaling an image by sub-pixel layer (a.k.a. Pixel shuffle). By using convolution, 4 channel (red, blue, orange and green) are obtained from the given input. Those obtained channels are used to create an upscaled input by extracting pixels from each channel and combining them into the sub-pixel blocks in the output. After the process, the output size is 9×9 pixels.

The sub-pixel layer technique is popular in super-resolution models [33, 5, 47, 35], and also is used in Robust UNet (RUNet) architecture [18], later mentioned in the section 4.3.5.

4.2 Super-resolution Basic Approaches

There are a lot of various architectures for solving super-resolution problems used today. Based on the chosen upsampling operation and the location of the operation in the model, they can be categorized using four basic approaches, which are described with visualizations (figures 4.3, 4.4, 4.5, 4.6). More detailed information can be found in the paper [40], with examples of architectures for each method.

4.2.1 Pre-Upsampling

Pre-upsampling uses basic interpolation methods (e.g. bicubic interpolation) to firstly obtain a „coarse“ HR image from an LR image. Then, convolutional neural networks (CNNs) are used to learn an end-to-end mapping from the LR image to the HR image. The most challenging operation to increase the resolution is performed in the beginning by the known interpolation method, and the other part of refining the image is done by CNN. By accordingly creating an HR image, the difficulty of learning requirements of CNNs is reduced. Furthermore, the advantage of the pre-upsampling method is that the input image of any size with a given scale factor can yield refined results with comparable performance to single-scale super-resolution models.

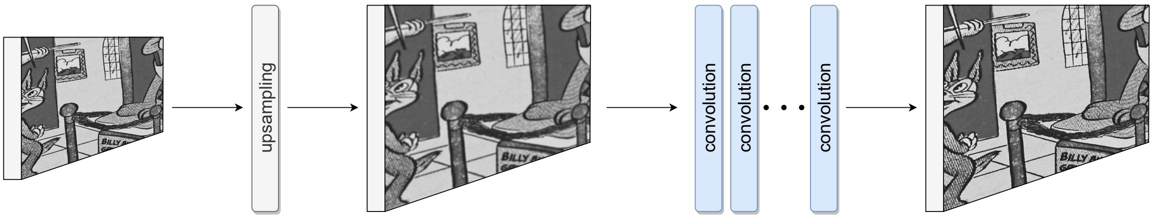


Figure 4.3: Example of classic **pre-upsampling** network with predefined upsampling method. Inspired from [40].

4.2.2 Post-Upsampling

The post-upsampling method is processed in the opposite direction as pre-upsampling. In the beginning, feature extraction is performed on low-resolution input, which is later upscaled by the chosen interpolation method. The procedure reduces the computational complexity of CNNs by performing operations on LR input rather than HR. Models based on the post-upsampling method also can be trained end-to-end if the learnable upsampling layer is used (e.g. Transposed Convolution Layer).

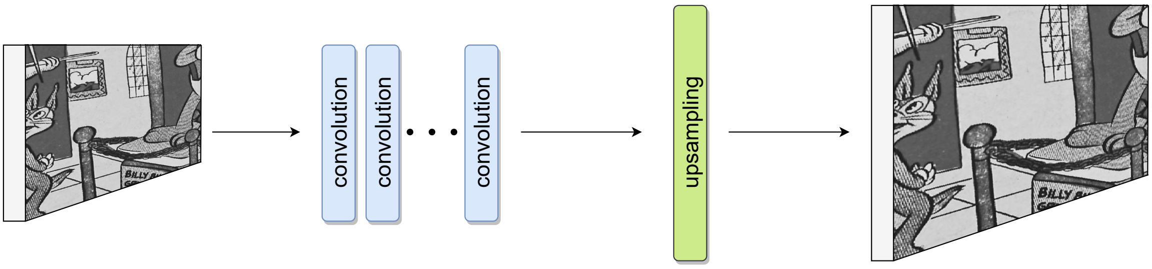


Figure 4.4: Example of classic **post-upsampling** network. Inspired from [40].

4.2.3 Progressive Upsampling

The post-upsampling method reduced computational complexity but brought some other disadvantages. The progressive upsampling method attempts to overcome these issues. In the post-upsampling model, the upsampling is performed only once, which means that the computational complexity is reduced. On the other hand, the learning difficulty increases for larger scaling factors. The progressive upsampling method decomposes the task into smaller tasks (steps), which caused a reduction in computational complexity. In progressive upsampling, CNNs are used to progressively reconstruct HR images at smaller scaling factors in each step.

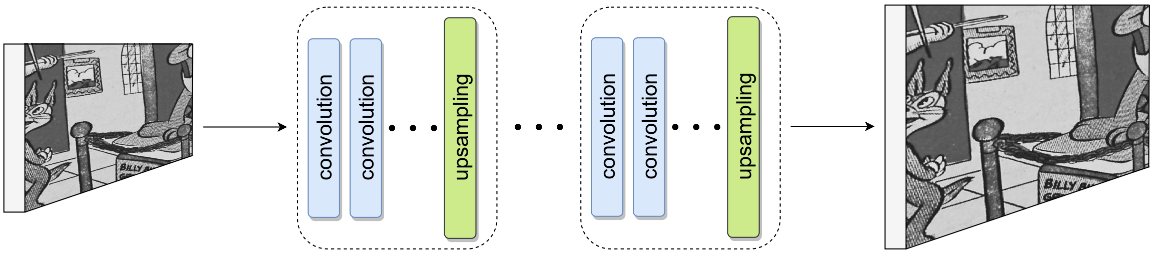


Figure 4.5: Example of classic **progressive upsampling** network. Inspired from [40].

4.2.4 Iterative Up and Down Sampling

The method involves alternating between upsampling and downsampling in order to identify better the deep relationships between the LR and HR image pairs, which allows for more accurate reconstruction. This iterative procedure, called back-projection, is becoming more popular in super-resolution networks, e.g. [14, 15]. The main point of the back-projection approach is to capture features at various different resolutions that could help with the reconstruction process.

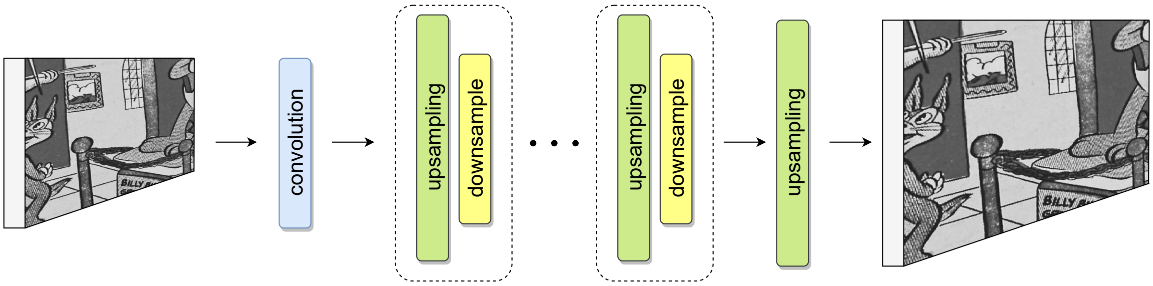


Figure 4.6: Example of classic **iterative up and down sampling** network. Inspired from [40].

4.3 U-Net Convolutional Neural Network

Ronneberger *et al.* [31] presented an architecture that resembles a U-shape, which is designed to segment biomedical images. It was only the initial purpose of the network, but as was mentioned in the conclusion of the U-Net paper, it could be easily applied to many other tasks, including super-resolution.

4.3.1 U-Net Architecture

As shown in Figure 4.7, the network architecture is composed of two parts: a contracting path on the left side, also known as the encoder, and an expansive path on the right side, also called the decoder. Based on the typical architecture of a convolutional network, the contracting path follows the same pattern. The contracting path consists of two 3×3 unpadded convolutions repeatedly applied, each followed by a Rectified Linear Unit (ReLU) activation function and downsampling operation provided by a 2×2 max pooling operation with stride 2. Each downsampling step doubles the number of feature channels. The expansive path performs an upsampling operation, followed by a 2×2 convolution („up-convolution") that halves the number of feature channels. Subsequently, a concatenation is performed with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each followed by a ReLU activation function. The final step is a 1×1 convolution that maps 64 feature channels to a desired number of classes.

4.3.2 Super-Resolution Using a U-Net network

U-Net architecture has quickly become popular, and over time, various U-Net modifications have been proposed. The modifications that touch upon the super-resolution field were presented, along with some new architectures based on U-Net. Among the U-Net based architectures, the ones worth mentioning are: UnetSR network [27], improved version Dense U-net Network [26], and Robust UNet [18] which uses residual blocks technique [16].

4.3.3 UnetSR Network

Lu *et al.* [27] proposed an improved network called Modified U-Net (**UnetSR**), and a mixed gradient loss function. The architecture is shown in Figure 4.8. There are three differences to the original U-Net architecture:

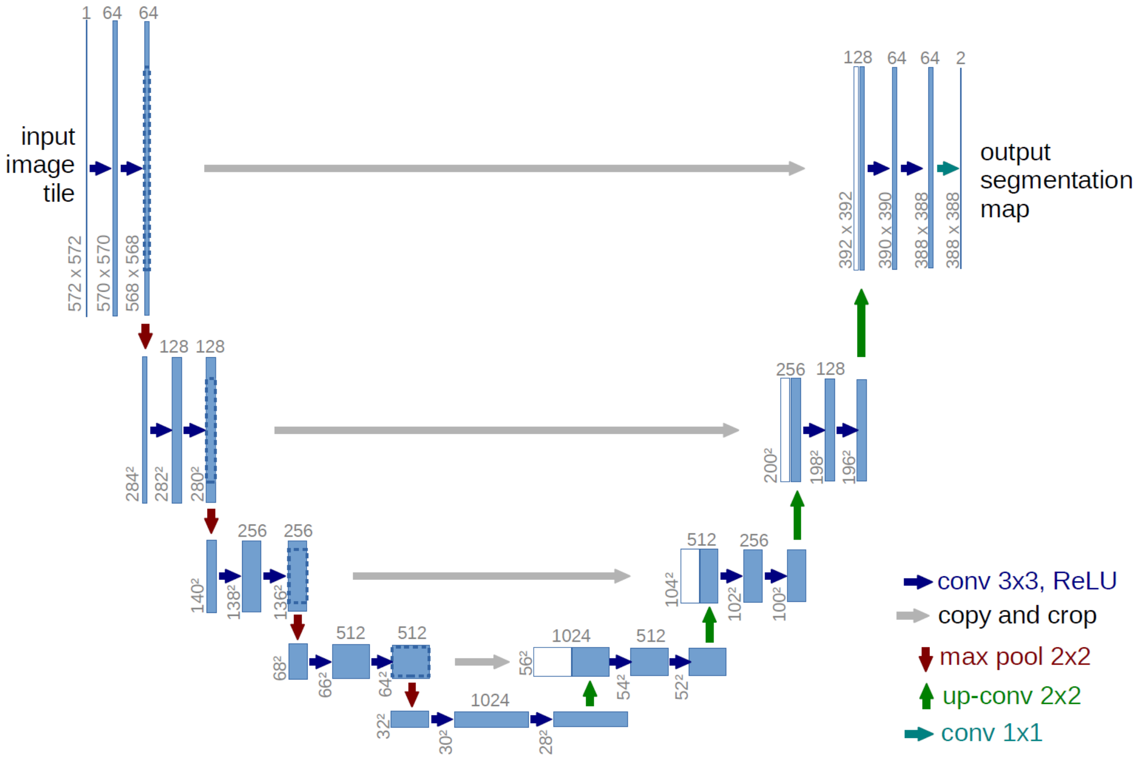


Figure 4.7: Visualization of U-Net architecture. The architecture consists of two parts: a contracting path and an expensive path. The number above the convolutional blocks denotes the number of features, and the bottom-left number denotes the input resolution size. In the original paper, biomedical images with a resolution of 572 × 572 were used as input for the model. The output consists of a 388 × 388 resolution segmentation map. The legend describes individual operations used in the U-Net architecture. Figure was adapted from [31].

1. *In each block, one of the convolution layers is removed.* Because the basic block does not require too complex mapping for low-dimensional tasks, it is reasonable to drop one convolution layer in each block.

2. *The input image is upscaled to the required output size and builds a new convolution block on a larger scale.* The newly built block has a skip connection with the output block on the same scale.

3. *The depth of the modified U-net architecture is set to four.* This means that the architecture contains four down-scaling blocks and four up-scale blocks. The depth was chosen based on computation cost to reconstruction accuracy.

For training, mixed gradient error was used (modified MSE), as described in the original paper. The evaluation method for this architecture was PSNR (3.3.2). The results proved that the network achieved a high value of PSNR with the trade-off of the depth of 5.

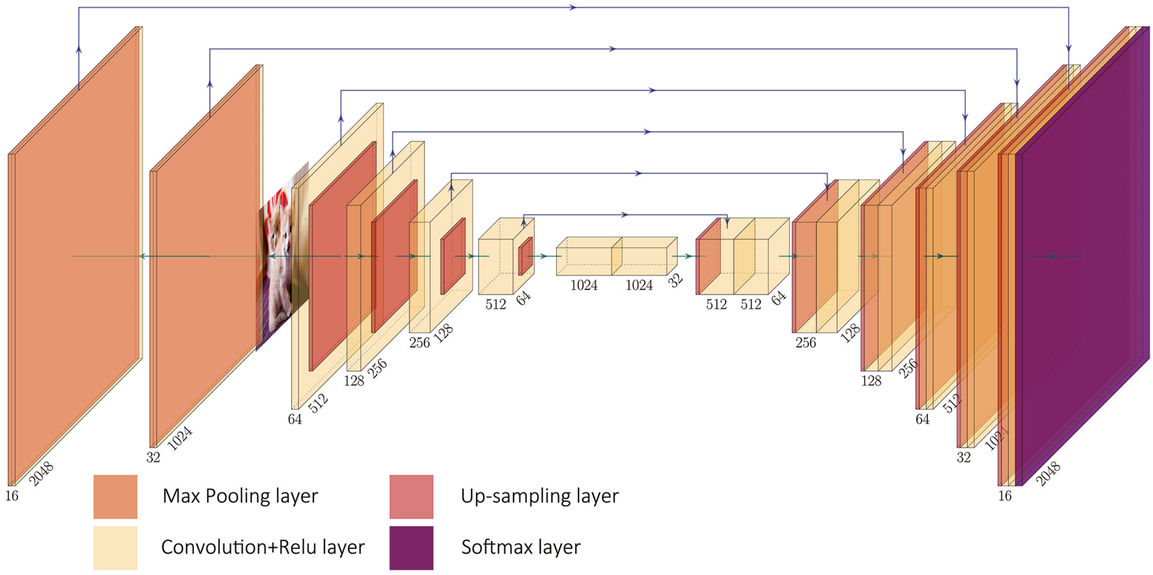


Figure 4.8: Visualization of the modified U-Net architecture. The arrows above blocks denote skip connections between blocks on the contracting and the expanding path (concatenation is performed with the corresponding feature map from the contracting path). The legend assigns the used operation to each colored block. Figure adapted from [27].

4.3.4 Dense U-net Network

Another modification of the original U-Net architecture is Dense U-Net [26]. Three main components make up the network: the upscale path which is used for pre-upsampling input, the contracting path for the feature extraction, and the expanding path for the image reconstruction. The Dense U-Net is an improved version of UnetSR architecture, enriched by dense blocks.

Network Architecture

Figure 4.9 illustrates the network architecture, which consists of 4 parts:

- **The contracting path** contains blocks made up of one convolution layer with a 3x3 kernel continued by ReLU layer, followed by max-pooling operation with 2×2 kernel and stride 2 used for downsampling. The modification to the original U-Net is the same as UnetSR. In each block, one of the convolution layers is dropped. The difference to the basic U-Net in this part is the improvement of the downsampling method. Instead of the basic max-pooling layer, the new shuffle pooling method doubles the feature every step instead of using a convolution layer (the method is described in [26]).

- **The expanding path** is made up of blocks of upsampling layer, followed by a convolution with a 2×2 kernel, which halves the number of features. Subsequently, one convolution layer with a 3×3 kernel is applied. In the end, the ReLU activation function layer is performed.

- **The upscale path** consists of an upsample layer, followed by a convolutional layer and a ReLU layer. The path is used to preserve the depth of the contracting path and the expanding path, and to build the symmetric feature extraction layer corresponding to the upsampling layer in the same depth as the expanding path part. The upsample layer performs bicubic interpolation (sec. 3.2.1).

- **The dense skip connection** part is the main modification of the network. Connections transfer feature maps from the blocks in all depths of the contracting path to blocks in the expanding path. The idea is to establish a multi-path data transmission and reduce the information transmission loss. The upsampling block on the expanding path gets information from essentially all combined feature maps instead of getting them from only the on-way downsampling path.

Along with that architecture, a mix loss function was presented. That loss function combines Mean square error, Structural Similarity Index, and Mean Gradient Error. In summary, the network reduces the information transmission loss because of the dense skip connection used. Furthermore, the paper [26] mentions in its conclusion that the network outperforms the state-of-the-art methods in Single Image Super-resolution fields (SET14, BSD300, ICDAR2003 datasets), especially in the text-based tasks.

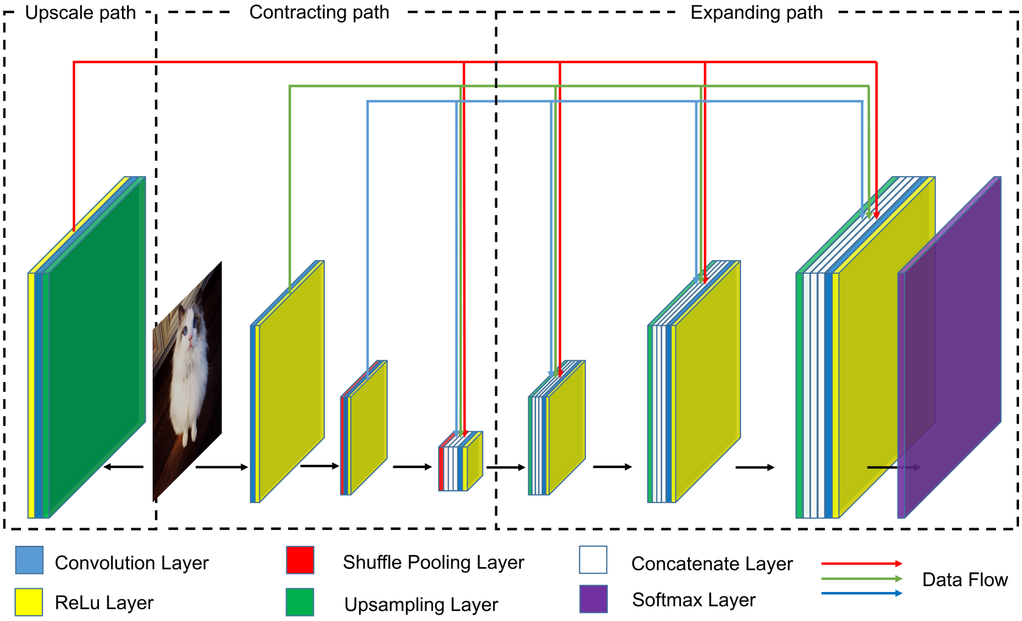


Figure 4.9: Visualization of Dense U-net architecture. The figures show the network divided into 4 parts. The depth of the network is 3 and the upscale factor used is 2. Legend denotes operations used and data flow of dense skip connection part. Figure adapted from [26].

4.3.5 Robust U-Net Network

Hu *et al.* [18] presented a new U-Net architecture that can learn the relationship between degraded low-resolution images and their corresponding original high-resolution images. Experimental results show that the presented RUNet enhances the visual quality of the obtained super-resolution images while it does not produce a significant reconstruction error.

Network Architecture

The used methods contain the network architecture shown in Figure 4.10, and the degradation module which prepares input images.

The degradation module has two modes, one for training and another one for testing. The network consists of contracting (decoding) and expanding (encoding) path. The contracting path in each depth contains a sequence of blocks. The block is made up of a convolutional layer, followed by batch normalization, ReLU activation function, another convolutional layer, and another batch normalization. The next block in the sequence gets to input the addition of output from the previous block and input from the subsequent block. An expanding path block consists of a batch normalization layer, a convolution layer, a ReLU, another convolution, and two ReLU activation functions. Expanding path uses the sub-pixel layer for upsampling. Then the next block gets input from sub-pixel layer output stacked with skip connection from a specific depth. In training mode, the downscaling bicubic operation is applied to the high-resolution image. The downscale factor is set to two. For better visualization, see Figure 4.10. Following the downscaling is a random blur using Gaussian filter operation. Afterwards, the blurred image is upscaled to its original size, using bicubic interpolation with a scale factor of two. In the testing mode, a low-resolution image is upscaled by a scale factor of two. Afterwards, a trained network is used to obtain the resulting super-resolution image.

The RUNet architecture consists of convolution layers, batch normalizations, ReLU (rectified linear unit) activation functions, tensor operations, max-pooling for downsampling and a sub-pixel convolutional layer to obtain efficient upscaling. For comparison to the U-Net architecture, the RUNet contracting path consists of a sequence block, each followed by a tensor addition to create the residual block [16]. The network uses the Perceptual loss function in a training phase for better perceptual performance.

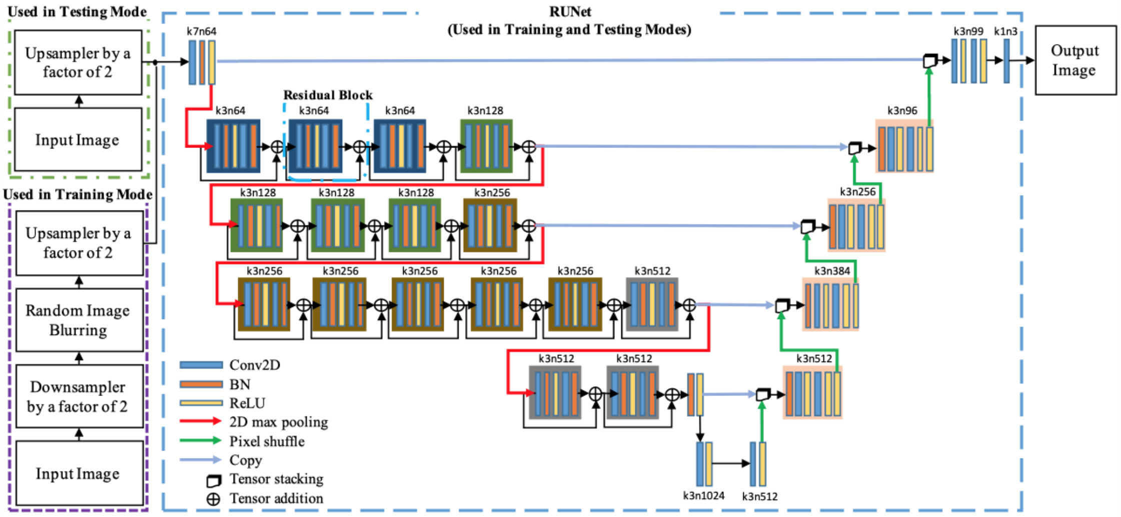


Figure 4.10: Visualization of Robust U-net architecture and Degradation module. Figure adapted from [18].

Perceptual Loss

The Perceptual loss function was presented in [19], to measure perceptual differences in content between images. The pre-trained loss network map predicted super-resolution image and high-resolution ground truth image to feature space. Then, the Perceptual loss calculates loos by measuring the distance between mapped images. For the mapping step, it is possible to extract a variable number of convolution layers from the loss network. The RUNet architecture in the mentioned paper uses five convolutional layers from the pre-trained loss network. The process of mapping images to feature space and calculating the loss is described in detail in [19] and [18].

The usage of the Perceptual loss function leads to worse results for SSIM and PSNR evaluation metrics. However, super-resolution results from RUNet architecture provide improved quality with noticeably sharper details. The paper [18] suggests the need to developing new evaluation metrics for super-resolution tasks. The reason is to get proper results when using Perceptual loss function.

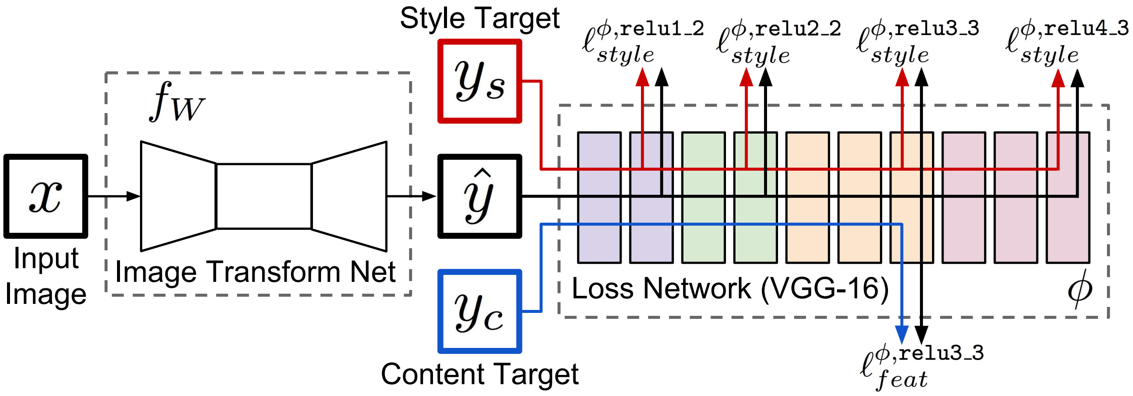

Style Target
$\ell_{style}^{\phi,relu1\_2}$
$\ell_{style}^{\phi,relu2\_2}$
$\ell_{style}^{\phi,relu3\_3}$
$\ell_{style}^{\phi,relu4\_3}$
y_s
f_W
x
$\hat{y}$
Input Image
Image Transform Net
y_c
Loss Network (VGG-16)
ϕ
$\ell_{feat}^{\phi,relu3\_3}$
Content Target

Figure 4.11: Figure shows on the left side the network used to transform input images into output images. The right side contains a pre-trained loss network, which defines Perceptual loss functions that measures perceptual differences in the content between images. Figure adapted from [19].

4.4 Generative Adversarial Networks

The generative adversarial networks, in comparison to CNNs, generate more appealing results for a human observer, as is shown in Figure 4.12. The basic architecture of the generative adversarial network consists of two models. Firstly, the generative model produces super-resolution images. And secondly, the discriminator model is trained to distinguish real images from the produced ones. The main idea is to train the generative model to produce images that can fool the discriminator model.

4.4.1 Super-resolution GAN-based network

Ledig *et al.* [25] proposed SRGAN, a GAN-based network for super-resolution optimized for a Perceptual loss. The network architecture of SRGAN model is composed of 2 networks, as shown in Figure 4.13.

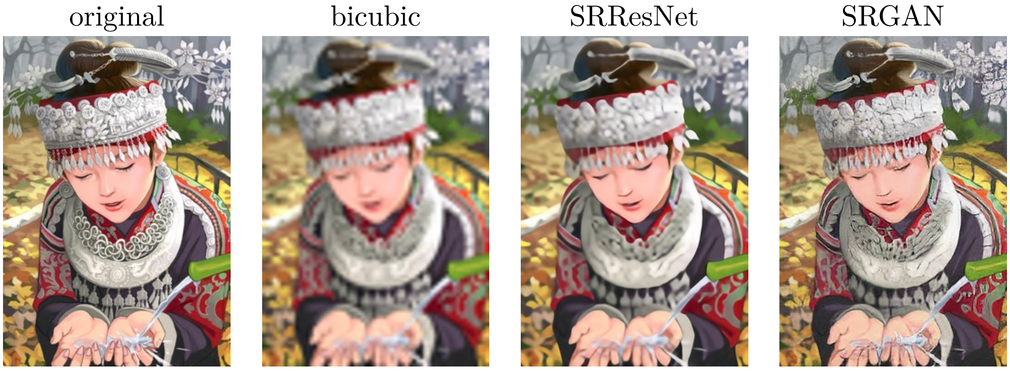


Figure 4.12: The preview of original and super-resolution obtained images from various techniques. The significantly better visual quality preformed the SRGAN model. Adapted from [25].

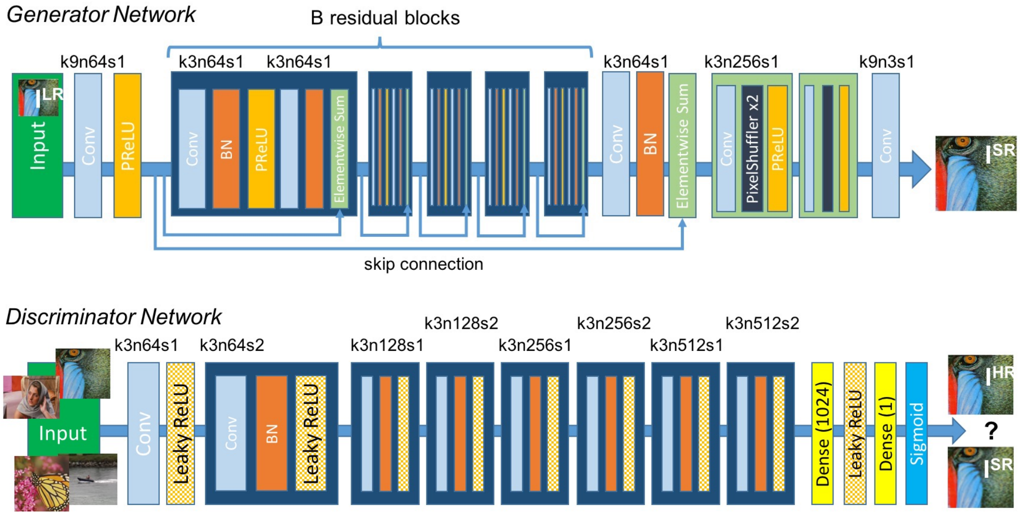


Figure 4.13: Architecture of Generator and Discriminator Network. The annotation above each convolution layer denotes the number of kernels (k) with the number of features (n) and step of stride (s). Adapted from [25].

A generator network is used to generate super-resolution images similar to real ones. The core of the generator network is made out of residual blocks with an identical layout. The residual block contains two convolutional layers with a 3x3 kernel and 64 feature maps, followed by a batch-normalization and ReLU activation function. For upsampling of input, two trained sub-pixel layers are used. To discriminate real images from generated ones, a discriminator network is used. The architecture uses the LeakyRelu activation function and avoids the max-pooling layer; therefore, strided convolutions perform downsampling.

The discriminator network consists of eight convolution layers with 3×3 kernel with an increasing number of features. Each block increases the number of features by a factor of 2 from 64 to 512 features. Each time the number of features is doubled, downsampling is performed. To obtain a sample classification, two dense layers followed by a sigmoid activation function are applied to the resulting 512 feature maps.

Chapter 5

Proposed Solution for Comic Images Super-Resolution using Deep Neural Networks

The comic-book images and illustrations have a unique composition. The combination of colourful drawings and white bubbles with text in them provides a somewhat tricky problem for the current methods of the super-resolution process. Neural networks have trouble with upscaling text inside white bubbles, which themselves are also located within hand-drawn pictures with colours and patterns. The same problem could also be extended to the world of video games. Graphic cards perform super-resolution on game scenes in video games, but they take the text out and perform super-resolution on the text and image content separately.

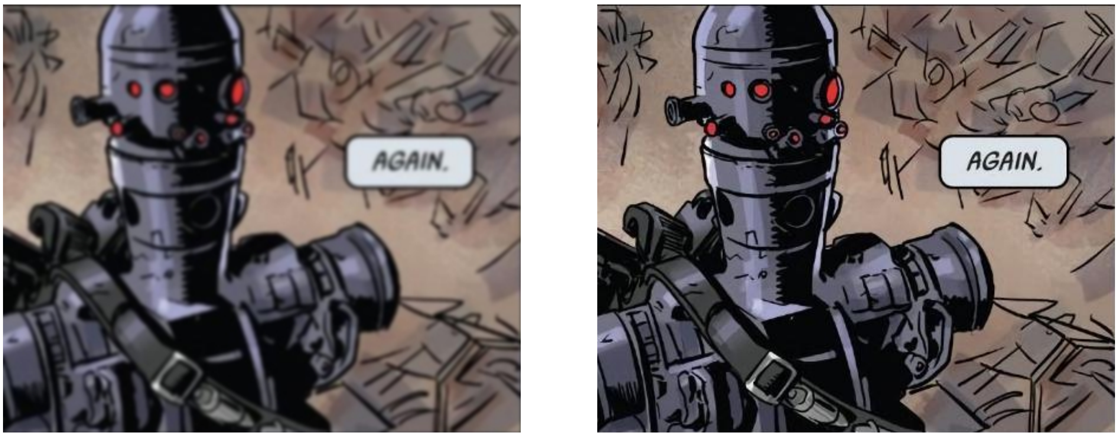
This work tries to approach the problem for both the text and the image simultaneously, which is also a part of the reasoning behind choosing the suitable architecture. It was important to focus on architectures proven to work with basic super-resolution. Additionally, new architectures are also explored with an effort to try to come up with some combination of a common and proven architecture and a newer, experimental one, possibly with a few modifications.

The chapter contains the proposed solution for the super-resolution process on comics images using deep neural networks. The first section defines the problem and suggests the main goal for the thesis to achieve. Afterwards, the dataset of comic images is presented, and a description of how to obtain, filter, and process them. Then the chapter talks about the chosen U-Net and RUNet architecture for this task, along with MSE and Perceptual loss function. In the end, the web application for image upscaling using the final trained model is proposed.

5.1 Task Definition and Main Goal

Super-resolution is an ill-posed problem of obtaining a high-resolution image from a low-resolution input. The goal in this thesis is to produce visually appealing and high-quality upscaled comic images. The important thing here was to increase the quality by approximately the same level for both images and text to avoid disproportional changes in quality. The dataset was also carefully chosen with the mindset of avoiding disproportionality, with further details described later in section 5.2.

Additional goal is to outperform the basic interpolation methods. Achieving better quality metrics of the resulting image is needed, and it is also very important to get the best possible visual impression of the observer. Figure 5.1 illustrates the expected results of the mentioned task. The chosen approach in regard to achieving the main goal of the thesis is to use the chosen architecture to increase the quality and leave the upscaling for the bicubic interpolation method.



(a) Example of an input image

(b) Example of an upsampled output image

Figure 5.1: Demonstration of the expected results

5.2 Preparing a Dataset of Images

A very important aspect of any deep learning task is to have a suitable working dataset. Preparing a set of suitable images that satisfy given criteria is essential for evaluating the success of the achieved results. In order to obtain the best possible dataset of suitable images, some criteria need to be established.

Firstly, the ratio between text area and drawings has to be preserved. An ideal image (a page from a comic book) for this dataset would consist of multiple blocks, each with a similar composition. That means that text and drawings parts would occupy the whole image with the same size. However, it is hard to obtain a large enough dataset of pictures with such ideal compositions. That is why the used criterion was to have a balanced ratio of text to images across a page. E.g. one block would contain only text, another block would be just an image (e.g. establishing shots of the scene), and then some blocks with a combination of text and image. Such a page would then be considered usable for training purposes. The final obtained dataset of such pages would create an ideal ratio of 1:1 for text and drawn parts. Not ensuring the balance between text parts and drawing parts leads to the neural network only learning better upscale the prevailing parts of the image, which is not desirable.

Secondly, a very important criterion is to have the images in digital form. Scanned images would also need to be properly digitized. Such images do not suffer from imperfections caused by poor scanning. They need to be precise and without any external deformations. The images from the dataset also need to be of high quality.

Another thing that had to be considered was the variety of comic-book styles and genres. The used style encompasses, among other things, drawing techniques, different curvatures of particular strokes (sharper vs rounder curves), character designs, choices of coloring and patterns, etc. The thesis focuses primarily on superhero comics, thanks to their popularity and accessibility. It was the best choice to get a large enough dataset of good quality images.

The dataset used for the task presented in this work was collected and made from the **getcomics.info**[1] site. The dataset consists of manually selected modern comic high-quality images. The set used for validation and the set used for testing are of the same genre and style. These images are supposed to be the ground truth for our model. Figure 5.2 shows the samples from the used dataset.

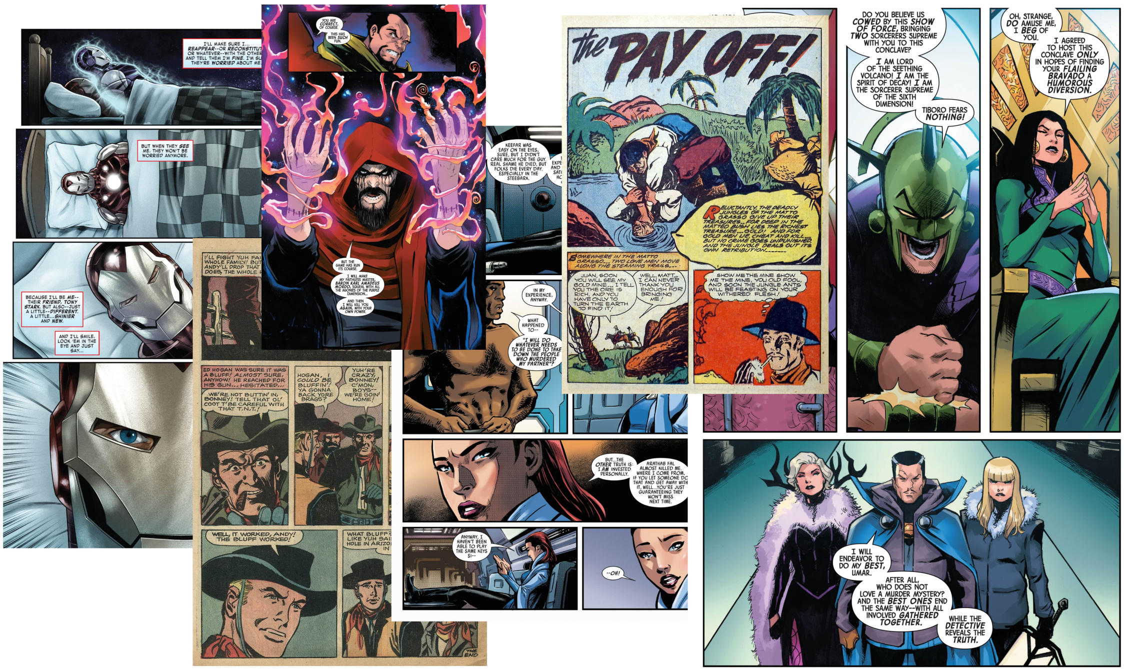

the PAY OFF!
DO YOU BELIEVE US COWED BY THIS SHOW OF FORCE, BRINGING TWO SORCERERS SUPREME WITH YOU TO THIS CONCLAVE?
I AM LORD OF THE SEETHING VOLCANO! I AM THE SPIRIT OF DECAY! I AM THE SORCERER SUPREME OF THE SIXTH DIMENSION!
TIBORO FEARS NOTHING!
OH, STRANGE, DO AMUSE ME, I BEG OF YOU.
I AGREED TO HOST THIS CONCLAVE ONLY IN HOPES OF FINDING YOUR FLAILING BRAVADO A HUMOROUS DIVERSION.
I WILL ENDEAVOR TO DO MY BEST, UMAR.
AFTER ALL, WHO DOES NOT LOVE A MURDER MYSTERY? AND THE BEST ONES END THE SAME WAY--WITH ALL INVOLVED GATHERED TOGETHER.
WHILE THE DETECTIVE REVEALS THE TRUTH.

Figure 5.2: Figure shows randomly selected images from the dataset.

The images are chosen from different issues of different comic book series. The samples from the dataset are divided into three sets.

1. The first set of images is used for training. The images were chosen such that they have an approximately equal proportion of text area as image area. Training set is approximately ten times larger than the validation set. So for ten images from training set, there is one image from validation set. The training set contains around **2000 images**.

2. The second set is for validation. It contains images from the same sources as the first set. The images have approximately the same composition, color palettes, but are different from any other image from the training set. The images are different, which ensures that the neural network has never seen or worked with these images, but at the same timel, makes sure that the images follow the same structure like the ones that it was trained on. The validation set contains approximately **200 images**.

[1] https://getcomics.info/

3. The last group of images is for testing. It includes a **few images** with special and different properties (edge cases). For example, an image that is a big text bubble with a minimal amount of drawings, or an image that contains no text whatsoever and is only a picture. Lastly, it also includes images with similar compositions as the training and validation set.

5.3 Dataset Preprocessing and Augmentation

One issue with the prepared dataset was its variability in resolutions and formats. In order to work with the later mentioned architectures (5.4), the images need to be uniform in some of their properties. This means that before they are used as an input to the architecture, they need to be further processed. Because of the hardware restraints, it was necessary to reduce the size of the used images. 128×128 resolution was used. It is small enough to satisfy the hardware and time restraints but large enough to contain still some level of detail from the original image. Randomly cropped 128×128 image segments were then used in different epochs of training (for each epoch, a different segment).

Another necessary step of preprocessing was normalization. For color images, the pixels were normalized to 8-bit colors (0-255 for each RGB channel). For training, grayscale images were used. The reason was to speed up the training process. Once the training was done, and the results were deemed sufficient on grayscale, then the work began on colored images.

For training and validation, both high-resolution and low-resolution images were needed. Initially, the dataset contained only high-resolution images (ground truth). It was therefore needed to arbitrarily obtain a low-resolution version of the dataset images. For getting the low-resolution images, the degradation module from RUNet paper [18] was used. The original (large) picture was shrunk by some scale factor. Then, a random Gaussian blur was applied, followed by adding some noise. The blurred and small image was then enlarged back to the original resolution.

In order to get better results during training, the number of samples needed to be increased. The augmentation is used for arbitrarily expanding the size of the dataset. It creates seemingly new inputs for the neural network using image transformations such as:

- random cropping of the image,
- random horizontal flipping,
- random vertical flipping,
- rotating the image,
- color changing (hue, saturation, lightness), etc.

Augmentation ensures that the dataset is larger and that the training process and subsequent validation yield a more precise product. The process of dataset augmentation only runs in the program, so the output generated from augmentation was not stored on a disk. The particular augmentation details are explained with their corresponding experiments in chapter 7.

5.4 Used Architectures

To obtain super-resolution images, two networks were used. The first architecture is the original U-Net architecture, utilising a degradation module from RUNet architecture. The second (primarily used) architecture is RUNet which uses the Perceptual loss function.

U-Net is the most well-known architecture (see 4.3). It was used for segmentation, but its applications are suited for many more things [31]. The vast possibilities of U-Net applications is why it was chosen for the comic-book super-resolution task to examine what kind of results it can produce. It also uses the degradation module from RUNet to get low-resolution inputs for training. Figure 5.3 shows the proposed setup for the U-Net architecture used for this task.

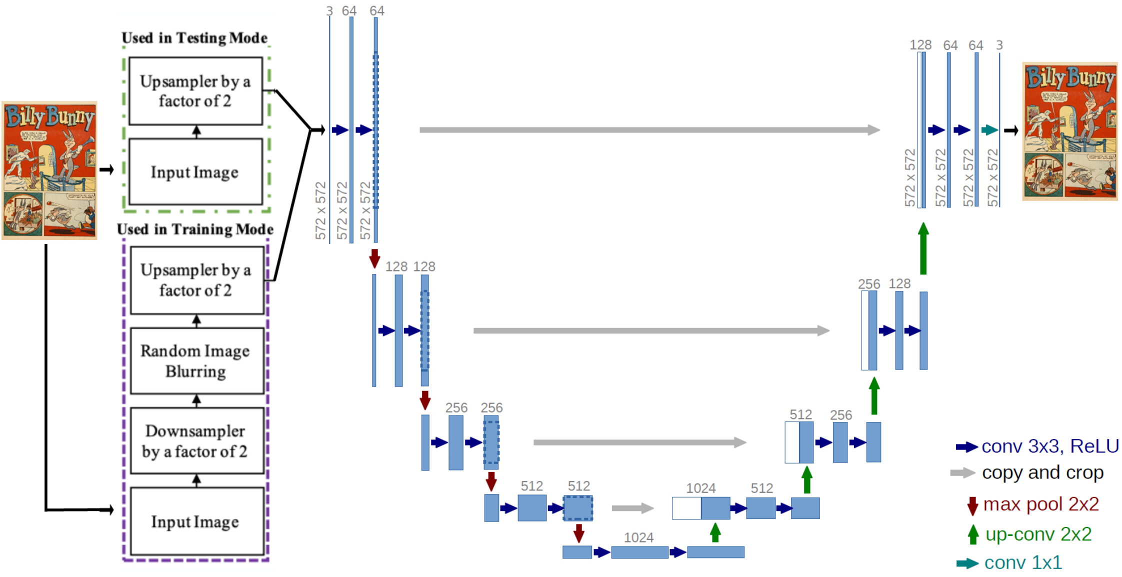


Figure 5.3: U-Net with a degradation module. Adapted from [31, 18] and edited.

RUNet was also used, thanks to the fact that it is fairly new architecture [18] and has achieved good results with regard to perceptual sight. RUNet uses Perceptual loss function and a residual block technique, which proved to have a better impact on learning more complex structures. Residual block technique uses information from the previous input in combination with the current output for the next operation in sequence.

5.5 Used Loss Functions

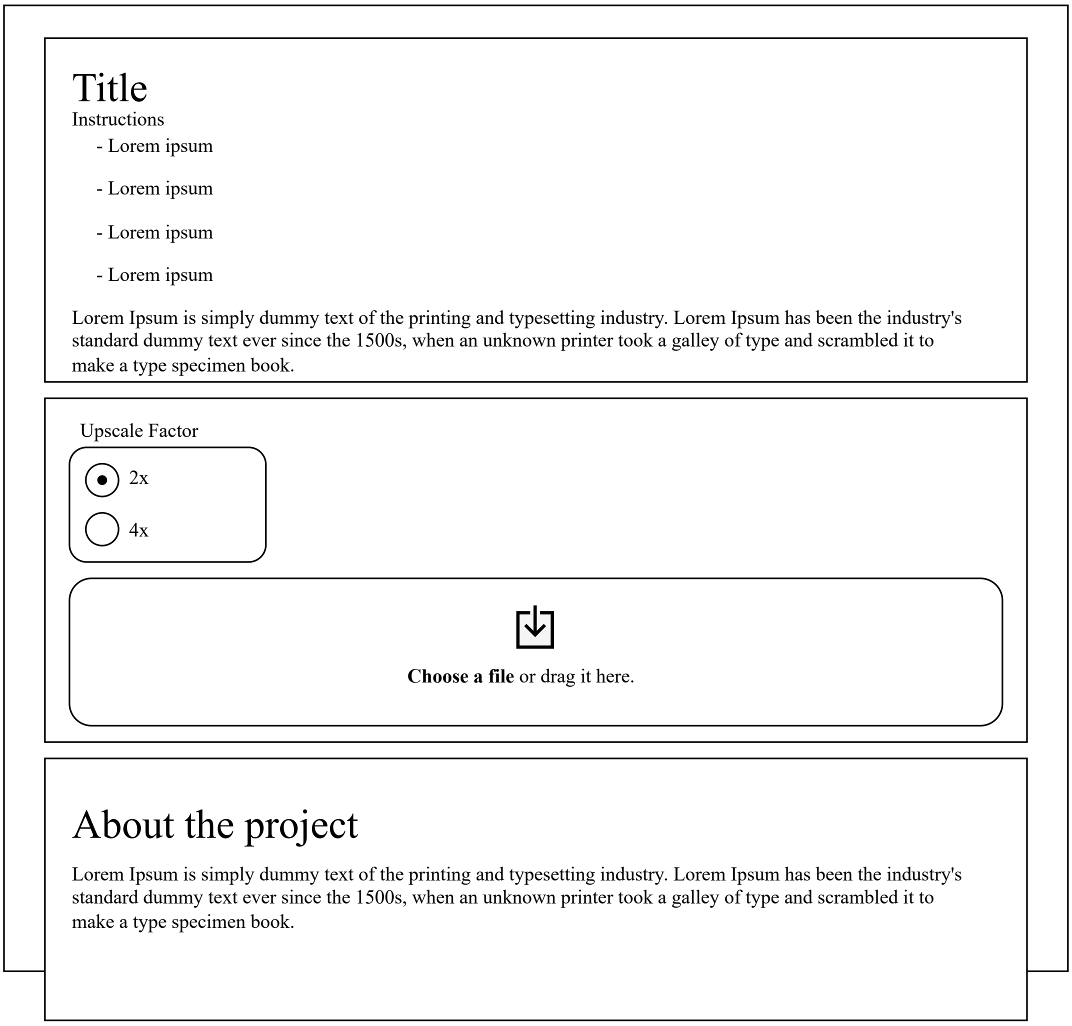
Choosing the proper loss function is crucial in guiding models towards optimum performance. During the training of selected architectures, Mean squared error (see 3.3.1) and Perceptual loss function (see 4.3.5) were used. MSE is a standard loss function utilized in many machines learning tasks. In short, it serves to compare pixel-color differences of the images. The Perceptual loss is more popular for super-resolution tasks, as it often provides more accurate results. It is used to compare images with a focus on their structure.

5.6 Web application

Most products today are provided to customers through web applications. That is the easiest way for many users to interact with applications, as most devices come with a web browser and an internet connection. Web application would make it easier for users to upscale comic images without manipulation with scripts, environment, etc.

The application should provide an elegant and intuitive user interface. The main element of the page should be a field where images could be uploaded, which would then be processed by a neural network, and later could be downloaded to the user's device. It is also necessary to include a text with instructions on how the application should be used. The process of upscaling images should be started right after files are uploaded. The application also should support basic image formats such as JPEG, JPG, PNG.

Figure 5.4 shows proposed mockups for the web application. The header of the page contains the instructions text. The following part provides input fields. The first one is to choose the upscale factor. The second one provides a space for the user to upload an image that he wants to upscale. The last part, the least important, contains a description of the project.



Title

Instructions

- Lorem ipsum
- Lorem ipsum
- Lorem ipsum
- Lorem ipsum

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Upscale Factor

● 2x

○ 4x

Choose a file or drag it here.

About the project

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Figure 5.4: Web application wireframe mockup.

Chapter 6

Implementation

The chapter describes the used techniques and technologies and explains the reasons for their selection. The chapter talks about a chosen programming language and libraries used for the task. Then, the program's structure, models and utilities are described and explained. The web application design is shown in the last part of this chapter.

6.1 Technologies

The **Python** programming language was chosen for this task. The reason was its simplicity and familiarity, along with the powerful options which were crucial for implementing the neural network. A big part of the benefits is the support of machine learning frameworks, which greatly sped up the implementation process. From the commonly used and available frameworks, **PyTorch**[1] was chosen. Another important library used is **Python Imaging Library (PIL)**[2] for working with images. The web application is built on the **Streamlit**[3] framework, used for a data application with PyTorch support.

PyTorch Framework

PyTorch is an open-source machine learning framework based on the Python programming language and the Torch library. The framework performs computations by using tensors that could be accelerated by graphics cards (GPU). The advantage that the framework brings is its good support on major cloud platforms, which makes it easy to scale. Another advantage is easy debugging, which is provided with additional tools.

Python Imaging Library (PIL)

Python Imaging Library is an open-source framework for working with images. It supports many commonly used image formats, such as PPM, PNG, JPEG, GIF, TIFF, and BMP. For the purposes of this thesis, PNG and JPEG formats were sufficient. The framework was primarily used for loading images to the model. Along with loading images, it gives the user a simple way to manipulate images. Opening, saving, scaling, sharpening images, adjusting their brightness, contrast, color, etc., are all handy features of the PIL framework.

[1] https://pytorch.org/

[2] https://pillow.readthedocs.io/

[3] https://streamlit.io/

Streamlit Framework

Streamlit is an open-source framework designed for creating and sharing web applications for data science and machine learning projects. The framework provides an easy way to deploy a web application for a machine learning project in PyTorch.

The 3 above-mentioned frameworks form a large part of the project. The other minor libraries which are worth mentioning are:

- **Torchvision**[4] – *part of the PyTorch project*
  Data augmentation and image transformation.
- **Ignite**[5] – *high-level library to help with training and evaluating neural networks*
  Image quality metrics for evaluation (SSIM, PSNR).
- **Matplotlib**[6] – **Seaborn**[7] – **Pandas**[8]
  Data visualization from training and evaluation process.

6.2 Models and Utilities Scripts

The source parts of the implementations are split into two groups, models and utilities. Source files of the models contain three implemented architectures: U-Net and RUNet. There are multiple scripts, each for a particular purpose. One is for the training process, another implements the validation process, and one is for trained model usage. Another script with grouped utilities contains a dataset class for loading images, image preprocesses functions and Perceptual loss implementation.

Models

Network architectures are implemented in separate files using classes. Clean code and good conventions are maintained by implementing each architecture block within its class before combining it into the main architecture class. An implementation of blocks uses a sequential container[9] to store layers in the order in which they were passed to the constructor. The main class containing the architecture also combines blocks using sequential containers. The features used for convolutional layers are stored in a list for easy manipulation when changes are needed. When implementing architectures, the main focus was on the robustness of the solution for the ease of making changes during testing.

Utilities

The utilities are contained within a single Python script file. There are several sections in the file, including the implementation of datasets, functions for transforming images, quality evaluation metrics, and Perceptual loss function. The quality evaluation metrics are included from the Ignite library and wrapped in the custom function for easy use. The

[4] https://pytorch.org/vision/stable/

[5] https://pytorch.org/ignite/

[6] https://matplotlib.org/

[7] https://seaborn.pydata.org/

[8] https://pandas.pydata.org/

[9] https://pytorch.org/docs/stable/generated/torch.nn.Sequential.html

implemented dataset class inherits properties from the torch dataset class. Subsequently, Torchvision functions for image transformation make up the transformation functions used on the dataset. The Perceptual loss function uses the VGG16[10] pre-trained network and is implemented as a class that inherits properties from the `Torch` `nn.module` class.

6.3 Web application

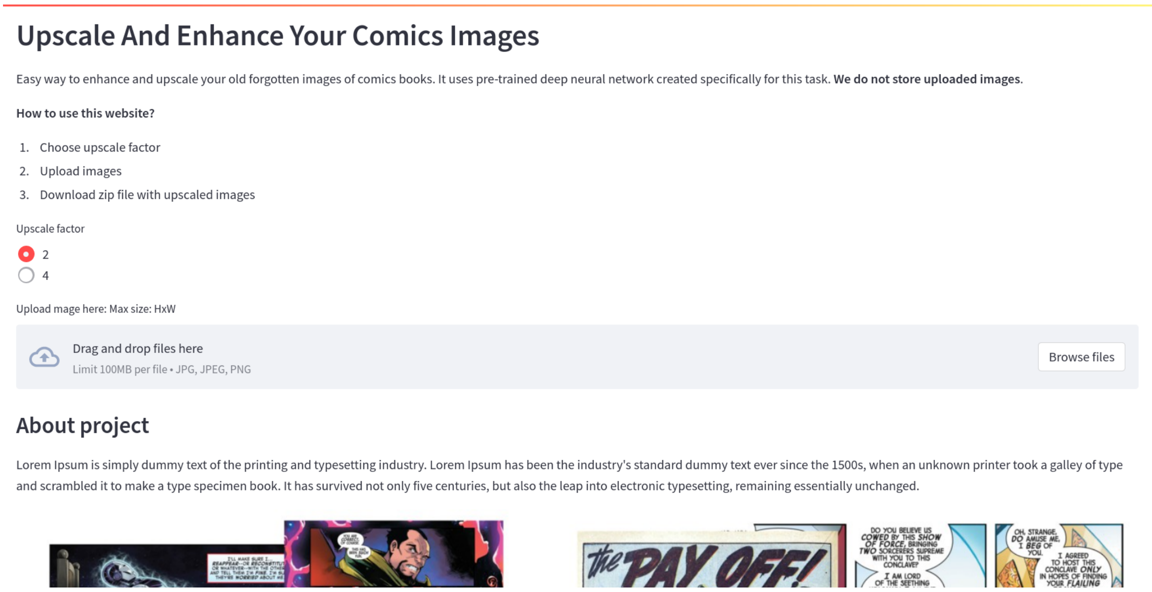
The work also includes a design for a web application built on the Streamlit framework. The application uses the final model with the best result to upscale comic images for users. Fast deployment and easy integration were key components of the building process of the application. Figures 6.1, 6.2 show the proposed designs for the application.

The web application contains a few essential parts:

- The **first** part of the web application includes an introduction with instructions for users on how to use the application (top of the 6.1 figure).
- The **second** and main part, contains the inputs field, the radio button for choosing upscale factor, and the file input, where the file is uploaded. The middle of Figure 6.1 visualizes the second part. After users upload the file (or multiple files), image preprocessing and upscaling is started. When the model upscales all the given files, a preview of the last uploaded file is shown, and the download button will appear, as Figure 6.2 shows. Download button downloads the zip file with upscaled images.
- The **third** part contains a description of the project.

The source codes for the web application are provided in shared files with this work. The model for the application is serialized using torchscript library and optimized for better performance. Due to the memory usage requirements of the application, it is not suitable for actual usage. Most of the memory usage is consumed within the network. One way to reduce the memory usage is to try to reduce the depth of the used architecture while simultaneously keeping the level of the image quality. After improving the memory usage of the deep neural network, the application would be usable. The other option is the usage of cache functionality provided by streamlit library. But at this moment, there are some problems with models that are run in torchscript.

[10] https://pytorch.org/hub/pytorch_vision_vgg/



Upscale And Enhance Your Comics Images

Easy way to enhance and upscale your old forgotten images of comics books. It uses pre-trained deep neural network created specifically for this task. **We do not store uploaded images**.

How to use this website?

1. Choose upscale factor
2. Upload images
3. Download zip file with upscaled images

Upscale factor

● 2
○ 4

Upload mage here: Max size: HxW

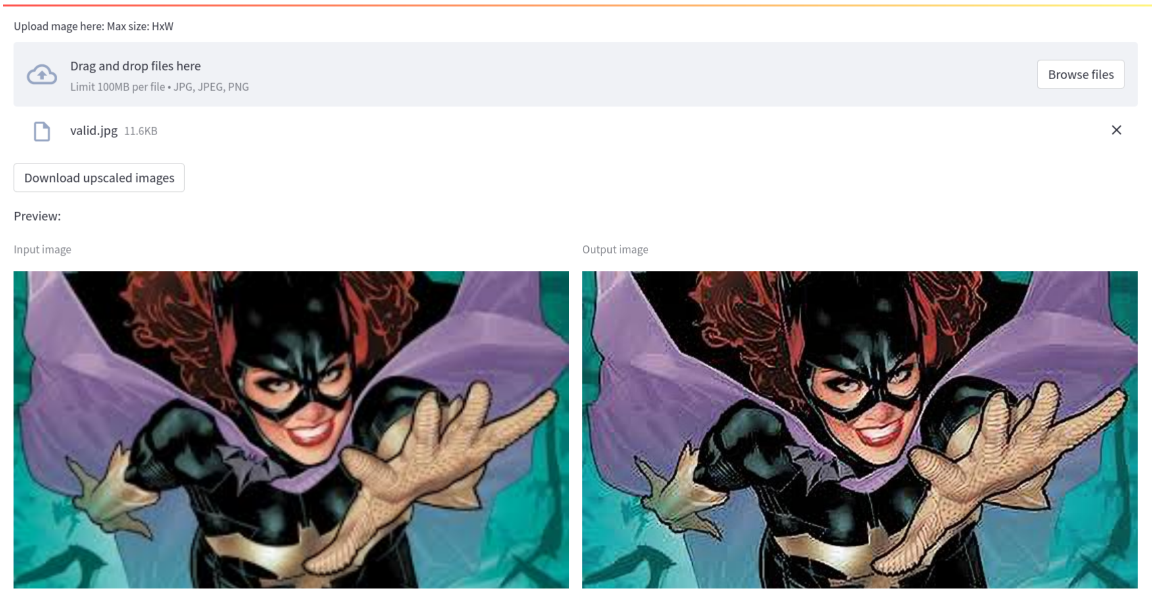
Drag and drop files here
Limit 100MB per file • JPG, JPEG, PNG

About project

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

The PAY OFF!
DO YOU BELIEVE US COWED BY THIS SHOW OF FORCE, BRINGING TWO SORCERERS SUPREME WITH YOU TO THIS CONCLAVE?
I AM LORD OF THE SEETHING
OH, STRANGE, DO AMUSE ME. I BEG OF YOU.
I AGREED TO HOST THIS CONCLAVE ONLY IN HOPES OF FINDING YOUR FLAILING

Figure 6.1: Web application design composed of three main blocks; introduction with instructions of app usage; input fields for images and upscale factor; final descriptions with details about the project and creator. The main design focus of the application was simplicity and minimalism.



Upload mage here: Max size: HxW

Drag and drop files here
Limit 100MB per file • JPG, JPEG, PNG

valid.jpg 11.6KB

Preview:

Input image

Output image

Figure 6.2: View of the main section of the application after input is provided. After upscaling the process of all images, the download button appears, also, with the preview of the results.

Chapter 7

Experiments

This chapter describes the conducted experiments and their results. These experiments led to the best combination of architecture and loss function for achieving the best results. The selected architectures U-Net and RUNet, mentioned in the proposed solutions section, were tested along with MSE and Perceptual loss function combinations. Another experiment focused on the number of the extracted blocks from the VGG16 loss network used for Perceptual loss.

For the evaluation, PSNR and SSIM metrics were used. But as mentioned in the paper [18], these metrics are significantly lower when using Perceptual loss function for training. For this reason, the test images related to the following experiments are provided so that it is possible to evaluate the quality by manual eye observation.

7.1 Training Process and Parameters

The inputs for the proposed networks are RGB (three color channels) images with 128×128 resolution due to memory restriction of used hardware. Adam Optimizer was used for training. The parameters used were defaulted from the PyTorch library, `betas` has a value of tuple (0.9, 0.999), `eps` is set to $1e^{-8}$ and weight decay to 0. Initially, the learning rate was set to 0.001. During the training process, the learning rate is dynamically reduced by the scheduler[1]. The learning rate value is reduced by a factor of 0.2 each time that validation loss has not improved in the 5 following epochs. In order to get the best possible model from training and to prevent dataset overfitting, early stopping[2] is used. Early stopping monitors validation loss and saves the model as the checkpoint if the loss has reached the new minimum. The early stop limit was set to 30 consecutive epochs. Essentially, the training ends when the validation loss has not reached a new minimum in 30 epochs.

The training procedure was performed on NVIDIA GeForce RTX 2060 Max-Q graphics card. Also, an important thing, convolutions are applied with padding to ensure that the image input and output sizes are the same.

7.2 Degradation Module and Data Preparation

The data augmentation technique was used to guarantee consistently new input for the neural network. The process of augmentation on ground truth images consists of random

[1] https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

[2] https://github.com/Bjarten/early-stopping-pytorch

cropping and flipping. The random crop operation produced the cropped result of the image of size 128 × 128 pixels. The following operations applied to the cropped image were random horizontal and vertical flips. Both operations had parameters set to a 25 percent chance of flipping the image.

After obtaining the ground truth augmented image, a degradation module was used to get the low-resolution one. In the beginning, the image is downscaled by bicubic interpolation by the chosen scale factor. Then the Gaussian blur is applied with a 3 × 3 kernel and sigma minimum set to 0.1 and maximum to 0.3. After the blur is applied, the image is upscaled to the original size, which means 128 × 128 pixels. The final step of the degradation is adding the Gaussian noise. Before the noise application, the image is converted to a tensor with values in a range from 0 to 1. The noise function uses a `torch.randn` function that produces a tensor with elements drawn from a Gaussian distribution of zero mean and unit variance. After that, the tensor is multiplied by 0.05 to have the desired variance.

7.3 Architectures Test

The following experiments test the selected architectures in their basic setup. The goal is to figure out which of the given architectures is more suitable for working with comic books.

7.3.1 U-Net vs RUNet

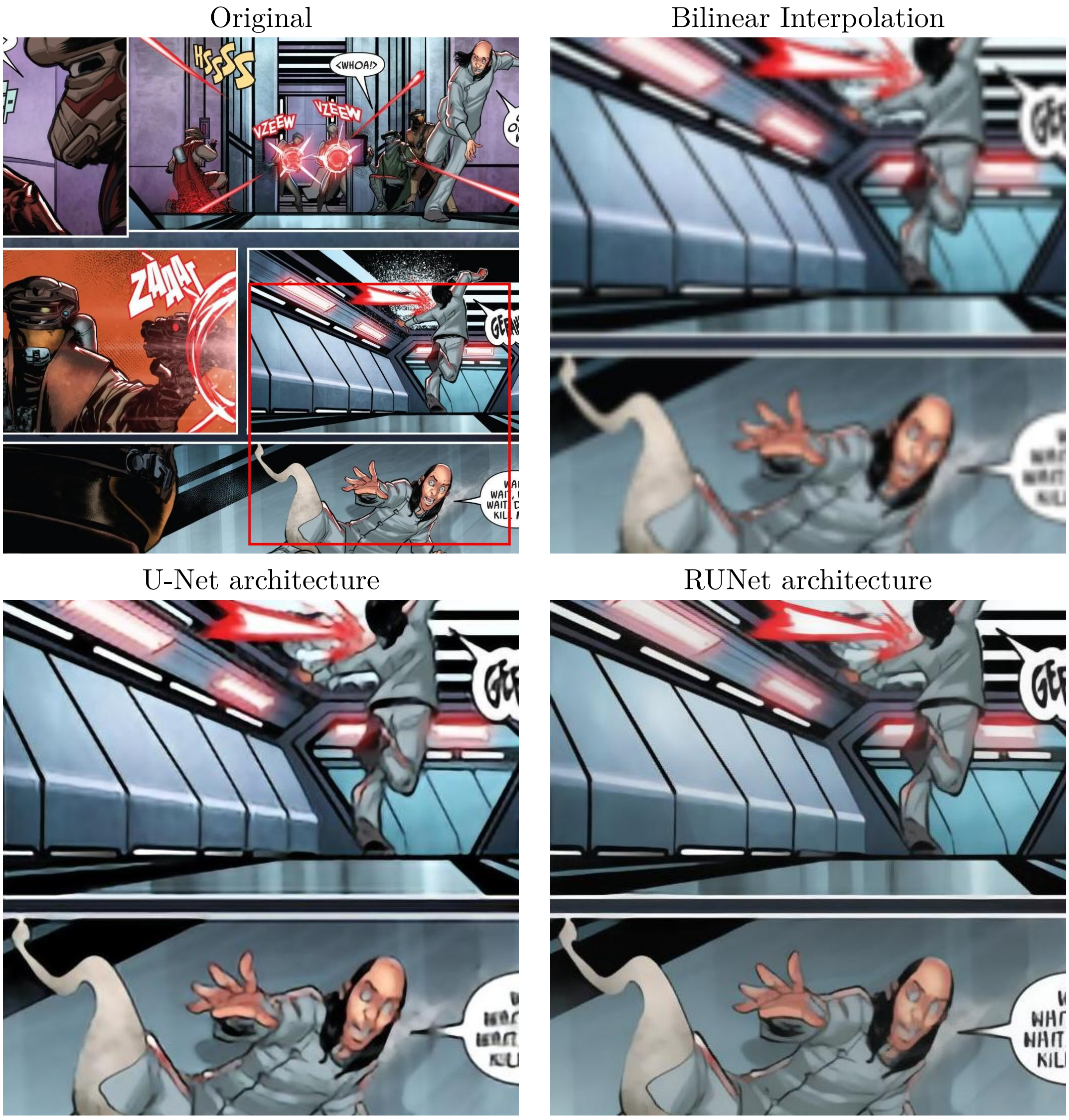
The experiment consists of comparing state-of-the-art U-Net and RUNet architectures. The U-Net architecture uses the Mean Squared Error loss function. The RUNet architecture uses a Perceptual loss function. The Perceptual loss function uses five extracted blocks from the VGG16 network.

Architecture & interpolation method	Scale factor			
	2x		4x	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM
U-Net	**31.218**	**0.890**	25.110	**0.831**
RUNet	27.031	0.888	**25.378**	0.825
Bilinear interpolation	23.982	0.870	20.093	0.721

Table 7.1: The resulting measurements for the selected architectures (scale factors of 2 and 4).

The U-Net architecture, unlike the RUNet, is in no way adapted to the super-resolution task. Thus, the RUNet results for the examined task should be better. As seen in Figure 7.1, the results from the trained network are outstanding. RUNet architecture provides more visually appealing results. However, Table 7.1 shows that U-Net architecture gets better metric results. SSIM of the upscaled images from both architectures got similar results. On the other hand, U-Net got significantly better results in the PSNR metric. Based on the discussion in the paper [18], those metrics results were expected to be worse for RUNet. But by visually observing the images, RUNet was better.

Both architectures achieved better results than bilinear interpolation. However, the U-Net architecture suffers from visible deformations in the text and blur amplification. On the other hand, RUNet achieved better text details, refined areas and sharp edges.


Original

Bilinear Interpolation

U-Net architecture

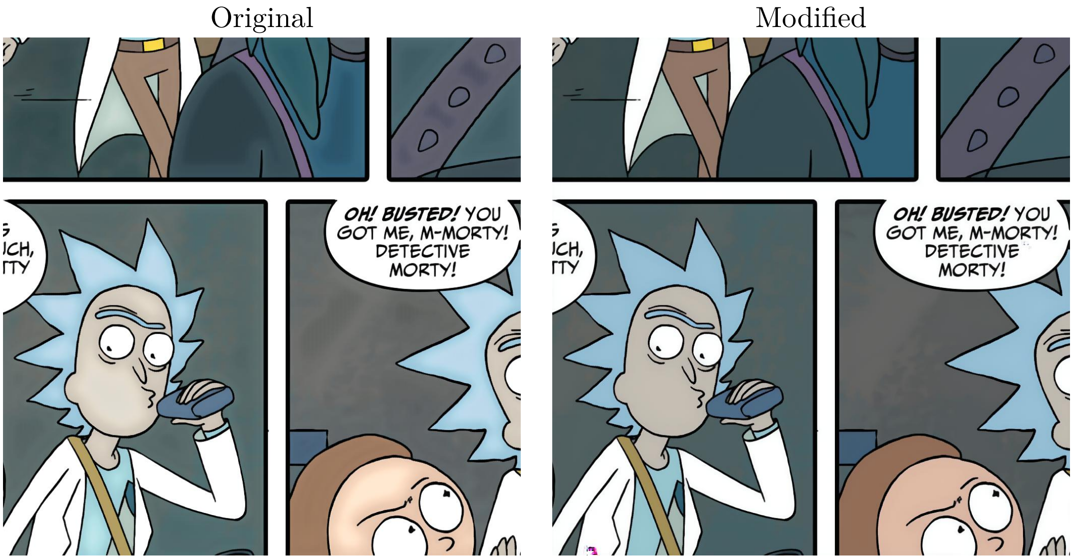
RUNet architecture

Figure 7.1: Example of qualitative visual results for bilinear interpolation, U-Net, and RUNet architecture (scale factor of 4).

RUNet modification

With a closer look at Figure 7.2, some weird refining and outline highlighting can be seen in the results achieved by RUNet. Therefore, a change to the architecture was proposed. The left contracting path of the architecture contained four max-pooling operations for downsampling. However, the expanding path contains five upsampling operations. The first change removes the pixel shuffle operation from the bottleneck to make both paths have an equal number of the sampling operations. Another modification realized to the RUNet was adding batch normalization between two ReLU layers in the expanding block. The results of applying these modifications are shown in Figure 7.2.

The modified architecture eliminated the problem of the original architecture, and therefore will be used in the following experiments. New metrics for the modified architecture are:


Original

Modified

OH! BUSTED! YOU GOT ME, M-MORTY! DETECTIVE MORTY!
OH! BUSTED! YOU GOT ME, M-MORTY! DETECTIVE MORTY!

Figure 7.2: Results of the modified RUNet architecture compared to the original one.

- scale factor 2 → **PSNR** = **28.714 dB** and **SSIM** = **0.902**
- scale factor 4 → **PSNR** = **25.151 dB** and **SSIM** = **0.834**

7.4 Loss Functions Test

The following experiments focus on the models using other loss functions as the default. The first experiment compares each architecture separately using the MSE and Perceptual loss functions. The second one examines the impact of various number of used blocks from the VGG16 network on the Perceptual loss function.

7.4.1 Mean Square Error vs Perceptual Loss

An essential aspect of the training process is to choose a suitable loss function. Therefore, the next experiment examines the impact of Mean Squared Error loss and Perceptual loss function on selected architectures.

Architecture	Loss Function	Scale factor			
		2x		4x	
		PSNR (db)	SSIM	PSNR (db)	SSIM
U-Net	Mean Square Error	**31.218**	0.890	25.110	0.831
	Perceptual Loss	29.857	**0.925**	24.086	**0.849**
RUNet	Mean Square Error	28.596	0.902	22.672	0.780
	Perceptual Loss	28.714	0.902	**25.151**	0.834

Table 7.2: The resulting measurements for the selected architectures with different loss functions. (scale factors of 2 and 4)

U-Net

Table 7.2 shows image quality metrics for U-Net architecture with different loss functions. The MSE loss function achieved the best results in the PSNR metric in both scale factors 2 and 4. On the other hand, Perceptual loss achieves better performance in SSIM. Figure 7.3 shows the visual results of the experiment. Using Perceptual loss function leads to smoother results. However, in both cases, it is still possible to observe a large deformation in the reconstruction of the text.

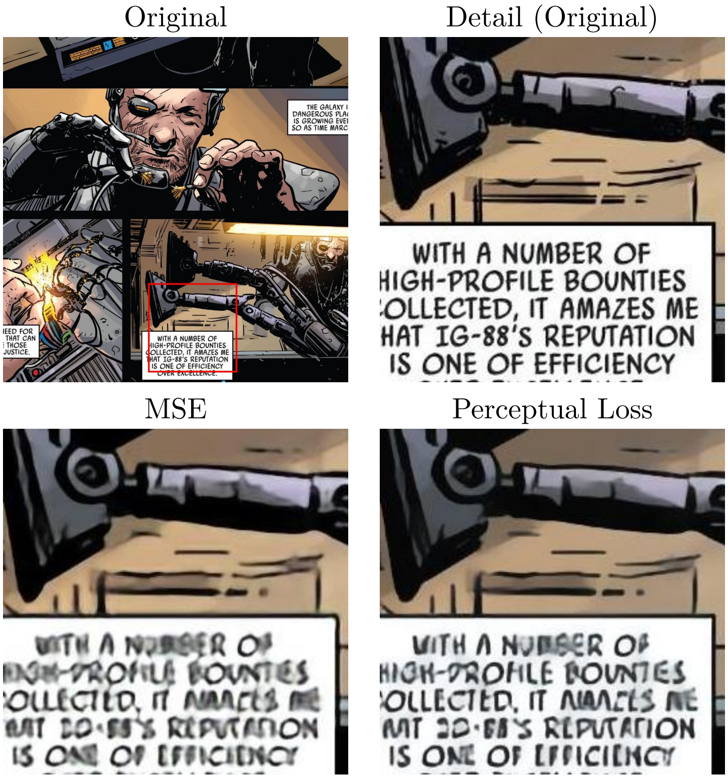


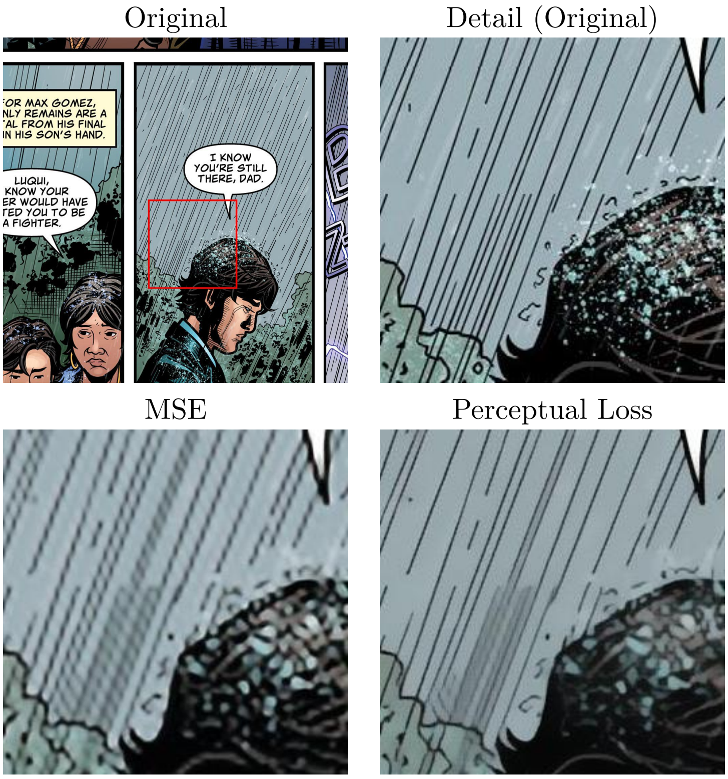
Figure 7.3: Visual results for **U-Net** architecture using MSE and Perceptual Loss functions (scale factor 4).

RUNet

The results of the RUNet architecture using the MSE loss function are straightforward. Figure 7.4 shows that usage RUNet architecture with MSE creates more noisy and worse quality outputs in comparison when using Perceptual loss. Also, the RUNet with Perceptual loss got significantly better results in metrics with a scale factor of 4. These metrics are quite similar when using a factor 2, but the Perceptual loss still achieves smoother and better images. Table 7.2 shows those metrics results.

7.4.2 Perceptual loss – various numbers of extracted blocks

The default number of blocks that RUNet uses in the Perceptual loss function is 5. This number was mentioned in the original paper, probably achieving the best results for the specific task used in the paper, which was super-resolution of television programs and


Original

Detail (Original)

MSE

Perceptual Loss

'OR MAX GOMEZ, NLY REMAINS ARE A 'AL FROM HIS FINAL IN HIS SON'S HAND.
LUQUI, KNOW YOUR ER WOULD HAVE TED YOU TO BE A FIGHTER.
I KNOW YOU'RE STILL THERE, DAD.

Figure 7.4: Visual results for **RUNet** architecture using MSE and Perceptual Loss functions (scale factor of 4)

movies. A different number of the extracted blocks is used in the following experiment to examine their impact on metric results and visual quality. The research range is from 1 to 5. **The RUNet architecture with a scale factor of 4 is used for this experiment due to the better observation of changes on the visual results.**

Figure 7.5 shows the visual results of the experiment with variable number of extracted blocks. The quality of the drawing part did not change much during the experiments, only some minuscule details that are not worth mentioning. However, observable changes have occurred in the text area. The architecture achieved the best results in text readability using the loss function with three and four blocks. Although it is still possible to observe text distortions, they are not as noticeable as in other cases.

Figures 7.6 and 7.7 show the image quality metrics of individually trained architecture with different numbers of blocks used. The best results were achieved by models trained with one block. Despite the best visual results, the model with four extracted blocks does not have as good image quality metrics as it does with fewer blocks. The drawn parts could be the reason, as the human observer does not notice small visual discrepancies. **Thus, the three and four extracted blocks produced the best results for maintaining the quality of the text and drawing.**

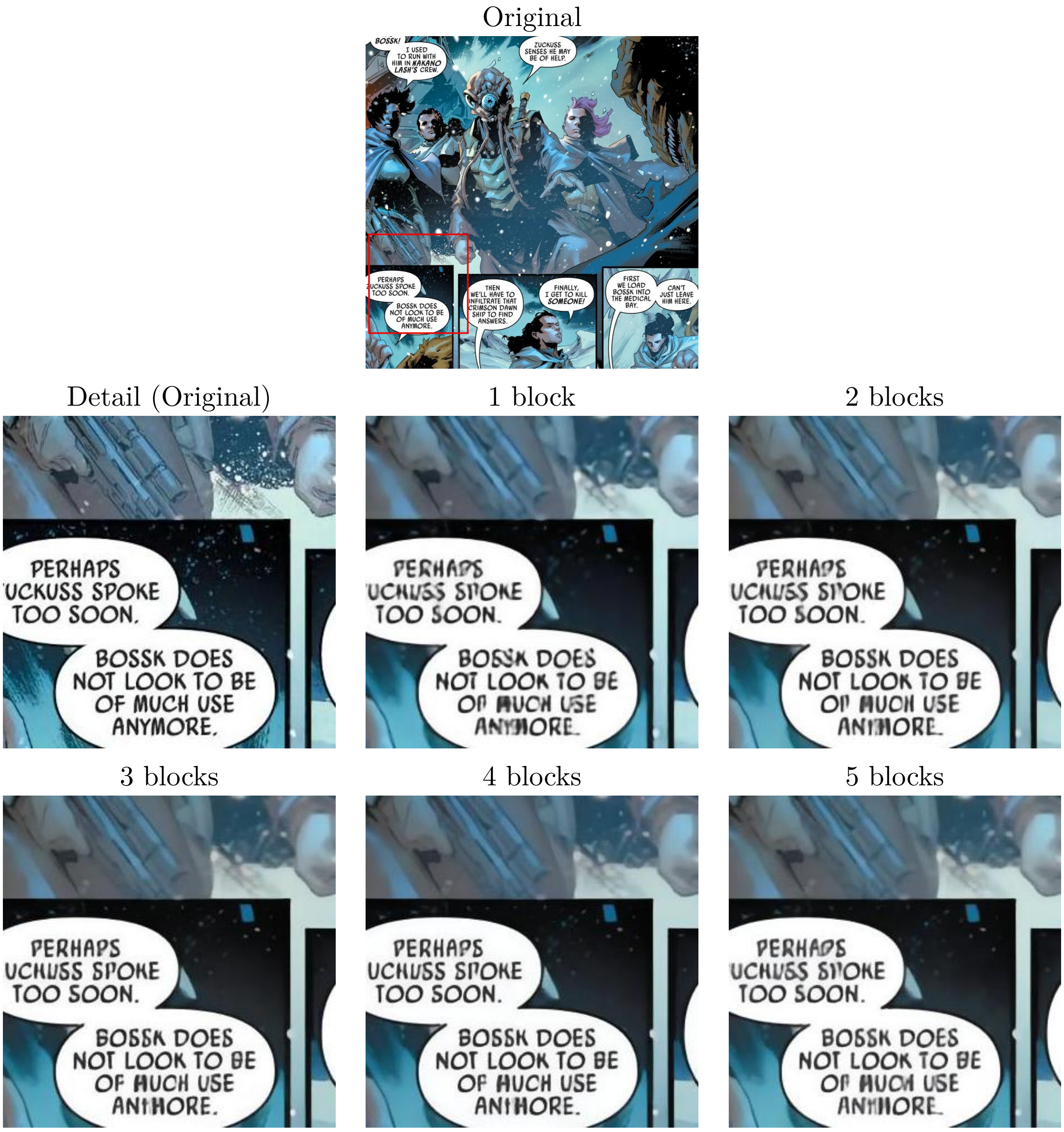

Original
BOSSK!
I USED TO RUN WITH HIM IN NAKANO LASH'S CREW.
ZUCKUSS SENSES HE MAY BE OF HELP.
PERHAPS ZUCKUSS SPOKE TOO SOON.
BOSSK DOES NOT LOOK TO BE OF MUCH USE ANYMORE.
THEN WE'LL HAVE TO INFILTRATE THAT CRIMSON DAWN SHIP TO FIND ANSWERS.
FINALLY, I GET TO KILL SOMEONE!
FIRST WE LOAD BOSSK INTO THE MEDICAL BAY.
CAN'T JUST LEAVE HIM HERE.
Detail (Original)
PERHAPS 'UCKUSS SPOKE TOO SOON.
BOSSK DOES NOT LOOK TO BE OF MUCH USE ANYMORE.
1 block
2 blocks
3 blocks
4 blocks
5 blocks

Figure 7.5: Visual image results using Perceptual loss function with various extracted blocks from the VGG16 network (RUNet architecture using scale factor of 4).

7.5 Summary

Initially, experiments focused on evaluating architectures with respect to their ability to produce super-resolution images. The results of both architectures are better than the results achieved with bilinear interpolation.

In line with expectations, RUNet's design was built for a super-resolution task and produced better results than U-Net. RUNet's ability to learn more complex structures mainly helped the text areas in the image, as Figure 7.8 shows. The experiments encountered a problem with the strange highlighting of areas when using the RUNet architect. Therefore, a modification has been proposed for RUNet, as described in the section 7.3.1. The adjustment has solved the problem and even improved the image quality metrics.

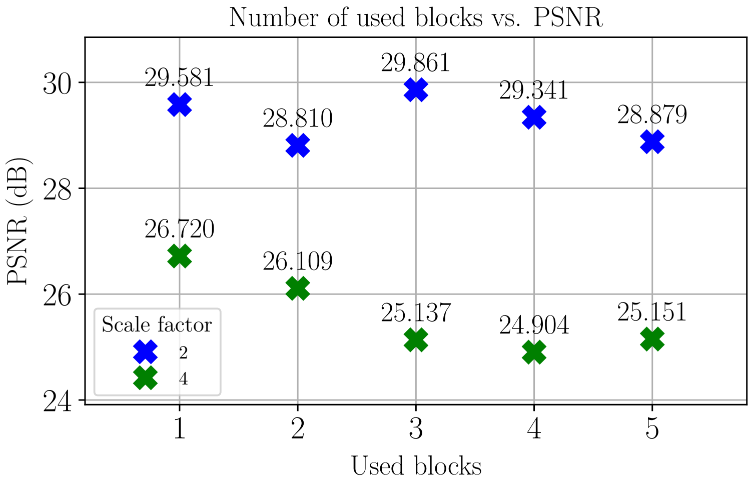


Figure 7.6: The graph shows the dependence of the number of used blocks from the VGG16 network on the **PSNR** metric. (RUNet)

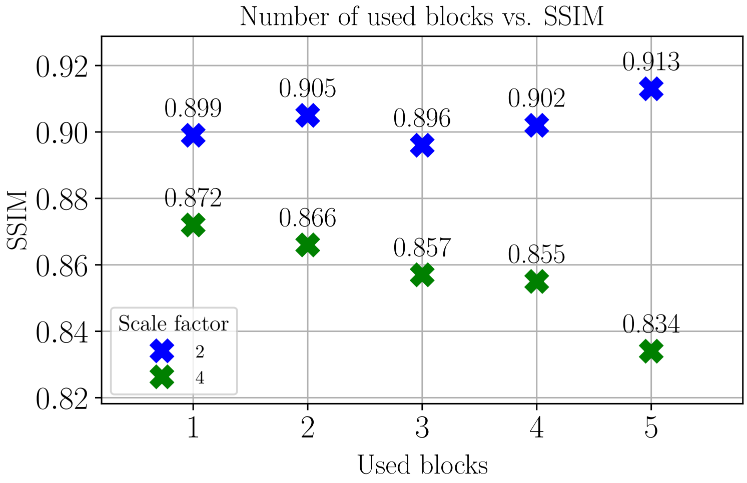


Figure 7.7: The graph shows the dependence of the number of used blocks from the VGG16 network on the **SSIM** metric. (RUNet)

Further experiments focused on examining the optimal loss function for a provided architecture. According to the experiments with loss functions, Perceptual loss outperformed MSE. The results for Perceptual loss were smoother and more readable in the text area, as can be seen in Figure 7.3 and 7.4, and also the results shown in Table 7.2 show that Perceptual loss performs better in the SSIM metric. With the loss function MSE, it was possible to see that RUNet performance had deteriorated significantly (see Figure 7.4). In terms of image quality metric results, Perceptual loss provides better results in SSIM (see Table 7.2). On the other hand, MSE outperforms Perceptual loss in PSNR.

The last experiments focused on the Perceptual loss function. The function uses the extracted blocks from the VGG16 network made of the convolutional layers to convert the image to feature space and calculate the euclidean distance between low-resolution and high-resolution feature maps. The experiment focused on finding the optimal number of blocks to achieve the best results. According to the findings, RUNet architecture is most effective when using the Perceptual loss function with three extracted blocks.

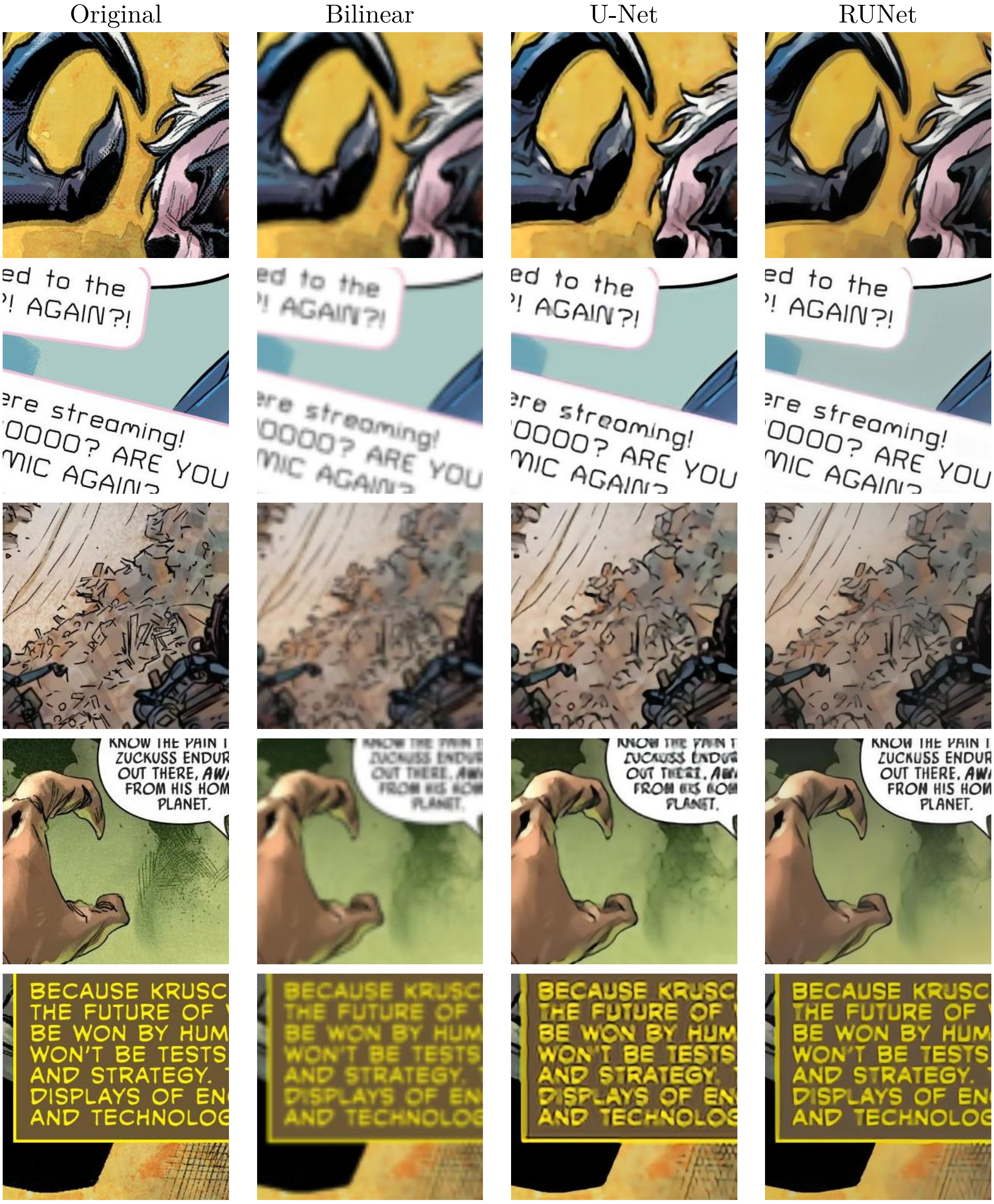

Original Bilinear U-Net RUNet

Figure 7.8: Additional visual results. RUNet using Perceptual loss with 3 extracted blocks and U-Net using MSE loss function.

Chapter 8

Conclusion

This Bachelor's thesis focused on upscaling and improving the quality of comic book images. The difficulty with comic book images lies in their mixed composition of text and drawn parts. The approach to upscaling the images was to use current super-resolution methods by using deep neural networks. Firstly, it was necessary to study deep neural networks and super-resolution techniques in order to select an appropriate architecture for upscaling comic book images. Secondly, a suitable dataset of comic book images was obtained for training, validating and testing the architecture. The dataset is composed of high-quality digital comic book images with a balanced content of text parts and drawn images. Experiments were conducted on two architectures – a U-Net network and a modified RUNet network, and two loss functions – Perceptual loss function and Mean Square Error function. Additionally, various combinations of these architectures and functions were also tested.

Results showed that the basic U-Net architecture is suitable for the comic book super-resolution. Using the scale factor 2, U-Net obtained promising results in both parts of a composition, but a problem was encountered in the text parts when using scale factor 4. The RUNet architecture achieved excellent results. Unlike U-Net, it did not have a problem with a scale factor of 4; no significant deformations of any part of the composition were found. Experiments that focused on the loss function used showed that the Perceptual loss function has a greater effect on the learning network to achieve better results. Experiments on the Perceptual loss function and the number of blocks used from the loss network showed that using 3 blocks achieves best visual results and image quality metrics.

Even though the experiments showed acceptable results for super-resolution tasks, there is still room for improvements. It would be interesting to explore generative adversarial networks (GANs)[13] in the future. The advantage of GANs is the utilization of two networks that train each other to produce enhanced, high-quality image details. Also, substantial improvements in the future should focus on the text parts of comic images. Although some text distortion is still visible in the results, it would be appropriate to examine the field of text oriented super-resolution techniques and combine them with the methods used in this thesis. The use of inception blocks [34], for example, may help with learning how to reconstruct the text and small details (within the range of a few pixels).

This work has shown that it is possible to apply deep learning and neural networks to more complex compositions of comic book images with acceptable results for a human observer. The work also offers a proposal for a user-friendly web application with the usage of a trained neural network.

Bibliography

[1] Ahn, N., Kang, B. and Sohn, K.-A. Fast, accurate, and lightweight super-resolution with cascading residual network. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, p. 252–268.

[2] Chang, H., Yeung, D.-Y. and Xiong, Y. Super-resolution through neighbor embedding. In: IEEE. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* 2004, vol. 1, p. I–I.

[3] Chen, G., Zhao, H., Pang, C. K., Li, T. and Pang, C. Image Scaling: How Hard Can it Be? *IEEE Access*. 2019, vol. 7, p. 129452–129465. DOI: 10.1109/ACCESS.2019.2940353.

[4] Donaldson, K. and Myers, G. K. Bayesian super-resolution of text in videowith a text-specific bimodal prior. *International Journal of Document Analysis and Recognition (IJDAR)*. Springer. 2005, vol. 7, no. 2, p. 159–167.

[5] Du, J., Zhou, H., Qian, K., Tan, W., Zhang, Z. et al. RGB-IR Cross Input and Sub-Pixel Upsampling Network for Infrared Image Super-Resolution. *Sensors*. 2020, vol. 20, no. 1. DOI: 10.3390/s20010281. ISSN 1424-8220. Available at: https://www.mdpi.com/1424-8220/20/1/281.

[6] Duchon, C. E. Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology and Climatology*. 1979, vol. 18, no. 8, p. 1016–1022.

[7] Dumoulin, V. and Visin, F. A guide to convolution arithmetic for deep learning. *ArXiv preprint arXiv:1603.07285*. 2016.

[8] Farsiu, S., Robinson, M. D., Elad, M. and Milanfar, P. Fast and robust multiframe super resolution. *IEEE transactions on image processing*. IEEE. 2004, vol. 13, no. 10, p. 1327–1344.

[9] Freedman, G. and Fattal, R. Image and video upscaling from local self-examples. *ACM Transactions on Graphics (TOG)*. ACM New York, NY, USA. 2011, vol. 30, no. 2, p. 1–11.

[10] Freeman, W. T., Jones, T. R. and Pasztor, E. C. Example-based super-resolution. *IEEE Computer graphics and Applications*. IEEE. 2002, vol. 22, no. 2, p. 56–65.

[11] Gholipour, A., Estroff, J. A. and Warfield, S. K. Robust super-resolution volume reconstruction from slice acquisitions: application to fetal brain MRI. *IEEE transactions on medical imaging*. IEEE. 2010, vol. 29, no. 10, p. 1739–1758.

[12] Glasner, D., Bagon, S. and Irani, M. Super-resolution from a single image. In: IEEE. *2009 IEEE 12th international conference on computer vision*. 2009, p. 349–356.

[13] Goodfellow, I. J., Pouget Abadie, J., Mirza, M., Xu, B., Warde Farley, D. et al. *Generative Adversarial Networks*. arXiv, 2014. DOI: 10.48550/ARXIV.1406.2661. Available at: https://arxiv.org/abs/1406.2661.

[14] Haris, M., Shakhnarovich, G. and Ukita, N. Deep Back-ProjectiNetworks for Single Image Super-Resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021, vol. 43, no. 12, p. 4323–4337. DOI: 10.1109/TPAMI.2020.3002836.

[15] Haris, M., Shakhnarovich, G. and Ukita, N. Deep Back-Projection Networks for Super-Resolution. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[16] He, K., Zhang, X., Ren, S. and Sun, J. Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, p. 770–778. DOI: 10.1109/CVPR.2016.90.

[17] Hore, A. and Ziou, D. Image quality metrics: PSNR vs. SSIM. In: IEEE. *2010 20th international conference on pattern recognition*. 2010, p. 2366–2369.

[18] Hu, X., Naiel, M. A., Wong, A., Lamm, M. and Fieguth, P. RUNet: A Robust UNet Architecture for Image Super-Resolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2019.

[19] Johnson, J., Alahi, A. and Fei Fei, L. *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*. 2016.

[20] Kennedy, J. A., Israel, O., Frenkel, A., Bar Shalom, R. and Azhari, H. Super-resolution in PET imaging. *IEEE transactions on medical imaging*. IEEE. 2006, vol. 25, no. 2, p. 137–147.

[21] Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1981, vol. 29, no. 6, p. 1153–1160. DOI: 10.1109/TASSP.1981.1163711.

[22] Keys, R. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*. Ieee. 1981, vol. 29, no. 6, p. 1153–1160.

[23] Kim, J., Lee, J. K. and Lee, K. M. Accurate image super-resolution using very deep convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 1646–1654.

[24] Kim, K. I. and Kwon, Y. Single-image super-resolution using sparse regression and natural image prior. *IEEE transactions on pattern analysis and machine intelligence*. IEEE. 2010, vol. 32, no. 6, p. 1127–1133.

[25] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A. et al. *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*. 2017.

[26] Lu, Z. and Chen, Y. *Dense U-net for super-resolution with shuffle pooling layer*. 2021.

[27] Lu, Z. and Chen, Y. Single image super-resolution based on a modified U-net with mixed gradient loss. *Signal, Image and Video Processing*. Springer. 2021, p. 1–9.

[28] Miravet, C., Rodri, F. B. et al. A two-step neural-network based algorithm for fast image super-resolution. *Image and Vision Computing*. Elsevier. 2007, vol. 25, no. 9, p. 1449–1473.

[29] Nguyen, C. D., Ardabilian, M. and Chen, L. Unifying approach for fast license plate localization and super-resolution. In: IEEE. *2010 20th International Conference on Pattern Recognition*. 2010, p. 376–379.

[30] Prajapati, A., Naik, S. and Mehta, S. Evaluation of different image interpolation algorithms. *International Journal of Computer Applications*. Citeseer. 2012, vol. 58, no. 12, p. 6–12.

[31] Ronneberger, O., Fischer, P. and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR*. 2015, abs/1505.04597. Available at: http://arxiv.org/abs/1505.04597.

[32] Samajdar, T., Quraishi, M. et al. Analysis and evaluation of image quality metrics. In: *Information Systems Design and Intelligent Applications*. Springer, 2015, p. 369–378.

[33] 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.

[34] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E. et al. Going Deeper with Convolutions. *CoRR*. 2014, abs/1409.4842. Available at: http://arxiv.org/abs/1409.4842.

[35] Talab, M. A., Awang, S. and Najim, S. A.-d. M. Super-Low Resolution Face Recognition using Integrated Efficient Sub-Pixel Convolutional Neural Network (ESPCN) and Convolutional Neural Network (CNN). In: *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*. 2019, p. 331–335. DOI: 10.1109/I2CACIS.2019.8825083.

[36] Tian, J. and Ma, K.-K. A new state-space approach for super-resolution image sequence reconstruction. In: IEEE. *IEEE International Conference on Image Processing 2005*. 2005, vol. 1, p. I–881.

[37] Tian, Y., Yap, K.-H. and He, Y. Vehicle license plate super-resolution using soft learning prior. *Multimedia Tools and Applications*. Springer. 2012, vol. 60, no. 3, p. 519–535.

[38] Tsai, R. Multiframe image restoration and registration. *Advance Computer Visual and Image Processing*. 1984, vol. 1, p. 317–339.

[39] 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, p. 864–873.

[40] Wang, Z., Chen, J. and Hoi, S. C. H. Deep Learning for Image Super-Resolution: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021, vol. 43, no. 10, p. 3365–3387. DOI: 10.1109/TPAMI.2020.2982166.

[41] Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*. IEEE. 2004, vol. 13, no. 4, p. 600–612.

[42] Watson, A. B. and Null, C. H. Digital images and human vision. In: *Electronic Imaging Science and Technology Conference*. 1997.

[43] Yu, L., Zhang, X. and Chu, Y. Super-Resolution Reconstruction Algorithm for Infrared Image with Double Regular Items Based on Sub-Pixel Convolution. *Applied Sciences*. 2020, vol. 10, no. 3. DOI: 10.3390/app10031109. ISSN 2076-3417. Available at: https://www.mdpi.com/2076-3417/10/3/1109.

[44] Zhang, H., Zhang, L. and Shen, H. A super-resolution reconstruction algorithm for hyperspectral images. *Signal Processing*. Elsevier. 2012, vol. 92, no. 9, p. 2082–2096.

[45] Zhang, Y., Wu, G., Yap, P.-T., Feng, Q., Lian, J. et al. Reconstruction of super-resolution lung 4D-CT using patch-based sparse representation. In: IEEE. *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, p. 925–931.

[46] Zhang, Y. Problems in the fusion of commercial high-resolution satelitte as well as Landsat 7 images and initial solutions. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*. Citeseer. 2002, vol. 34, no. 4, p. 587–592.

[47] Zhao, S., Zhang, L., Shen, Y., Zhao, S. and Zhang, H. Super-Resolution for Monocular Depth Estimation With Multi-Scale Sub-Pixel Convolutions and a Smoothness Constraint. *IEEE Access*. 2019, vol. 7, p. 16323–16335. DOI: 10.1109/ACCESS.2019.2894651.

Appendix A

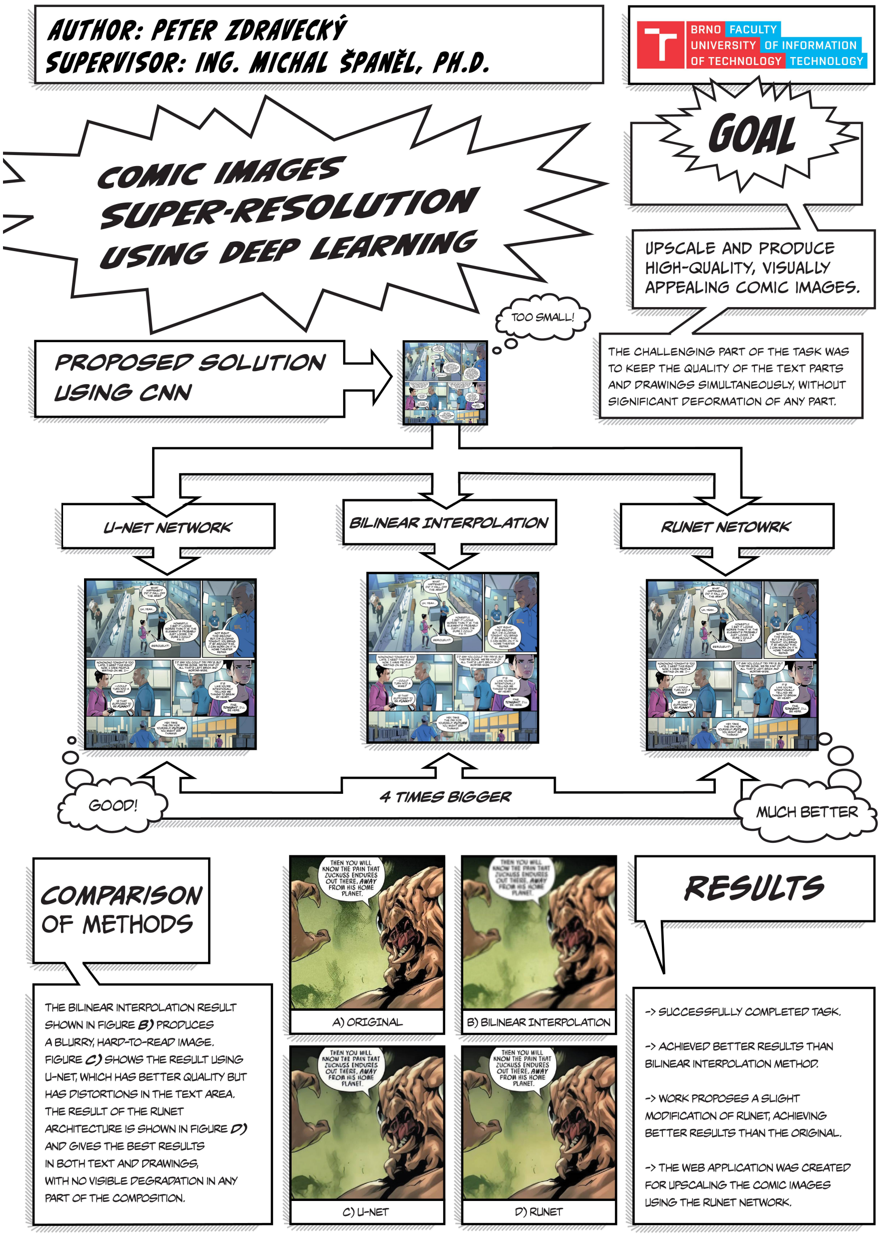
Contents of the Included Storage Media

• `src/`	Folder with source files.
• `saved-models/`	Folder with pretrained models.
• `latex/`	Folder with LaTeX source files.
• `datasets/`	Folder with dataset for training, validation, and testing.[1]
• `configs/`	Folder with example config files.
• `LICENCE`	Project licence.
• `README.md`	README file for the project.
• `requirements.txt`	Python libraries dependencies.
• `poster.jpg`	Poster image.
• `poster-print.pdf`	Poster for print.
• `thesis.pdf`	Thesis report file.
• `thesis-print.pdf`	Thesis report file for print.

[1] Due to inaccurate license terms of the images from the `getcomics.info` site, only a few images are uploaded to show the images' structure.

Appendix B

Poster



AUTHOR: PETER ZDRAVECKÝ
SUPERVISOR: ING. MICHAL ŠPANĚL, PH.D.

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

COMIC IMAGES SUPER-RESOLUTION USING DEEP LEARNING

GOAL

UPSCALE AND PRODUCE HIGH-QUALITY, VISUALLY APPEALING COMIC IMAGES.

THE CHALLENGING PART OF THE TASK WAS TO KEEP THE QUALITY OF THE TEXT PARTS AND DRAWINGS SIMULTANEOUSLY, WITHOUT SIGNIFICANT DEFORMATION OF ANY PART.

PROPOSED SOLUTION USING CNN

COMPARISON OF METHODS

THE BILINEAR INTERPOLATION RESULT SHOWN IN FIGURE *B)* PRODUCES A BLURRY, HARD-TO-READ IMAGE. FIGURE *C)* SHOWS THE RESULT USING U-NET, WHICH HAS BETTER QUALITY BUT HAS DISTORTIONS IN THE TEXT AREA. THE RESULT OF THE RUNET ARCHITECTURE IS SHOWN IN FIGURE *D)* AND GIVES THE BEST RESULTS IN BOTH TEXT AND DRAWINGS, WITH NO VISIBLE DEGRADATION IN ANY PART OF THE COMPOSITION.

RESULTS

-> SUCCESSFULLY COMPLETED TASK.

-> ACHIEVED BETTER RESULTS THAN BILINEAR INTERPOLATION METHOD.

-> WORK PROPOSES A SLIGHT MODIFICATION OF RUNET, ACHIEVING BETTER RESULTS THAN THE ORIGINAL.

-> THE WEB APPLICATION WAS CREATED FOR UPSCALING THE COMIC IMAGES USING THE RUNET NETWORK.