Investigation of Using RGB-Based Real-World Image Super-Resolution to Achieve Real-World Thermal Image Super-Resolution

Thesis Report

Aalborg University
Electronics and IT

Study Board of Electronics and IT
Fredrik Bajers Vej 7B
DK-9220 Aalborg

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Investigation of Using RGB-Based Real-world Image Super-Resolution to Achieve Real-World Thermal Image Super-Resolution

Theme:
Computer Vision

Project Period:
Spring semester 2021

Project Group:
VGIS, 10th Semester

Participant(s):
Moaaz M. J. Allahham

Supervisor(s):
Kamal Nasrollahi
Andreas Aakerberg

Copies: 1

Page Numbers: 64

Date of Completion:
June 3, 2021

Abstract:

Thermal cameras are used in various domains where the vision of RGB cameras is limited. Thermographic imaging enables the visualizations of objects beyond the visible range, which enables its use in many applications like autonomous cars, nightly footage, military, or surveillance. However, the high cost of manufacturing this type of camera limits the spatial resolution that it can provide. Real-World Super-Resolution (RWSR) is a topic that aims to solve this problem by using image processing techniques that enhance the quality of a real-world image by reconstructing lost high-frequency information. This work adapts an existing RWSR framework that is designed to super-resolve real-world RGB images. This framework estimates the degradation parameters needed to generate realistic LR and HR image pairs. The SR model learns the mapping between the LR and HR domains using the constructed image pairs and applies this mapping to new LR thermal images. The experiment results show a clear improvement in the perceptual quality, which surpasses the performance of the current SotA method for thermal image SR.

Preface

This report documents a Master's thesis at the Master's Programme in Vision, Graphis, and Interactive Systems (VGIS) at Aalborg University (AAU). The aim of this thesis is to investigate the possibility of using RGB-based Real-world image Super-resolution (RWISR) to achieve real-world thermal image super-resolution. The thesis starts with chapter Introduction that explains the motivation behind the project and introduces the initial problem formulation. Chapter Problem Analysis narrows down the initial problem formulation resulting in a final problem formulation at the end of the chapter. The Theory chapter explores some key concepts in the image processing field as well as the methodology that was utilized during the project. The Design and Implementation chapter explains the training details that were taken into consideration when training the utilized method. The Evaluation and Results chapter talks about the experiments that were carried out during the work and presents the results gathered from the experiments. The Discussion chapter discusses the results of the evaluation chapter and introduces some ideas that could be added to the project to improve its quality. Finally, the Conclusion chapter summarizes the overall conclusion of the thesis.

The citation style used during this thesis is the IEEE reference style. Meaning that the author's names are not always visible, but instead, the source is referred to using square brackets.

I would like to take this opportunity to thank my supervisors Kamal Nasrollahi and Andreas Aakerberg for the guidance and the feedback that helped me during the process of writing this thesis.

Aalborg University, June 3, 2021

Moaaz M. J. Allahham
<mallah16@student.aau.dk>

Contents

Acronyms

ANN Artificial Neural Networks. 25

BRISQUE Blind/Referenceless Image Spatial Quality Evaluator. 20

CNN Convolutional Neural Networks. 19, 28, 29

DL Deep-learning. 25

ESRGAN Enhanced Super-Resolution Generative Adversarial Networks. 30, 33, 34, 36, 37, 42, 47, 48

GT ground truth. 20

HR High-resolution. iv, 6, 7

IQA Image Quality Assessment. 16, 20, 47, 54, 57

LeakyReLU Leaky ReLU. 26

LPIPS Learned Perceptual Image Patch Similarity. 18–21, 47–49, 54

LR Low-resolution. iv, 6–8, 24

MOR Mean-Opinion-Rank. 54

MOS Mean-Opinion-Score. 54

NIQE Natural Image Quality Evaluator. 19

ORB Oriented FAST and Rotated BRIEF. 37, 38

PIQE Perception based Image Quality Evaluator. 19

PSNR Peak-Signal-to-Noise-Ratio. 17, 18, 20–22, 47–49, 53

RealTISR Real Thermal Image Super-Resolution. 48, 49, 53, 54

ReLU Rectified Linear Unit. 26

RGB Red-Green-Blue. iv, 5, 6, 12, 13, 15, 16, 20, 21, 23, 24, 42, 54, 57

RRDB Residual-in-Residual Dense Block. 33

RWISR Real-world image Super-resolution. vii

RWSR Real-World Super-Resolution. iv, 6, 8, 9, 11, 15, 20, 24, 29, 57

SotA state-of-the-Art. iv, 6, 9, 10, 12, 15, 20–25, 47, 48, 50, 53, 54, 57

SR Super-resolution. iv, 5–8

SSIM Structural Similarity index. 18, 20–22, 47–49, 53

TherISuRNet . 48, 49, 53, 57

Chapter 1

Introduction

We have probably all seen one of those movies, where investigators try to identify a criminal in very low-quality surveillance footage. The next they usually do is that they seek the help of an IT specialist that, with the use of some image processing tools reconstructs a very enhanced image of the person from that footage. Some of us might think that this is just a science fiction movie, and the zoom and enhance technology is not reality. However, this is a topic that has been heavily researched for the past two decades or so and is referred to as Super-resolution (SR). Super-resolution has been an attractive research topic for many years, and it has been used in real-life applications like regular video information enhancement, surveillance, medical diagnosis, satellite, and aerial imaging, and other applications that require image resolution enhancement [51, 32]. In recent years, thermal imaging has grown considerably and is being used in various domains where a typical RGB camera can not get the job done, like nightly footage, surveillance, or in autonomous cars. However, thermal images generally have some shortcomings like insufficient details and blurred edges, and most importantly considerably low-resolution. This makes it too hard to observe the structure and recognize objects in an image as illustrated in figure 1.1.

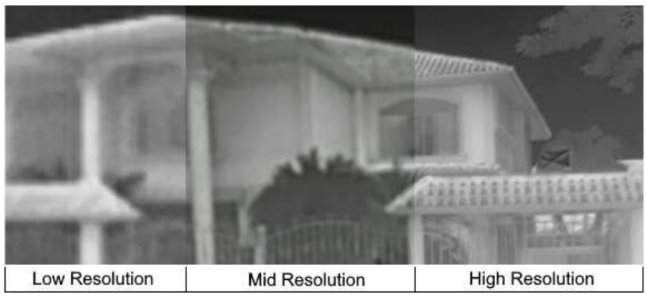


Figure 1.1: An illustration of the significant difference in structural details observed in three images of the same building captured with three different resolutions. Figure from [35].

However, having a thermal camera that is capable of capturing high-resolution images is not as affordable as using RGB cameras. This is caused by the high cost of manufacturing this type of camera. Thermal cameras use a larger sensor due to the larger individual pixels necessary to capture the infrared light that has another wavelength than the light captured by normal RGB cameras. With a larger sensor, a larger lens is needed, which drives the price of these cameras very high. Those lenses are most often made of Germanium, which is an expensive material that can reflect most of the visible light and passes the infrared light. Therefore, having a higher sensor size leads to larger lenses that are much more expensive than the ones used in normal RGB cameras [17]. Even the most expensive thermal cameras, which can vary from $200.00 to more than $20000.00 [36], still can not deliver sufficient resolutions. To the best of our knowledge, the highest resolution that a thermal camera can provide as for today is 1920×1200 pixels for the [42]. Getting back to the problem, enhancing real images captured by thermal cameras is therefore important. However, although increasing the resolution of a thermal image with an image processing algorithm would not compensate for the true information that is not captured by the camera's sensor, having an enhanced and higher resolution image makes it easier to recognize objects and structure in an image. The efficiency of this process can be improved by taking advantage of computer vision techniques that can assist in enhancing these images. Many methods were developed to perform image Super-resolution, however, most of these methods perform poorly when used on real LR images. This is because they follow the approach of downsampling high-quality images to construct Low-resolution (LR) and High-resolution (HR) pairs and then they super-resolve the LR image to match the HR image quality. Such methods fail when given a real-world image as the degradation process is unknown. Therefore recent studies have been working on developing methods that would be more robust to previously unseen real-world images that are acquired directly from cameras with unknown degradation parameters. This RWSR issue also applies to the thermal imaging domain, making it an interesting area to investigate since it has not been widely explored. Hence, the goal of this project is to explore the state-of-the-Art (SotA) SR algorithms that deal with RGB images and investigate its usability in the thermal imaging domain, and explore the possibility of tuning these methods to fit the thermal domain.

1.1 Problem Formulation

Based on the current knowledge about the given problem, the initial problem formulation was formulated as follows:

Can the recent advancements in RGB-based real-world super-resolution be applied to the thermal image domain with the goal of improving the perceptual quality?

Chapter 2

Problem Analysis

The goal of this chapter is to narrow the initial problem formulation down, to formulate the final problem formulation. The chapter will include a deeper dive into the different image processing techniques used to handle the super-resolution problems, and explore the different related studies that have been done in this field.

2.1 Traditional VS Deep-Learning Super-resolution

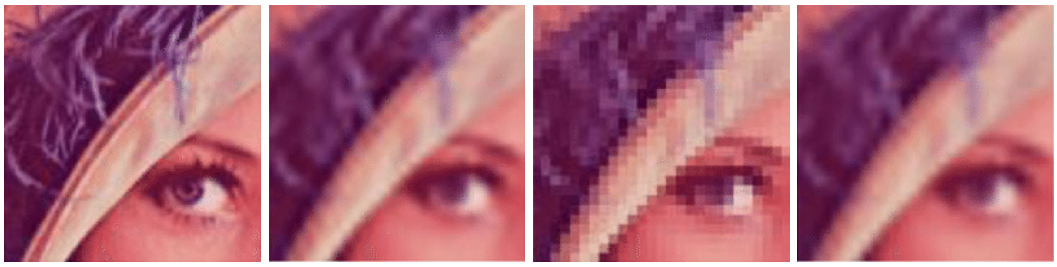
Super-resolution is the process of constructing one or more high-resolution images from their low-resolution counterpart. This means increasing the resolution of an image by increasing the number of pixels and enhancing details by enhancing high-frequency components. But, SR is an ill-posed problem, since there exist many HR images that correspond to a single LR image. Super-resolution methods can be classified based on the employed domain (Frequency, Spatial), the number of the LR images (Single, Multiple), and the actual reconstruction method [32]. In the *Frequency* domain, images are transformed into frequency distribution prior to processing the images. The frequency components of images are split into high-frequency that correspond to edges, and low-frequency that correspond to smooth regions. Whereas in the *Spatial* domain, images are dealt with as matrices.

Traditional methods have been around for decades now, however, these methods have been outperformed by their deep learning-based counterparts[49]. There are different traditional SR methods, but the standard and most popular methods that have been used are:

- **Nearest neighbour interpolation:** replicates the pixels in the super-resolved image as it scales up.
- **Bilinear interpolation:** takes the weighted average of the 4 surrounding pixels to calculate its interpolated value.

- **Bicubic interpolation** takes the weighted average of the 16 surrounding pixels to calculate its interpolated value.

As seen in figure 2.1 it is possible to see differences in the performance between the mentioned traditional SR methods.



(a) Original. **(b)** Bicubic. **(c)** Nearest. **(d)** Bilinear.

Figure 2.1: An example showing the performance of the most common traditional SR methods.

However, researchers have proposed a wide variety of models that use deep-learning to estimate the extra information to be added to a LR image to generate an image with a higher amount of high-frequency details. These models are split into mainly two categories, supervised SR an unsupervised SR.

Supervised SR approaches focus on training the models using pairs of LR images and their corresponding HR ones. Those LR-HR image pairs are typically obtained by synthetically reducing the resolution of the HR image using Matlab *Imresize* function, which applies bicubic interpolation with anti-aliasing on the given image[49]. This is done because it is physically difficult to obtain a pair of LR-HR images of the same scene with different resolutions. The SR models are then trained to reverse the process of descaling the image by trying to reconstruct the original HR image given the LR one. By doing this, the models learn the relationship between the LR and HR domains and then use this knowledge to super-resolve a new given LR image. The supervised-SR approach performs well when applied to LR images that are synthetically produced using a degradation model that is similar to the one used during training. However, supervised SR algorithms perform poorly when applied to real LR images because real images are affected by external factors that are neglected when synthetically generating LR images, such as blurring, noise, and compression artefacts[49, 41]. These factors can even vary between images taken by the same camera, where lighting conditions might differ. Therefore, the degradation parameters are unknown given real LR images. This is why researchers are leaning more and more towards unsupervised SR, or so-called Blind super-resolution, approaches to build algorithms that would potentially be more robust when it comes to real-world images. RWSR falls under this category, where researchers address this problem in different ways. Some [11,

41, 30, 14] try to exploit the internal image information by utilizing the similarities within the same image. Some others[23, 48, 46], try to adapt the HR images' domain by learning its characteristics, like high-frequency information, and apply it to the super-resolved images to make it more realistic.

2.2 Related work in Real-world Super-resolution

The main focus of this section is to explore the current SotA algorithms within RWSR.

2.2.1 Zero-shot Methods

In 2017, ZSSR[41] was introduced as the first blind SR algorithm (self-learned-based) that performed SR on LR real-world images without relying on any prior image examples or prior training. Instead, ZSSR trains an image-specific CNN using the recurrence of small patches across different scales within the same image at test time. This was done by downscaling the test image to smaller versions of itself, then applying data-augmentation (rotations/flip) to the smaller versions to fulfill the need of having multiple examples as a training dataset. The image-specific CNN learns to reconstruct the original LR image using the downscaled examples, then they finally apply the trained CNN to the original test image to construct the desired HR output. The overall structure of the ZSSR algorithm can be seen in figure 2.2. ZSSR outperformed external-based SotA methods in some regions when tested on images with salient recurrence of information. A drawback of ZSSR is the fact that the learning process fully depends on the internal information in the test image, which makes it require thousands of back-propagation gradient updates. This yields slow testing time as well as poor results in some regions compared to other external-based methods[44].

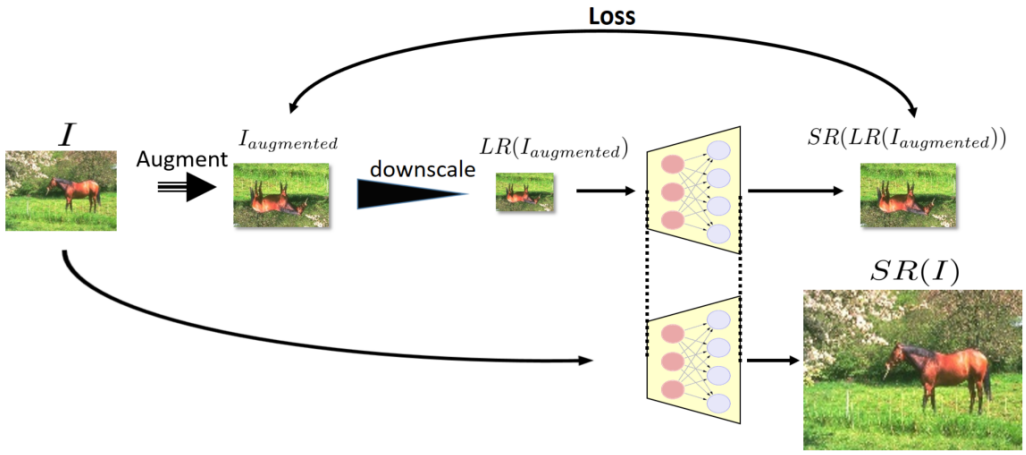


Figure 2.2: The overall structure of the ZSSR algorithm. Figure from [41]

"Meta-Transfer Learning for Zero-Shot Super-Resolution" (MZSR)[44] is another zero-shot algorithm that is heavily inspired by ZSSR. The authors of MZSR utilize the powerful parts of ZSSR and improve upon it by introducing the concept of Meta-Transfer learning. Without diving deep into how meta-learning works, the idea behind it is to make the model adapt fast to new blur kernel scenarios by adding a meta-training step, then utilize transfer-learning by pre-training the SR network using a large-scale dataset DIV2K[1]. The combination of Meta-transfer learning and ZSSR exploits both the internal (the test image) and external (the DIV2k) information. The main advantage that was introduced in the MZSR work, is the flexibility and fast running time compared to the ZSSR method, as well as outperforming other supervised SotA algorithms such as CARN[2] and RCAN[53]. Figure 2.3 shows an overview of the different learning steps involved in the MZSR method.

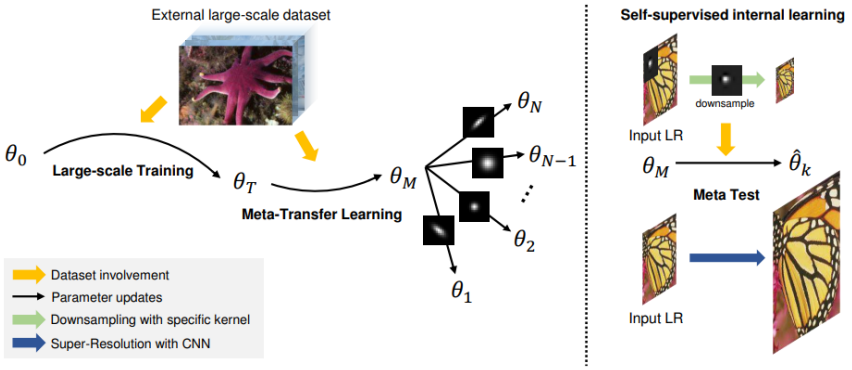


Figure 2.3: The overall structure of the MZSR algorithm. Figure from [44]

Dual Super-resolution (DualSR)[14] is another self-learning-based approach that addresses the RWSR problem in a similar way to the way it was addressed in the ZSSR work, where they learn the image-specific LR-HR relations by training their proposed network at the test time using patches extracted from the test image. Their proposed network is split into mainly two parts as shown in figure 2.4, the downsampler which learns the degradation process using a generative adversarial network(GAN), and an upsampler that learns to super-resolve the LR image. Both the up-sampler and down-sampler are trained simultaneously by improving each other using the cycle-consistency loss, the masked interpolation loss, and the adversarial loss.

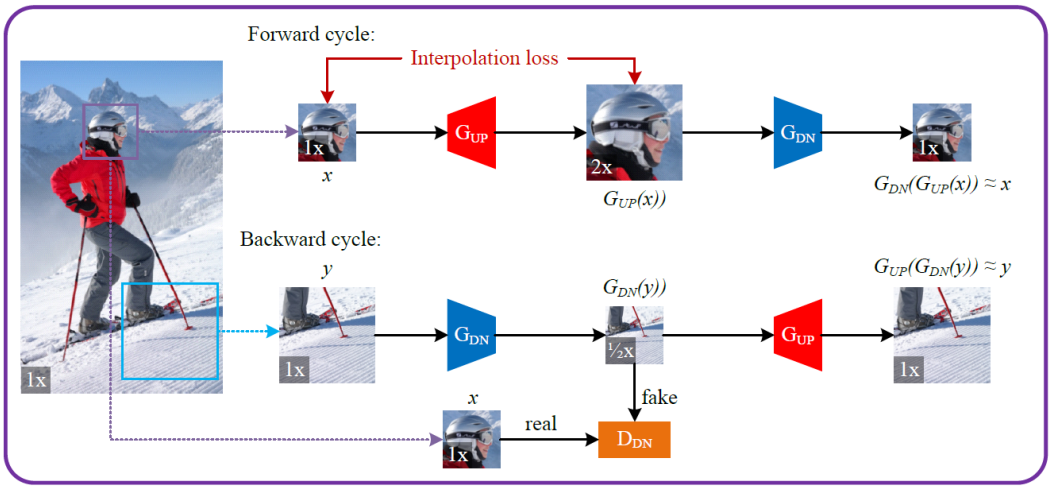


Figure 2.4: The overall structure of the DualSR algorithm. Figure from [14]

2.2.2 Learned Degradation based Super-resolution

Many supervised SR approaches make the assumption that LR images are a bicubicly downscaled version of their HR counterpart, and that Gaussian noise is usually used to simulate the sensor noise. However, these approaches fail when tested on real images because those images were not degraded using ideal degradation operation (bicubic kernel + Gaussian noise). For this reason, Fritsche et al. [16] introduced DSGAN(the winner of AIM2019 RWSR challenge[31]), which is a GAN network that learns to generate the appropriate LR images, which have the same corruptions as the original HR images. DSGAN inspired Umer et al.[46] to build a Super-Resolution Residual Convolutional Generative Adversarial Network (SRresCGAN) that exploits the power of DSGAN by combining it with another GAN network that super-resolve the degraded images generated by DSGAN. A full overview of the networks is shown in figure 2.5. The authors proposed their

work at the NTIRE2020 challenge[31] and were able to achieve results that are competitive with other SotA methods.

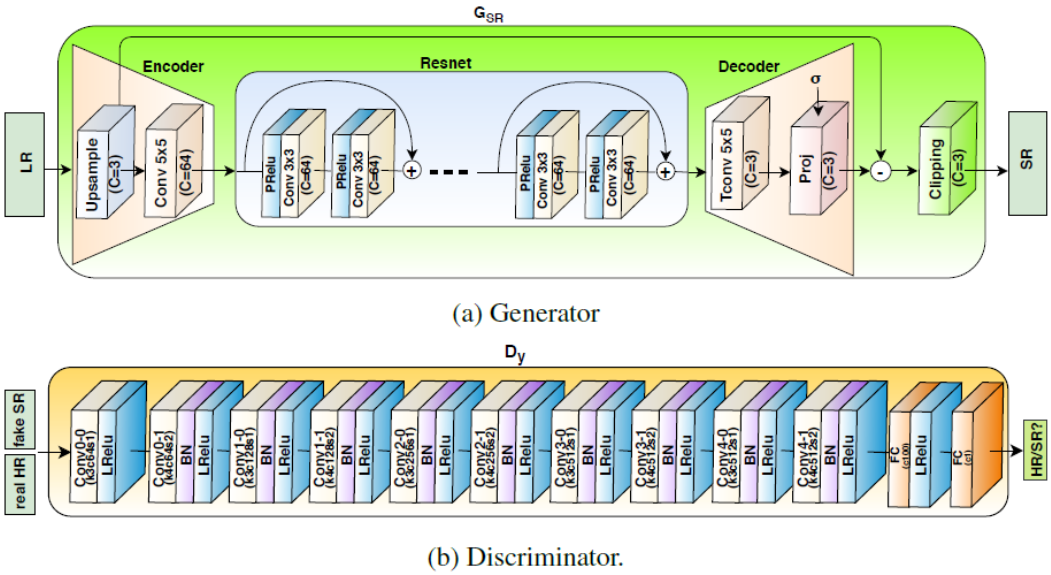


Figure 2.5: The SRresCGAN network architecture that was proposed by [46]. Figure from [46]

Bell-Kligler et al.[5] introduced another realistic degradation method KernelGAN, an image-specific Internal-GAN, which trains solely on the LR test image at test time and learns its internal distribution of patches. The generator of the network is trained to produce a lower resolution image such that the network's discriminator can not distinguish between the patch distribution of the generated image and the patch distribution of the original LR image. Ji et al.[23] proposed their method RealSR, which is divided into two stages. They first use KernelGAN to estimate the degradation from the real data and use it to construct the LR images, and then they train an SR model based on the constructed data. The degradation process in the RealSR method consists of two steps, first, they utilize KernelGAN to build a pool of kernels, and then they extract high-frequency noise patches from the original real-world image. The noise patches are meant to be used to compensate for the lost high-frequency information that is lost during the downsampling process. After having the kernel pool and the noise patches, the LR images are constructed by downsampling the original image using a randomly picked kernel, and then apply noise injection using the extracted noise patches. Finally, they train an SR model that is based on ESRGAN[48], using the constructed paired data. RealSR method was the winner of the NTIRE 2020 challenge [31], and by the time of doing this work, RealSR is considered to be the SotA in the real-world super-resolution field for RGB images. An overview of the degradation

process that RealSR used to construct the image pairs can be seen in figure 2.6.

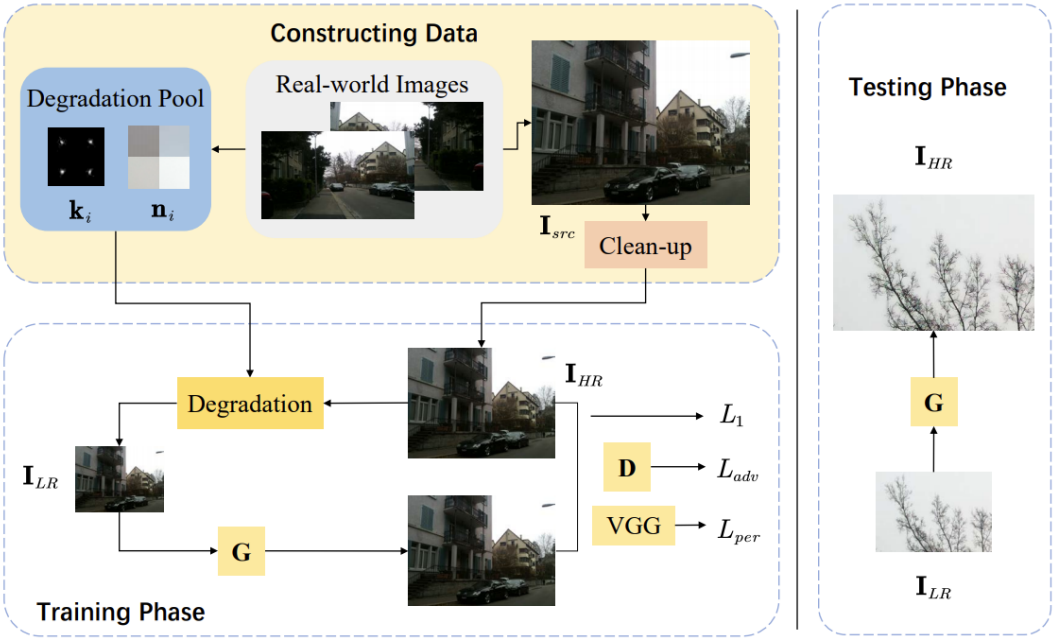


Figure 2.6: The RealSR degradation proposed by [23]. Figure from [23]

2.3 Related work in Thermal Real-world Super-resolution

All the methods mentioned in 2.2 are examples of super-resolution methods that deal with images in the RGB spectrum. However, there are only a few studies that developed methods for super-resolving LR thermal images. This section will be about exploring the available thermal SR approaches.

A study was conducted by Cho et al.[8] where they tried to enhance thermal images by training a CNN using different image spectrums aiming to find the best representation that would fit the thermal domain. They found that a grayscale trained network provided the best enhancement. Lee et al.[27] proposed a similar CNN-based on enhancement for thermal images, where they evaluated four RGB-based domains with a residual-learning technique. That improved the enhancement in comparison to the previous work by [8]. Rivadeneira et al.[9] was motivated by the two previously proposed methods, so he proposed Thermal Enhancement Network (TEN), which was the first CNN-based method to be trained specifically using thermal dataset unlike the two previous proposals by [27, 8]. TEN was based on the SRCNN model[13] that utilizes the residual net and dense connections technique that is shown in figure 2.7. TEN was able to outperform

the previously proposed methods, which was due to training the network using thermal images instead of RGB-based domains.

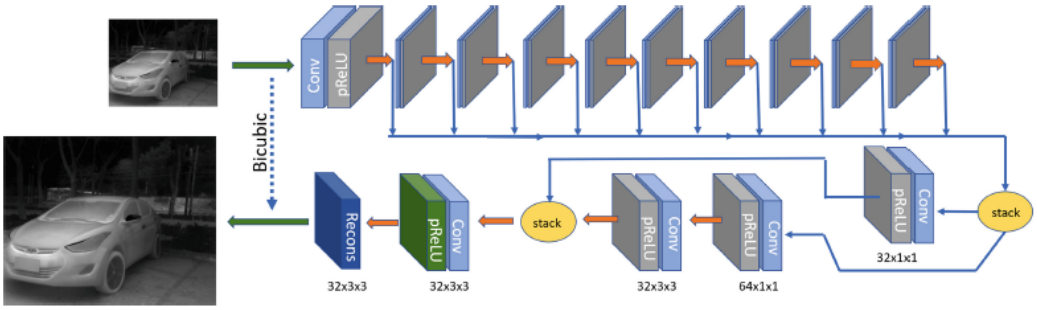


Figure 2.7: The overall architecture of the TEN method that was proposed by [9].

Recently, Rivadeneira et al.[36] proposed another thermal SR method that is based on the well-known CycleGAN[54] architecture. The proposed method was designed specifically for thermal images. Two-way generative-Adversarial-network (CycleGAN) is a technique that is used to map information from one domain to another. So the authors of [36] used the CycleGAN network to map information from the LR domain to the HR domain. They trained their proposed network, which can be seen in figure 2.8 to perform x2 scale SR following two scenarios, LR to medium-resolution (MR) and MR to HR. It is worth mentioning that they trained the network on a dataset that was proposed at the same work, and will be explained in more detail in section 2.4.

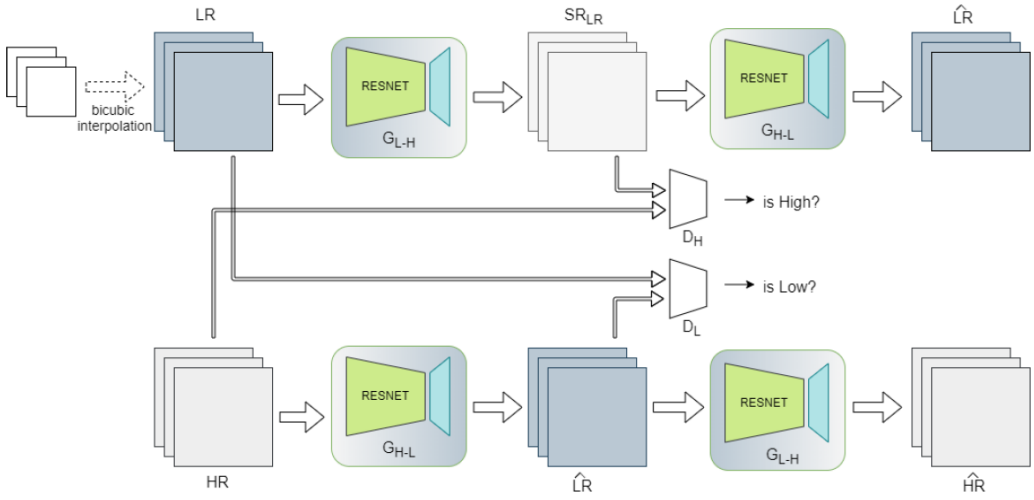


Figure 2.8: The CycleGAN network architecture that was proposed by [36].

Chudasama et al.[10] proposed another method to super-resolve thermal images by progressively upscaling the LR test image to obtain the final SR image. They achieve different upscaling factors (x2,x3, and x4) by applying residual learning. The TherISuRNet network consists of 4 main modules, low-frequency feature extraction modules, high-frequency feature extraction modules, second high-frequency feature extraction modules, and finally an image reconstruction module that is responsible for reconstructing the final SR image. They measured the performance of their proposed method by comparing its performance to the most common SotA methods [30, 53, 9, 29, 34] and bicubic interpolation, and they were able to surpass all the other methods when testing on thermal images. An overview of the entire architecture can be seen in figure 2.9. TherISuRNet was the winning method for the Thermal Image Super-Resolution Challenge PBVS 2020, which makes the TherISuRNet the SotA method for the thermal image SR domain.

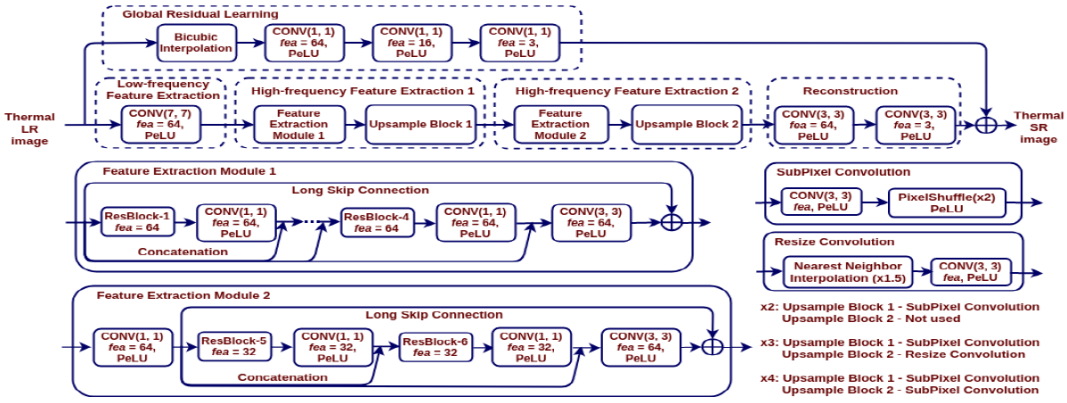


Figure 2.9: The TheISuRNET network architecture that was proposed by [10].

2.4 Datasets Exploration

There is a wide variety of available datasets that consist of RGB images and could be used for RWSR methods that are designed for RGB images. However, only a few datasets were built using real-world thermal images. This section shows an overview of the current benchmark thermal image datasets. It also explores the possibility of utilizing some of these datasets later in this work for both training and evaluating the performance of the proposed work.

PBVS[36, 35] This dataset consists of a set of 1021 thermal images, that were acquired using three distinct thermal cameras using three different resolutions, resulting in 3063 thermal images (low-mid-high resolution). The dataset contains a mixture of indoor and outdoor footage at various lighting conditions at different

points of time during the day. Table 2.1 shows an overview of the cameras' parameters that were used to build this dataset. The cameras were mounted on a rig while minimizing the distance between them to acquire three identical scene images.

Image Description	Camera Brand	Resolution
Low-Resolution	Axis Domo P1290	160×120
Mid-Resolution	Axis Q2901-E	320×240
High-Resolution	FC-632O FLIR	640×512

Table 2.1: This table shows the cameras that were used to collect the PBVS dataset, with their corrosponding resolutions (Note: The high-resolution images were cropped to 640×480)[36].

KAIST[20] is an annotated multispectral pedestrian dataset that is mainly used as a benchmark dataset for pedestrian detection problems. It consists of 95k color-thermal pairs of urban traffic environment. The dataset was acquired using a stereo-camera setup that consisted of three main parts, RGB camera (PointGrey Flea3), thermal camera (FLIR-A35), and beam-splitter hardware that was used to physically align the footage of the two cameras. The color camera has a resolution of 640×480 pixels, where the thermal camera has 320×256 pixels of resolution, and both cameras were recording at 20 Hz framerate. As the authors of the dataset stated in their paper, the data gathering was conducted by mounting the stereo-camera setup on the roof of a car and roaming the city to capture various traffic scenes day and night.

FLIR[15] consists of a combination of annotated thermal and non-annotated RGB images, which were acquired via a thermal and RGB cameras that were mounted on top of a vehicle. The dataset contains a total of $\tilde{1}$4.5k annotated thermal images that have been sampled from short videos. The videos were taken on streets under clear-sky weather conditions both day and night over a period of 7 months. The thermal images were acquired using the FLIR Tau2 thermal camera at a resolution of 640×512 pixels and a sampling rate of 1-2 images per second (original videos were captured at 30 frames per second).

2.5 Image Quality Assessment Methods

Image Quality Assessment (IQA) is a simple task that we as humans can accurately perform, at least to some considerable extent. However, this task is not as simple when it comes to computers due to the lack of perceptual ability that humans have. This made it challenging for image-processing researchers to assess the performance of image-processing tasks such as image enhancement, image restora-

tion, and image quality assessment in general. During the past two decades, few methods have been developed to fulfill this need. These methods can be split into mainly two categories, human perception-based subjective evaluation and quality metrics-based objective evaluation methods[50, 7], which can also be divided into three: no-reference-based (NIQE, PIQE, BRISQUE), reduced-reference-based and, fully-referenced based (PSNR, SSIM, LPIPS) methods. As the name suggests, subjective evaluation methods can vary in results based on personal preferences, and conducting such methods can often be costly and can not be automated. However, objective-based methods can be more convenient, although different assessment matrices can give inconsistent results in comparison to other matrices, or when compared to subjective-based methods. Super-resolution is a field that relies heavily on such quality assessment methods, therefore, for the purpose of this work, we will be looking at the most common methods that are currently being utilized and investigate the usability of some of these methods when it comes to the evaluation phase.

2.5.1 Reference-Based IQA Metrics

Peak-Signal-to-Noise Ratio

Peak-Signal-to-Noise-Ratio (PSNR) is an objective assessment method that is full-reference metric. It measures the difference between two given images by measuring the ratio between the maximum possible value (255 in our case) and the power of the noise that affects the quality of a given image. This method relies on having a set of two images, the processed image and the original image that is used as a reference. The main idea behind it is that the higher the PSNR is, the closer the processed image is to the original image[19]. However, this assessment method does not consider the perceptual quality of the image, and it is very sensitive to the pixels' values, as translating the image 1 pixel to any direction can result in a drastic drop in the PSNR value. Given a test image I and a ground truth image ref, PSNR is measured in decibel (dB) and calculated as shown in formula 2.1:

$$PSNR(ref, I) = 20log_{10}(\frac{L}{\sqrt{MSE(ref, I)}}) \tag{2.1}$$

Where MSE(Mean-Squared-Error) is as follows:

$$MSE(ref, I) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(ref(i,j) - I(i,j))^2 \tag{2.2}$$

Where M,N is the size of the image, and L represents the maximum pixel value (255 for 8-bit images).

Structural Similarity Index Measure

Structural Similarity index (SSIM) is a newer image quality assessment method that is fully-reference-based method. It is based on the combination of three factors, luminance (l), contrast (c), and structure (s). The first factor, luminance, is a comparison function that measures the closeness of two images' mean luminance (μ_{ref},μ_I). The second factor, contrast, is a comparison function that measures the closeness of two images' mean contrast (σ_{ref},σ_I). The final factor is the structure comparison function that measures the correlation coefficient between two images ref and $I(\sigma_{ref,I})$. The three factors are defined respectively as:

$$l(ref,I) = \frac{2\mu_{ref}\mu_I + C1}{\mu_{ref}^2 + \mu_I^2 + C1} \quad c(ref,I) = \frac{2\sigma_{ref}\sigma_I + C2}{\sigma_{ref}^2 + \sigma_I^2 + C2} \qquad s(ref,I) = \frac{\sigma_{ref,I} + C3}{\sigma_{ref}\sigma_I + C3} \tag{2.3}$$

Where C1, C2, and C3 are used to stabilize the division by avoiding null denominator. SSIM takes values in the range [0,1], where a value of 0 means no correlation between the two images and a value of 1 means that both images are identical[19]. SSIM is a metric that is less sensitive than PSNR, as it measures the textural structure of two given images. This makes SSIM a more preferred method than PSNR, especially in cases where the perceived quality is a key.

Mean Opinion Score

As mentioned before, image quality assessment is split into two main categories, objective and subjective, where computers can not assess the quality as well as a human can. For this purpose, subjective methods like mean-opinion-score(MOS) can be utilized. MOS in general is a numerical measure of human judged overall quality of an event or experience. This method has been widely utilized within the image-processing field in cases where other objective measures, such as PSNR and SSIM, can perform poorly in comparison to the human eye. MOS is expressed as a single number from 1 to 5 (1-Bad, 2-Poor, 3-Fair, 4-Good, 5-Excellent), that corresponds to an average opinion score of some observers that are given a set of images with various distortions/qualities. The observers are usually asked to rate each image by giving it a score that describes how close

Learned Perceptual Image Patch Similarity

Learned Perceptual Image Patch Similarity (LPIPS) is a full-reference-based learned metric that was found by Zhang et al.[52]. This metric measures the distance between two given images by looking at the $L2$ distance between the reference and the test images in a deep feature space. LPIPS outperforms the widely used PSNR and SSIM metrics. Zhang et al. found that deep network activations work well as a perceptual similarity metric that correlates well with human perceptual judgments.

The way they discovered that, is that they first created a dataset that contained a large number of images with had different types of corruption to it. Followed by that, they picked some off-the-shelf convolutional neural networks, which they used to classify the images in the created dataset by finding the most similar image pairs (original+corrupted). They found that most of these networks that were trained in a meaningful manner achieved similar scores to a similar human survey they conducted. Where they asked the participants had to perform the same task that the CNNs were asked to do, which was finding the most similar images in the built dataset. Networks architectures like VGG[43], AlexNet[38], SqueezeNet[21] provided similar performance and are the most used architectures when using the LPIPS metric. An illustration showing how the feature space distance is calculated can be seen in figure 2.10, where it can be seen how x and x_0 are fed into CNNs that create feature stacks. Those features get subtracted and then averaged over the spatial dimensions, and finally resulting in a single number that represents the distance between the two input images in the feature space.

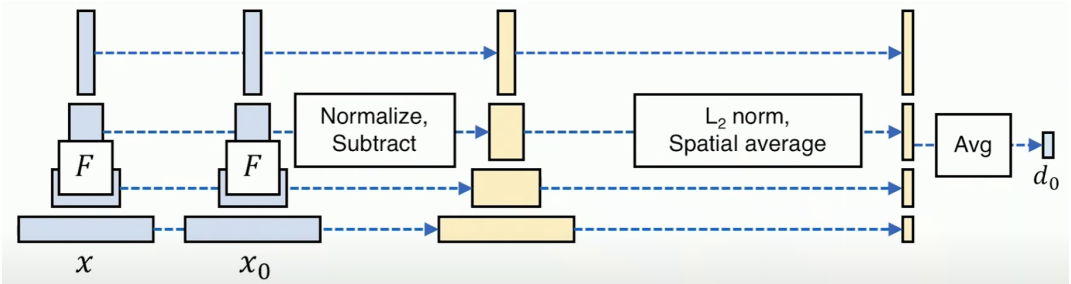


Figure 2.10: An illustration showing how LPIPS calculates the distance between two given images x and x_0. Figure from [52].

2.5.2 No-Reference-Based IQA Metrics

Perception based Image Quality Evaluator (PIQE) is a completely blind metric that does not require a trained model. Instead, it assesses the image quality by estimating block-wise distortion, by dividing the test image into non-overlapping blocks, and then it measures the local variance of each block to compute the final quality score. Natural Image Quality Evaluator (NIQE) is another blind metric that assesses the image quality without knowledge of anticipated distortions or human opinions of them. Unlike PIQE, NIQE uses a pretrained model that is trained on a dataset of pristine images. It uses a multivariate Gaussian model to fit quality features extracted from images. These features include parameters of the generalized Gaussian distribution and asymmetric generalized Gaussian distribution that characterize the behavior of image patches. Then the quality of an image is measured using the distance between the two Gaussian models fitting the evaluated image

and natural images that the metric was previously trained on. Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) is different from the last two metrics since it is pretrained on a dataset of images with known distortion parameters. The problem with this metric is that it can evaluate images that are similar to those it was pretrained on. Therefore, it is recommended to train a custom BRISQUE model from a set of quality-aware features, before using it to evaluate test images.

2.6 Performance Overview

After having looked at the current RWSR SotA methods for both RGB and thermal images SR, it is now time to evaluate the performance of some of these methods to draw a baseline for which method to be utilized during this project. The classic way to perform a comparison of SR methods is to usually super-resolve an image using each method and compare the reconstructed image to the ground truth. This is usually feasible for the supervised method that relay on image pairs, where the LR images are usually synthetically degraded, and the SR version of these images is then compared to the original image. However not as simple when working with RWSR methods since ground truth (GT) images are unavailable in the case of RWSR. However, even when having the GT images available, as in the PBVS dataset 2.1, those images were captured using different cameras with different resolutions, which introduced few challenges. Some of these challenges are misalignment, the sensor's intrinsic and extrinsic settings, light conditions, and different possible factors that may affect the acquired images. This makes it hard to compare SR images with GT images that were taken using a different camera. Hence, the performance of each method is evaluated using both reference and non-reference-based IQA methods. For the reference-based methods, PSNR, SSIM and LPIPS will be used. Where PIQE, NIQE, and BRISQUE will be used the non-reference-based methods used to assess the quality of the super-resolved images generated using each of the SR methods. Figure 2.11 shows a visual comparison of three different resolution thermal images from the PBVS dataset.
The results shown in the PBVS challenge paper [35] are unfortunately not possible to be replicated when comparing the RWSR methods. This is because the set of 10 images used for evaluation during the challenge are not available to the public. Therefore, we will be adapting their evaluation method that is shown in figure 2.12, but the PBVS validation set will be used for testing.
The quantitative comparison in terms of PSNR, SSIM and LPIPS measures of both the SotA thermal SR method and some of the SotA RGB SR are presented in table 2.2. The RealSR DPED, DF2K, DF2K_JPEG are different pretrained models, which were provided by the RealSR authors, and are pretrained using datasets that consist of RGB images. Looking at the quantitative results, it is hard to judge which method performs best in terms of PSNR, SSIM, and LPIPS, and this is because

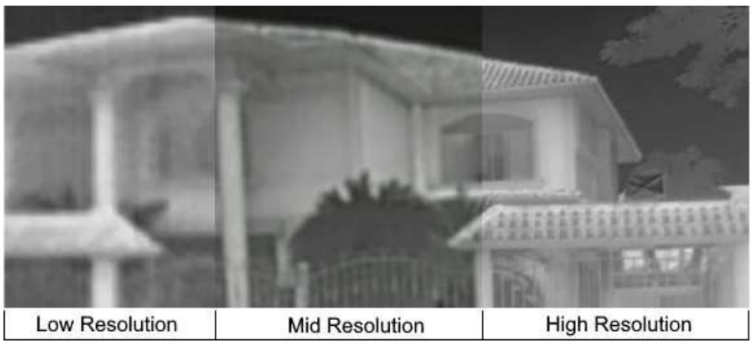


Figure 2.11: A comparison showing the visual differences between images acquired with three resolutions using three distinct cameras of the same scene[35].

these different metrics are not positively correlated. For example, if we compare the SRrescGAn and the RealSR-DF2K, we can see that each of the methods is best at one of the metrics. Where RealSR-DF2K is given the best PSNR values, and SRrescGAN gives the best SSIM and LPIPS values (excluding the TherISuRNet, as it is considered a baseline for comparison). TherISuRNet is considered to be the baseline for this comparison since it achieved the SotA performance for thermal imaging SR according to the results reported in the PBVS challenge paper[35].

	PSNR↑	SSIM↑	LPIPS↓	PIQE↓	NIQE↓	BRISQUE↓
Bicubic	20.11	0.70	0.4661	67.39	5.55	57.20
RGB SotA SR methods						
RealSR-DPED	18.64	0.51	0.4333	36.10	4.10	29.11
RealSR-DF2K	**20.26**	0.65	0.4219	**22.72**	3.54	41.66
RealSR-DF2K_JPEG	20.07	0.64	0.4011	19.34	**2.59**	**20.23**
DualSR	18.77	0.59	0.4328	56.48	4.18	43.03
ZSSR+KernelGAN	19.01	0.57	0.4404	60.79	5.71	46.14
SRrescGAN	19.98	**0.66**	**0.3416**	30.78	3.78	30.94
Thermal SotA SR method						
TherISuRNet	20.10	0.71	0.4273	88.69	5.20	55.34

Table 2.2: The quantitative comparison of the SotA methods in both the RGB and Thermal SR domains in terms of PSNR, SSIM and LPIPS.

Looking at figure 2.13, it is easier to observe the difference in performance between the different methods. It is fair to say, that the perceived quality can vary a lot based on the used method, where some methods, like DualSR and K-ZSSR, enhance the noise that originates from the camera used to acquire the Domo images and result in a negative impact on the overall quality of the SR images by introducing some hallucination and ghosting artefacts to the image. On the other hand, some methods, like RealSR-DPED, remove most of the high-frequency information

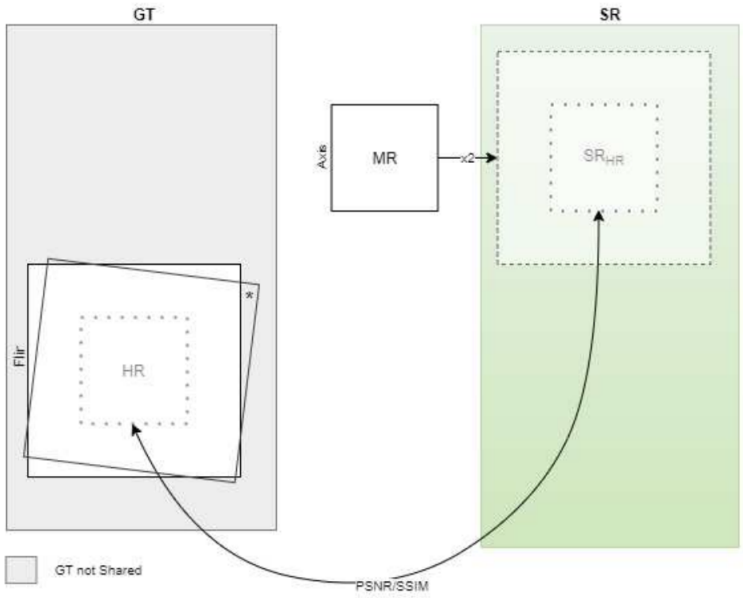


Figure 2.12: An illustration of the evaluation process performed during the PBVS challenge[35].

in the image, resulting in over-smoothed images. Looking at both table 2.2, we can see that the Bicubic method delivered one of the best results in terms of PSNR and SSIM. However, looking at the rest of the metrics that are driven by the perceptual quality, the Bicubic method delivered the worst results in comparison to the other SotA competing methods. Even though both Bicubic and TherISuRNet yield high PSNR values, they delivered over-smoothed images that lack high-frequency details. It is arguably fair to say that RealSR, in general, was one of the best performing methods in terms of visual quality, and quantitative results. Which makes it the most promising method to utilize in this work, and explore the possibility of tuning it to deliver better perceptual quality results than the TherISuRNet, and still maintain good fidelity that would match the target GT images.

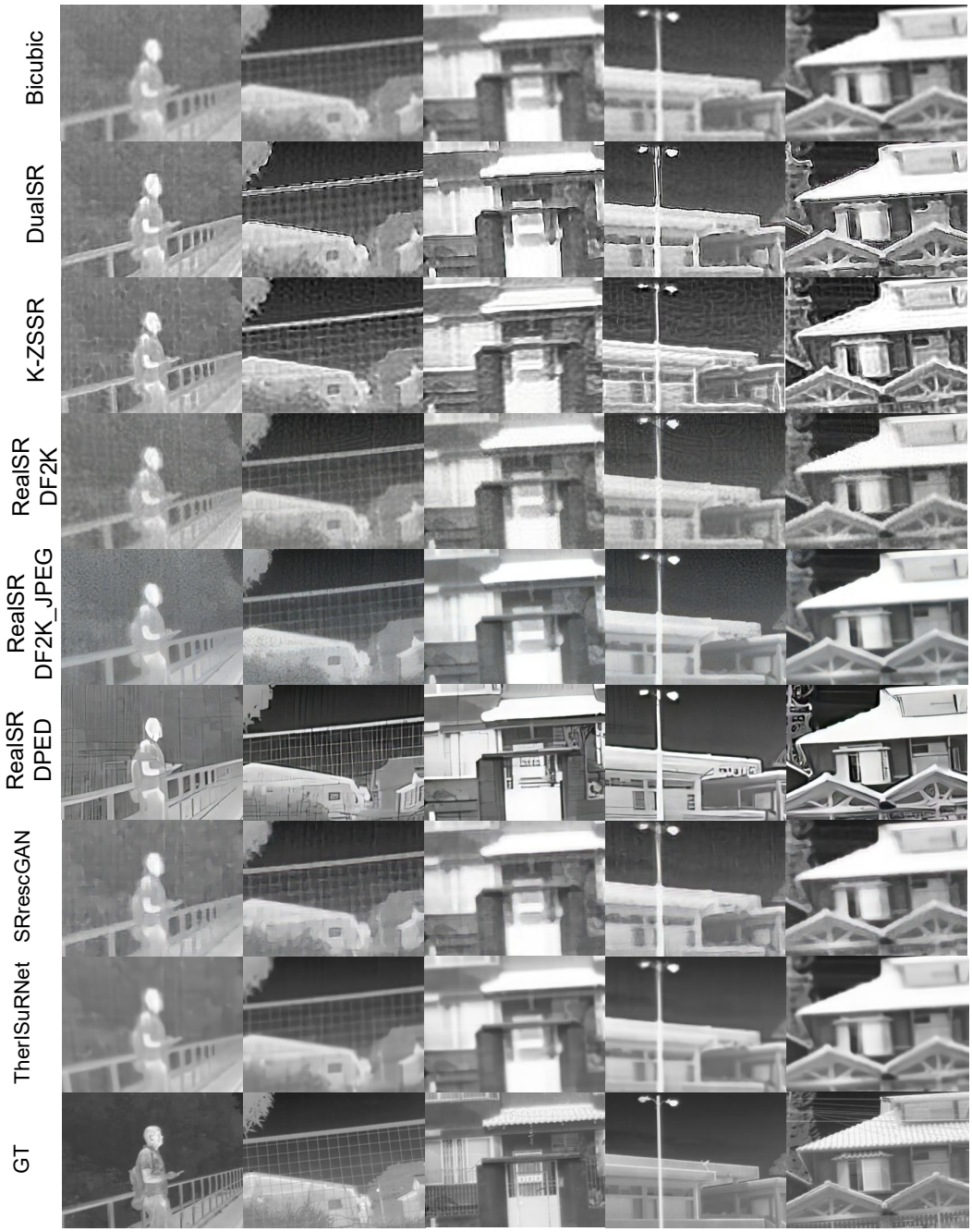


Figure 2.13: Qualitative comparison between the SotA methods in both the RGB and Thermal SR domains.

2.7 Summary

In this section, the possibility of using existing SR methods that deal with RGB images to be used with thermal images, was explored. Challenges when dealing with thermal images super-resolution were identified as well. Moreover, the TherISuRNet method was considered a baseline during this work, since it achieves SotA results in the thermal domain. RealSR was chosen as the target method to be tuned to fit the thermal domain. It was also chosen to use the PBVS dataset since it is the only dataset that offers image pairs with different naive resolutions of the same scene, which enables the possibility of assessing the super-resolved images in comparison to the ground-truth target images.

2.8 Final Problem Formulation

Based on the problem analysis, it was found that recent studies have proposed powerful deep learning-based algorithms that are able to achieve SotA performance on LR images. However, the majority of these algorithms have been developed to deal with visible RGB images, and their performance has not yet been explored in the thermal imaging domain. Since a very limited number of RWSR algorithms were designed specifically for thermal images, it was therefore interesting to investigate how SR algorithms that are designed to work with RGB images would perform on thermal images. It was found that the performance of RealSR[23] pre-trained on RGB images is comparable with the current SotA thermal SR method. Based on that, the problem formulation was narrowed down into the following:

Is it possible to surpass the quantitative and qualitative performance of the SotA thermal RWSR method by adapting the RealSR method to fit the thermal domain?

Chapter 3

Theory

This chapter will explore the theory behind the RealSR method to understand how to adapt to the thermal domain with to achieve better performance than the SotA thermal SR algorithm(TherISuRNet). A brief introduction into Deep-learning and artificial neural networks will also be introduced, as those topics are necessary for understanding the RealSR.

3.1 Artificial Neural Networks

Neural networks or so-called Artificial Neural Networks (ANN) is a piece of a computing system that is built to function like the human brain. The general concept when ANN was designed to utilize the process of training or learning rather than using handcrafted rules [12]. The building block of ANNs is called neuron (figure 3.1), which is designed to take input or multiple from a previous set of neurons forming a set of layers as seen in figure 3.2. At each neuron, a weighted sum is calculated by multiplying each input by a weight (w) that specifies how much impact each input will have on the final network's output. A bias (b) is then added to the weighted sum before passing it to an activation function (f) that calculates the firing rate of each neuron. Finally, the output (o)of the layer is calculated as follows:

$$o = f(\sum_{i=1}^{n} x_i w_i + b) \tag{3.1}$$

An activation function is chosen based on the specific problem one is trying to solve. There exist a number of activation functions, but the 5 most used functions are presented in table 3.1.

Binary Step: This function is suitable for binary problems, where the function can output a 0 or 1 based on whether the input value is lower or higher than a specific threshold.

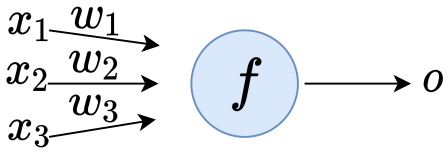


Figure 3.1: An illustration of a neuron that has 3 inputs x_1, x_2, x_3 associated with 3 weights w_1, w_2, w_3, activation function f and output o

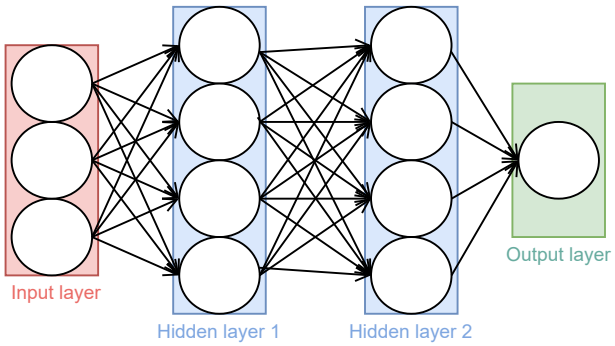


Figure 3.2: An example of an artificial neural network with 2 fully connected hidden layers.

Sigmoid: This function outputs values that are in the range $[0 - 1]$ and is used in the case of problems that require predicting a probability, like a multi-class classification problem.
Softmax: This function's output is very similar to Sigmoid, except that all resulting probabilities have to add up to 1, where the sum of Sigmoid's output can be above 1.
Rectified Linear Unit (ReLU): This function has been used within hidden layers and is probably the most used activation function. It passes the input values if they are above zero and rectifies values below 0 thereby forcing them to zero.
Leaky ReLU (LeakyReLU): This function has an identical result when compared to ReLU except that it outputs a small value for any given input instead of outputting zero for all the negative values.

3.1.1 Backpropagation

We mentioned in the previous section that the special part of neural networks is the ability to train or learn. This is usually done using an **optimizer** that tells the network how to adjust its weights to achieve the desired outcome. In a supervised learning situation, we usually look at the error of the network's output, by comparing it to the true class of the input. The idea is to backpropagate the error through the network to calculate the gradient descent (GD)[37], one way of doing it, that

Activation Function	Equation	Range
Binary	$f(x) = \begin{Bmatrix} 0, if x < 0 \\ 1, otherwise \end{Bmatrix}$	{0,1}
Sigmoid	$f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$	(0,1)
Softmax	$f(x) = \frac{exp(x_i)}{\sum_j exp(x_j)}$	(0,1)
ReLU	$f(x) = \begin{Bmatrix} 0, if x < 0 \\ x, otherwise \end{Bmatrix}$	$(0, -\infty)$
LeakyReLU	$f(x) = \begin{Bmatrix} 0.01 * x, if x < 0 \\ x, otherwise \end{Bmatrix}$	$(-\infty, \infty)$

Table 3.1: The most used activation functions [33].

can then be used to adjust the network to fit the results better. The amount of how much that should be adjusted each iteration is controlled by different hyperparameters, that can be set before the training starts. However, optimizers are usually used for this purpose.

3.1.2 Optimiser

The GD, calculated in the backpropagation, is as mentioned used to adjust trainable parameters in the network towards a correct output. The optimizer updates the weights by controlling how they are adjusted for each iteration.

For the GD there are typically defined three different categories: Batch-, Stochastic- and Mini-batch gradient descent. The batch GD usually referred to as GD, computes the derivatives to decide of how weights and biases should be adjusted. The 3 different versions of GD are based on how often these are calculated. In batch GD it is only done for each epoch, which is not memory efficient. The stochastic version calculates for each data point, where the mini-batch calculates for each mini-batch. This means that stochastic- and mini-batch GD converges faster and uses less memory.

Adam [**ADAM**] is also an optimization algorithm that can be used instead of the GD. Instead of keeping a constant learning rate, Adam adjusts the learning rate during the training and keeps a separate learning rate for each parameter. Typically you would start with a high learning rate and decay it during training as it is desired to reach global minima. However, there is a risk of getting stuck local minima, there is therefore a trade-off between high- and low learning rate. It is not only Adam that uses the adaptive learning rate a method only for this also exists called learning rate scheduler, which we will explain later.

3.1.3 Convolutional Neural Networks (CNN)

CNN is a subcategory of artificial neural networks, it takes its name from the mathematical linear operation between matrices. CNNs consist of multiple layers, including convolution layers, max-pooling layers, and fully connected layers, and are used to take an image as an input and assign different weights and biases to different parts of the image, in which we refer to as features, and then use them to differentiate one from another[3].

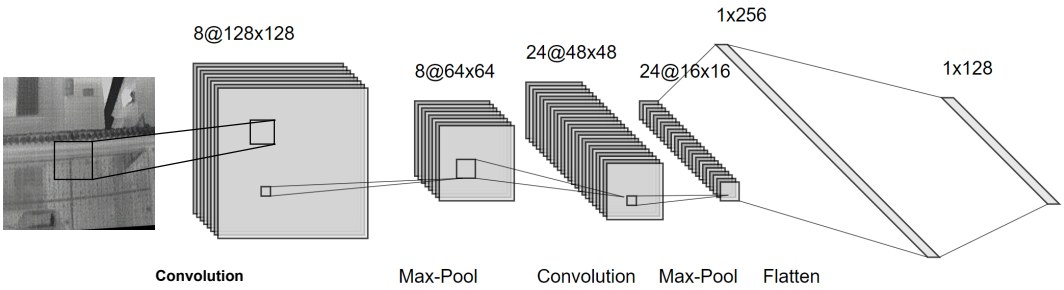


Figure 3.3: A simple LeNet[26] that illustrates the different types of layers used in a CNN.

Convolutional Layer

Convolution is a simple operation that is used to filter a given image by extracting its features using a feature detector resulting in a feature map. Feature maps summarize the presence of specific features in the input image. Generally speaking, this can be achieved by handcrafting a kernel that can preserve specific features, like edges. However, convolutional neural networks learn to calculate those feature detectors in a way that lets them extract feature maps that are important for the network. Figure 3.4 illustrates how the feature detector is used as a sliding window to iterate over the image.

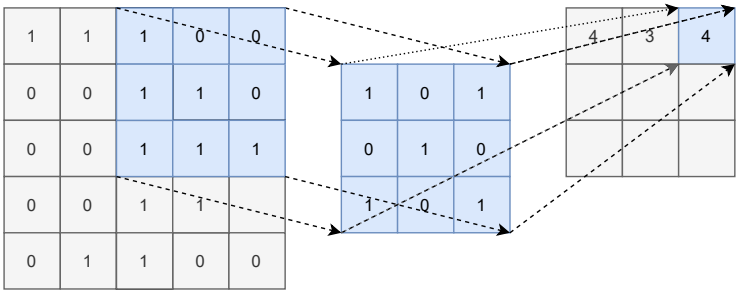


Figure 3.4: An illustration of the convolution operation with a stride step equal to 1.

Max Pooling Layer

Max pooling is another essential layer that is often used when building a CNN. This type of layer is used to reduce the size of the previous layer by preserving the important information and discarding the rest. Max pooling is performed in a similar way as the convolutional layer, where a sliding window iterates over a feature map resulting in a new smaller feature map. The amount of movement between each step of the kernel over the input image is referred to as the stride, and it is almost always symmetrical in height and width dimensions. For instance, figure 3.5 illustrate the pooling operation with a stride step equal to 2.

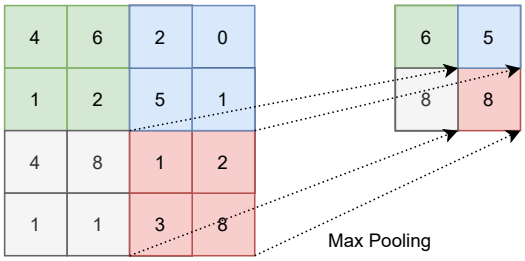


Figure 3.5: An illustration of the max pooling operation with a stride step equal to 1.

Flattening Layer

As the name suggests, this type of layer is used to flatten a feature map by converting a matrix with a size of n rows and m columns to a single-dimensional vector with a size of $m \times n$ dimension as shown in figure 3.6. This step is usually used to prepare the feature maps for the final layer in the network. For instance, the flattened layer can be connected to a final layer with two nodes that are used for binary classification.

3.2 RealSR

RealSR[23] is an unsupervised SR pipeline that was the winner of the Real-World Super-Resolution NTIRE2020 challenge[31]. This method was designed to overcome the challenges of real-world super-resolution. Different studies tried to artificially construct blurry and noise-added data with the aim of furtherly enhancing the robustness of an SR model, however, it was a requirement to have sufficient prior about blur and noise, which made the application of such methods limited. Some of the studies that we have explored in section 2.2 addressed some of these issues in different ways. However, according to the RealSR authors, most of these methods paid the price by increasing the interference time drastically, which moti-

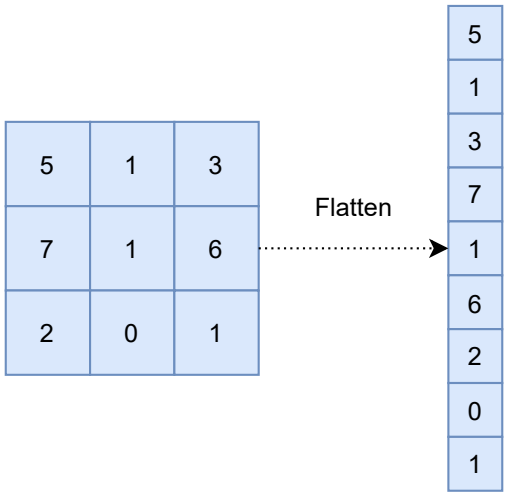


Figure 3.6: An illustration of the flatten operation.

vated the RealSR authors to propose their method that aims for explicitly estimating the blurring kernel and noise from the input images, and use it to construct realistic image pairs that can be used to train a more robust SR model. The authors then utilize ESRGAN[48] to generate images that are upscaled with a factor of 4 using the constructed image pairs.

3.2.1 Realistic Degradation using KernelGAN and Noise Injection

In general, KernelGAN is an image-specific Internal-GAN[40] that trains solely on a given LR image at test time and learns its internal distribution of patches. Its generator (G) is trained to generate a downscaled version of the given image, such that its discriminator (D) can not distinguish between the patch-distribution of the generated image and the patch distribution of the original image. D is trained to output a heat map, referred to as *D-map*, indicating for each pixel how likely is its surrounding patch to be drawn from the original patch-distribution. The loss is the pixel-wise MSE difference between the output *D-map* and the label map. Where the label map is all the ones in the crops extracted from the original image, and all the zeros in the crops extracted from the downscaled image[5]. Looking at figure3.7, it is possible to see how the entire pipeline works, and how the *D-map* looks like.
According to the RealSR authors, the estimated kernel needs to meet the following:

$$arg\min_{\mathbf{k}} \|(I_{src} * k) \downarrow_s - I_{src} \downarrow_s\|_1 + \left|1 - \Sigma k_{i,j}\right| + \left|\Sigma k_{i,j} \cdot m_{i,j}\right| + |1 - D((I_{src} * k) \downarrow_s)| \tag{3.2}$$

Where $(I_{src} * k) \downarrow_s$ is a downsampled LR image with kernel k, and $(I_{src} * k) \downarrow_s$ is the same LR image downsampled with the ideal kernel, therefore encouraging the downsampled image to preserve low-frequency information. The second term is

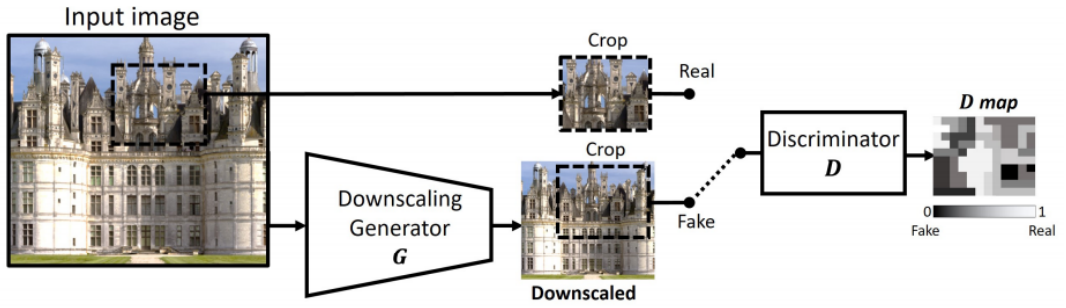


Figure 3.7: KernelGAN trains on patches of a single real image, where D tries to distinguish between patches cropped from the real image and patches of the image generated by G. G learns to fool D by generating downscaled images that have the same distribution of images as the input real image[5].

to constrain k to sum to 1. The third term is to penalty boundaries of k. The fourth and last term D is to ensure consistency of the source domain.

To better understand how the kernel degradation process is achieved, let's assume an LR image is obtained following the degradation method:

$$I_{LR} = (I_{HR} * k) \downarrow_s + n \tag{3.3}$$

Where k denotes the kernel used to blur the image, n denotes the noise added to the image, and s denotes the downscaling factor. Instead of using ideal kernels (e.g. Bicubic downscaling), RealSR explicitly utilizes KernelGAN to create a pool of kernels that can be used to construct the LR-HR image pairs.

Architecture

The architecture shown previously in figure 2.6 illustrates how the overall method takes an input image and outputs a *D-map*. However, we can see that the GAN network consists of two main parts, the discriminator D and the generator G. Figure 3.8 shows the architecture of the D used in the KernelGAN network.

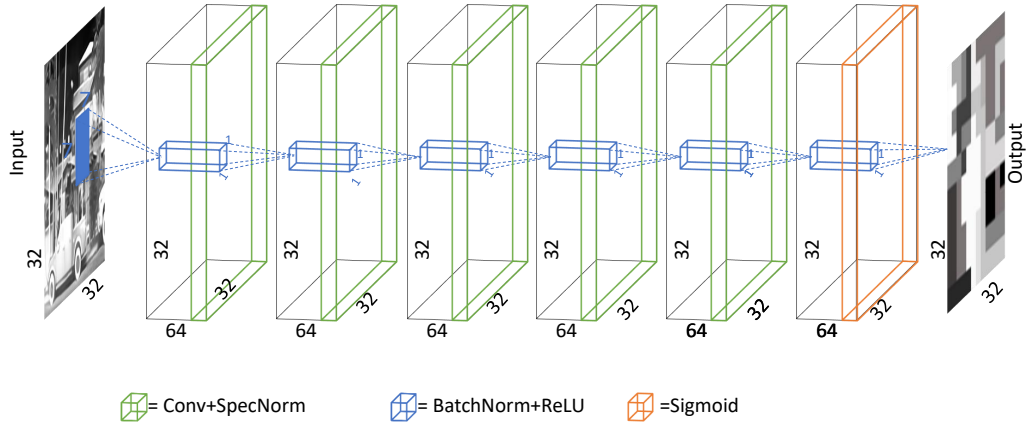


Figure 3.8: Convolutional Patch Discriminator: An input crop of size 32×32 that is convolved with a 7×7 convolutional filter followed by six 1×1 convolutions + Spectral normalization + ReLU activation, except the last hidden layer using Sigmoid for activation, and it outputs a 32×32 D-map with pixel values in the range [0,1].

The other part of the KernelGAN network is the generator G that constitutes the downscaling model. The generator network shown in figure 3.9 consists of 5 hidden convolutional layers with 64 channels each and those layers are without non-linear activation functions. The first 3 filters are 7×7, 5×5, 3×3 and the rest are 1×1, which results in a receptive field[1] of 13×13.

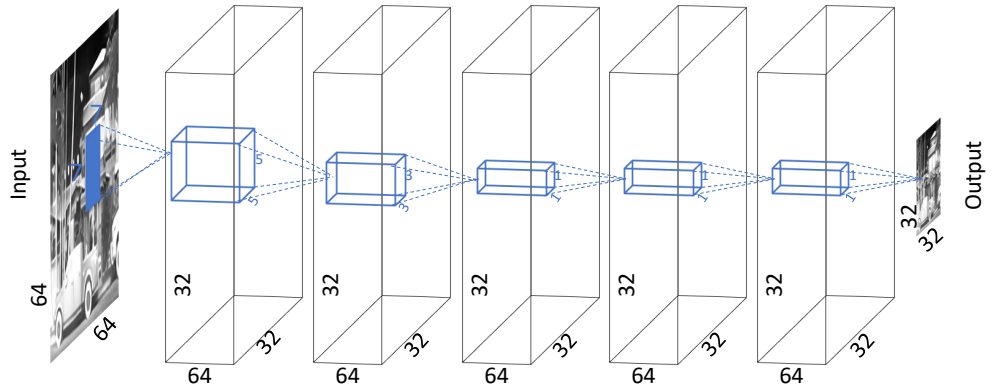


Figure 3.9: The downscaling Generator: An input image is convolved through 5 hidden convolutional layers without any activation functions. The output of the generator is a downscaled version of the input image with a downscaling factor of 2.

[1] "**Receptive field** is defined as the region in the input space that a particular CNN's feature is looking at (i.e. be affected by)"[18]

Noise Extraction

In addition to creating the kernel pool, RealSR introduces a simple filtering rule for extracting noise patches from source images. The idea behind extracting these noise patches is to inject them into the degraded images (LR) so both HR and LR have similar noise distribution. The filtering rules used to choose the relevant noise patch is using the following rule:

$$\sigma(n_i) < v \tag{3.4}$$

Where $\sigma(\cdot)$ denotes the function used to calculate the noise variance, and v is the max value of variance.
Having created a series of kernels $\{k1, k2 \cdots k_l\}$ and a series of noise patches $\{n1, n2 \cdots n_m\}$, the degradation process is performed as follows:

$$I_{LR} = (I_{HR} * k_i) \downarrow_s + n_j, i \in 1, 2 \cdots l, j \in 1, 2 \cdots m \tag{3.5}$$

Where s denotes the sampling stride. Figure2.6 summarises how the image pairs are constructed following the workflow explained above. It is also worth mentioning that the *Clean-up* process is a simple method that is used to generate sharper images that are noise-free by applying a bicubic downsampling operation which tends to make images sharper. Another point that can be noticed, is that the noise injection step is not visible in the image-pairs construction pipeline as the noise injection is combined during the degradation training phase. This, according to the authors, makes the noise more diverse and regularises the SR model to distinguish content from noise.

3.2.2 Super-Resolution Model

As mentioned in section 2.2.2, RealSR consists of two phases, the first is constructing the realistic image pairs using KernelGAN and the second phase is training the SR model that is based on ESRGAN with some modification. To understand the RealSR SR backbone, we need to first understand how ESRGAN work, and then understand how RealSR adjust the ESRGAN architecture to make it more flexible to different image sizes.

ESRGAN

ESRGAN[48] stands for Enhanced Super-Resolution Generative Adversarial Networks, which is a generative adversarial network that is based on SRGAN. SRGAN is a GAN network that is capable of generating realistic textures during single-image SR, where its discriminator aims aim is to make its prediction based on perceptual quality. However, ESRGAN improves SRGAN by adjusting the SRGAN architecture where they introduce their Residual-in-Residual Dense Block (RRDB)

without batch normalization, as well as improving the SRGAN discriminator by making it judge whether an image is more realistic than another rather than judging whether an image is real or fake. ESRGAN improvement over SRGAN resulted in sharper and more visually pleasing results[48] as shown in figure 3.10.

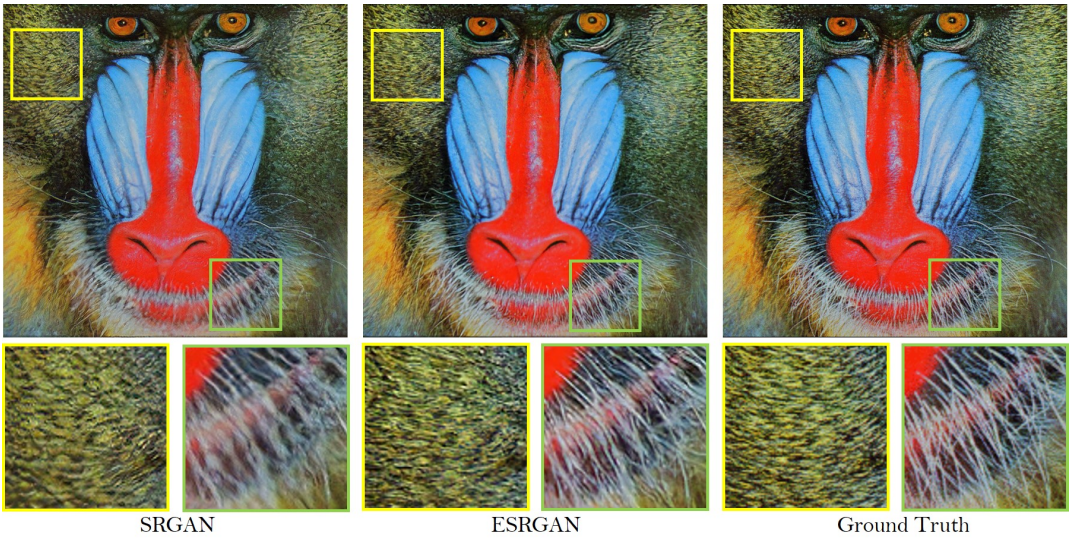


Figure 3.10: Qualitative comparison showing how ESRGAN outperforms SRGAN in sharpness and details[48].

From the name Enhanced Super-Resolution GAN, we can tell that the architecture should contain the two main modules, discriminator D and generator G networks. The G network takes a low-resolution image (LR) as input, and it passes it through a 2D convolutional layer (Conv1)with small 3×3 kernels and 64 feature maps. It is then passed through 23 Residual in Residual Dense Block (RRDB). The image is then passed through another convolutional layer (Conv2) in which its output is summed with the output of the first (Conv1). At this stage, the image gets upscaled with a factor of 4 by passing it through an upsampling block that consists of two convolutional layers for reconstruction, with LeakyReLU (LReLU) activation ($\alpha = 0.2$) on each layer. After upsampling, the image is passed through another convolutional layer (Conv3) with LReLU activation ($\alpha = 0.2$). Finally, the image is passed through the final convolutional layer (Conv4) that final super-resolved image. An overview of the G network can be seen in figure 3.11.

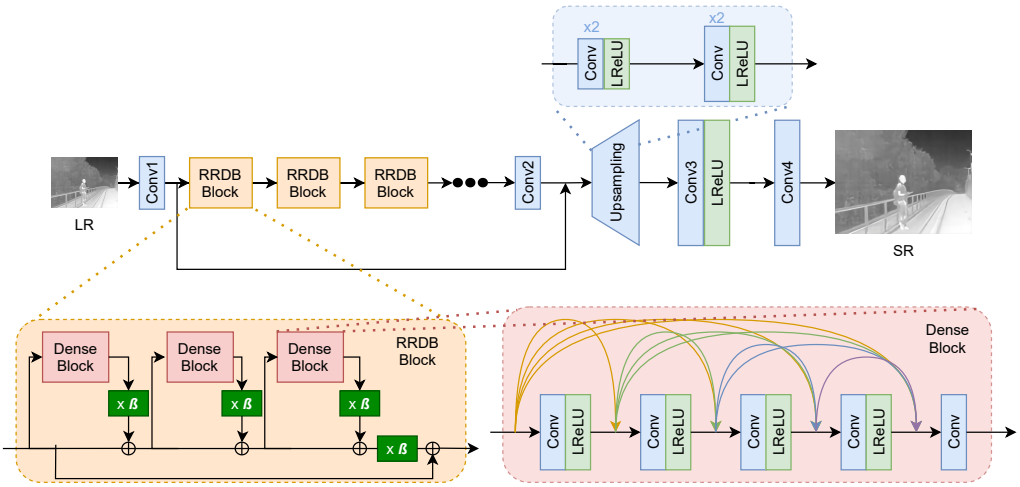


Figure 3.11: Full architecture of the ESRGAN G network, where the upper part is the network that takes a LR image as input, and generate an HR image (SR).

The other part of the network is the discriminator D, and to be more specific it is called the Relativistic Discriminator[24]. The ESRGAN authors have chosen to use this specific discriminator rather than using the standard discriminator used by the SRGAN author. This is because the relativistic discriminator estimates the probability that a real image x_r is relatively more realistic than a fake one x_f. Where a standard discriminator estimates only whether an image x is natural enough to be real. Figure 3.12 shows the difference between the relativistic and a standard discriminator.

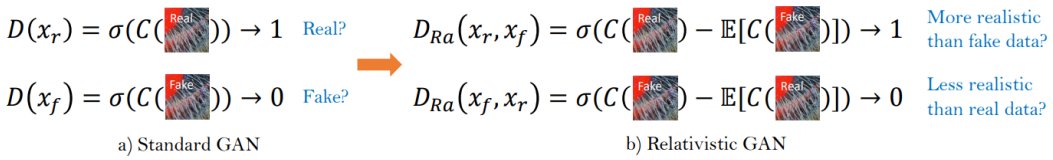


Figure 3.12: Difference between standard discriminator D and a relativistic discriminator RaD.

Where $D(x_r)$ denotes the standard discriminator and D_{Ra} denotes the relativistic discriminator shown in figure 3.13. α is the sigmoid function used to obtain a probability in the range of $[0, 1]$, and $C(x)$ is the non-transformed discriminator output

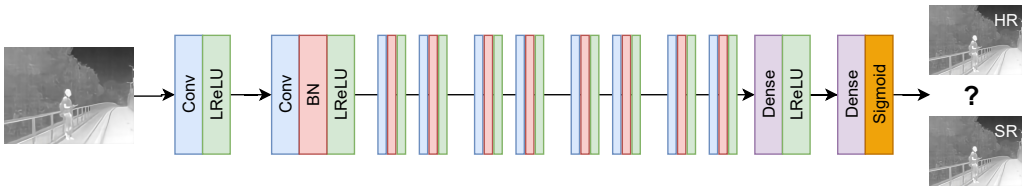


Figure 3.13: Full architecture of the ESRGAN D network, which takes an input image, and outputs a probability of whether the image is a real HR image or a SR image.

RealSR adapted the ESRGAN structure and trained it using the constructed paired data$\{I_{LR}, I_{HR}\}$. Several losses were used during the training including:

- **Pixel loss** L_1: or so called Mean Absolute Error (MAE), which measures the mean absolute pixel difference of all pixels in two given images.
- **Perceptual loss** L_{per}: proposed to enhance the visual quality by minimizing the error in feature space instead of pixel space. It uses the inactive features of VGG-19[43] and aims to enhance the visual quality of low-frequency information like edges.
- **Adversarial loss** L_{adv} This loss is used to enhance the texture details to make the image look more realistic.

The final loss function was the weighted sum of all the above losses as follows:

$$L_{total} = \lambda_1 \cdot L_1 + \lambda_{per} \cdot L_{per} + \lambda_{adv} \cdot L_{adv} \tag{3.6}$$

Where λ_1, λ_{per}, and λ_{adv} are constants used to specify the weight of each of the losses on the total loss.

PatchGAN Discriminator

The RealSR authors reported that the discriminator (VGG-128) used in the ESRGAN may introduce many artefacts, so instead, PatchGAN [22] was used instead for two reasons. The first reason is that VGG-128 used by ESRGAN limits the size of the generated image to 128, making multi-scaling training not as simple. The second reason is that the VGG-128 fixed fully connected layer makes the discriminator pays more attention to the global features and ignore the local ones. Where the PatchGAN has a fully convolutional structure that maintains a fixed receptive field that restricts the discriminator's attention to the local image patches. The structure of PatchGAN only penalizes structure at the scale of patches, meaning

that it tries to classify if each $N \times N$ patch in an image is real or fake. The responses of all patches get averaged afterward forming the final D output to guarantee global consistency, then gets fed back to the generator. PatchGAN has a very simple structure (figure 3.14), as it consists of three hidden convolutional layers, each of them followed by a batch normalization layer and LeakyReLU is used for activation.

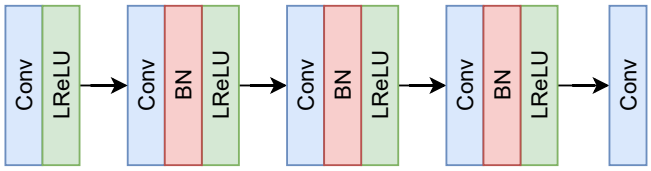


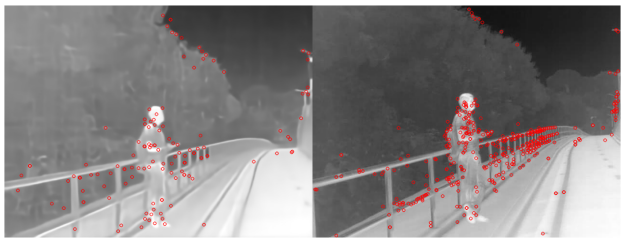
Figure 3.14: The architecture of the PatchGAN used by RealSR as a substitute for the ESRGAN' discriminator.

3.3 Image Registration

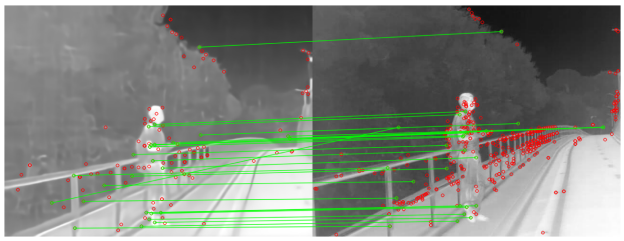
Image Registration is the process of overlapping images that could be taken at different times and from different viewpoints of a scene. The differences between such images are the result of different imaging conditions[55]. The topic of image registration is not within the scope of this work, therefore, a brief explanation of how it was done to serve this work will be explained. Given the two images I_i and I_ref the process of registering the two images can be divided into 4 main steps that can be seen in figure 3.15 and are as follows:

- **Feature Descriptors:** First, distinctive objects, edges, lines intersections, contours, or any key features that can be shared by the two given images (I_i, I_ref)are extracted. Then descriptors are used to find the similar features between the two images.
- **Feature Matching:** This step is about finding the correspondences between the features found in I_i, and those found in the reference image (I_i) using the features descriptors.
- **Mapping Function:** The parameters needed to build a function that can be used to align the established feature correspondence are computed at this step.
- **Image Transformation:** The mapping functions are then used to transform I_i.

There exist a wide variety of feature detection algorithms, one of which is Oriented FAST and Rotated BRIEF (ORB). ORB was chosen as the feature descriptor



(a) Features extraction.



(b) Features matching

(c) Features mapping and transformation

Figure 3.15: An illustration showing the steps followed to register two given images.

to be used for the purpose of this work, as it is one of the most standard used methods nowadays. Based on multiple experiments that were done during this work, ORB was also proven to perform best in comparison to other popular methods like SIFT[25], SURF[4], BRISK[28]m and BRIEF[6]. ORB uses the combination of FAST[47] key point detector and BRIEF descriptor. It uses FAST to extract key points, then it uses harris corner measure[45] to find top N points among them.

Chapter 4

Design and Implementation

This chapter describes both the design of the system and its implementation process. It also goes through the considerations made to achieve the proposed system.

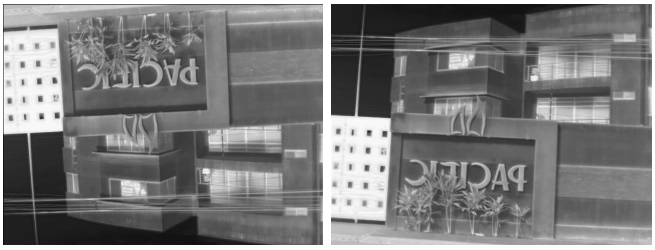
4.0.1 Data Preprocessing

In section 2.7, we chose to use the PBVS dataset during this work. The dataset has three subsets modules called *Domo, Axis* and *GT* with different resolutions that were reported in table 2.1. For this work, the Axis subset is discarded, and the Domo & GT subsets are used as the source and target domains respectively. Each of these subsets includes a total of 951 training images and 50 images for validation. The reason behind discarding the Axis subset, was because the goal of this work is to super-resolve a given resolution with an upscaling factor of $s = 4$ and later evaluate the performance by comparing it to the ground-truth, which has a native resolution that matches the SR output images. So the plan was to super-resolve the input images (Domo validation subset) and compare the output with the ground truth (GT validation subset). However, one of the problems with the PBVS dataset is the limited number of images in each subset, which is considered too little to be used for training a neural network. Therefore, it was decided to use the augmented version of the PBVS dataset, which was provided by the authors of the TherISuRNet[10]. The augmentation operations they apply on the original dataset are horizontal flipping, 180°rotation, and two affine operations. An example of these augmented images can be seen in figure 4.1.

4.1 Training Details

This section explains the different modules that RealSR consists of, as well as the adjustments that were done to achieve the final results. Figure 4.2 shows the system pipeline, and how the individual modules are connecting.

(a) Original **(b)** Affine1 **(c)** Affine2



(d) 180°Rotation **(e)** Horizontal flip

Figure 4.1: Examples showing the augmentation operations that were applied on both the Domo and GT subsets.

4.1.1 Kernel Estimation and Noise Injection

The used methods have been previously explored in section 3.2. However, an explanation of the implementation will be covered in this section.
The KernelGAN network was trained on a total of 740 images out of the original Domo subset using the settings shown in table 4.1 resulting in a pool of kernels that contains 740 kernels. Some of these kernels are shown in figure 4.3.

Parameter	Value
Iterations	3000
Learning rate	$2e^{-4}$
Learning rate decay	0.1 every 750 iterations
Optimizer	ADAM($\beta_1 = 0.5, \beta_2 = 0.999$)

Table 4.1: The parameters' values used during the KernelGAN network training for each image.

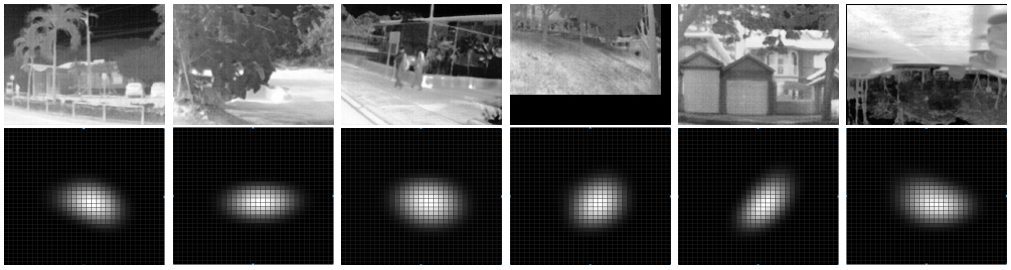


Figure 4.3: Few examples showing the different estimated kernels using KernelGAN associated with their corresponding images from the LR Domo subset.

Noise Collection

The next step was to create a pool of noise patches from the Domo subset (source domain) and use it during the noise injection step as shown in step 2 in figure 4.2. This was done using a simple rule that was proposed by the RealSR authors as explained in section 3.2.1 ($\sigma(n_i) < v$). The main strategy behind choosing the right noise patch was to extract patches of smooth areas such as walls, sky, or any area that does not have any patterns that would be confused with noise. After experimenting with different variance ranges and different patch sizes, the following values were chosen:

- Patch size = 70×70
- Max variance value = 70
- Min variance value = 0

It is worth mentioning that those values can be different based on the given images. Looking at figure 4.4 we can see some noise patches that were extracted using the above variance range. Another thing to keep in mind is that the noise injection step was not done simultaneously with the downsampling operation. The noise injection step was done during the training phase of the SR model, as this makes the noise more diverse and it regularises the SR model to distinguish between the noise and the content.

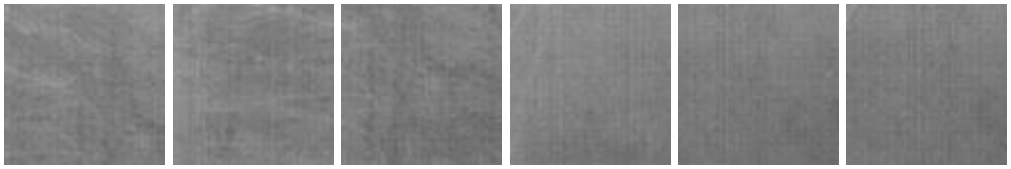


Figure 4.4: Few examples showing the different noise patches that were extracted from the Domo subset.

4.1.2 Super-Resolution Model

Step 3 in figure 4.2 is the step where the constructed image pairs are used to train a SR model. As discussed before in section 3.2.2, ESRGAN is used as the backbone for training the SR model while replacing its discriminator with the PatchGAN discriminator. All experiments were done with a scaling factor of $\times 4$ between the constructed LR and HR images. The mini-batch size was set to 16, where the size of the cropped HR patches used during the training was set to 128×128. The ESRGAN authors suggested in their paper that the network can benefit from using a larger patch size, as a larger patch size can help the network capturing more semantic information. However, using higher mini-batch and patch sizes was not an option due to the limited resources utilized during this work[1]. All the trainings done during this work was split into two stages. For the first training, the models' weights were initialized using the PSNR-oriented pretrained RRDB, and the objective of the network was set to use the pixel loss (L1) only. This was done to fine-tune the RRDB model to the thermal images, as the RRDB model was trained on RGB images only as reported in the [48]. The learning rate was initialized as 1×10^{-4} and decayed by a factor of 2 every 5×10^{3} iteration. This model was trained for 250 epochs[2] and then used as initialization for the next training. The second training's objective was changed to be focused towards better perceptual quality by change the training objective to the loss function in Eq. 3.6 with $\lambda_1 = 1 \times 10^{-2}$, $\lambda_{per} = 1$ and $\lambda_{adv} = 5 \times 10^{-3}$. For optimisation, ADAM with $\beta_1 = 0.9$, $\beta_2 = 0.999$.

[1] All the experiments were performed on a computer with a Ryzen 7 5800X processor @3.8GHz x16 running on 32GB RAM and an NVIDIA GeForce 3070 with 8GB of memory.

[2] 1 epoch = $\approx$ 300 iterations). The number of iterations is calculated by dividing the total number of image pairs 4755 on the mini-batch size.

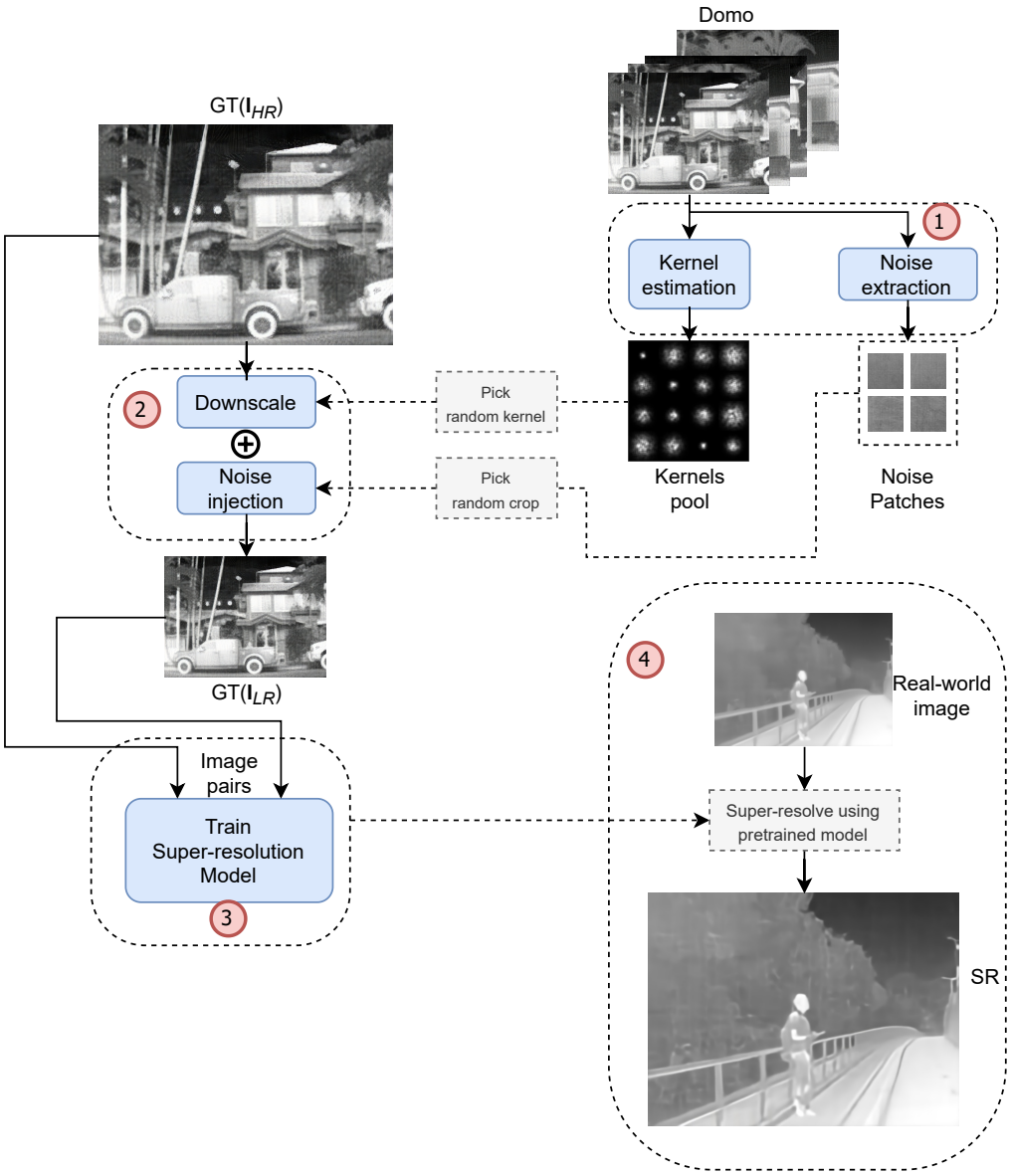


Figure 4.2: The figure presents an overall overview of the different modules that RealSR consists of, and how the final super-resolved images are obtained. 1. The Domo subset is used to extract the blurring kernels and the noise patches from the source domain. 2. The extracted kernels and noise patches are used to generate realistic low-resolution images (GT_{LR}) that are used to create image pairs (LR-HR). 3. The image pairs are then used to train an SR model (ESRGAN). 4. The pretrained SR model is used to super-resolve LR images.

Chapter 5

Evaluation and Results

To evaluate the performance of the adapted RealSR in comparison to the other methods, different experiments have been carried out. Detailed analyses of the experiments are presented in this section. The overall evaluation process, which was adapted from the PBVS challenge[1] was done following the pipeline shown in figure 5.3, which will be explained in details in section 5.1.1.

5.1 Testing

In section 2.6, the challenges that are present with the utilized dataset were explained. These challenges varied from different camera settings, light conditions, different sensor noise, resulting in differences in brightness and contrast, and most importantly the misalignment. When addressing these issues, it was important to be careful not to manipulate the images in a way that would change the content of these images. Especially since the evaluation methods used to evaluate the performance of the SR models, and modifying the content of the images will result in wrong results. However, the only problem to be addressed without modifying the images was the misalignment and addressing it was considered to be the most important issue to be fixed before evaluating the different methods. Hence, the first step before being able to acquire the quantitative results of the trained SR models was to apply the image registration to align SR images with the GT images. The ORB detector (explained in section 3.3) with a target number of features N=5000 was used to align the super-resolved images, using the GT validation dataset for reference. Despite the efforts to achieve the best image alignment possible, it was observed that the chosen image registration method did not always give the most satisfactory results, and in some cases, the registration failed. If we look at figure

[1]The evaluation process according to the PBVS challenge paper is to register the GT image and use the SR image for reference, but the opposite is done in this work, where the GT image is used as a reference instead.

5.1 we can see few examples where the registration was successful. However, other SR images like those shown in figure 5.2 the registration operation completely failed to align the two images due to the lack of key points in the input image.

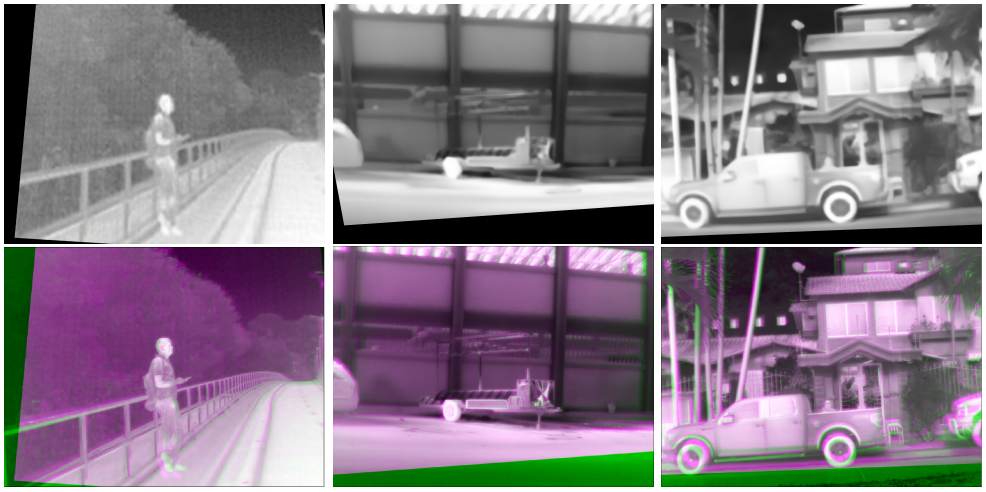


Figure 5.1: Examples of successfully registered SR images, where the magenta and blue colours indicate the good and bad image matching respectively.

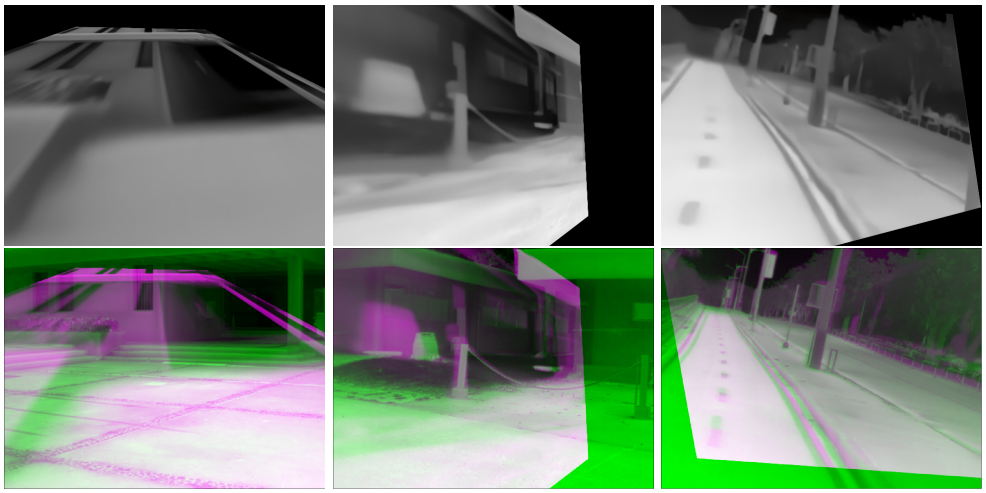


Figure 5.2: Examples showing the failed registration process, where the magenta and blue colours indicate the good and bad image matching respectively.

To keep the comparison fair and to avoid bias to any of the SR methods, the failed registration cases were accepted, as the poor registration was considered to be due to the poor performance of the specific algorithms. Especially that the same images that were super-resolved using another algorithm had no registration problems.

5.1.1 Results

After having the registered sets of images, it was decided to use a combination of both the reference-based and non-reference-based IQA methods. The reason for this decision was the imperfect alignment of the super-resolved and ground-truth images. A method such as PSNR, will penalize the performance in case the registered image is shifted one pixel in any direction, and we know for sure that this is most likely the case with our data. Therefore, utilizing non-reference-based IQA methods would give a better idea about the performance in this context in terms of image quality. The image quality was evaluated on the central crop (50%) of both the SR and GT images. This was done to discard the empty areas (black background) in the registered images as illustrated in figure 5.3.

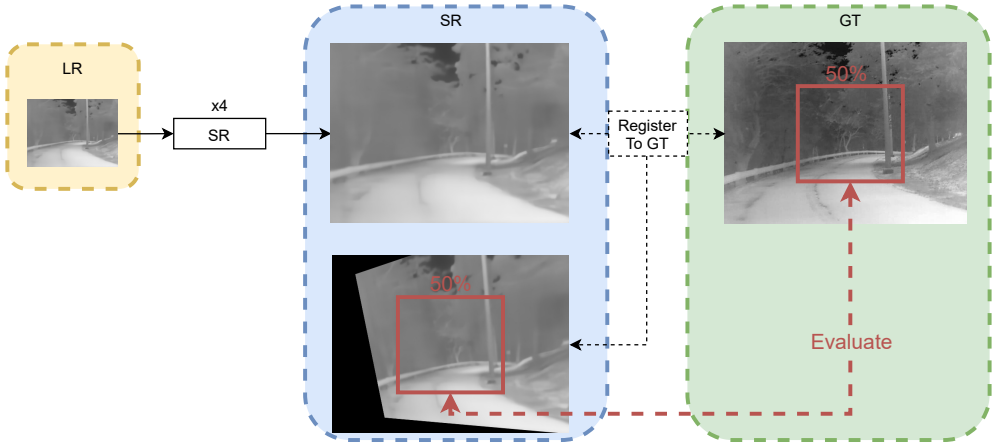


Figure 5.3: The evaluation pipeline used to evaluate the super-resolved LR image in comparison to the GT.

The quantitative results for the best model we trained, as well as the SotA methods in terms of the reference and no-reference-based metrics, is shown in table 5.1. The results for the competing methods were obtained by either retraining the models as the authors of each method suggested, or in case of the methods like DualSR and ZSSR-KernelGAN training were not required. As DualSR and ZSSR-KernelGAN are trained during the inference phase. For ESRGAN, we use the pre-trained weights provided by the authors. Figure 5.4 shows random crops that were extracted from some of the super-resolved images to get a closer look at the differences in the visual quality reconstructed by the different SotA as well as the adapted method. The crops from our model show significant improvement in the perceptual quality in terms of sharpness and image clarity. It was explained in section 2.5 that the no-reference-based metrics as well as the LPIPS measure the perceived image quality unlike the PSNR and SSIM that measure how similar the

two given images to each other (the SR and GT in our case), which explains the contrast in the quantitative results.

Backbone	PSNR↑	SSIM↑	LPIPS↓	PIQE↓	NIQE↓	BRISQUE↓
Bicubic	20.11	0.700	0.46	67.39	5.55	57.20
DualSR[14]	18.77	0.591	0.43	56.48	4.18	43.03
ZSSR-KernelGAN[5]	19.01	0.572	0.44	60.79	5.71	46.14
ESRGAN[48]	**20.18**	0.664	0.42	64.34	6.44	55.52
TherISuRNet[10]	20.10	**0.719**	0.42	88.69	5.20	55.34
RealTISR#4	18.78	0.520	**0.37**	**36.33**	**3.31**	**34.31**

Table 5.1: Comparison between the best performing model we trained and the SotA methods that have been tested. The best values are in bold text.

5.2 Ablation Study

In this work, the goal was to surpass the performance of TherISuRNet by adapting the RealSR pipeline to fit the thermal domain. As we have seen in table 5.1, RealTISR surpassed the performance of TherISuRNet in terms of perceptual quality metrics. To understand the reason behind the superiority of RealTISR, extra experiments were needed.

- If we revise the differences between the two methodologies, we can see that the two methods use different degradation techniques. Where RealTISR uses realistic noise injection and kernel estimation, and TherISuRNet uses Gaussian noise and bicubic downsampling, which are factors that can affect the performance. To get an understanding of the influence of each of these factors on the final performance, several scenarios were considered. We alternate between the two degradation techniques, as well as trying to enable and disable the noise injection and the perceptual loss. Regarding the perceptual loss, it was explained in section 4.1.2 how the models were trained with two stages by first setting the objective loss of the network to be PSNR oriented to tune the pretrained RRDB model provided by the ESRGAN authors. Then the network's loss function was set to Eq.3.6 to let the SR model focus on generating images with better perceptual quality.

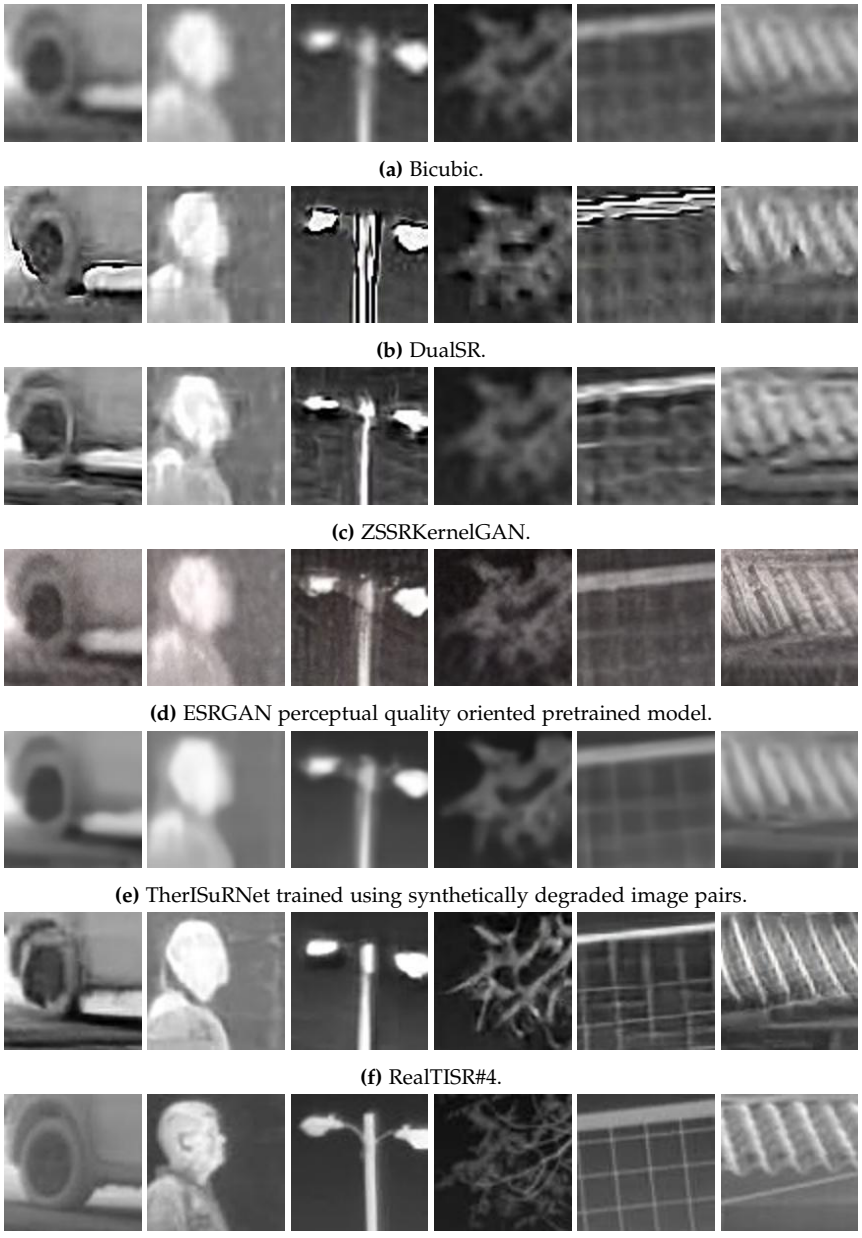
- The first experiment was about checking the influence of the degradation technique on the models that we trained. However, this experiment is about testing the effect of the degradation technique on the TherISuRNet. This was done to determine whether the difference in performance was a result of the different degradation techniques or due to the different SR backbones. The experiment was about training the TherISuRNet using the image pairs that

were constructed using the realistic degradation pipeline. The results of this experiment are visible in figure 5.4 where we can see how the sharpness of the images generated by TherISuRNet increased significantly when we trained using the realistically degraded image pairs, which shows the downside of training a model using ideal downsampling operations like bicubic interpolation. The models trained using the realistically degraded image pairs became more robust to new unseen images.

Figure 5.5 contains image crops that were obtained from the different models that we trained while conducting the ablation Study, and the quantitative results for those models are reported in table 5.2.

Backbone	D	Initialisation Model	Noise Injection	PSNR↑	SSIM↑	LPIPS↓	PIQE↓	NIQE↓	BRISQUE↓
Bicubic	-	-	No	**20.11**	0.700	0.46	67.39	5.55	57.20
TherISuRNet	SD	-	No	20.10	**0.719**	0.42	88.69	5.20	55.34
TherISuRNet	RD	-	No	19.06	0.640	0.44	85.82	5.72	53.88
RealTISR#1	SD	RRDB	Yes	19.71	0.718	0.46	89.81	6.41	55.24
RealTISR#2	SD	RealTISR#1	Yes	19.63	0.714	0.40	51.31	3.86	51.17
RealTISR#3	RD	RRDB	Yes	19.17	0.670	0.42	85.89	5.35	53.14
RealTISR#4	RD	RealTISR#3	Yes	18.78	0.520	**0.37**	36.33	**3.31**	**34.31**
RealTISR#5	RD	RRDB	No	19.24	0.636	0.58	84.59	5.85	51.21
RealTISR#6	RD	RealTISR#5	No	17.65	0.397	0.58	**26.30**	3.90	35.34

Table 5.2: Comparison between the different training scenarios that have been tested. The best values are in bold text. (D) stands for degradation, where (SD) and (RD) stand for synthetic degradation and realistic degradation respectively.



(a) Bicubic.

(b) DualSR.

(c) ZSSRKernelGAN.

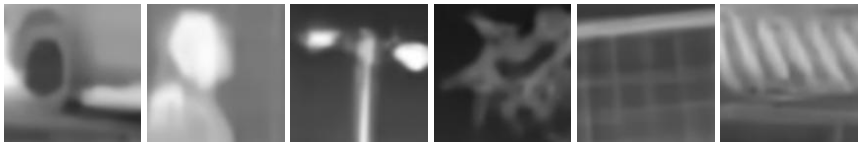
(d) ESRGAN perceptual quality oriented pretrained model.

(e) TherISuRNet trained using synthetically degraded image pairs.

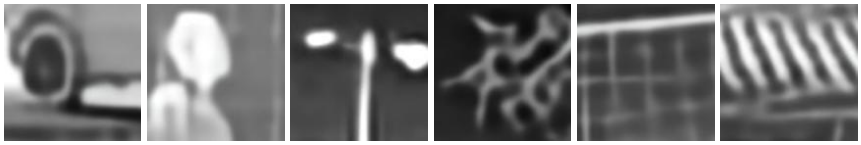
(f) RealTISR#4.

(g) GT.

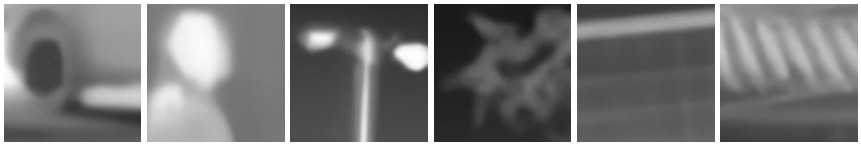
Figure 5.4: Qualitative comparison of SotA methods for x4 SR of LR images from the Domo validation subset.



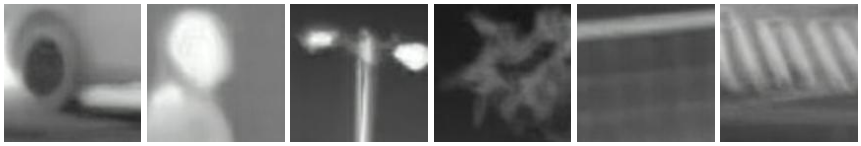
(a) TherISuRNet trained using synthetically degraded image pairs.



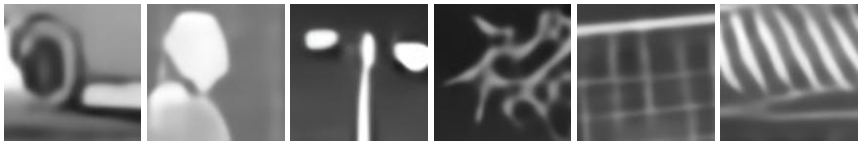
(b) TherISuRNet trained using realistically degraded image pairs.



(c) RealTISR#1.

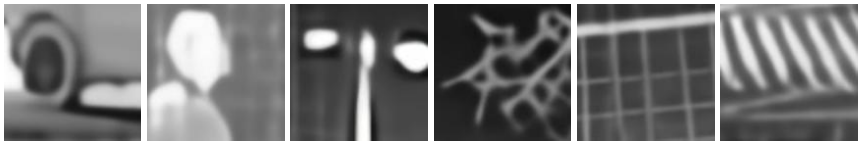


(d) RealTISR#2.



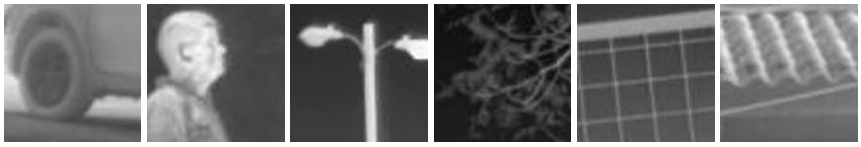
(e) RealTISR#3.

(f) RealTISR#4.



(g) RealTISR#5.

(h) RealTISR#6.



(i) GT.

Figure 5.5: An overview of all the trained models that were shown in table 5.2.

Chapter 6

Discussion

The following sections discuss the results obtained during the evaluation chapter 5, and some optimizations that could be added to the adapted method to achieve better performance.

Based on the results reported in section 5.1.1, it was uncovered how the proposed RealTISR was able to achieve SotA results in terms of perceptual quality. The proposed method delivers images with sharper edges,a more defined image structure, a balanced level of noise, and not over-smoothed images. Since images that are super-resolved using some competing methods like ZSSR+KernelGAN and DualSR suffered from high amount of noise that had a very negative impact on the perceived image quality.

After conducting the experiments, it became clear how PSNR and SSIM metrics do not correlate with the perceptual quality of an image. Hence, choosing the best-performing model based on these two metrics, as done at the beginning of this work, should be reconsidered. Another observation was that high PSNR and SSIM values correlate with blurry images that lack high-frequency details. This observation is illustrated in figure 5.5 where it is possible to see that the most blurry reconstructed images were generated with the models that achieved the highest PSNR and SSIM values (e.g. TherISuRNet(SD), RealTISR#1, RealTISR#2, and most importantly the bicubically interpolated images). All the mentioned models were trained using the synthetically degraded image pairs, which explains one of the reasons achieved high PSNR and SSIM values in the first place. A challenging problem that was faced during this work was related to the utilized dataset. The Domo and GT subsets were captured using different cameras, which introduced a couple of problems, the first being the misalignment issue. However, another issue that adds to the overall low PSNR and SSIM values obtained during this work is the brightness and contrast gap between the two subsets. Since these

two metrics measure the absolute difference between images [39].

6.1 Improvements

We explained in section 2.5 the reason behind choosing each of the IQA metrics to be utilized in this work when evaluating the performance of each model that was tested during this work. We have mentioned as well, how no-reference-based methods as well as the LPIPS metrics, were designed and trained on RGB images. To the best of our knowledge, there are no IQA metrics that were designed specifically for evaluating thermal images, which made it a tough decision to select the best overall performing model out of both the SotA and the proposed RealTISR. Instead, we chose to evaluate the models based on the qualitative results, where the image quality was judged by how close the reconstructed images were to the ground-truth images. The downside of this choice is the subjectivity of our evaluation, and to account for that, it would have been more valid to utilize the MOS and MOR metrics. This generalizes the amount of subjectivity on how super-resolved images are perceived.

6.2 Future Work

This section discusses some ideas and additions that can be added to the overall solution, which might improve the quality of the reconstructed images.

In the last chapter, we mentioned the challenges that were faced working with the PBVS dataset, which was mainly the misalignment, brightness, and contrast between the Domo and GT subsets. One way to account for these challenges is to pre-register the two subsets and use them directly as image pairs for training the RealSR method. This would introduce the model to both domains, encouraging the model to become better at mapping images from the Domo domain to the GT. In case of succeeding with integrating this into the RealSR pipeline, it would replace the need of using the realistic degradation pipeline.

Another idea that is still related to the alignment issue is to investigate the possibility of integrating an alignment module in the RealSR SR model, which would make the method more flexible and robust to new data that is not perfectly aligned.

The current IQA metrics utilized during this work are not designed to work specifically with thermal images. So, finding a method that would measure thermal

images more accurately can probably give a better understanding of how to evaluate the thermal image SR methods.

Chapter 7

Conclusion

In this work, we investigated the possibility of adapting RGB based SotA RWSR methods to fit the thermal imaging domain. The goal was to achieve a higher perceptual quality that would surpass the performance of TherISuRNet, which is the SotA thermal super-resolution method. The final problem statement was:

Is it possible to surpass the quantitative and qualitative performance of the SotA thermal RWSR method by adapting the RealSR method and tuning it to fit the thermal domain?

Based on that, we utilized the RealSR pipeline for this matter. We trained the mentioned method using a dataset that consists of multiple subsets of thermal images taken of the same scene, but with different image resolutions. and we were able to partially reconstruct images that have a better perceptual quality than those reconstructed using TherISuRNet.

In the evaluation phase, we were able to achieve quantitative results that did not fully surpass the thermal SotA results. However, the reconstructed images were sharper and more enhanced, which was considered satisfactory. Especially that it was found during the work that not all the utilized IQA matrices were suitable, at least not for this type of images.

With that said, we find our results promising, as it is an initial step towards building a solution that would surpass the SotA thermal super-resolution methods both quantitatively and qualitatively. Section 6.1 introduces few ideas that can be taken into consideration to improve the quality of this work. The next chapter will explain few new ideas that can be a potential step towards a better thermal super-resolution method.

Bibliography

[1] Eirikur Agustsson and Radu Timofte. "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2017.

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. *Fast, Accurate, and Lightweight Super-Resolution with Cascading Residual Network*. 2018. arXiv: 1803.08664 [cs.CV].

[3] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network". In: *2017 International Conference on Engineering and Technology (ICET)*. IEEE. 2017, pp. 1–6.

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded Up Robust Features". In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. ISBN: 978-3-540-33833-8.

[5] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. *Blind Super-Resolution Kernel Estimation using an Internal-GAN*. 2020. arXiv: 1909.06581 [cs.CV].

[6] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. "BRIEF: Binary Robust Independent Elementary Features". In: vol. 6314. Sept. 2010, pp. 778–792. ISBN: 978-3-642-15560-4. DOI: 10.1007/978-3-642-15561-1_56.

[7] Honggang Chen, Xiaohai He, Linbo Qing, Yuanyuan Wu, Chao Ren, and Ce Zhu. *Real-World Single Image Super-Resolution: A Brief Review*. 2021. arXiv: 2103.02368 [eess.IV].

[8] Youngjun Cho, Nadia Bianchi-Berthouze, Nicolai Marquardt, and Simon J. Julier. "Deep Thermal Imaging". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018). DOI: 10.1145/3173574.3173576. URL: http://dx.doi.org/10.1145/3173574.3173576.

[9] Yukyung Choi, Namil Kim, Soonmin Hwang, and In So Kweon. "Thermal image enhancement using convolutional neural network". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 223–230.

[10] Vishal Chudasama, Heena Patel, Kalpesh Prajapati, Kishor P. Upla, Raghavendra Ramachandra, Kiran Raja, and Christoph Busch. "TherISuRNet - A Computationally Efficient Thermal Image Super-Resolution Network". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2020.

[11] Zhen Cui, Hong Chang, Shiguang Shan, Bineng Zhong, and Xilin Chen. "Deep Network Cascade for Image Super-resolution". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, 2014, pp. 49–64. ISBN: 978-3-319-10602-1.

[12] Josh Cutler and Matt Dickenson. "Introduction to Machine Learning with Python". In: *Computational Frameworks for Political and Social Research with Python*. Cham: Springer International Publishing, 2020, pp. 129–142. ISBN: 978-3-030-36826-5. DOI: 10.1007/978-3-030-36826-5_10.

[13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. *Image Super-Resolution Using Deep Convolutional Networks*. 2015. arXiv: 1501.00092 [cs.CV].

[14] Mohammad Emad, Maurice Peemen, and Henk Corporaal. "DualSR: Zero-Shot Dual Learning for Real-World Super-Resolution". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2021, pp. 1630–1639.

[15] *FLIR Thermal Dataset for Algorithm Training*. FLIR ADK DATASHEET. FLIR Systems, Inc. Aug. 2019. URL: https://www.flir.com/oem/adas/adas-dataset-form/.

[16] Manuel Fritsche, Shuhang Gu, and Radu Timofte. *Frequency Separation for Real-World Super-Resolution*. 2019. arXiv: 1911.07850 [eess.IV].

[17] Rikke Gade and Thomas B. Moeslund. "Thermal Cameras and Applications: A Survey". English. In: *Machine Vision Applications* 25.1 (Jan. 2014), pp. 245–262. ISSN: 0932-8092. DOI: 10.1007/s00138-013-0570-5.

[18] Dang Ha The Hien. *A guide to receptive field arithmetic for Convolutional Neural Networks*. 2018. URL: https://blog.mlreview.com/a-guide-to-receptive-field-arithmetic-for-convolutional-neural-networks-e0f514068807.

[19] Alain Hore and Djemel Ziou. "Image quality metrics: PSNR vs. SSIM". In: *2010 20th international conference on pattern recognition*. IEEE. 2010, pp. 2366–2369.

[20] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. "Multispectral Pedestrian Detection: Benchmark Dataset and Baselines". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[21] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*. 2016. arXiv: 1602.07360 [cs.CV].

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. *Image-to-Image Translation with Conditional Adversarial Networks*. 2018. arXiv: 1611.07004 [cs.CV].

[23] Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. "Real-World Super-Resolution via Kernel Estimation and Noise Injection". In: *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2020.

[24] Alexia Jolicoeur-Martineau. *The relativistic discriminator: a key element missing from standard GAN*. 2018. arXiv: 1807.00734 [cs.LG].

[25] Yan Ke and R. Sukthankar. "PCA-SIFT: a more distinctive representation for local image descriptors". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 2. 2004, pp. II–II. DOI: 10.1109/CVPR.2004.1315206.

[26] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE*. 1998, pp. 2278–2324.

[27] Kyungjae Lee, Junhyeop Lee, Joosung Lee, Sangwon Hwang, and Sangyoun Lee. "Brightness-based convolutional neural network for thermal image enhancement". English. In: *IEEE Access* 5 (Nov. 2017). Funding Information: This work was supported by the Institute for Information and Communications Technology Promotion through the Korea government (MSIP) under Grant 2016-0-00197. Publisher Copyright: © 2013 IEEE., pp. 26867–26879. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2769687.

[28] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. "BRISK: Binary Robust invariant scalable keypoints". In: Nov. 2011, pp. 2548–2555. DOI: 10.1109/ICCV.2011.6126542.

[29] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. "Multi-scale Residual Network for Image Super-Resolution". In: *The European Conference on Computer Vision (ECCV)*. 2018.

[30] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. *Enhanced Deep Residual Networks for Single Image Super-Resolution*. 2017. arXiv: 1707.02921 [cs.CV].

[31] Andreas Lugmayr, Martin Danelljan, Radu Timofte, Namhyuk Ahn, Dongwoon Bai, Jie Cai, Yun Cao, Junyang Chen, Kaihua Cheng, SeYoung Chun, Wei Deng, Mostafa El-Khamy, Chiu Man Ho, Xiaozhong Ji, Amin Kheradmand, Gwantae Kim, Hanseok Ko, Kanghyu Lee, Jungwon Lee, Hao Li, Ziluan Liu, Zhi-Song Liu, Shuai Liu, Yunhua Lu, Zibo Meng, Pablo Navarrete Michelini, Christian Micheloni, Kalpesh Prajapati, Haoyu Ren, Yong Hyeok Seo, Wan-Chi Siu, Kyung-Ah Sohn, Ying Tai, Rao Muhammad Umer, Shuangquan Wang, Huibing Wang, Timothy Haoning Wu, Haoning Wu, Biao Yang, Fuzhi Yang, Jaejun Yoo, Tongtong Zhao, Yuanbo Zhou, Haijie Zhuo, Ziyao Zong, and Xueyi Zou. *NTIRE 2020 Challenge on Real-World Image Super-Resolution: Methods and Results*. 2020. arXiv: 2005.01996 [eess.IV].

[32] Kamal Nasrollahi and Thomas B Moeslund. "Super-resolution: a comprehensive survey". In: *Machine vision and applications* 25.6 (2014), pp. 1423–1468.

[33] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 2018. arXiv: 1811.03378 [cs.LG].

[34] Seong-Jin Park, Hyeongseok Son, Sunghyun Cho, Ki-Sang Hong, and Seungyong Lee. "SRFeat: Single Image Super-Resolution with Feature Discrimination". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[35] R. E. Rivadeneira, A. D. Sappa, B. X. Vintimilla, L. Guo, J. Hou, A. Mehri, P. B. Ardakani, H. Patel, V. Chudasama, K. Prajapati, K. P. Upla, R. Ramachandra, K. Raja, C. Busch, F. Almasri, O. Debeir, S. Nathan, P. Kansal, N. Gutierrez, B. Mojra, and W. J. Beksi. "Thermal Image Super-Resolution Challenge - PBVS 2020". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 432–439. DOI: 10.1109/CVPRW50498.2020.00056.

[36] Rafael E Rivadeneira, Angel D Sappa, and Boris X Vintimilla. "Thermal Image Super-Resolution: a Novel Architecture and Dataset". In: *International Conference on Computer Vision Theory and Applications*. nn. 2020, pp. 1–2.

[37] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv e-prints* (Sept. 2016). arXiv: 1609.04747 [cs.LG].

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[39] Ya-Hui Shiao, Tzong-Jer Chen, Keh-Shih Chuang, Cheng-Hsun Lin, and Chun-Chao Chuang. "Quality of Compressed Medical Images". In: *Journal of digital imaging : the official journal of the Society for Computer Applications in Radiology* 20 (July 2007), pp. 149–59. DOI: 10.1007/s10278-007-9013-z.

[40] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. *InGAN: Capturing and Remapping the "DNA" of a Natural Image*. 2019. arXiv: 1812.00231 [cs.CV].

[41] Assaf Shocher, Nadav Cohen, and Michal Irani. *"Zero-Shot" Super-Resolution using Deep Internal Learning*. 2017. arXiv: 1712.06087 [cs.CV].

[42] SIERRA-OLYMPIC. *VAYU HD Feature Specefication*. accessed: 02/06-2021. URL: https://sierraolympic.com/wp-content/uploads/2020/06/2020_VayuHD_Sell-Sheet_FINAL.pdf.

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

[44] Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. *Meta-Transfer Learning for Zero-Shot Super-Resolution*. 2020. arXiv: 2002.12213 [cs.CV].

[45] Javier Sánchez, Nelson Monzón, and Agustín Salgado. "An Analysis and Implementation of the Harris Corner Detector". In: *Image Processing On Line* 8 (Oct. 2018), pp. 305–328. DOI: 10.5201/ipol.2018.229.

[46] Rao Muhammad Umer, Gian Luca Foresti, and Christian Micheloni. *Deep Generative Adversarial Residual Convolutional Networks for Real-World Super-Resolution*. 2020. arXiv: 2005.00953 [eess.IV].

[47] Deepak Geetha Viswanathan. "Features from accelerated segment test (fast)". In: *Proceedings of the 10th workshop on Image Analysis for Multimedia Interactive Services, London, UK*. 2009, pp. 6–8.

[48] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. *ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks*. 2018. arXiv: 1809.00219 [cs.CV].

[49] Zhihao Wang, Jian Chen, and Steven CH Hoi. "Deep learning for image super-resolution: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* (2020).

[50] Zhou Wang and Alan C Bovik. "Modern image quality assessment". In: *Synthesis Lectures on Image, Video, and Multimedia Processing* 2.1 (2006), pp. 1–156.

[51] Linwei Yue, Huanfeng Shen, Jie Li, Qiangqiang Yuan, Hongyan Zhang, and Liangpei Zhang. "Image super-resolution: The techniques, applications, and future". In: *Signal Processing* 128 (2016), pp. 389–408. ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2016.05.002.

[52] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 2018. arXiv: 1801.03924 [cs.CV].

[53] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. *Image Super-Resolution Using Very Deep Residual Channel Attention Networks*. 2018. arXiv: 1807.02758 [cs.CV].

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

[55] Barbara Zitová and Jan Flusser. "Image registration methods: a survey". In: *Image and Vision Computing* 21.11 (2003), pp. 977–1000. ISSN: 0262-8856. DOI: https://doi.org/10.1016/S0262-8856(03)00137-9.