# Exploring super resolution combined with an object detector on optimizing building detection in Greenland

VGIS

Group 1046

Vision, Graphics and Interactive Systems
Aalborg University

**Electronics and IT**
Aalborg University
http://www.aau.dk

**AALBORG UNIVERSITY**
STUDENT REPORT

**Title:**
Exploring super resolution combined with an object detector on optimizing building detection in Greenland

**Theme:**
Computer Vision

**Project Period:**
Spring Semester 2022

**Project Group:**
VGIS 1046

**Participant(s):**
Oscar Mäkinen

**Supervisor(s):**
Mark Philip Philipsen
Anders Skaarup Johansen
John Kamper

**Copies:** 1

**Page Numbers:** 45

**Date of Completion:**
June 1, 2022

**Abstract:**

As high resolution imagery can be very expensive, super resolution is becoming more and more researched. Remote sensing is one of the fields using super resolution to up sample freely available satellite data. Generative adversarial networks or GANs are usually used for super resolution and has proven to yield good results. This project looks into GANs, more specifically Real-ESRGAN with a dataset based on Greenlandic aerial imagery, to best optimize the algorithm for Greenlandic environment. The results are run through an object detector, a Mask-RCNN, to detect buildings. Although the super resolution images produced by the GAN network aren't visually appealing, an improvement in building detection has emerged. The new up sampled images find buildings better than a low resolution and bicubic outputs. From previous images the detector had trouble finding small buildings next to big buildings, this problem was solved by introducing super resolution outputs.

# Contents

# Preface

This project was written during the forth semester of the masters program of Vision, Graphics and Interactive Systems or VGIS. The idea behind the project is to learn about super resolution which is a hot topic in today's computer vision world and how it could upgrade the performance of a building detector. Python, PyTorch and ArcGIS was used in this project.

The author would like to thank Mark Philip Philipsen, Anders Skaarup Johansen and Frederik Hass for their support and feedback during this project.

Aalborg University, June 1, 2022

Oscar Mäkinen
<omakin20@student.aau.dk>

# 1 - Introduction

The Government of Greenland is obligated to keep an up to date technical map of it cities and settlements, the challenge becomes massive. As Greenland is the largest island in the world the distance between cities and settlements are often hours away or days even by airplane and reaching these remote places is time consuming and costly As close to real time satellite becomes more and more freely available a super resolution solution combined with a object detector could reduce cost for keeping such a map up to date. Super resolution is used in many fields, eg for medicine use, self-driving vehicles or remote sensing, what this project will focus on. Generative adversarial networks are often used in super resolution problems [16]. The project will go through some of them like SRGAN[29], ESRGAN[53] and Real-ESRGAN[54]. The project will then focus on Real-ESRGAN [54] as it is the most developed and has achieved the best result of the previous mentioned GAN's. The Real-ESRGAN is trained on aerial data from cities in Greenland to make it more robust for the use in Greenlandic environment. Mask-RCNN makes for a great detector when it comes to building detection as it overlays a mask on the building footprint not just en-capturing it with a bounding box[19]. As the technical map needs to be as accurate as possible and the outlines of the house as perfect as can be, the Mask-RCNN is a great candidate

A delimitation is set as the world of super resolution is wide. This project focuses on GANs and more specifically the SRGAN family with the Real-ESRGAN implementation. The object detector has not been trained together with this project although was trained by the author in an earlier project[39].

## 1.1 Initial problem description

*How can super resolution and object detection work together for an optimized result in Greenlandic conditions?*

# 2 - Theory

## 2.1 Super-resolution

Super-resolution is an imaging technique for enhancing the resolution of an image. Super-resolution,2.1, is widely used in media up-sampling, medicinal research, surveillance and remote sensing. In this project remote sensing will be explored. The idea behind super resolution is to create a realistic high resolution images from its low resolution counterpart. There are many methods for doing this, the deep learning method using GAN networks will be discussed in this project. To get a low resolution out of a high resolution can be done using a number of techniques, such as adding noise, blur downsamlping and JPEG compression[11, 46, 54].

The equation 2.1 shows an image degradation formula, where $I_x$ is the low resolution image, $I_y$ is the high resolution image, $\sigma$ is the noise and $D$ is the degradation function. The two later variables are unknown and the algorithm is used to inversely find these parameters.

$$I_x = D(I_y; \sigma) \tag{2.1}$$

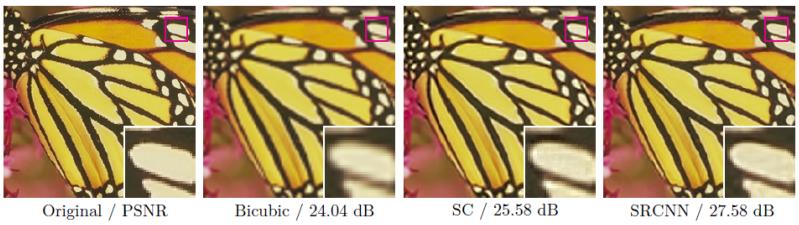


**Figure 2.1:** Comparison between different methods and their corresponding Peak Signal-to-Noise Ratio,PSNR, metric [11]

The simplest and most common ways of upsampling are nearest neighbour, bi-linear and bicubic interpolation. The nearest neighbour method spreads out the

pixels and fills the holes with the values of the neighbouring pixels. Bilinear interpolation is an improvement of the previous method, it takes the weighed average of the neighbouring pixels and fills out the empty pixels. This creates a more smooth looking image than the previous method. The third option is bicubic interpolation, which is an improved bilinear interpolation. Bicubic takes the 16 closet diagonal pixels in contrary to bilinear which only takes the 4 closets. This creates an even better outcome [17][59] examples can be seen in figure 2.2.

**Figure 2.2:** Comparing different simple super-resolution techniques, from left to right, nearest neighbour, bilinear, bicubic and ground truth. Depicted by Xu et. al.[59]

A simplified image of super resolution process can be seen in figure 2.3.

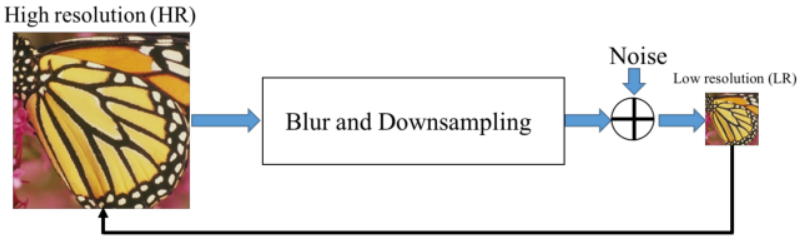


**Figure 2.3:** The main objective of super resolution is for the network to find a high resolution from its low resolution counterpart [60]

Super resolution leans up on the data processing inequality concept, which states that post-processing cannot increase data, as shown by the Markov chain $X \rightarrow Y \rightarrow Z$ where Z depends only Y and is conditionally independent of X [28]. This concepts states that super resolution is not possible without further information. By using of neural networks super resolution is indeed possible. The network gathers information by training on datasets that are build for super resolution problems, like the DIV2K dataset [4, 24]. The DIV2K dataset **??** is built up of 800 pictures for training, 100 pictures for validation and 100 pictures for testing. Each high resolution image has 3 corresponding low resolution image with different downscaling factors 2, 3 and 4.

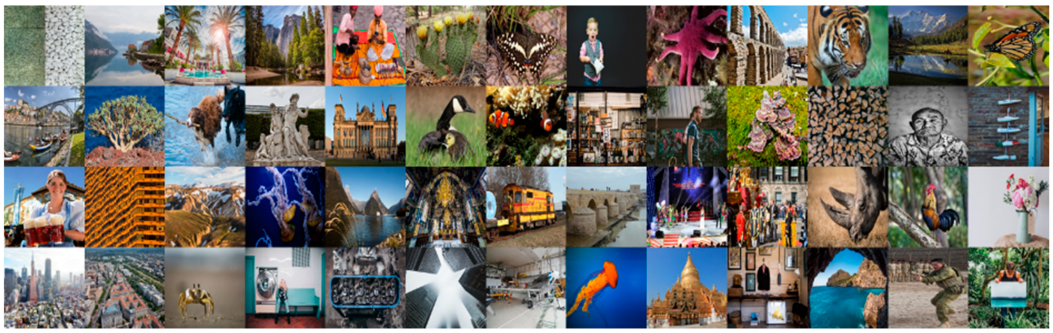

**Figure 2.4:** An extract of the DIV2K dataset [4]

## 2.2 Generative adversarial network - GAN

Generative adversarial networks or GANs, where introduced by Goodfellow et al. [16] and works by training two sub-models at the same time, a generative model and a discriminative model. The discriminative model is trained as an image classifier and its purpose is to classify the images, that are produced by the generative model, as fake or real data. A diagram of a GAN can be seen in figure 2.6. GANs are most notable in the image-to-image translation field, such as translating images from sunny to rainy and in generating realistic looking objects, people, etc that humans can not distinguish if they are fake or not. Another GAN usage is creating fake data or just creating data. A good example is face creation, the network creates faces that do not belong to real people. A picture of generated face by Nvidia [26] can be seen in figure 2.5

**Figure 2.5:** These people do not exist in real life and where created by a GAN designed by Nvidia [26]

The discriminator is a neural network and is trained with real images from a dataset and fake data from the generator. The loss function is maximised by

increasing the gradient. The generator is trained by generating fake data and passing it into the discriminator in order to trick it. The generator gradient is decreased in order to minimize the loss function. The generator is a neural network that creates fake data to be used in the discriminator. It takes data in form of a fixed length vector and adds noise to create fake data. The fake data is then run thought the discriminator, also a neural network, to be classified as real or fake data. A value of 0 to 1 is given to the data, 0 being fake and closer to 1 being real. The losses gives a penalty to the generator for failing to produce data that is real enough. Minmax loss was used by Goodfellow et al. [16].

Different models exists and GANs will be discussed and explored further in this project.

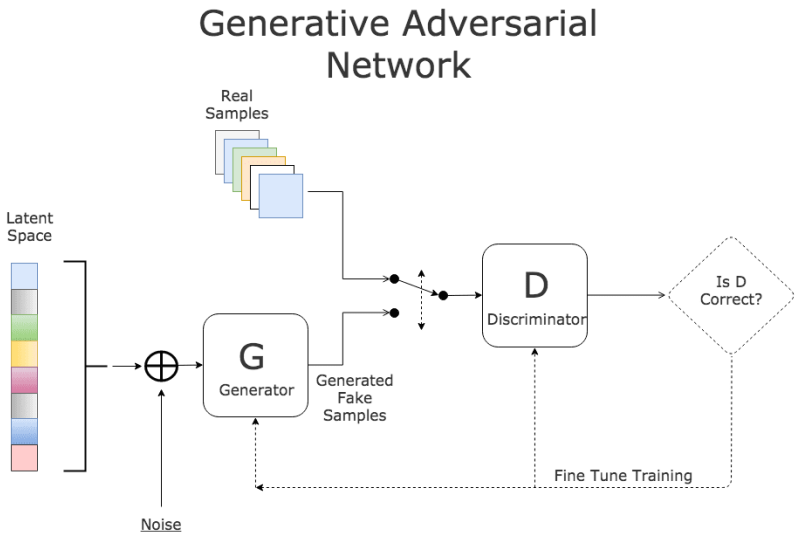


**Figure 2.6:** Noise is added to the latent space and passed thorough the generator, G, and forwarded to the discriminator, D, for evaluation. Tuning of the network is done by back propagation [15]

### 2.2.1 SRGAN

In 2017 Ledig et al. [29] proposed a new idea for super resolution problem and introduced GANs to the super resolution world, achieving state of the art 4x upscaling. The SRResNet uses a simple MSE loss[51], but Ledig et al. proposed a perceptual loss and content loss in their GAN version. The problem with earlier super resolution project where that the level of details where lacking, even though the metrics, such as PSNR[21] (peak-signal-to-noise-ratio), showed good results[23], this was solved by the introduction of GANs. By simply using SRResnet combined with MSE loss, shows that the results are a little blurry, to fix this issue Ledig et al. [29] proposed a SRGAN that is a GAN network combined with

a perceptual loss. A comparison of results can be seen in figure 2.7. The SRGAN is a GAN network and comprises of 2 networks competing against each other and thus improving themselves along the way.

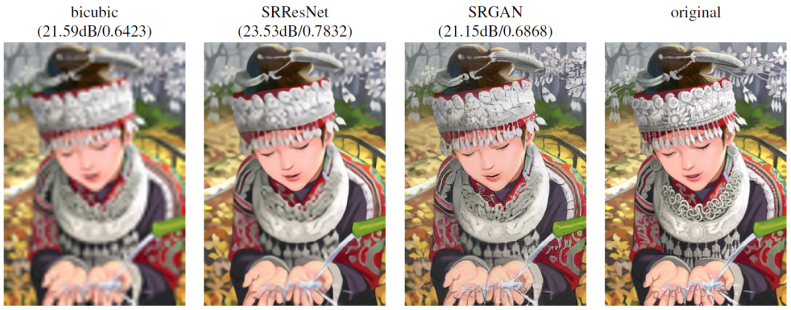


**Figure 2.7:** The SRGAN version outperforms the SRResNet by visually looking at it although the metrics states otherwise and thus PSNR cannot be used as a standalone validation tool. [29]

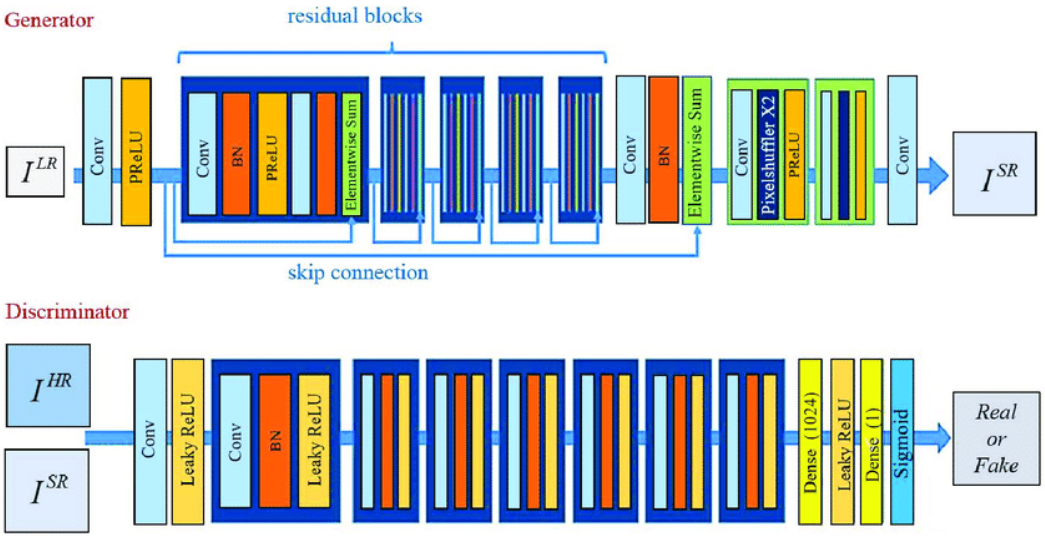


**Figure 2.8:** Divided into two parts the first is the generator, which generates the super resolution images and the second is the discriminator whose job is to determine the quality of the output from the generator. [29]

An illustration of the SRGAN can be seen in figure 2.8 with kernels sizes, feature maps and strides[29].The SRGAN takes a low resolution image as input and passes it through a convolution layer follow by a parametric rectified linear unit or PReLU. The idea of using PRelu[18] in the generator and Leaky ReLU[33] in the discriminator comes from the finding of Radford et. al.[44], who found that using these activation functions in this order gave good result, the SRGAN follows that order. The Leaky ReLU differs from the normal ReLU in such way that it fixes

the problem of a so called dead neuron. ReLU is an activation function where the negative dimension is zero and the positive dimension is positive. On the negative side all inputs become zero which results in a the so called dead neuron. To fix that issue an improved ReLU was proposed, the Leaky ReLU. The Leaky ReLU incorporates a slope for negative values and thus will have other values than zero. This parameter is usually a small number, eg. 0.02, used in the SRGAN model[29]. The PReLu is a form of Leaky ReLU but the parameter is learned by the use of backpropagation when training the network. The equation for ReLU, Leaky ReLU and PReLU can be seen in equation 2.2, 2.3 and 2.4 and the slopes can be seen in figure 2.9. In the first equation it can be seen that if everything up to zero, is zero[58]. The second equation tells us that a small value, in this case 0.02, can be used as a parameter used for multiplying every instance below zero, thus creating a value other than zero[33]. From the last equation we can derive that if $x$ is larger then zero the output is linear and if $x$ is less than zero, the output is $a$ times $x$. $a$ is a trainable value and thus it becomes a generalized ReLU function[18].

$$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases} \tag{2.2}$$

$$f(x) = \begin{cases} 0.02x & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases} \tag{2.3}$$

$$f(x) = \begin{cases} ax & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases} \tag{2.4}$$

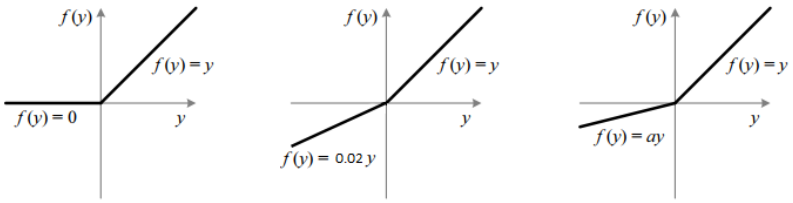


**Figure 2.9:** As can be seen PReLU has a value of 0 for all negative in contrast to LeakyReLU which multiplies the negative values with a small value and lastly PReLU which has the ability to learn the slope parameter. [18]

The network was trained on ImageNet [10] pictures using bicubic kernel to downsample the high resolution images to get the low resolution, a downsample factor of 4 was used. The authors of SRGAN[29] proved that by using GAN loss the results improved over MSE based loss. The MSE loss gives a better PSNR score but is much smoother and not visually appealing, MSE takes the average of all potential solution while the GAN wishes to trick the discriminator and creates one solution that could be realistic, as seen in image 2.10[29].

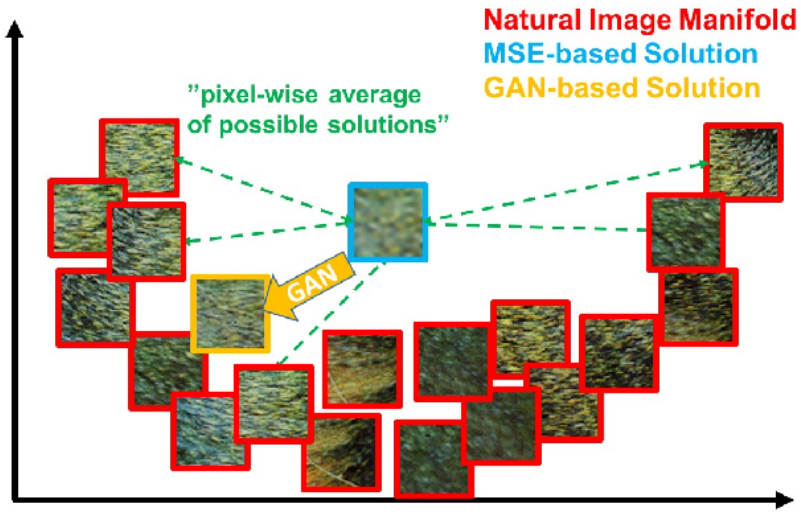


**Figure 2.10:** The GAN based solution only outputs one image as for the MSE solution, the loss takes an an average of all possible solutions [29]

### 2.2.2 ESRGAN

Xinntao et. al.[53] proposed in 2018 an improved version of the SRGAN, which was described in the previous section, called Enhanced SRGAN or ESRGAN. Some main components where improved e.g. the architecture, losses etc, some of them will be discussed in this section. Image 2.11 shows that the ESRGAN and has better texture detail than the previous SRGAN.

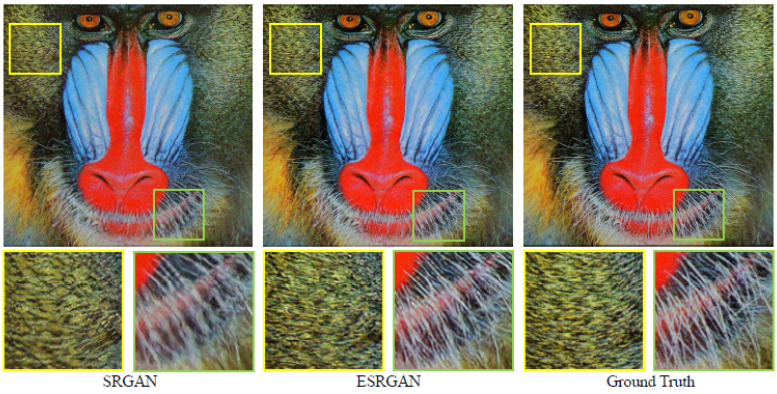


**Figure 2.11:** ESRGAN outperforms SRGAN in detail reconstruction.[53]

One of the main improvements is the change in architecture, namely the Residual Block is changed to a Residual-In-Residual dense block where the batch normalization is removed. Residual-in-Residual dense network (RRDB) was proposed by Zhang et. al. [61] and showed great improvement in super resolution problems.

The RRDB is based on the DenseNet model which connects all the layers within a residual block to each other [22], it can be seen in figure 2.12 which is also the proposed architecture for ESRGAN.

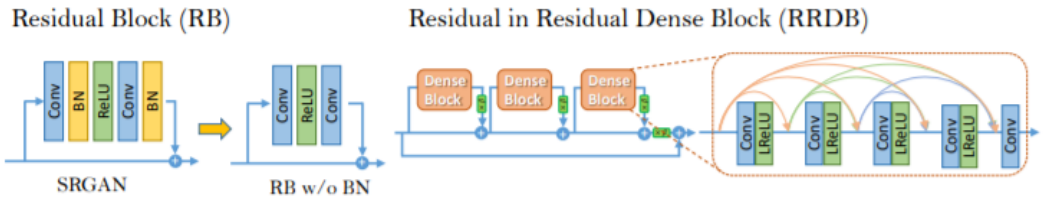


**Figure 2.12:** As can be seen the batch normalization (BN) is removed and dense blocks are introduced. This increases greatly the amount of convolutional layers by a great deal. The SRGAN had 32 convolutional layers versus the ESRGAN which has 345 layers.[53, 61]

By removing the batch normalization in the RRDB the computational complexity was removed and the performance was increased. Furthermore BN may introduce artifacts when the network is deep and when the training and datasets differs a lot, as seen in figure 2.13 [53, 61].

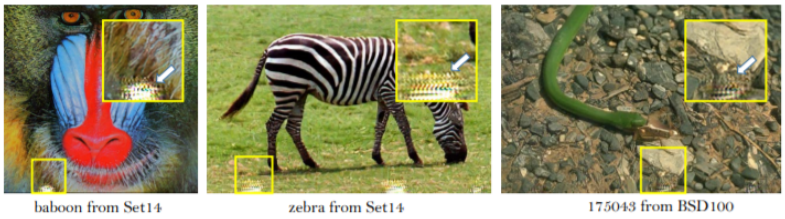


**Figure 2.13:** Artifacts can be minimized without batch normalization[53, 61].

A modified perceptual loss is also introduced, by measuring the loss before activation and by this the authors of ESRGAN overcame two problems. The first issue is the sparsity of the activated features, as is illustrated in figure 2.14. The second issue was that the brightness wasn't reconstructed properly when comparing to the ground truth[53].

### 2.2.3 Real-ESRGAN

As ESRGAN uses a bicubic downsampling that differs from the real world degradation, Xinntao et. al.[54] adopted an upgraded version of ESRGAN, called Real-ESRGAN. The authors of the new Real-ESRGAN introduced a so called second order degradation process, incorporating a second degradation process directly after the first one. The standard degradation first introduces blur, Gaussian blur, next a downsampling operation, noise is then added to simulate camera sensor artifacts, lastly JPEG compression is added to the image, [54, 12, 30]. Figure 2.15 shows the process follow by a second order degradation.

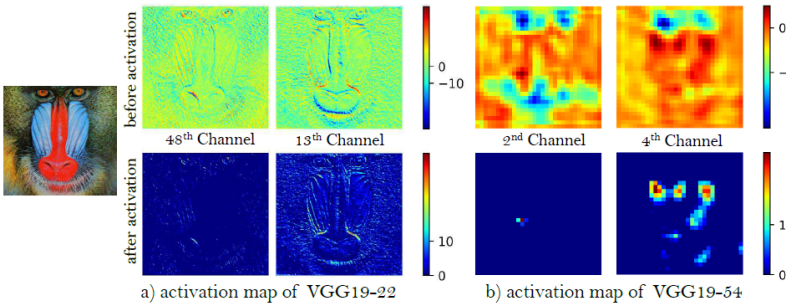


**Figure 2.14:** Feature maps before activation are more prominent than after.[53].

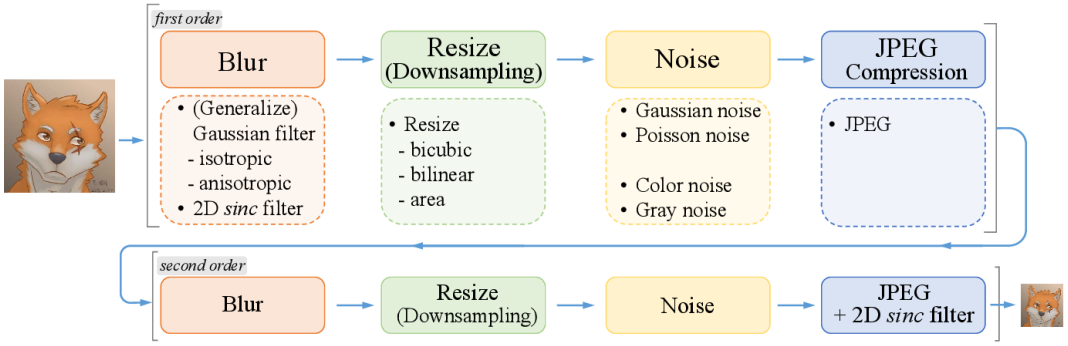


**Figure 2.15:** To improve super resolution results the creator of Real-ESRGAN introduced a high order degradation process to better simulate real life degradation[54].

The equation 2.5 shows the degradation process, where $x$ depicts the low resolution image, D is the degradation process of $y$. $K$ denotes the blur, while $r$ is the downsampling process and $n$ is noise, lastly *jpeg* compression is used.

$$x = D(Y) = [(y \circledast k) \downarrow_r + n]_{jpeg}[54] \tag{2.5}$$

The degradation process in super resolution is used to form pairs of image in a dataset. As single image super resolution is based on only one image the degradation process is necessary to create a complete dataset. Xinntao et. al.[54] argued that a high order degradation was needed to replicate real life image degradation as close as possible. Real world loss of quality comes from multiple combined degradation over several different steps. E.g. images are captured using low quality camera sensors, thus introducing the first step of loss of quality. Other degradation process may include uploading to a social media account, editing using software, old photos being digitally scanned etc. In the process a lot of noise, blur, compression and other artifacts are introduced. Ringing artifacts are a common issue in image processing. It appears on sharp edges of objects in an image. The artifacts usually evolves from sharpening algorithms like the jpeg compression [37]. The sinc filter[9] is introduced as a method of creating ringing artifacts

for image pairs in the dataset. The filter is used in the blurring process and as the last degradation method in the degradation model[54]. Illustration 2.16 shows the ringing and over shoot artifacts as well as the sinc filter with different cutoff frequencies, marked $\omega_c$.

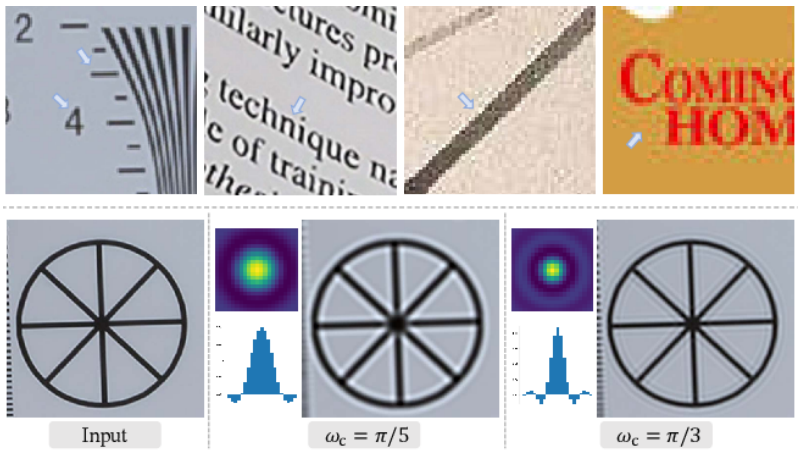


**Figure 2.16:** The top image sequence shows ringing and overshoot artefacts. The bottom sequence shows how different ringing problems appear when using different cutoff frequencies. [54]

Based on the ESRGAN [53] the Real-ESRGAN needs to handle larger degradation processes and thus an update to the discriminator was needed. The U-Net [49, 47] structure 2.17 was proposed for the discriminator architecture. The U-Net uses skip connections and provides feedback on a pixel level[54].

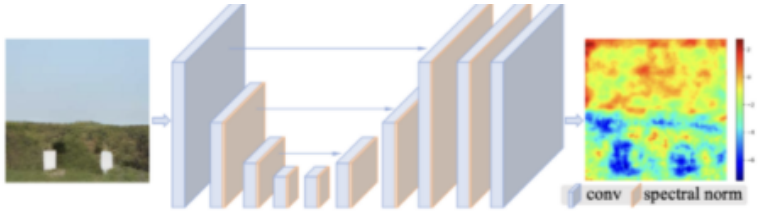


**Figure 2.17:** The U-Net, a spectral normalization was also added for network stability[54, 36].

### 2.2.4 Other super resolution methods

**A-ESRGAN** 2.17 is based on the Real-ESRGAN architecture and incorporates the same generator as earlier models but uses a Attention U-Net Discriminator. The Attention U-Net is usually used for 3D medical image segmentation [41], but the authors of A-ESRGAN have modified it to be used with 2D pictures [56]. The attention U-Net focuses, or gives attentions, to the target details in an image on the contrary of a standard U-Net.[36] This results in sharper edges and less distortion.

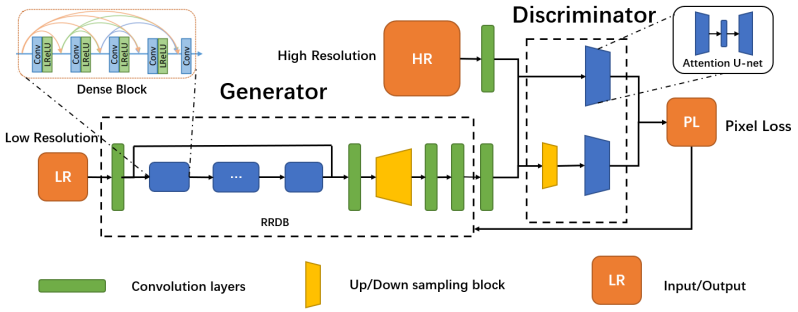


**Figure 2.18:** The generator design is the same as in Real-ESRGAN but a portion of the discriminator is changed into a Attention U-Net [56]

**Multiple Image Super resolution** is a super resolution task solved by using multiple images of the same object, eg. satellite imagery taken at different times of the same location. By combining and comparing theses a super resolution image can be created. Multiple image super resolution is not as widely studied as single image, there are many challenges involved with multi-image super resolution, eg. the scenery may change, the quantity of low resolution images may vary etc. To overcome these challenges Nalepa et.al.[40] proposed a StatNet that could be used for multiple image super resolution. StatNet includes a so called StatBlocks that computes different pixel-wise statistics eg. Min, Max , etc. and from there the feature maps are calculated, an illustration of the network can be seen in figure 2.19.

### 2.2.5 PSNR, SSIM and other quality metrics

**Peak-Signal-to-Noise-Ratio - PSNR** is one of the most widely used metrics and is an expression of the ratio between signal and noise. PSNR can be expressed as the equation2.8. $L$ is the signal part and would be 255 in standard 8-bit image. PSNR is measured in decibel and the higher the decibel the better se figure2.20 [27]. The short fall of PSNR is that it incorporates MSE, which looks at images on pixel level, i.e. it may give a high error for an image that looks, in human eyes, almost perfect.

$$psnr = 10\log_{10}(L^2/MSE) \tag{2.6}$$

**Structural Similarity Index Measure - SSIM** 2.21 is a quality metrics that looks at the image structure and is thus better for assessing super resolution images as it is closer to what humans perceive an image. SSIM looks at groups of pixels and is a product of luminance, contrast and structure as seen in equation 2.7, where $x$ and $y$ denotes the input and the output respectively. SSIM has a defined range and is always between -1 and 1. where 1 is a perfect match and -1 is an imperfect

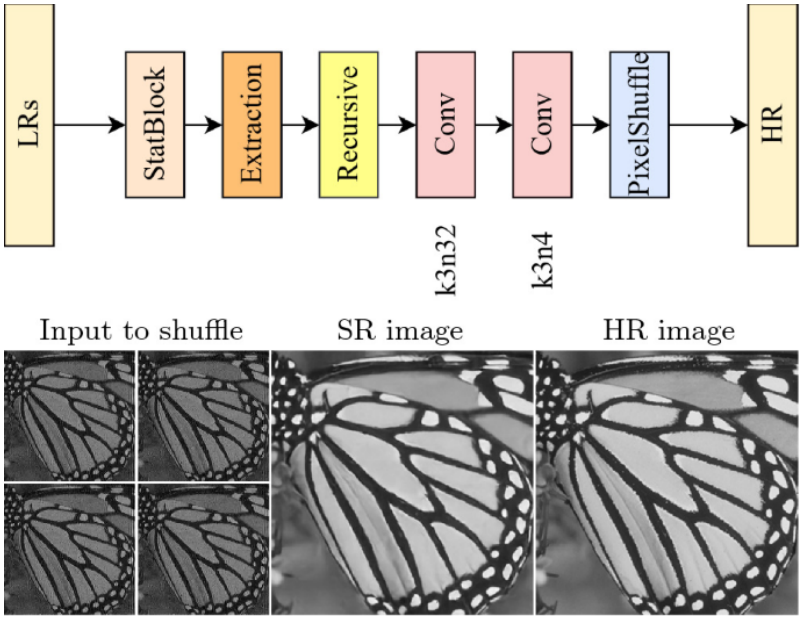


**Figure 2.19:** The StatNet starts by inputting several LR images that passes through a StatBlock followed by an ExtractionBlock which is superseded by RecursviceBlock. Two convolution layers and a pixelshuffler ends the network architecture [40]

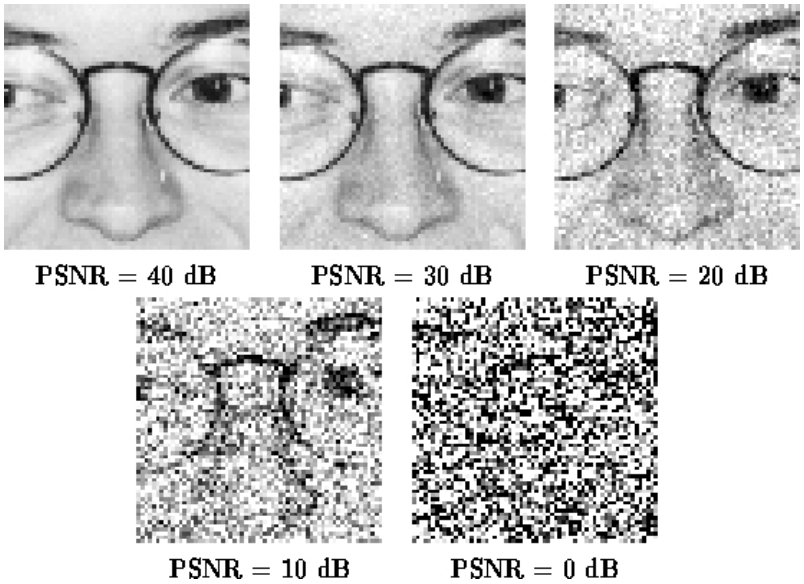


**Figure 2.20:** Increased decibel improves the result[52].

match[55].

$$SSIM(x,y) = [l(x,y)] * [c(x,y)] * [s(x,y)] \tag{2.7}$$

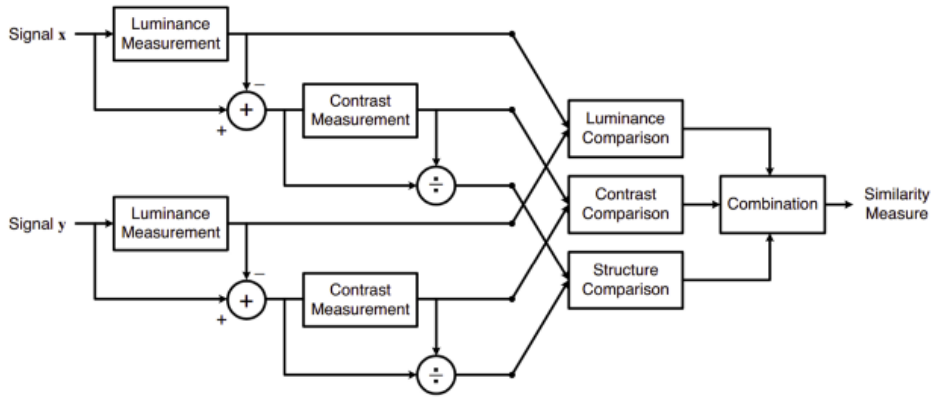


**Figure 2.21:** SSIM work flow represented in an image[55].

**Perceptual-Distortion** is a measure of rating super resolution results on an x-y graph. Where the root mean squared error is on the x-axis and the perceptual index is on the y-axis. Figure 2.22 shows different algorithms being ranked and it can be determined that no algorithm is both good in distortion (RMSE) nor perceptual quality, this is known as the Perception-Distortion trade off[35].

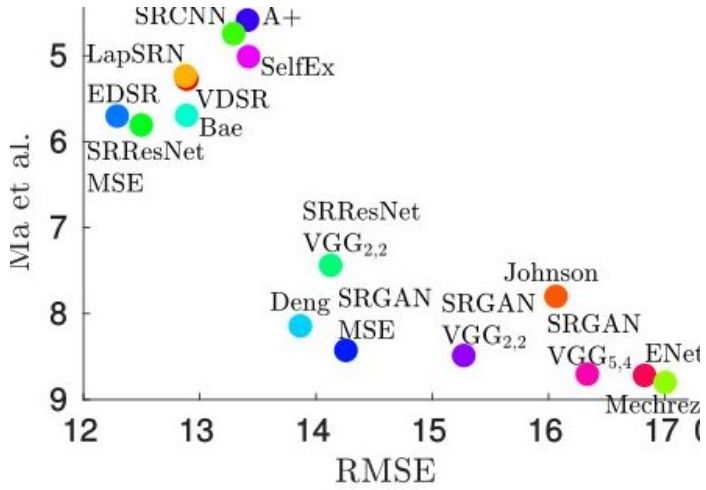


**Figure 2.22:** The x-axis shows the distortion measured in RMSE and the y-axis shows the Natural Image Quality Evaluator, NIQE, an metric that is a close as possible to human perception.[35, 7, 32].

### 2.2.6 Loss

One of the most used loss functions in machine learning is mean squared error loss. In super resolution problem the MSE function computes the average of the squares of each pixel error[31] . The loss equation can be seen in equation 2.8.

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} - y_{ij})^2 \tag{2.8}$$

$i$ and $j$ indicates the pixel location in the image, $x$ and $y$ are the the compared images and $m$ and $n$ are their size[31]. The issue addressed by Ledig et. al. [29] is that the MSE loss gives high value on the PSNR metric but visually looks overly smooth[29]. This can be seen in figure 2.7, the second image from the left uses only MSE loss. The authors of SRGAN[29] combined content loss and adversarial loss to create a perceptual loss. Johnson et. al.[25] proposed in 2016 a perceptual loss for super resolution and image transfer problems. The perceptual loss is based on the idea of being closer to perceptual similarity[53, 8]. The authors of[29] used a VGG-16 network to output a loss function. They used because of the good mapping feature VGG-16 network has, it encodes information about the image and has the capability of distinguishing one image from another thus giving it perceptual abilities [29] 2.23 .

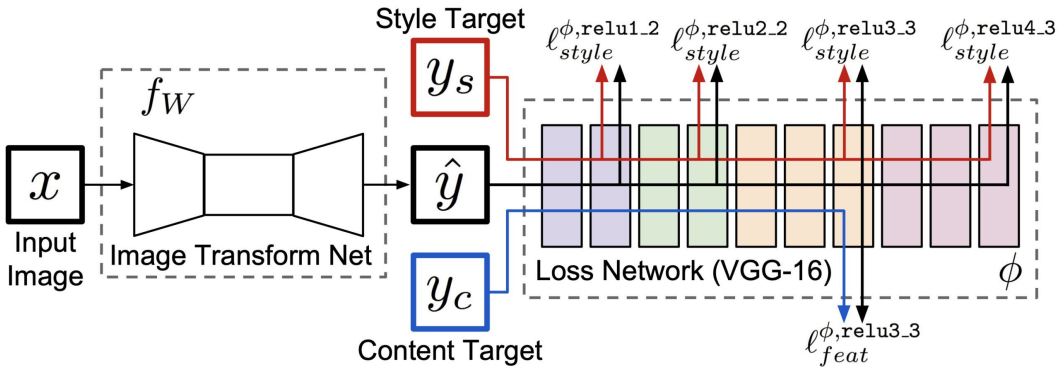


**Figure 2.23:** The image shows the architecture behind the VGG loss. [25]

Figure 2.24 shows a comparison between results of perceptual loss in both super resolution and style transfer. In the style transfer case the pixel, in the result, vary quite a bit and the MSE loss would give a low score on the other hand the perceptual loss gives a high score[25].

## 2.3 Mask-RCNN

Mask-RCNN, which stands for Masked Regional Convolutional Neural Network, is a instance segmentation neural network made by He et. al[19]. The Mask-RCNN works by extracting regions of interest then passing them through a neural network that predicts the class and showing the end results in a mask form. This differs from other RCNN architectures by working on a pixel level and thus not just

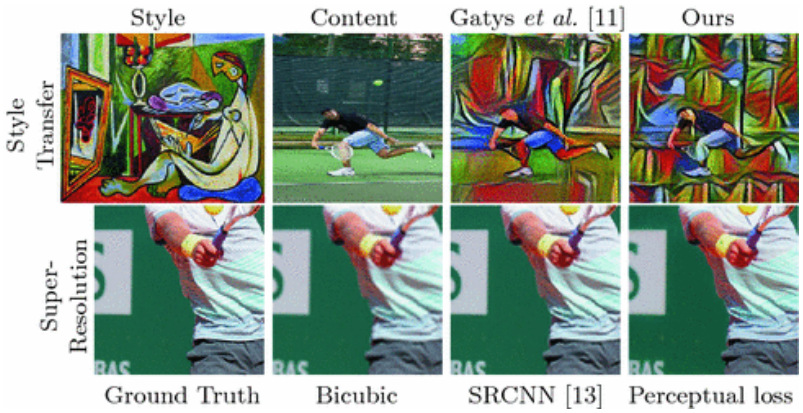


**Figure 2.24:** A comparison between results of the VGG loss mapping. [25, 14]

outputting the bounding box but outputting the outlines of the object detected[39, 45]. Mask-RCNN is an image segmentation architecture, image segmentation is a technique for partitioning images into different segments contain the object of interest. Instance segmentation on the other hand is to differentiate the the objects segmented in image segmentation, thus combining semantic segmentation and object detection, figure 2.25. Mask-RCNN can be used in many applications, eg. COVID-19 detection in X-Ray images of patient lungs it was found that it was far better then other deep learning methods [42], another good example would be building detection or more specific to find the exact footprint of that building [39].

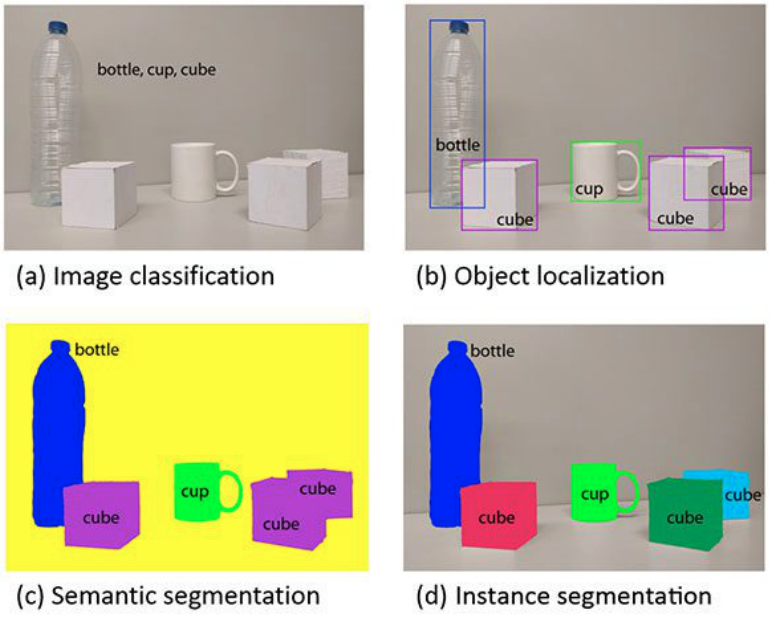


**Figure 2.25:** The figure shows the different types of classification and segmentation. [13]

# 3 - Implementation

## 3.1 Method

As this project focuses on satellite data and upscaling it, the Real-ESRGAN[54] algorithm was chosen for the project. ESRGAN [53] was initially explored but further investigation showed Real-ESRGAN proved to be superior in terms of training data handling, as it uses a degradation solution to create synthetic dataset. Due to the lack of paired training data this is optimal, often the freely available satellite data is usually of low resolution and high resolution data could be expensive. The degradation process is explained in section 2.2.3 and the first training process follows the standard degradation parameters as described in Wang et. al.[54]. The algorithm is then finetuned by introducing different datasets, results are compared and checked.

**Figure 3.1:** The ground truth shown in red and the trained Mask-RCNN shown in white.[39, 6]

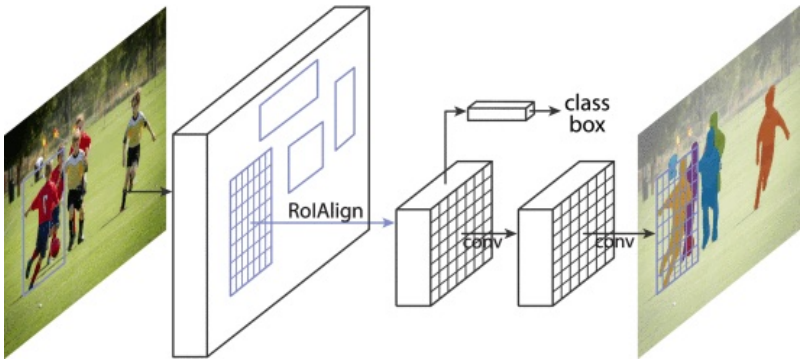


**Figure 3.2:** A Mask RCNN outputs the mask of the detected object and not the bounding box, which is ideal for detecting house footprints.[19, 39]

As the final step the images are run through a building detection algorithm, a Mask-RCNN [19], to asses if the super resolution images have an increase performance of detecting buildings in Greenland's landscape. The Mask-RCNN is trained on buildings in Greenland and was not trained as a part of this project 3.1[39]. Illustration 3.3 shows the flow of processes in this project.

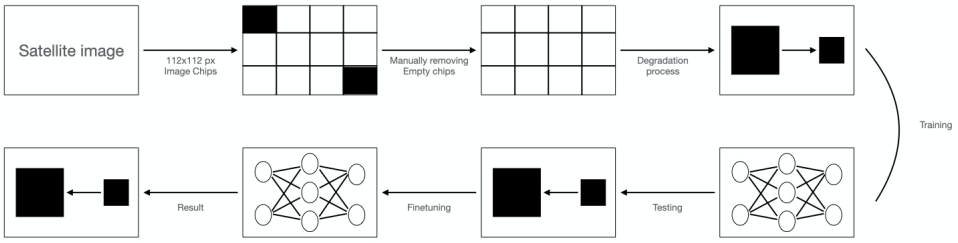


**Figure 3.3:** A single satellite image is first converted to 112x112 pixels sized image crops and the process continues with manually removing blank crops followed by the Real-ESRGAN degradation process[54] after this step, the training begins. The blank corps are found on the edges of the satellite image where no or very little data exists. Results are checked afterwards and finetuning is applied if necessary and the end results are then investigated.

## 3.2 Dataset

Asiaq[6]has during many years acquired a lot of data, either drone-, satellite or other remote sensing imagery. As the Real-ESRGAN [54] has been pretrained on DIV2K dataset[4, 24], which contains daily life image such as image of landscape, people etc, high resolution aerial imagery was chosen as dataset. Asiaq acquires high resolution aerial data with a spatial resolution of 0.1m in combination with

technical map updating. The aerial imagery are orthophotos, meaning they are geo-accurate and can be used as geo-referencing. The city of Ilullisat was chosen as the training dataset area due to its large variety of landscape with a combination of buildings, figure 3.4 shows a part of the dataset.

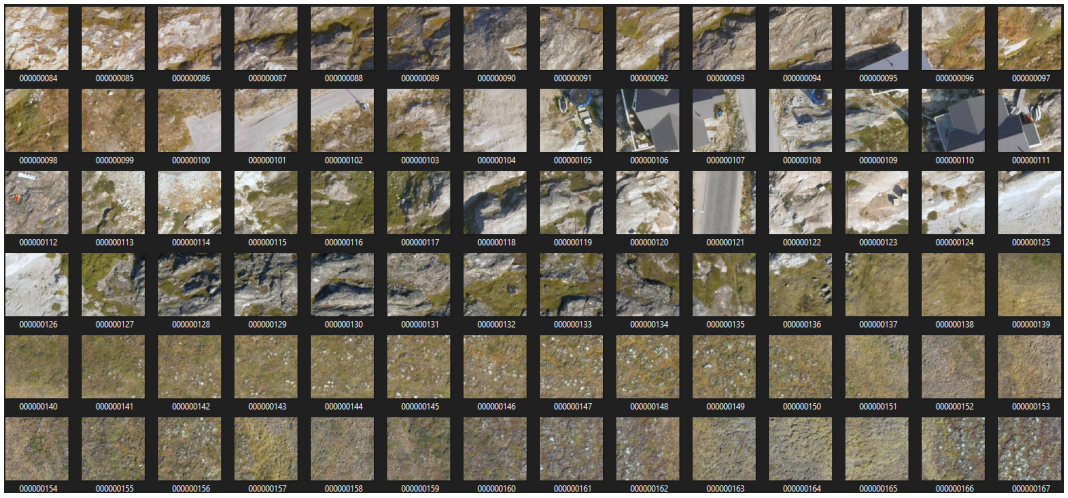

**Figure 3.4:** Ilullisat was chosen due to its variety of landscape and buildings[6].

The whole orthophoto of Ilullisat is 70000x90000 pixels so image crops where exported using ArcGIS Pro [5]. The image crops where chosen to be 112x112 pixels, as earlier experiments showed that larger image crops would fill the GPU ram and the computer was not able to process them. The data was then checked for images not containing data, eg. in a corner of the orthophoto where no info exits, and deleted. Images containing half and half where also deleted. The end result was a dataset with more than 43000, 112x112 image crops. Around 30 images where chosen as test images, the figure 3.5 shows some of these test images.

An extension to the previous dataset was also created, figure 3.6. This set contains more buildings and was used to fine tune the algorithm to make it more robust when upsampling buildings, a total of 13198 images where produced. Although buildings where present in the first dataset less than 1% of image crops contained buildings.

As the original plan was to upsample Sentinel 2 satellite imagery, Sentinel 2 dataset 3.7 was also created. Sentinel 2 imagery is freely available on Sentinel EO hub[50]. The data was downloaded and imported into ArcGIS where it was extracted to the desired dataset format which was explained earlier in this section. The freely available Sentinel 2 data has a spatial resolution of 10m/pixel whereas the earlier mentioned aerial imagery has a resolution of 0.1m/pixel [1, 6]. This dataset contains 13608 images.

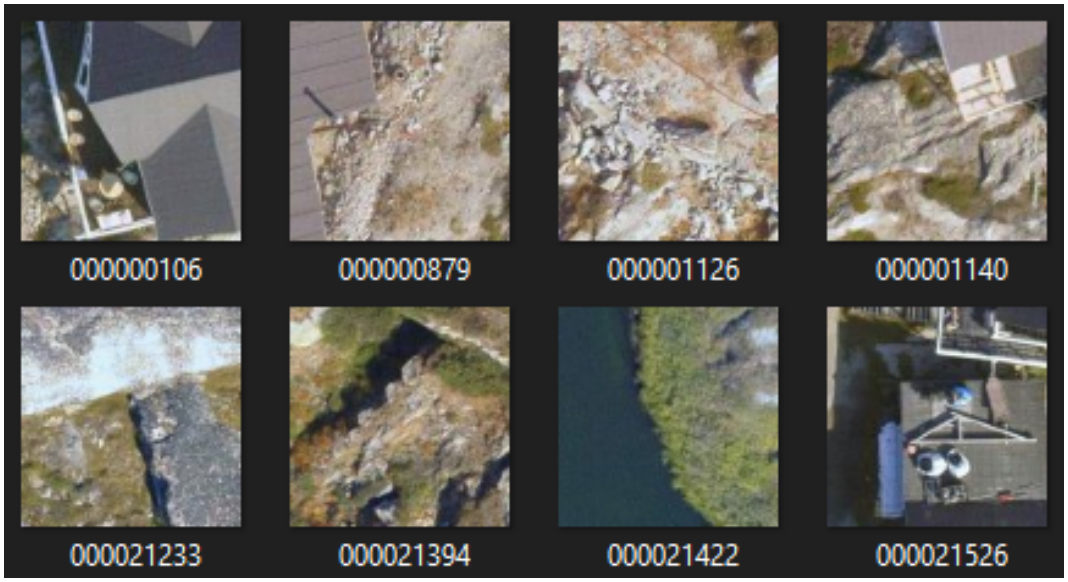


**Figure 3.5:** Image crops from the test dataset[6].

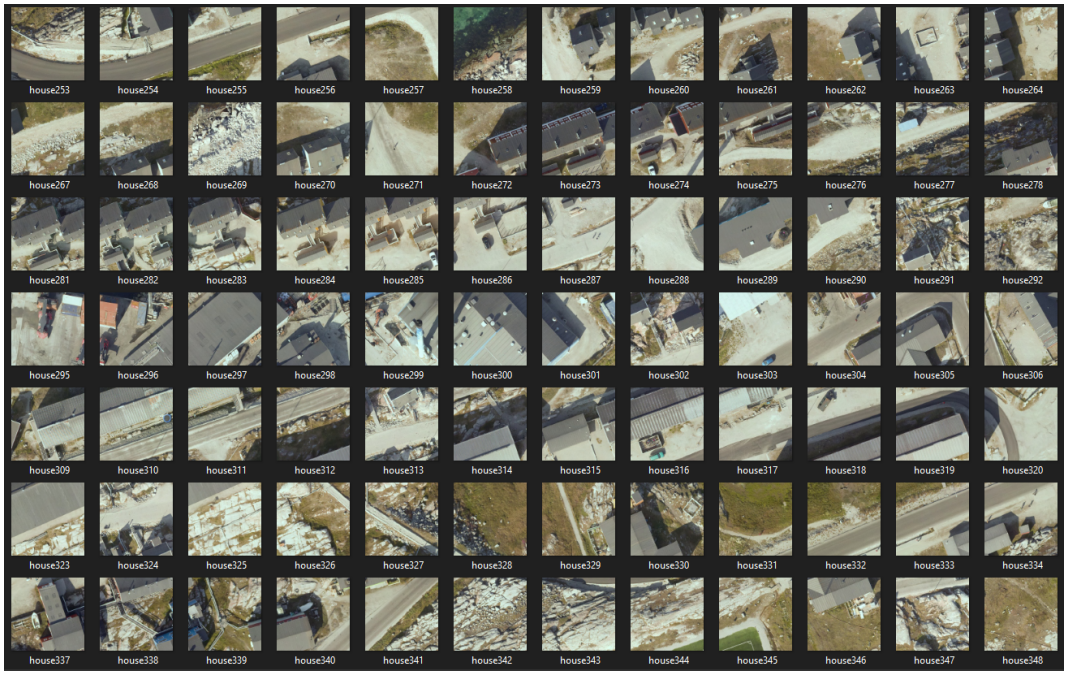


**Figure 3.6:** More buildings where included in this dataset expansion.[6].

An additional dataset was also created from SPOT 3.8 satellite data. Unlike Sentinel 2 data, SPOT is not freely available and has to be purchased. SPOT, specifically SPOT 6 and 7 has a resolution of 1.6m/pixels and covers the whole

**Figure 3.7:** Freely available Sentinel 2 data has a spatial resolution of 10m/pixels and has 13 different bands to choose from.[50, 1, 20].

coastal area of Greenland[48, 3]. This dataset was created for further training of the algorithm and for testing consisting of 18122 images.

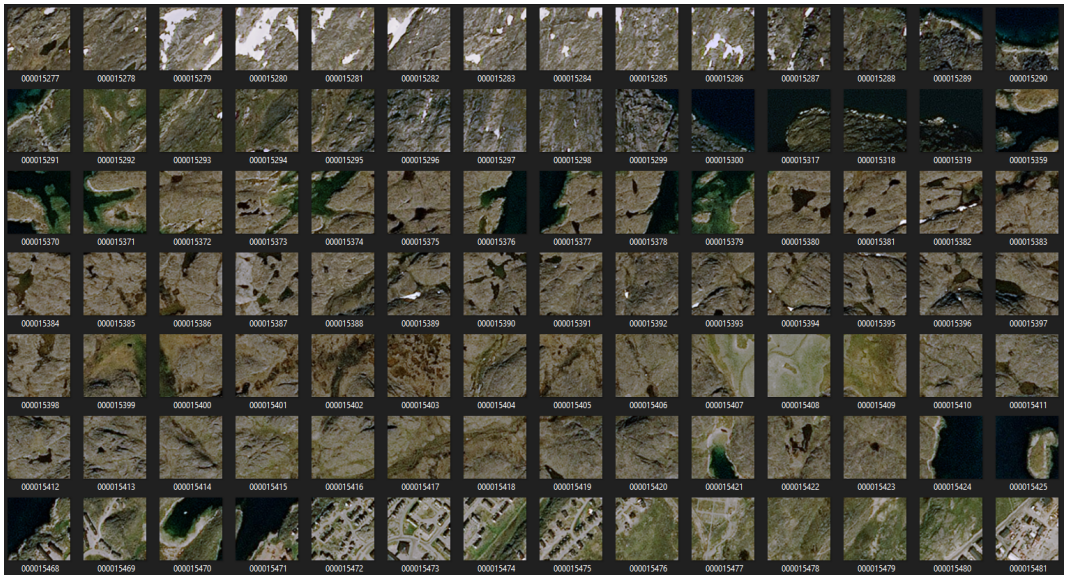

**Figure 3.8:** SPOT satellites 6 and 7 cover the whole coast of Greenland .[50, 1, 20].

### 3.2.1 Data preparation

As there are no image pairs in the dataset, the pairs are made on the go. A degradation process is used to degrade a high resolution image by adding noise, blur etc and thus creating a low resolution image as close as possible to real life degradation. Two full degradation process where used in sequence of each other. By having two full processes after one another insures that the degraded image is a close as real life degradation, that can come from many different factors, eg. sensor artifacts, jpeg compression, uploading to the internet etc. In reality an image degradation doesn't follow a specific pattern and the source can be completely unknown, thus resulting in a number of unknown degradation steps.

## 3.3 Training

Training was done on a Windows 10 operated computer with an Intel Xeon CPU, an Nvidia Quadro K2200 GPU with 4gb of RAM and 68gb of RAM memory. PyTorch[43] was used as the python framework along with a diversity of python libraries. 400k iterations was chosen initially and took around 17 days to train. Due to hardware limitation a batch size of 8 was used and images of a size of 112x112 where used. Adam was used as an optimizer for both the generator and the discriminator with a learning rate of 1e-4 and a multi step learning decay function [38], that reduces the learning rate by half every 100.000th iteration. The other iterations follow the same parameters except for iterations, they where reduced to 100k to reduce time cost and with multi step learning rate decaying the rate every 33000th iteration.

## 3.4 Results

The model was tested on some of the 30 images randomly chosen to be test dataset. The PSNR and SSIM where used as quality metrics as well as visually accessing the images. PSNR and SSIM where calculated using Matlab[34], both metrics where explained in section 2.2.5.Figure 3.9 shows 4 images, from left to right, the low resolution image, a 4x bicubic upsampling, a 4x upsampling using Real-ESRGAN [54] first iteration and a 4x upsampling using Real-ESRGAN second iteration. The first iteration implies a training of the algorithm using a dataset mostly containing landscape whereas the second iteration contains buildings, making the algorithm more robust on buildings. The baseline is also compared against these two iterations.

At first glans the results looks good, it outperforms the bicubic upsampling and is visually more fine than the coarse bicubic upsampling. A closer look reveals that fine details are lost and have become rather smooth, as the project aims to

**Figure 3.9:** Low resolution, bicubic, first iteration result and second iteration result. SSIM for the figures are 0.82, 0.84 and 0.84, respectively The PSNR is 29.5dB, 29.4dB and 28.9dB

improve a building detector, this should not effect the results, the detection part will be discussed later in the project. Some details on the house are lost in the first iteration but are somewhat retrieved in the second. Ringing and overshoot problem have been solved with the addition of houses to the training, as can bee seen on figure 3.9 which is a shadow projected on a roof from figure 3.10.

**Figure 3.10:** Before and after the addition of houses to the dataset.

As the overall results look smooth the edges of the roof has been sharpened in the second iteration. The transition between building and ground is more notable in the later, this may greatly affect the outcome of the detector, figure 3.11.

A second example can be seen in figure 3.12, here a house with a terrace, where some terrace details have been improved 3.13. This can also be seen in figure 3.14

The important factors for an object detector is that the edges of the houses needs to be clearly distinguishable from the surrounding environment. Wherever or not a patch of grass is visible or recognizable is not so important in this matter.

More examples can be seen in figures below.

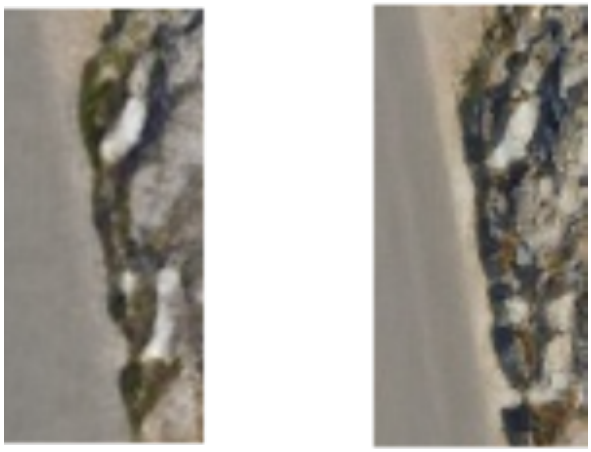

**Figure 3.11:** The edge has been improved in the second iteration.

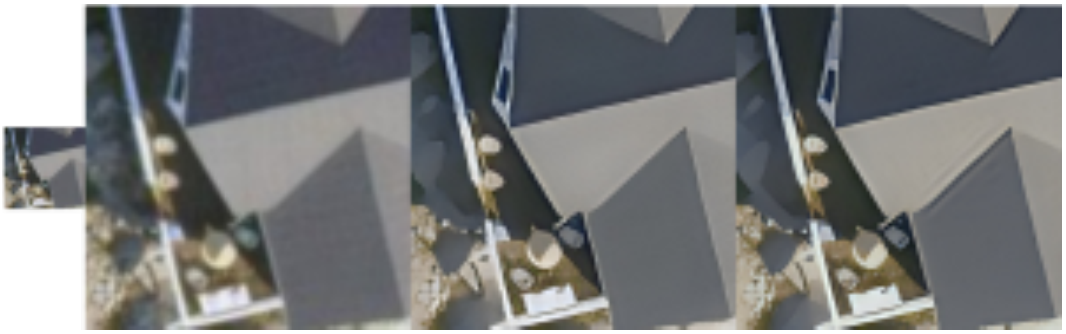

**Figure 3.12:** Sharper edges means more accurate object detection

**Figure 3.13:** Some of the details on the terrace have been improved.

In figure 3.15 we can see that the bicubic upsampling has made swirls in the image and the results looks unnatural. The first and second iteration has yielded

**Figure 3.14:** An example of a house with a terrace.

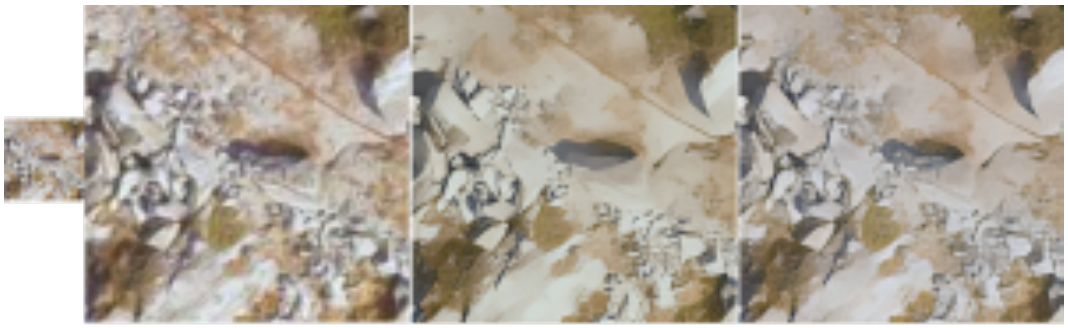

**Figure 3.15:** A good example where bicubic has having a hard time reconstructing images.

good results. Although this figure shows non man made objects there has been an improvement between iteration one and iteration two. Cut outs from the respective image can be seen in figure 3.16 and 3.17.

**Figure 3.16:** The rock appear less smooth on the right.

The figures 3.18 to 3.26 shows additional results, the low resolution and its super resolution counterpart.

A finetuning of the algorithm was also performed by adding Sentinel 2 images to the dataset. Even though the initial project was directed towards Sentinel 2 image upsampling. The finetuning was conducted with a Sentinel 2 dataset containing around 13000 images and was trained for 100000 iterations. The outcome was not as good as expected and resulted in discolouring of the super resolution

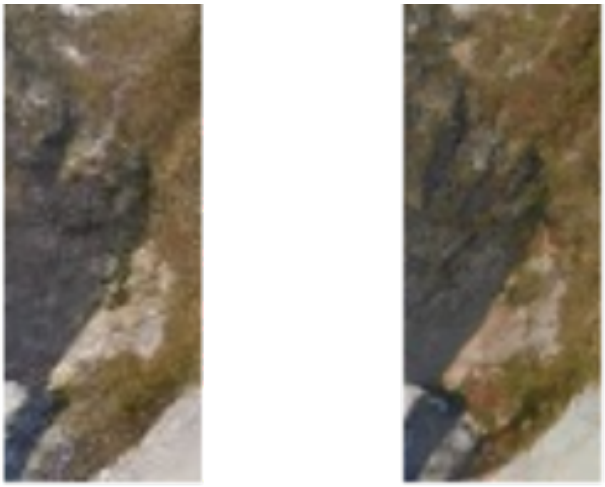

**Figure 3.17:** Less pixelated on the right.

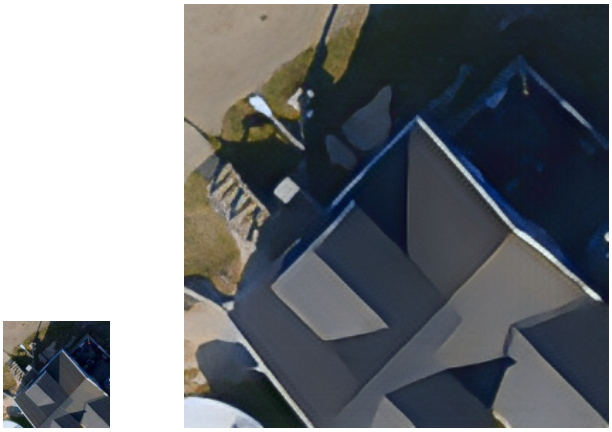

**Figure 3.18**

images. The results can be seen in the figures 3.27.

The main object was to investigate if a dataset containing aerial imagery from Greenland could improve an object detector for detecting building in Greenland environments. The object detector is a Mask-RCNN trained on buildings in the Greenlandic city of Ummannaq and has proven to yield good results. A minor flaw that was found during this project was the detector has problems detecting small building when next to bigger houses. This problem was helped by introducing super resolution imagery. Images of the settlement of Kangaamiut was chosen as testing grounds for the detector. Image crops of 300x300 pixels where extracted over different parts of the settlement. The crops where manually chosen to diversify the objects found in the images, images without buildings where also chosen to check the performance of the detector. Figure 3.28 shows the low resolution

**Figure 3.19**

**Figure 3.20**

**Figure 3.21**

Figure 3.22

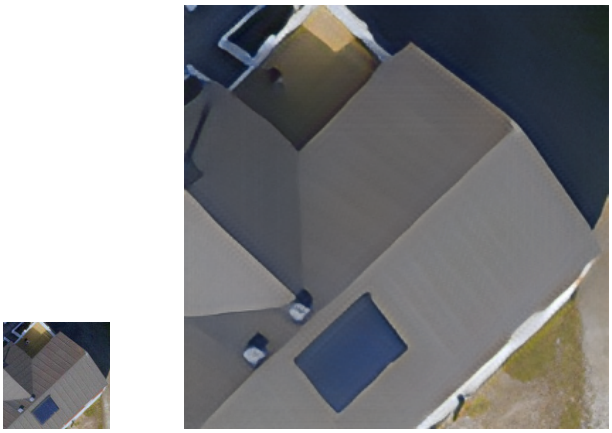

Figure 3.23

Figure 3.24

**Figure 3.25**

**Figure 3.26**

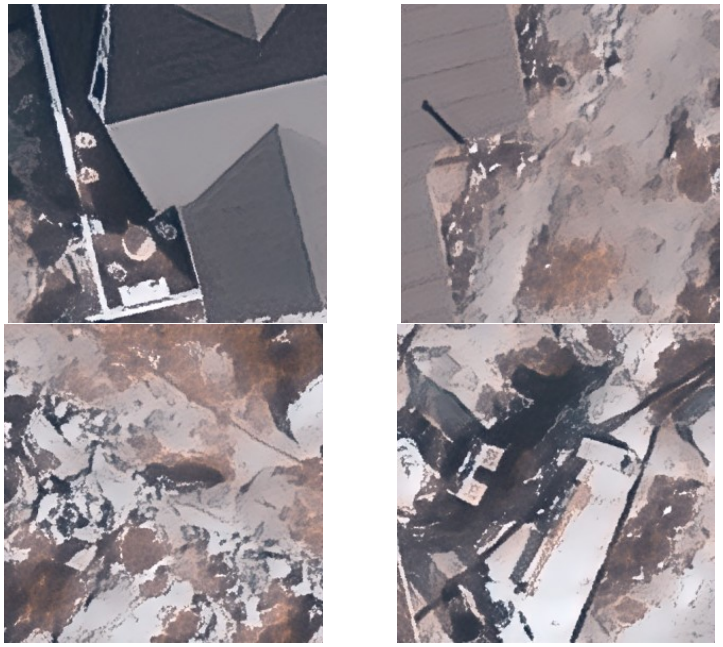

**Figure 3.27:** The color mismatch originates from the Sentinel 2 dataset

compared to the super resolution, in both cases the detector does a descent job of finding the house, but has trouble accurately outlining the buildings, this could be a result of having trained the algorithm on low resolution training data and thus not making it as accurate on the newly produced super resolution imagery.

In some cases the detector didn't succeed in finding small buildings, especially when close or next to a larger structure, when using low resolution images. The figures 3.29 and 3.30 show that when including a super resolution image the problem was solved.

The asses the object detector the precision and recall was calculate with a 50% Intersect over Union or IoU. The precision, 3.2 is the ratio of detected true positives and all the detected objects and the recall, 3.3 is the ratio between detected objects and actual amount of objects. A true positive needs to be in the right place and have the right classification, in this project there is only one class, building The IoU, figure 3.31, on the other hand is the ratio of overlap between the detected object and the ground truth. A 50% overlap is sufficient for assessing a detector model.

$$IoU(b_{pred}, b_{gt}) = \frac{Area(b_{pred} \bigcap b_{gt})}{Area(b_{pred} \bigcup b_{gt})} \tag{3.1}$$

$$Precision(P) = (TP/(TP + FP))x100 \tag{3.2}$$

**Figure 3.28:** The detector has found all buildings in both cases(The low resolution image is enlarged for visual convenience).

**Figure 3.29:** Super resolution solved the problem of finding this small building(The low resolution image is enlarged for visual convenience).

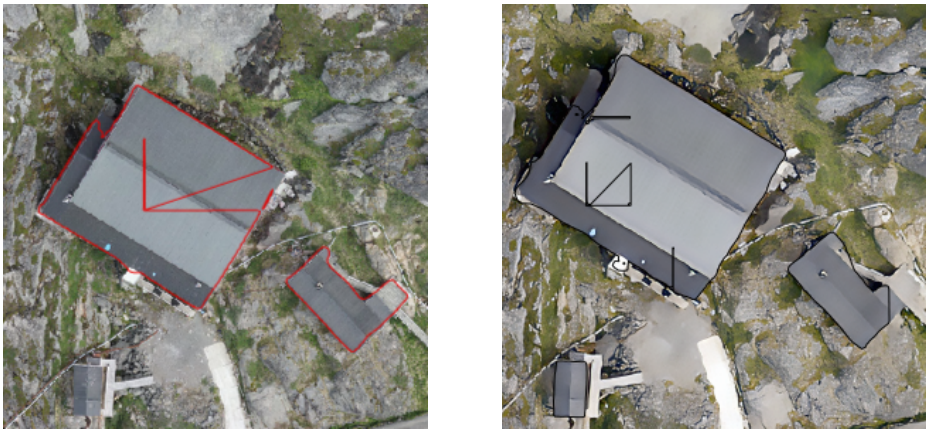

**Figure 3.30:** As can be seen in the lower left corner the low resolution(The low resolution image is enlarged for visual convenience) the building ha not been found, but in the super resolution the building is found.

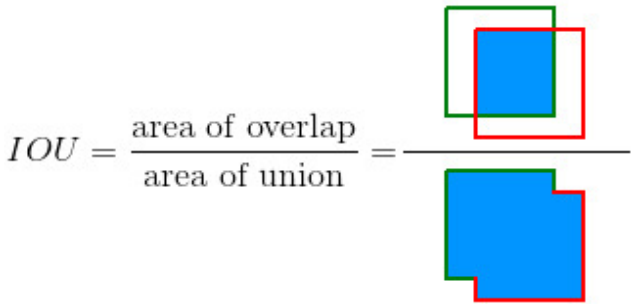


**Figure 3.31:** The IoU is the area of overlap divided by the area of union. [2]

$$Recall(R) = (TP/(TP + FN))x100 \tag{3.3}$$

$b_{gt}$ represents the ground truth mask and $b_{pred}$ is the max for the predicted result of the algorithm. The IoU threshold is usually set to 50% [39, 57]. The results can be seen in table 3.1

| Method: | Precision: | Recall: |
|---|---|---|
| Low resolution | 81.8% | 92.5% |
| Bicubic | 82.7% | 87.9% |
| Real-ESRGAN | 85.3% | 90.2% |
| Real-ESRGAN GL | 87.5% | 97.5% |

**Table 3.1:** The precision for the different methods

In the following figures shows different methods of super resolution are compared and and run through the Mask-RCNN building detection.

**Figure 3.32:** The first figures shows a bicubic upsampling the second a plain Real-ESRGAN and lastly the Real-ESRGAN with Greenland data.

**Figure 3.33:** Another example where the detector fails in finding buildings in the bicubic image and the plain Real-ESRGAN. But the later found "houses" along the dirt road.

In figure 3.32 we can see that the bicubic version fails to find the building in the right upper corner and only finds half a house in the right center but finds a "house" in the water. The second figure finds two out of three houses and the last finds all houses. Anothe example can be seen in figure 3.33.

In some cases all three algorithms fail to find a specific building. In figure 3.34 the building with the green roof has gone unnoticed, but the building next to it with grey roof is detected. This could be caused by similarity, since the majority of houses in the Mask-RCNN training dataset has grey roof. This can be solved by adding a Digital Surface Model, DSM, data to the dataset, as explained by Mäkinen et. al. [39]. The DSM is not color specific and only accounts for elevation data.

**Figure 3.34:** In this example all three models failed to find the building with the green roof.

# 4 - Conclusion

As more and more freely low resolution satellite data is available and the Greenland government needs to keep an up to date technical map, a combination of super resolution and object detection was investigated, that could solve for a more accurate building detector. A 4x upsampling was achieved by combining Real-ESRGAN and aerial data from different cities in Greenland. Firstly the super resolution algorithm was trained and when results where satisfactory the up sampled images where tested on a Mask-RCNN object detector. The detector was not trained in conjunction with this project. The overall results where good, with an improvement when using super resolution image on the Mask-RCNN, even though the detector was not trained on super resolution images it achieved good results. Some buildings went undetected, as seen in figure 3.34, the building with a green roof has been missed by the detector. This is probably caused by the similarity in the detector dataset. As the majority of houses have grey roofs, a roof with a different colour could cause the detector to miss that building.

## 4.1 Future work

Although the images are not visually perfect to the human eye, the object detector does a descent job of finding buildings. A perfect outline of the image was not accomplished, as seen in figures 3.29 and 3.30, but could be resolved by re training or finetuning the detector with super resolution images, as the detector is now trained on low resolution imagery. Digital Surface Model could be introduced to the dataset when training the detector. As explained by Mäkinen et. al. [39] the DSM adds elevation data as an extra band to a satellite image and has shown to improve the Mask-RCNN detector. A DSM accounts only for elevation data and is a powerful extra data in the Greenlandic landscape as there are minimal trees and other high bushes to confuse with buildings. Figure 4.1 shows a DSM and a Digital Terrain Model.

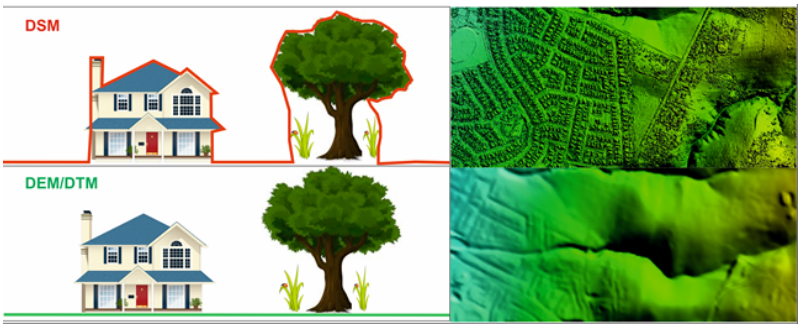


**Figure 4.1:** The DSM is a powerful tool in the tree-free Greenlandic environment.

# Bibliography


[1] URL: https://www.sentinel-hub.com/.

[2] *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. 2020. DOI: 10.1109/iwssip48289.2020. URL: https://doi.org/10.1109/iwssip48289.2020.

[3] European Space Agency. *Spot*. URL: https://earth.esa.int/eogateway/missions/spot.

[4] Eirikur Agustsson and Radu Timofte. "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 1122–1131. DOI: 10.1109/CVPRW.2017.150.

[5] *ArcGIS pro*. URL: https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview.

[6] *Asiaq*. URL: https://www.asiaq-greenlandsurvey.gl/frontpage/.

[7] Yochai Blau and Tomer Michaeli. "The Perception-Distortion Tradeoff". In: (2017). DOI: 10.48550/ARXIV.1711.06077. URL: https://arxiv.org/abs/1711.06077.

[8] Joan Bruna, Pablo Sprechmann, and Yann LeCun. *Super-Resolution with Deep Convolutional Sufficient Statistics*. 2015. DOI: 10.48550/ARXIV.1511.05666. URL: https://arxiv.org/abs/1511.05666.

[9] W. Chou, T.H. Meng, and R.M. Gray. "Time domain analysis of sigma delta modulation". In: *International Conference on Acoustics, Speech, and Signal Processing*. IEEE. DOI: 10.1109/icassp.1990.115820. URL: https://doi.org/10.1109/icassp.1990.115820.

[10] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[11] Chao Dong et al. "Image Super-Resolution Using Deep Convolutional Networks". In: *CoRR* abs/1501.00092 (2015). arXiv: 1501.00092. URL: http://arxiv.org/abs/1501.00092.

[12] M. Elad and A. Feuer. "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images". In: *IEEE Transactions on Image Processing* 6.12 (Dec. 1997), pp. 1646–1658. DOI: 10.1109/83.650118. URL: https://doi.org/10.1109/83.650118.

[13] Alberto Garcia-Garcia et al. *A Review on Deep Learning Techniques Applied to Semantic Segmentation*. 2017. DOI: 10.48550/ARXIV.1704.06857. URL: https://arxiv.org/abs/1704.06857.

[14] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. *A Neural Algorithm of Artistic Style*. 2015. DOI: 10.48550/ARXIV.1508.06576. URL: https://arxiv.org/abs/1508.06576.

[15] *Generative adversarial networks - hot topic in machine learning*. URL: https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html.

[16] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[17] Dianyuan Han. "Comparison of Commonly Used Image Interpolation Methods". In: *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*. Atlantis Press, 2013. DOI: 10.2991/iccsee.2013.391. URL: https://doi.org/10.2991/iccsee.2013.391.

[18] Kaiming He et al. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015. DOI: 10.48550/ARXIV.1502.01852. URL: https://arxiv.org/abs/1502.01852.

[19] Kaiming He et al. "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

[20] *Home*. URL: https://sentinel.esa.int/web/sentinel/missions/sentinel-2.

[21] Alain Horé and Djemel Ziou. "Image Quality Metrics: PSNR vs. SSIM". In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 2366–2369. DOI: 10.1109/ICPR.2010.579.

[22] Gao Huang et al. *Densely Connected Convolutional Networks*. 2016. DOI: 10.48550/ARXIV.1608.06993. URL: https://arxiv.org/abs/1608.06993.

[23] Youyou Huang et al. "Deep Learning-Based Inverse Scattering With Structural Similarity Loss Functions". In: *IEEE Sensors Journal* 21.4 (2021), pp. 4900–4907. DOI: 10.1109/JSEN.2020.3030321.

[24] Andrey Ignatov, Radu Timofte, et al. "PIRM challenge on perceptual image enhancement on smartphones: report". In: *European Conference on Computer Vision (ECCV) Workshops*. 2019.

[25] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*. 2016. DOI: 10.48550/ARXIV.1603.08155. URL: https://arxiv.org/abs/1603.08155.

[26] Tero Karras et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *CoRR* abs/1710.10196 (2017). arXiv: 1710.10196. URL: http://arxiv.org/abs/1710.10196.

[27] Onur Keleş et al. *On the Computation of PSNR for a Set of Images or Video*. 2021. DOI: 10.48550/ARXIV.2104.14868. URL: https://arxiv.org/abs/2104.14868.

[28] Peter E. Latham and Yasser Roudi. *Mutual information*. URL: http://www.scholarpedia.org/article/Mutual_information.

[29] Christian Ledig et al. *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*. 2016. DOI: 10.48550/ARXIV.1609.04802. URL: https://arxiv.org/abs/1609.04802.

[30] Ce Liu and Deqing Sun. "On Bayesian Adaptive Video Super Resolution". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.2 (2014), pp. 346–360. DOI: 10.1109/TPAMI.2013.127.

[31] Zhengyang Lu and Ying Chen. *Single Image Super Resolution based on a Modified U-net with Mixed Gradient Loss*. 2019. DOI: 10.48550/ARXIV.1911.09428. URL: https://arxiv.org/abs/1911.09428.

[32] Chao Ma et al. *Learning a No-Reference Quality Metric for Single-Image Super-Resolution*. 2016. DOI: 10.48550/ARXIV.1612.05890. URL: https://arxiv.org/abs/1612.05890.

[33] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.

[34] *Matlab*. URL: https://se.mathworks.com/products/matlab.html.

[35] A. Mittal, R. Soundararajan, and A. C. Bovik. "Making a "Completely Blind" Image Quality Analyzer". In: *IEEE Signal Processing Letters* 20.3 (Mar. 2013), pp. 209–212. DOI: 10.1109/lsp.2012.2227726. URL: https://doi.org/10.1109/lsp.2012.2227726.

[36] Takeru Miyato et al. *Spectral Normalization for Generative Adversarial Networks*. 2018. DOI: 10.48550/ARXIV.1802.05957. URL: https://arxiv.org/abs/1802.05957.

[37] Ali Mosleh, J. M. Pierre Langlois, and Paul Green. "Image Deconvolution Ringing Artifact Detection and Removal via PSF Frequency Analysis". In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 247–262. DOI: 10.1007/978-3-319-10593-2_17. URL: https://doi.org/10.1007/978-3-319-10593-2_17.

[38] *Multisteplr¶*. URL: https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.MultiStepLR.html.

[39] Oscar Mäkinen. *Exploring Digital Surface Model as an extra image band in building detection*. 2022.

[40] Jakub Nalepa, Krzysztof Hrynczenko, and Michal Kawulok. "Multiple-Image Super-Resolution Using Deep Learning and Statistical Features". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2021, pp. 261–271. DOI: 10.1007/978-3-030-73973-7_25. URL: https://doi.org/10.1007/978-3-030-73973-7_25.

[41] Ozan Oktay et al. *Attention U-Net: Learning Where to Look for the Pancreas*. 2018. DOI: 10.48550/ARXIV.1804.03999. URL: https://arxiv.org/abs/1804.03999.

[42] Soumyajit Podder et al. "An efficient method of detection of COVID-19 using Mask R-CNN on chest X-Ray images". In: *AIMS Biophysics* 8.3 (2021), pp. 281–290. DOI: 10.3934/biophy.2021022. URL: https://doi.org/10.3934/biophy.2021022.

[43] *Pytorch*. URL: https://pytorch.org/.

[44] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2015. DOI: 10.48550/ARXIV.1511.06434. URL: https://arxiv.org/abs/1511.06434.

[45] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2015. DOI: 10.48550/ARXIV.1506.01497. URL: https://arxiv.org/abs/1506.01497.

[46] Yaniv Romano, John Isidoro, and Peyman Milanfar. *RAISR: Rapid and Accurate Image Super Resolution*. 2016. DOI: 10.48550/ARXIV.1606.01299. URL: https://arxiv.org/abs/1606.01299.

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. DOI: 10.48550/ARXIV.1505.04597. URL: https://arxiv.org/abs/1505.04597.

[48] *Satellitbilleder*. URL: https://sdfe.dk/saadan-arbejder-vi-med-data/satellitbilleder.

[49] Edgar Schönfeld, Bernt Schiele, and Anna Khoreva. *A U-Net Based Discriminator for Generative Adversarial Networks*. 2020. DOI: 10.48550/ARXIV.2002.12655. URL: https://arxiv.org/abs/2002.12655.

[50] *Sentinel-Hub EO-browser3*. URL: https://apps.sentinel-hub.com/eo-browser/.

[51] Wenzhe Shi et al. *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*. 2016. DOI: 10.48550/ARXIV.1609.05158. URL: https://arxiv.org/abs/1609.05158.

[52] Todd Veldhuizen. *Grid Filters for Local Nonlinear Image Restoration.* 1998. URL: http://hdl.handle.net/10012/943.

[53] Xintao Wang et al. *ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks*. 2018. DOI: 10.48550/ARXIV.1809.00219. URL: https://arxiv.org/abs/1809.00219.

[54] Xintao Wang et al. *Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data*. 2021. DOI: 10.48550/ARXIV.2107.10833. URL: https://arxiv.org/abs/2107.10833.

[55] Z. Wang et al. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612. DOI: 10.1109/tip.2003.819861. URL: https://doi.org/10.1109/tip.2003.819861.

[56] Zihao Wei et al. *A-ESRGAN: Training Real-World Blind Super-Resolution with Attention U-Net Discriminators*. 2021. DOI: 10.48550/ARXIV.2112.10046. URL: https://arxiv.org/abs/2112.10046.

[57] Xiongwei Wu, Doyen Sahoo, and Steven C. H. Hoi. *Recent Advances in Deep Learning for Object Detection*. 2019. DOI: 10.48550/ARXIV.1908.03673. URL: https://arxiv.org/abs/1908.03673.

[58] Bing Xu et al. *Empirical Evaluation of Rectified Activations in Convolutional Network*. 2015. DOI: 10.48550/ARXIV.1505.00853. URL: https://arxiv.org/abs/1505.00853.

[59] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. "Single-Image Super-Resolution: A Benchmark". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 372–386. ISBN: 978-3-319-10593-2.

[60] Wenming Yang et al. "Deep Learning for Single Image Super-Resolution: A Brief Review". In: (2018). DOI: 10.48550/ARXIV.1808.03344. URL: https://arxiv.org/abs/1808.03344.

[61] Yulun Zhang et al. *Residual Dense Network for Image Super-Resolution*. 2018. DOI: 10.48550/ARXIV.1802.08797. URL: https://arxiv.org/abs/1802.08797.