TECHNISCHE UNIVERSITÄT
ILMENAU

Environment Based Transformation of Outdoor Images using Neural Networks

Research Project

Adeel Ahmed

Matriculation Number: 62748

Masters in Research in Computer and Systems Engineering

In cooperation with
Virtual Worlds and Digital Games Group

Supervisors:
Prof. Dr. Wolfgang Broll
M. Sc. Christoph Gerhardt

Fakultät Wirtschaftswissenschaften und Medien
Technische Universität Ilmenau

ABSTRACT

The translation of image-to-image is a significant and difficult task in the field of computer vision. The goal is to learn the distribution of corresponding images in the target domain given an image in the source domain. In the past, various methods have been used to address this problem, but it becomes particularly challenging when dealing with multiple modalities. The development of deep learning and the growth of technology have allowed researchers to use deep learning techniques, rather than traditional computer vision algorithms, to solve this problem. In 2014, Ian Goodfellow introduced Generative Adversarial Networks (GANS), which can generate realistic images and effectively learn distributions. This was a significant advancement in image-to-image translation, as the resulting images were highly realistic.

In this project, we will train four different networks - Pix2Pix (3.1), Pix2PixHD (3.2), CycleGAN (3.3), and StarGAN (3.4) - on a custom dataset called "Seasons", which includes both paired and unpaired images of the four seasons. The goal is to determine which model most effectively translates the input image to a summer, winter, spring, and autumn setting. We trained the Pix2Pix and Pix2PixHD models on a paired dataset, and the CycleGAN and StarGAN models on an unpaired dataset. After training, we tested the models on unseen images and generated output images. To evaluate the performance of each model, we conducted a public survey in which participants were shown the generated images from each model and asked to rate the image quality and desired season attribute. 105 participants completed the survey, and the results were analyzed.

The results of the comparison between Pix2PixHD and CycleGAN show that both networks performed similarly in terms of producing high-quality images and desired attributes. If a strict decision must be made, Pix2PixHD may be considered the winner based on the highest number of votes. However, it should be noted that CycleGAN's ability to operate without paired datasets makes it a more versatile option for tasks where such data is not available. We recommend using CycleGAN for the task of transforming outdoor images using neural networks based on the environment.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

Many problems in computer vision involve translating images from one domain to another, such as image colorization, inpainting, attribute transfer, and style transfer. The image-to-image translation (I2I) problem has received significant attention due to its numerous applications. When we have a dataset with paired training data, the problem can be solved using a regression model or a conditional generative model (CGAN). A significant amount of work has been done in the field of computer vision to address the image translation problem. The I2I task is one of the primary tasks in computer vision, in which an image from the source domain is translated to the target domain while maintaining the originality of the source image.

A Generative Adversarial Network (GAN) is currently the best candidate for image-to-image (I2I) tasks as the generated results are significantly superior to those produced by earlier approaches. GANs consist of two networks: a generator and a discriminator. The generator learns from the input image and then generates a fake image. The discriminator learns the loss from the difference between real and fake images and determines whether the generated image is fake or real. Depending on the network architecture, the discriminator or generator may be punished.

In the paper, 'Paired Translation with Conditional GAN' (Pix2Pix) [1], a conditional GAN (cGAN) is used to learn a mapping between paired images in the source and target domains. The paper 'HD Image Synthesis and Semantic Manipulation with cGANs' (Pix2PixHD) [2] is an improved version of Pix2Pix that can learn the mapping at higher resolutions. BicycleGAN [3] learns a distribution of possible outputs and uses it as a cGAN model setting, enabling it to generate multimodal outputs for a fixed input. SPA-GAN [4] introduces an attention mechanism directly into the GAN architecture and proposes a novel spatial attention GAN model for image-to-image translation tasks. Consistent Embedded Generative Adversarial Networks (CEGAN) [5] aims to learn conditional generation models for generating perceptually realistic outputs and captures the full distribution of potential multiple modes of results by enforcing connections in both the real image space and latent space.

In unsupervised learning, the source and target image sets are completely independent from each other between the two domains. This is because acquiring paired training data can be expensive or even impossible for various applications. The solution for this unpaired learning is to use unsupervised learning approaches, such as CycleGAN [3], DualGAN [6], and MUNIT [7] and StarGAN [8].

We will discuss four of these techniques in detail, the Pix2Pix (3.1), Pix2PixHD (3.2), CycleGAN (3.3) and the StarGAN network (3.4). We will later implement these four techniques on a Seasons dataset (4.1) containing paired and unpaired images of all four seasons. We will use this dataset to train our models to generate images in various modalities. In total, we have trained 37 models, including 12 Pix2Pix models, 12 Pix2PixHD models, 12 CycleGAN models, and 1 StarGAN model.

After training these models, we evaluated their performance on the test dataset (4.2) by generating images of the desired season. As we do not have ground truth images to compare the results of our networks, we conducted a public survey (4.3.1) to gather subjective evaluations of the generated images from human participants. In the survey, the participants were shown 48 images and asked to rate the quality and desired generated attribute of each image. A total of 105 complete responses were collected before the survey ended.

We analyzed survey results and used the data to compare the generated images. We evaluated paired and unpaired models separately and plotted the results. We also compared all models to determine the best overall model according to human perception. Our analysis in Section (4.3.2) revealed a tie between Pix2PixHD and CycleGAN, both receiving a similar number of votes. Additionally, results for StarGAN and Pix2Pix were also similar. However, we conclude that CycleGAN is a superior model as it is unsupervised and can also handle paired datasets.

2 LITERATURE

This chapter focuses on the basic building blocks used for defining models in image-to-image translation. These networks learn the relationships between different modalities and generate unique outputs. We will cover Conditional Generative Adversarial Networks (2.1.1), U-NET (2.1.2), PatchGAN (2.1.3), and ResNet (2.1.4).

2.1 NETWORKS

In this section, we will discuss the networks that are used to translate the image from one modality to another.

2.1.1 *Conditional Generative Adversarial Networks (CGAN)*

A Conditional Generative Adversarial Network (CGAN) is a type of GAN that can generate images based on labels or auxiliary information during training. The generator and discriminator models are both conditioned on this additional information, which allows the model to learn a mapping between inputs and outputs using various contextual information. In a CGAN, the generator and discriminator are conditioned on auxiliary information from other modalities, allowing the model to learn a multi-modal mapping from inputs to outputs using different contextual information [9].

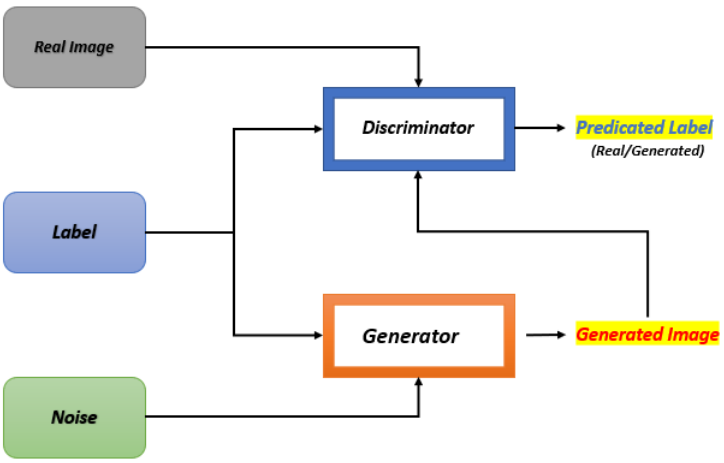

Figure 2.1.1: CGAN Basic Architecture
[10]

2.1.2 *UNET*

UNET is a U-shaped encoder-decoder network architecture, which consists of encoder-decoder blocks that are connected via a bridge.

The encoder is a traditional convolutional neural network (CNN) that processes the input image and extracts high-level features. The decoder is then responsible for upsampling the feature maps produced by the encoder and generating the segmentation mask. The decoder also uses skip connections that concatenate the feature maps from the encoder at the same resolution to the upsampled feature maps produced by the decoder, which helps to preserve spatial resolution and fine details in the segmentation mask.

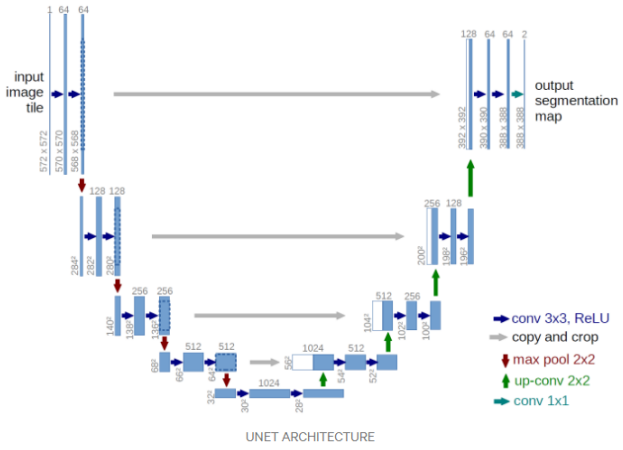

Figure 2.1.2: UNET Model
[11]

This network is mainly used in image segmentation tasks and also in medical tasks as it is capable of generating very good fine structures. The model of UNET can be defined as:

- Each block in the encoder is: Convolution -> Batch normalization -> Leaky ReLU.
- Each block in the decoder is: Transposed convolution -> Batch normalization -> Dropout (applied to the first 3 blocks) -> ReLU.
- There are skip connections between the encoder and decoder

The advantage of UNET is that the network is able to handle small, highly unbalanced datasets. The U-Net is able to preserve spatial resolution and fine details in the segmentation mask due to the skip connections that concatenate the feature maps from the encoder at the same resolution to the upsampled feature maps produced by the decoder. The U-Net is able to learn rich feature representations due to the encoder-decoder structure, which can be useful for downstream tasks such as object detection and classification.

2.1.3 *PatchGAN*

In a PatchGAN, the discriminator network is trained to classify whether a small patch in the image is real or synthetic, rather than the entire image as a whole. The patch size is typically much smaller than the size of the input image. For example, if the input image is 256x256 pixels, the patch size might be 70x70 pixels [12]. The discriminator network consists of a series of convolutional layers that process the image patches and output a single scalar value indicating the probability that the patch is real. The generator network is trained to produce synthetic images that are indistinguishable from real images to the discriminator. As an example from the paper [1] the authors have used a 70 by 70 pixels PatchGAN. If we have an image of size (256x256), the operation is similar to convolution. As we are using a 70 by 70-pixel patch, the patchGAN will check on the patch whether it is real or fake. Then it will move by a stride of one and the process will be repeated on the entire image and the results are averaged. Instead of comparing the whole image, the PatchGAN will compare one patch at a time such that it runs in a convolution way across the image and averages all the responses. So we have numerous patches and we classify if either of the patches is real or fake. This is what a simple discriminator does but with PatchGAN as the discriminator, it does this on all the patches.

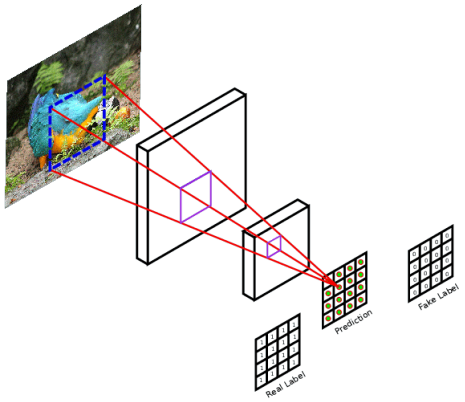

Figure 2.1.3: PatchGAN
[12]

The principle of PatchGAN can be defined as:

- Each block in the discriminator is: Convolution -> Batch normalization -> Leaky ReLU.
- Each 30 x 30 image patch of the output classifies a 70 x 70 portion of the input image.
- The discriminator receives 2 inputs:
  - The input image and the target image, which it should classify as real.
  - The input image and the generated image should classify as fake.

The advantages of PatchGAN architecture for image generation are that they are designed to generate high-resolution images by training the discriminator to classify small patches in the image rather than the entire image. This allows the model to learn fine details and texture at a local level. PatchGANs can be applied to a wide range of image generation tasks and have been used to generate high-resolution images of faces, landscapes, maps, and other image-to-image translation tasks [1].

2.1.4 *Residual Neural Networks*

Residual Neural Networks (ResNets) is a type of deep learning model that was introduced to address the issue of vanishing gradients in deep neural networks. As the number of layers in a deep neural network increases, it becomes more difficult to train the model due to vanishing gradients, where the gradients of the weights become very small and hinder the model's ability to learn. ResNets address this issue by introducing a shortcut connection or skip connection, that bypasses one or more layers and allows the model to directly access lower layers, improving the model's ability to learn efficiently and the flow of gradients through the network [13].

ResNets are known for their ability to improve performance by increasing the depth of the network, which is not typically possible with traditional deep neural networks due to the issue of vanishing gradients. ResNets are able to overcome this limitation and continue to improve performance as the depth of the network increases, making them useful for tasks that require deep networks, such as image classification and object detection.

One of the key components of ResNets is the use of residual blocks, which consist of multiple layers and a skip connection. The skip connection allows the model to bypass one or more layers and directly access lower layers, improving the flow of gradients through the network. The layers in the residual block are usually convolutional layers with non-linearities (such as ReLUs) and batch normalization applied in between. The output of the residual block is then added to the input, resulting in a residual mapping.

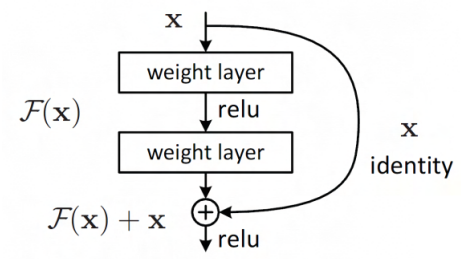

Figure 2.1.4: ResNet
[13]

Many problems can be addressed using ResNets. They are easy to optimize and achieve higher accuracy when the network's depth increases, producing better results than the previous networks. In most cases, ResNets will simply stop improving rather than decrease in performance so it is an excellent network to use [14] when training GANs as we do not converge in most cases. In Summary, Resnet can be described as:

- Very deep neural networks are not practical to implement as they are hard to train due to vanishing gradients.
- The skip-connections help to address the Vanishing Gradient problem. They also make it easy for a ResNet block to learn an identity function.
- There are two main types of ResNets blocks: The identity block and the convolutional block.

In summary, ResNets are a type of deep learning model that utilize skip connections and residual blocks to improve the training of deep neural networks and improve performance on tasks that require very deep networks.

3 METHODOLOGY

This chapter covers four papers that have significantly impacted the field of image-to-image translation, including a discussion of the network architectures used in each. The papers will be organized by paired and unpaired image-to-image translations.

3.1 PAIRED TRANSLATION WITH CONDITIONAL GAN: *pix2pix*

The paper, titled "Image-to-Image Translation with Conditional Adversarial Networks" commonly referred to as Pix2Pix, was authored by Isola et al. [1] and presented in 2016 at the Conference on Computer Vision and Pattern Recognition (CVPR). It utilizes a variant of Generative Adversarial Networks (GANs) and is trained on pairs of images, with the input being an image from one domain and the output is the corresponding image in another domain. The model is designed to generate outputs that closely resemble ground truth images.

The generator in this model is trained using a CGAN architecture, as shown in figure (3.1.1).

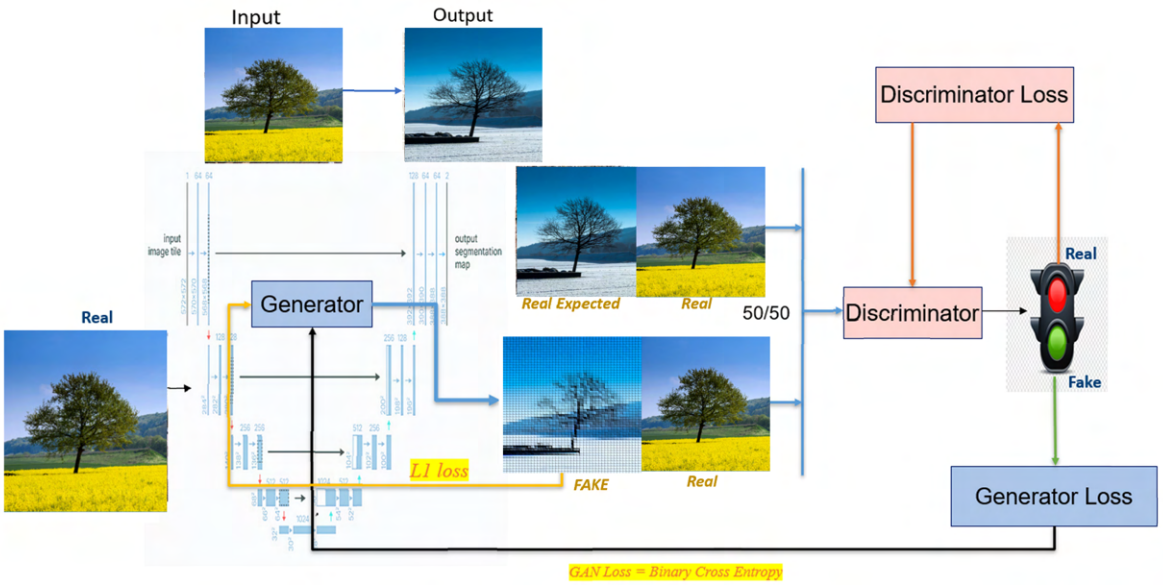

Figure 3.1.1: cGAN Architecture Used to Train the Network

We have a UNET-based generator (2.1.2), which takes a real image as input and produces a fake image as output. In our case, the generator is fed a real summer image and is expected to output a fake winter image. The discriminator takes two inputs: the real summer image and the fake winter image generated by the generator. The discriminator task is to distinguish between these two images. To do this, we provide the discriminator with pairs of real images and their corresponding fake images. As shown in figure (3.1.1), these image pairs are concatenated and fed

into the discriminator in a 50:50 ratio. We label the real images as "ONE" and the fake images as "ZERO". We use a PatchGAN architecture (2.1.3) for the discriminator.

The generator in our model is trained using a combination of two loss functions: the binary cross-entropy loss and the mean absolute error or L1 loss. The binary cross-entropy loss is used to update the generator's weights and is calculated based on the fake image produced by the generator. The L1 loss, which is the mean absolute error between the fake image and the corresponding real image, is also included in the generator's loss calculation. The L1 loss is given a weight of 100 times the GAN loss (referred to as the "Lambda" loss or λ) when combined with the GAN loss and used to update the generator's weights.

The authors of Pix2Pix suggested not to use batch normalization on the first "Convolution 64" layer. In our implementation, we have followed this recommendation and set the batch size to 1. We used the Adam optimizer to train the network. To ensure the reproducibility of our research, we have used the code from the official pix2pix repository, modifying it to meet our specific needs. We have trained 12 different models, each representing a different season using this architecture.

To evaluate the performance of the trained models, we tested them on a set of test images. An example of the results produced by one of the trained models is shown in figure (3.1.2). These results will be used later in our research to evaluate the performance of the trained models.

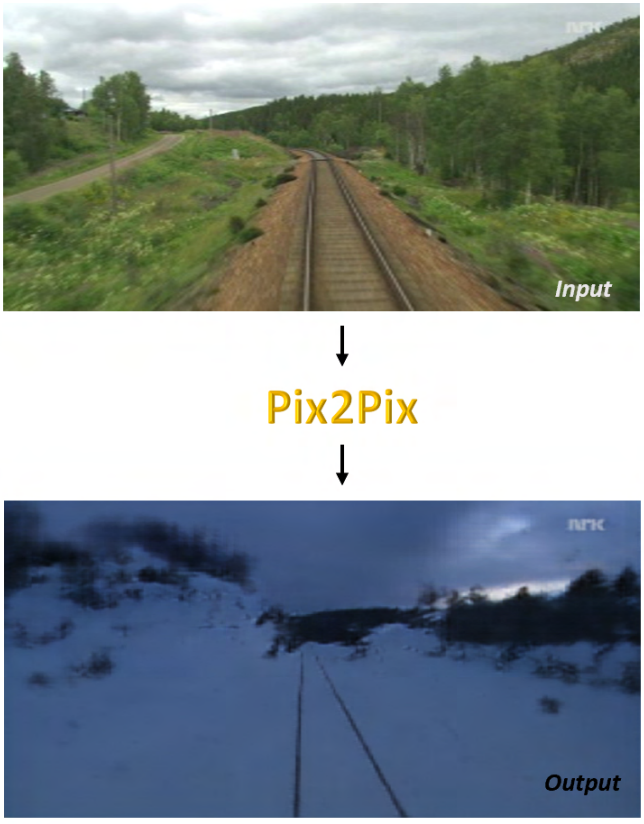

Figure 3.1.2: Trained Pix2Pix Example

3.2 HD IMAGE SYNTHESIS AND SEMANTIC MANIPULATION WITH CGANS: *pix2pixhd*

The paper "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs" (Pix2PixHD) authored by Wang et al. [2] was presented in 2017 at the Conference on Computer Vision and Pattern Recognition (CVPR). Pix2PixHD is an extension of the Pix2Pix architecture, a GAN-based technique for generating high-resolution images from semantic label maps. The Pix2Pix model, originally introduced by Isola et al. [1] in 2016, is a GAN designed for image-to-image translation tasks such as converting a grayscale image to a color image or a semantic label map to a photograph. Pix2PixHD improves upon the original Pix2Pix architecture by incorporating a multi-scale generator and discriminator, allowing for the synthesis of higher-resolution images and the handling of more complex image translation tasks.

One limitation of the original pix2pix model is that it struggles to generate high-resolution images with realistic textures and details due to a lack of data at the lower layers of the encoder-decoder architecture. This leads to difficulties in properly decoding the features, resulting in a failure to upsample the encoded features to a high-resolution image.

The Pix2PixHD network addresses this issue by introducing a new generator architecture, a multi-scale discriminator architecture, and an additional loss term. These improvements allow Pix2PixHD to support image-to-image translation at a resolution of 2048x1024 with higher quality than the original pix2pix model. The authors of Pix2PixHD also proposed a two-stage approach for training the networks, as illustrated in figure (3.2.1).

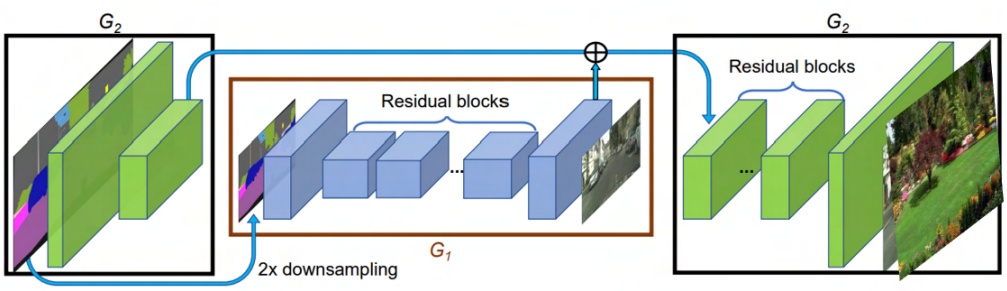

Figure 3.2.1: Pix2PixHD Generator Network

The Pix2PixHD training process involves using two generator networks: a global generator G1 and a local enhancer network G2. The global generator generates a low-resolution image of 1024x512, which is then enlarged by the local enhancer network to a resolution of 2048x1024. The local enhancer network is composed of three components: a convolutional front-end, a series of residual blocks, and a transposed convolutional back-end. The feature map is also modified by introducing global information through element-wise sum before it is passed to the residual blocks. During training, the global generator is trained first, followed by the local enhancer network in order of their resolutions. Finally, all of the networks are jointly fine-tuned together.

To synthesize high-resolution images, the discriminator in the Pix2PixHD model needs a large receptive field. This can be achieved by either increasing the depth of the network or using larger convolutional kernels, but these approaches may result in overfitting due to increased network capacity. To avoid this, the authors of Pix2PixHD propose using three discriminators with identical network structures that operate at different image scales. These discriminators, referred to as D1, D2, and D3, are trained to distinguish real and synthesized images at the corresponding scales, creating an image pyramid. By using multiple discriminators that view the image at different

levels of the receptive field, the generator is able to learn about both the large structures and fine details in the image.

The Pix2PixHD paper introduces an improved adversarial loss function that combines the standard GAN loss, consisting of a discriminator-induced loss, with an MAE loss calculated with respect to the target image. However, this function has a limitation in that it cannot consider the intermediate representations of the generator network. To address this limitation, the paper proposes an updated loss function that incorporates intermediate representations from both the discriminator and generator networks. This allows the network to focus on finer details during downsampling and improves the quality of the synthesized images. The updated loss function is achieved by extracting the intermediate representations of both the discriminator and generator networks.

Pix2PixHD enables manipulation of the attributes of the generated images by using an additional encoder to extract features from real images and applying instance-wise average pooling. This process involves taking the average of all pixels in an object and broadcasting it back to the pixels on the feature map. These extracted features are then used as input for the generator. Additionally, K-means clustering is applied to the features of all objects in each class, allowing users to select different textures or colors for the objects during inference.

The modified network can generate realistic, high-resolution images that contain proper structures and fine details that are not present in simple Pix2Pix models. As an example, figure (3.2.2) shows the result of converting a summer image to a winter image using Pix2PixHD.

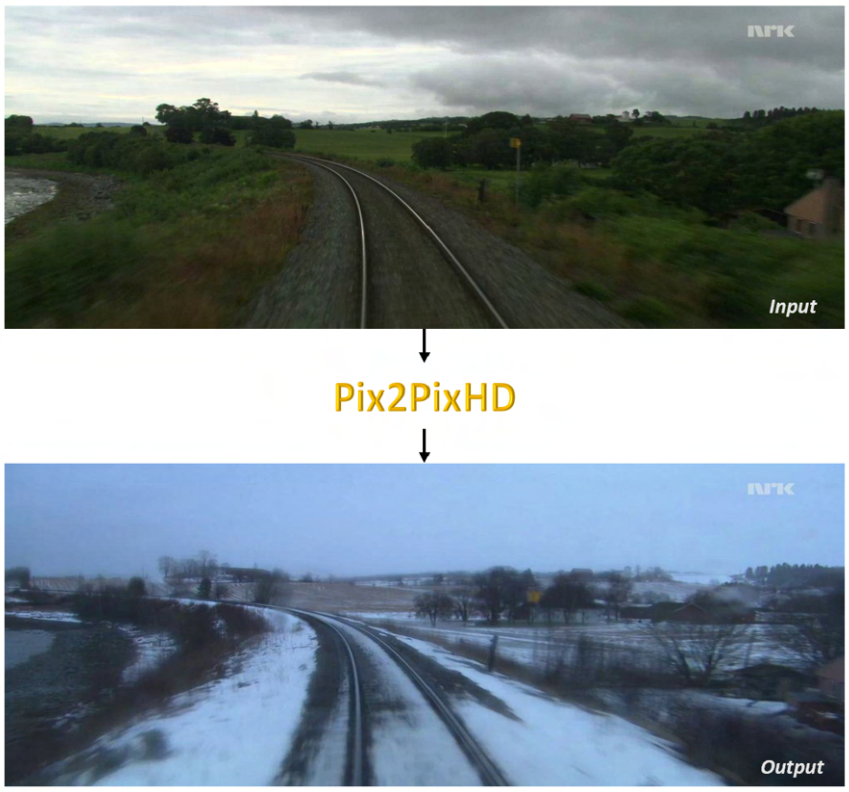

Figure 3.2.2: Trained Pix2PixHD Example

3.3 UNPAIRED TRANSLATION USING CYCLE-CONSISTENT GAN: *cyclegan*

The paper "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" (CycleGAN) authored by Zhu et al. [3] was presented at the 2017 IEEE International Conference on Computer Vision (ICCV).

CycleGAN is a technique for translating images from one domain to another without the need for large amounts of labeled data. It does this using two GANs: a generator that translates images from one domain to the other, and a discriminator that tries to determine whether an image is real or translated. The model is trained using a cycle consistency loss, which ensures that the generated images maintain the content of the input image while being translated to the target domain.

The authors of the CycleGAN paper explore the idea of capturing the unique characteristics of one set of images (e.g. "Autumn" images) and finding similarities in the characteristics of another set of images (e.g. "Summer" images) in order to understand how the images from the first set can be translated into the second set. By examining the common characteristics of the two image sets, the model can learn to translate images between the two domains and produce high-quality images that capture the essential features of the target domain.

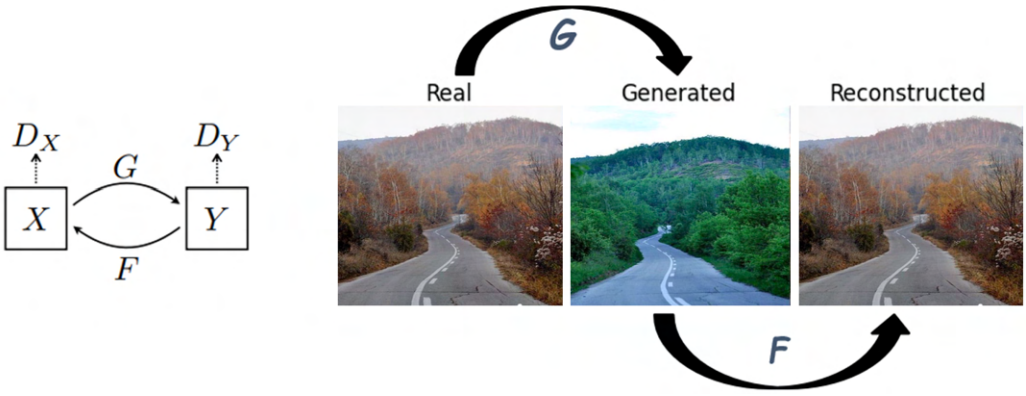

Figure 3.3.1: cGan Cycle Mapping

The concept of "cycle consistency" refers to the idea that it should be possible to translate an image from one domain to another and then back to its original domain, resulting in the original image. For example, if starting with an "Autumn" image and using a generator to translate it to a "Summer" image, it should be possible to use a second generator to translate the "Summer" image back to the original "Autumn" image. This process is illustrated in figure (3.3.1), The loss associated with this process is known as the "cycle consistency loss" and is calculated by comparing the original "Summer" image to the "Summer" image generated by the first generator, and the original "Autumn" image to the "Autumn" image generated by the second generator. The goal of the CycleGAN model is to learn a mapping function that can translate images between the "Autumn" and "Summer" domains while preserving the content and structure of the original images.

The CycleGAN model employs the use of "Instance Normalization" for the discriminators and a patchGAN (2.1.3) with a 70x70 size. Instance normalization normalizes the values of each output feature map independently, as opposed to normalizing across all feature maps in a batch as done in batch normalization. To prevent oscillation in the model, the discriminator is updated using a history of 50 generated images instead of just a single image. The Adam solver and a batch size

of 1 are also used in this model. Additionally, an "Identity Mapping" loss term is introduced to encourage the generator to preserve the color of the input image. This means that if an "Autumn" image is input to the generator, which has been trained to generate "Autumn" images, it should output the same "Autumn" image without making any changes.

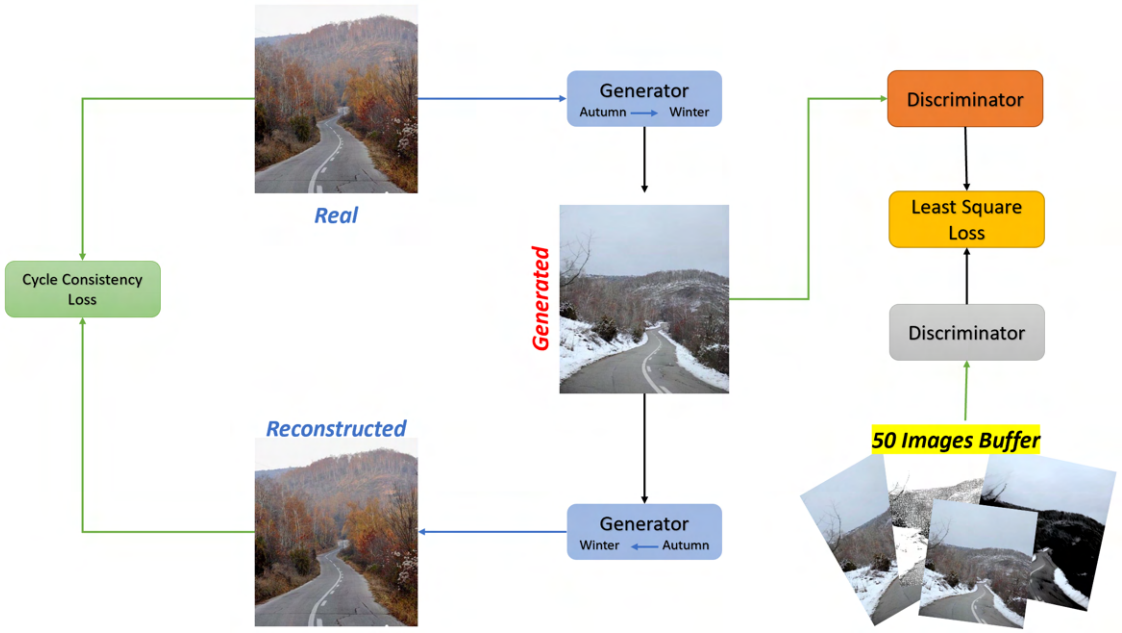

Figure 3.3.2: Training of Cycle GAN

The Generator in CycleGAN is a ResNet (as referenced in (2.1.4). The model has two generators, as shown in figure (3.3.2). The first generator takes an "Autumn" image as input and produces a "Winter" image, while the second generator takes the generated "Winter" image as input and produces an "Autumn" image. The goal of the model is to learn the mapping between "Autumn" and "Winter" images, allowing the generators to translate images between the two domains.

The discriminators are trained to classify images as real or fake. The first discriminator determines whether the generated "Winter" image is real or fake, while the second discriminator determines whether the generated "Autumn" image is real or fake. These errors are used to calculate the loss for the model, as described in the next section.

Error Calculations

In cycle GAN the generator is trained with the combined model to minimize the Four different losses. The losses are computed as:

- **Identity Loss** *(L1)*: Output source image as is it without translation. It is given by:
  $\mathrm{IdentityLoss} = \frac{1}{n}\sum_{i=1}^{n}(y_{\text{true}} - y_{\text{predicted}})^2$
- **Adversarial Loss** *(L2)*: Minimize loss predicted by discriminator for generated images marked as 'real'. It is given by:
  $\mathrm{AdversarialLoss} = (y_{\text{true}} - y_{\text{predicted}})^2$

- **Cycle Loss Forward Loss** *(L1)*:Regeneration of a source image. (Autumn image to Winter image 3.3.3). It is given by:

  $\text{CycleLoss}_{\text{forward}} = \frac{1}{n}\sum_{i=1}^{n}(y_{\text{true}} - y_{\text{predicted}})^2$

- **Cycle Loss Backwards** *(L1)*:Regeneration of a source image. (Winter image to Autumn image 3.3.3) It is given by:

  $\text{CycleLoss}_{\text{backward}} = \frac{1}{n}\sum_{i=1}^{n}(y_{\text{true}} - y_{\text{predicted}})^2$

In the CycleGAN model, there are four losses used during training: adversarial loss, identity loss, forward cycle loss, and backward cycle loss. These losses are weighted as $(1, 5, 10, 10)$, respectively, according to the authors of the CycleGAN paper [3]. An example of the results of a trained CycleGAN model for both the forward and backward generators can be seen in figure (3.3.3).

Figure 3.3.3: Trained CycleGAN model forward and backward example

3.4 MULTI-DOMAIN TRANSLATION WITH UNIFIED GAN: *stargan*

The paper "Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation" (StarGAN) authored by Choi et al. [8] was presented in 2018 at the Conference on Computer Vision and Pattern Recognition (CVPR).

StarGAN uses a conditional GAN architecture, where the generator receives an additional input representing the desired domain for the output image. The model is trained using a single dataset containing images from multiple domains, and the generator learns to translate images from one domain to another by predicting the domain label for each image. The output of StarGAN [8] is somewhat similar to CycleGAN (3.3). The aim is to translate unpaired image data between different domains. We use one generator to translate between multiple domains instead of using the 'n' number of generators.

One of the key contributions of the StarGAN paper is the introduction of the concept of domain transferability. Domain transferability refers to the ability of the model to translate images between domains that it has not seen during training. StarGAN demonstrates that the model has strong domain transferability, meaning that it can translate images between domains that it has not been explicitly trained on.

The significance of StarGAN can be understood by considering the task of translating an image into one of four seasons (Summer, Winter, Spring, and Autumn) using CycleGAN. To translate between all four seasons using CycleGAN, we would need two generators and one discriminator for each translation. This means that we would need to have a total of 24 different models to be able to convert from any season to any other season, as shown in figure (3.4.1).

Figure 3.4.1: Traditional Approach for Multi-domain Translations using CycleGAN

To avoid the need for training multiple generators, we can use StarGAN. With StarGAN, we can use a single generator and discriminator to convert between any of the four seasons, rather than the 24 generators and 12 discriminators required by CycleGAN.

The architecture of StarGAN is similar to that of CycleGAN. The generator is usually an autoencoder, and the discriminator still outputs a score indicating the authenticity of the generated images. The model takes in and produces images with changes in the form of inputs and outputs.

In StarGAN, the generator receives not only an image but also a label indicating the target class to which it should attempt to convert the image. This label is represented using a one-hot vector, where the value is zero for all classes except the target class, which is represented with a value of one. This is illustrated in figure (3.4.2).

Figure 3.4.2: One Hot Vector

According to the authors of "StarGAN", it is recommended to replicate the label spatially and append it as an additional channel to the input image, in addition to the standard red, blue, and green channels. The discriminator in StarGAN is designed to output a score indicating the authenticity of the generated image, as well as a classification in the same format as the label. The learning objectives for the generator in this case include making the generated image look as realistic as possible, which is evaluated using the discriminator's output on whether the image is real or fake. The generator performs well when it can fool the discriminator into believing that the generated image is real.

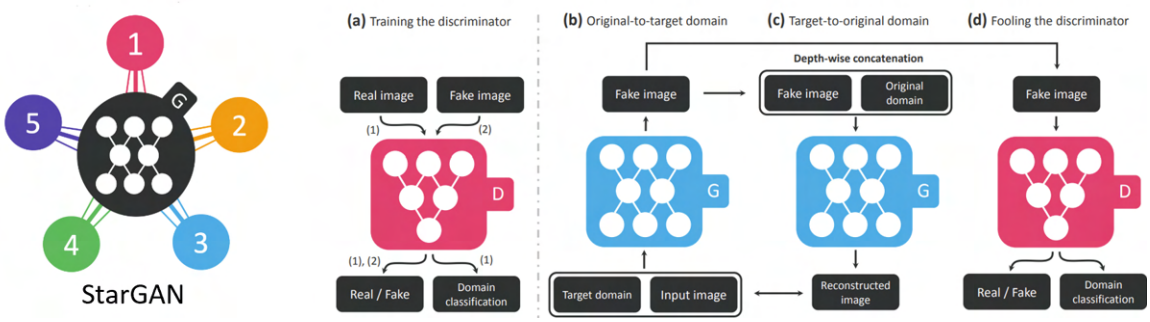

Figure 3.4.3: StarGAN Architecture
[8]

The second goal of the generator in StarGAN is to make the generated image as similar as possible to the target domain, as indicated by the target label. This is evaluated by comparing the discriminator's classifier output to the original target label. The generator performs well when it can fool the discriminator's classifier into believing that the generated image belongs to the target class. Additionally, the generator in StarGAN uses a cycle consistency loss, similar to the one used in the CycleGAN architecture. This loss involves passing the generated image back through the

same generator, but with the source label as input, in order to reconstruct the original image. The generator is then penalized for any differences between the original and reconstructed images.

The discriminator in StarGAN has two main objectives. The first is to correctly distinguish real images from those generated by the generator. The second objective is to accurately classify an image based on its class, which is evaluated by comparing the discriminator's classification output to the real label for a real image. In our implementation, we followed the instructions provided in the "StarGAN" paper [8] for using the model to translate an input image into one of the four seasons. An example of the generated output from StarGAN is shown in the image in figure (3.4.4).

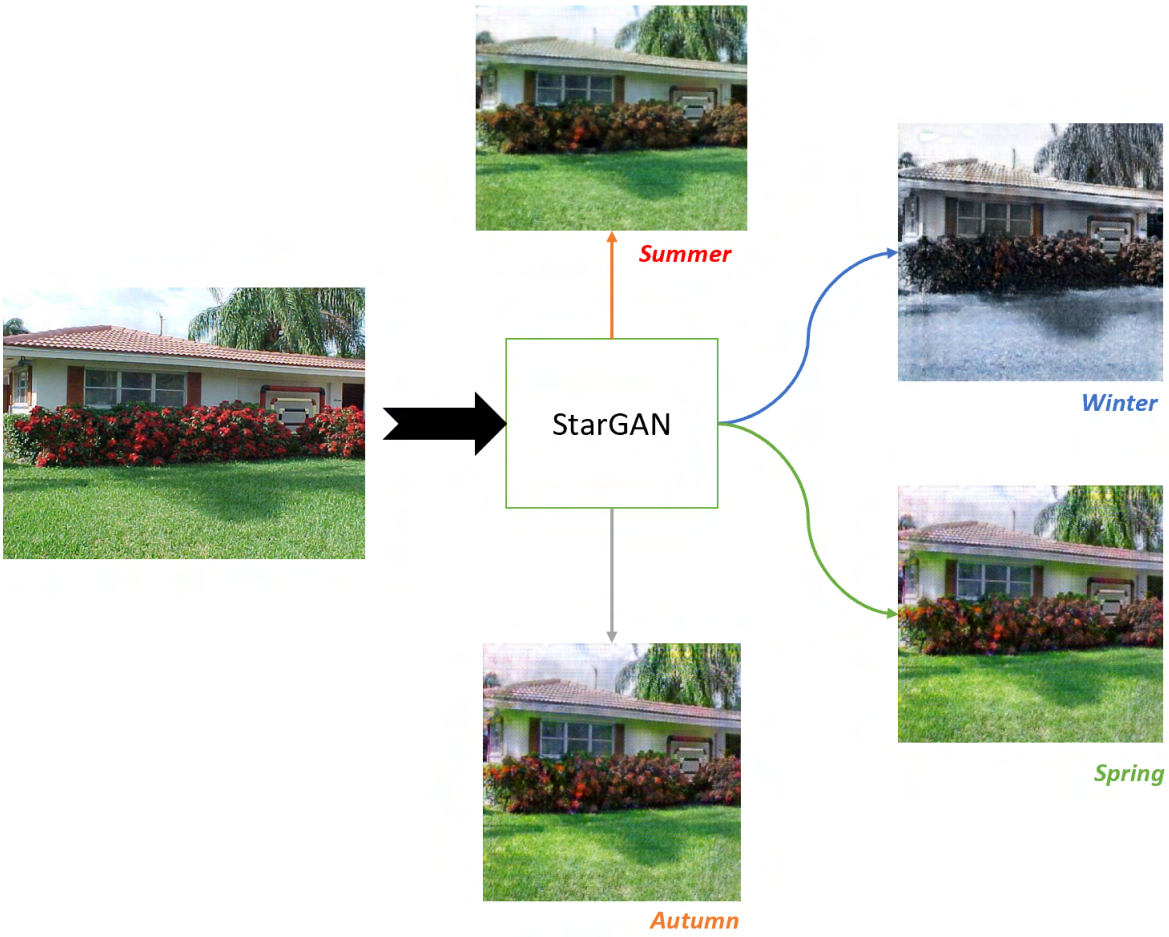

Figure 3.4.4: StarGAN Generated Example

4 EXPERIMENT AND EVALUATION

In this chapter, we will train the four networks Pix2Pix (3.1), Pix2PixHD (3.2), CycleGAN (3.3), and StarGAN (3.4) on the custom made dataset called Seasons. Our goal is to generate images in different domains using these networks and compare their performance. After training the models, we will use a test dataset to evaluate their results. First, we will describe how the dataset is created and split. Then, we will present the results of training the Pix2Pix, Pix2PixHD, CycleGAN, and StarGAN models. Finally, we will discuss the creation of a public survey and the results obtained from it.

4.1 CREATING THE DATASET

In order to train the Pix2Pix (3.1) and Pix2PixHD (3.2) networks, we used a paired dataset called 'Single-View Place Recognition under Seasonal Changes' from the paper [15]. This dataset is composed of images of the same location taken in all four seasons and are aligned with each other, as depicted in figure (4.1.1). For training, we utilized 3000 images and set aside an additional 300 images for testing.

Figure 4.1.1: Paired Dataset for Pix2Pix and Pix2PixHD

For the unpaired models CycleGAN (3.3) and StarGAN (4.2.4), we have used a different dataset. The images were collected from Google photos and other freely available datasets on the internet. We have used 3000 images for each season and a separate pool of 300 images for testing. Example images from the seasons can be seen in figure (4.1.2).

Figure 4.1.2: Unpaired Dataset for CycleGAN and StarGAN

4.2 TRAINING AND TESTING MODELS

We want to compare the performance of different models in translating images from one modality to another. Our comparison is between Pix2Pix (3.1) and Pix2PixHD (3.2), and between CycleGAN (3.3) and StarGAN (3.4). We have trained each model and generated output in each modality. We have also carried out extensive testing on the models to obtain the best results, and in some cases, we have retrained the models with different hyperparameters. We will discuss this further when discussing each specific model.

4.2.1 *Pix2Pix Model*

As previously discussed, we used the code provided by the authors of the "Pix2Pix" paper [1] to train our network. Our goal was to convert images from one season to the other three seasons. After training, we tested the trained network on unseen images, and the results are shown in figure (4.2.1).

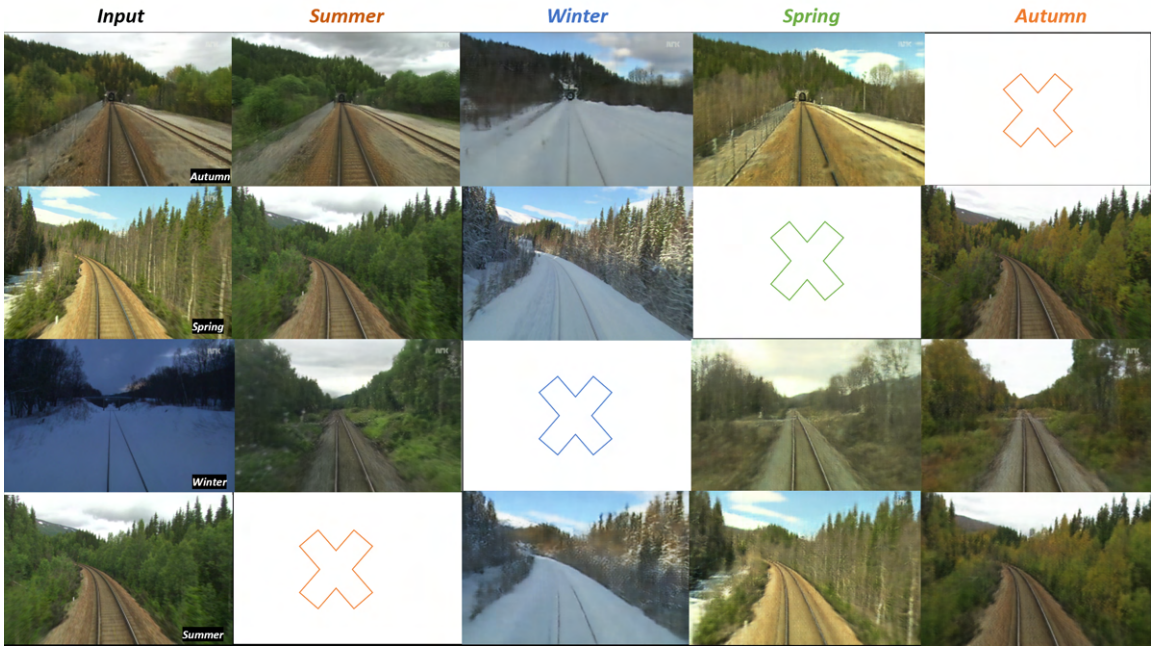

Figure 4.2.1: Testing the Pix2Pix models

Pix2Pix performed well in translating images from one modality to another using the paired dataset. The translations are accurate, although the images are somewhat blurry and lack clarity. The model also struggles to accurately represent individual features, such as trees. Despite these limitations, the model performs well overall. When we tested the model using unpaired data, it performed poorly and was unable to generate an accurate image, as shown in figure (4.2.2).

Figure 4.2.2: Testing the Pix2Pix models on Unpaired Dataset

4.2.2 *Pix2PixHD Model*

Similarly to Pix2Pix, we used the code provided by the authors of the "Pix2PixHD" paper [2] to train the Pix2PixHD model. Our goal was to convert images from one weather condition to the other three. After training, we tested the trained network on unseen images, and the results are shown in figure (4.2.3).

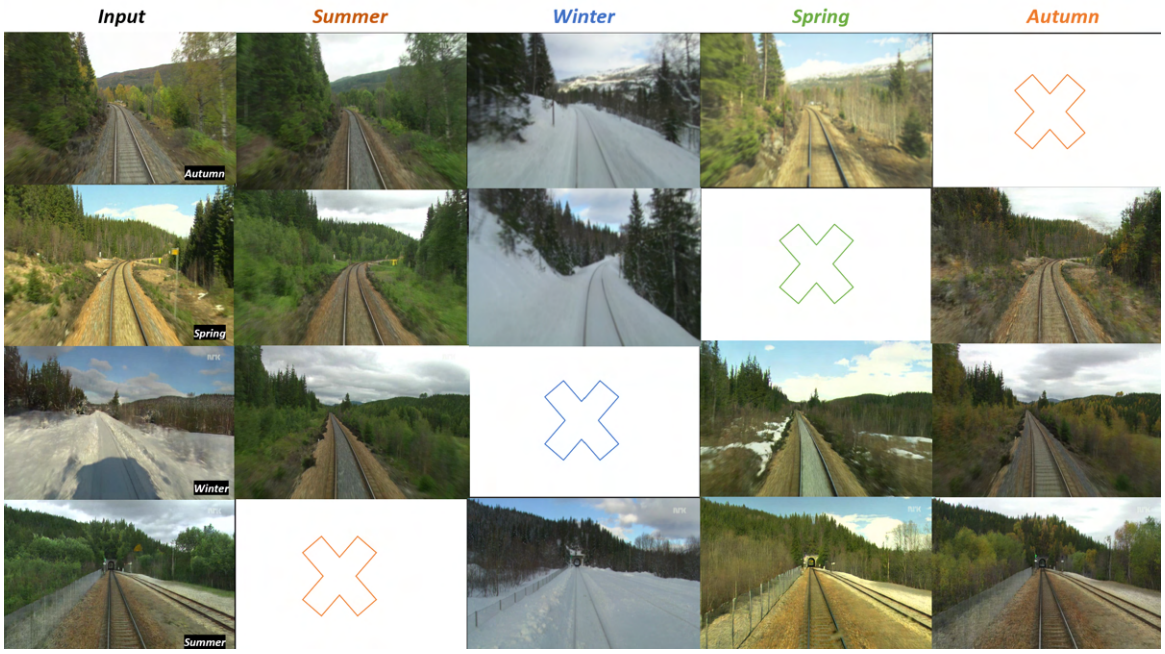

Figure 4.2.3: Testing the Pix2PixHD models

Pix2PixHD performed excellently in translating images from one modality to another using the paired dataset. The translations are crisp and the images are very sharp. The model performed well, producing accurate representations of individual features such as trees. When we tested the model using unpaired data, it performed poorly but still produced recognizable translations. The results are much better than those of Pix2Pix, as we can see that the model attempted to translate the image to the desired modality, as shown in figure (4.2.4).

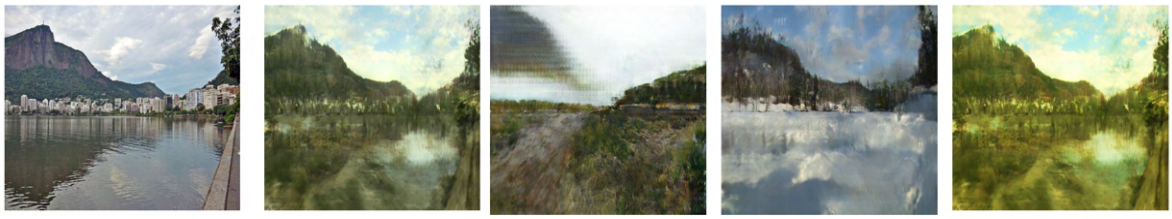

Figure 4.2.4: Testing the Pix2PixHD models on Unpaired Dataset

4.2.3 *CycleGAN Model*

We trained the CycleGAN model on an unpaired dataset as illustrated in figure (4.1.2). We used the code provided by the original paper to train the model with the aim of converting images from one season to any of the other three seasons. Upon testing the trained network on previously unseen images, we obtained the results shown in figure (4.2.5).

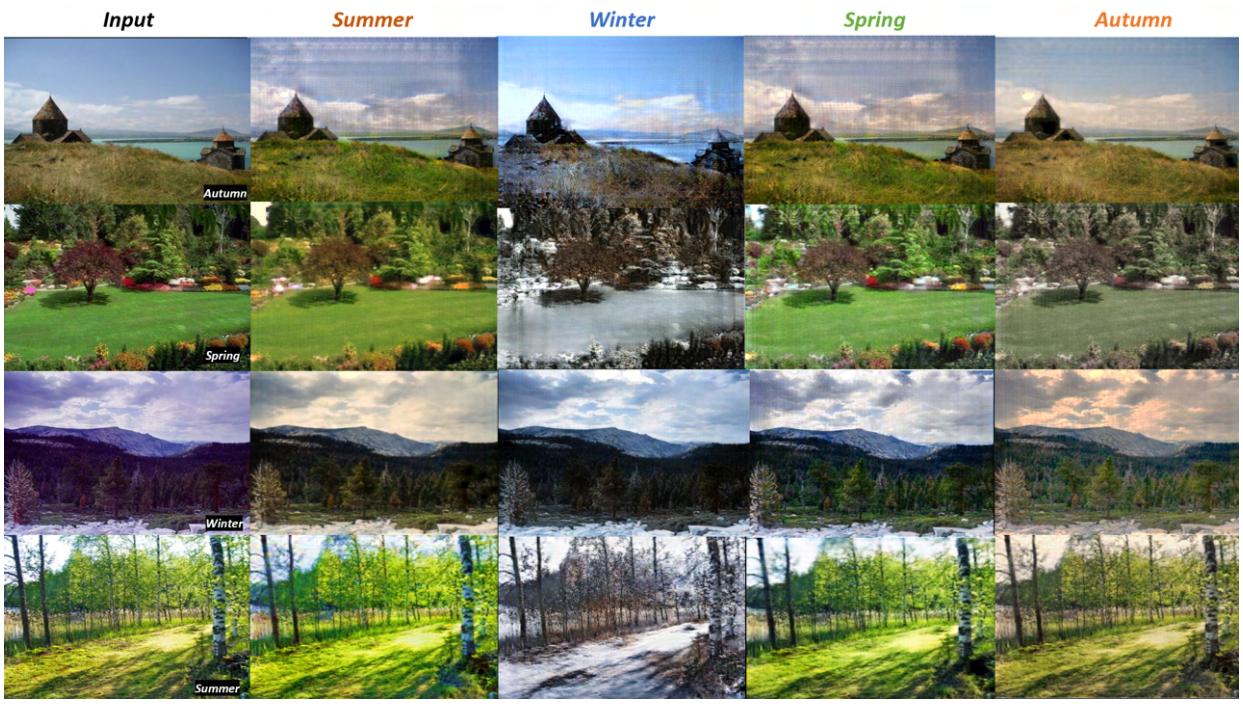

Figure 4.2.5: Testing the CycleGAN models

The CycleGAN model performed well in translating images from one modality to another using the unpaired dataset. The translations were clear and the image quality was good. There were some instances of noise in the generated images, particularly in the summer images, but overall the model performed well and produced satisfactory results.

4.2.4 *StarGAN*

As previously mentioned in section (3.4), the StarGAN model is able to generate multiple modalities using a single generator. During our training process, we found that using a batch size of 1 resulted in better performance than the recommended batch size of 16 (as suggested in the original code for StarGAN [16]). We also tried using a batch size of 8 (as shown in figure 4.2.8), which produced better results than the batch size of 16 (shown in figure 4.2.7), but the output was still not satisfactory. In our work, we employed the most favorable output of StarGAN, as depicted in figure (4.2.6), which was obtained using a batch size of 1.

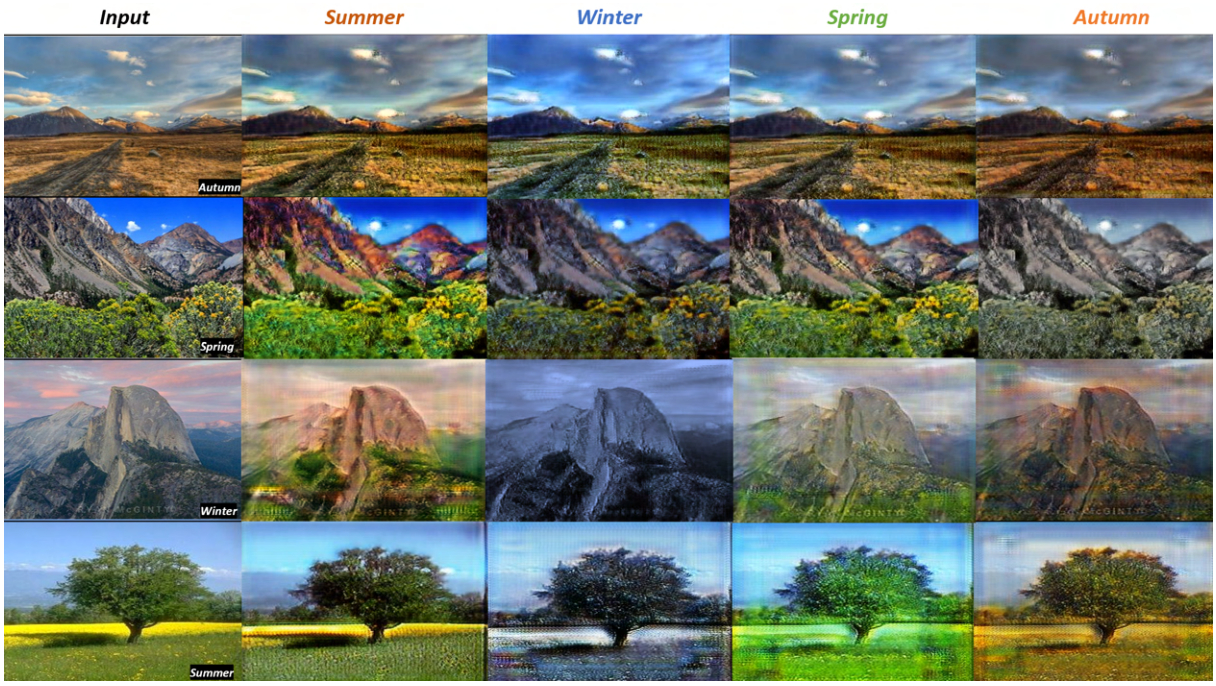

Figure 4.2.6: Training the StarGAN model Batchsize 1

The original configuration of the StarGAN model was disappointing because it produced poor translations of images, as demonstrated in the training snapshot shown in figure (4.2.7). Despite increasing the number of iterations, we were unable to improve the quality of the generated images. Therefore, we decided to try modifying the batch size in an attempt to improve the performance of the network. As a result, we observed an improvement in the quality of the generated images.

Figure 4.2.7: Training Snapshot of the Original StarGAN model (batchsize16)

We experimented with changing the batch size of the model in an attempt to improve its performance. We set the batch size to 8 and retrained the model. The resulting performance is shown in figure (4.2.8).

Figure 4.2.8: Testing the StarGAN model with Batchsize 8

In summary, the performance of the StarGAN model was disappointing as it did not produce clear, high-resolution images. Instead, it only altered the color of the input image without effectively learning to generate sharp output images.

4.2.5 *Testing Networks Using Same Inputs*

We will be using a single input image to compare the output generated by both Pix2Pix and Pix2PixHD across all seasons, as shown in figure (4.2.9). We will also compare the output of the StarGAN model with that of the CycleGAN model using the same input image. The results generated by these networks are depicted in figure (4.2.10).

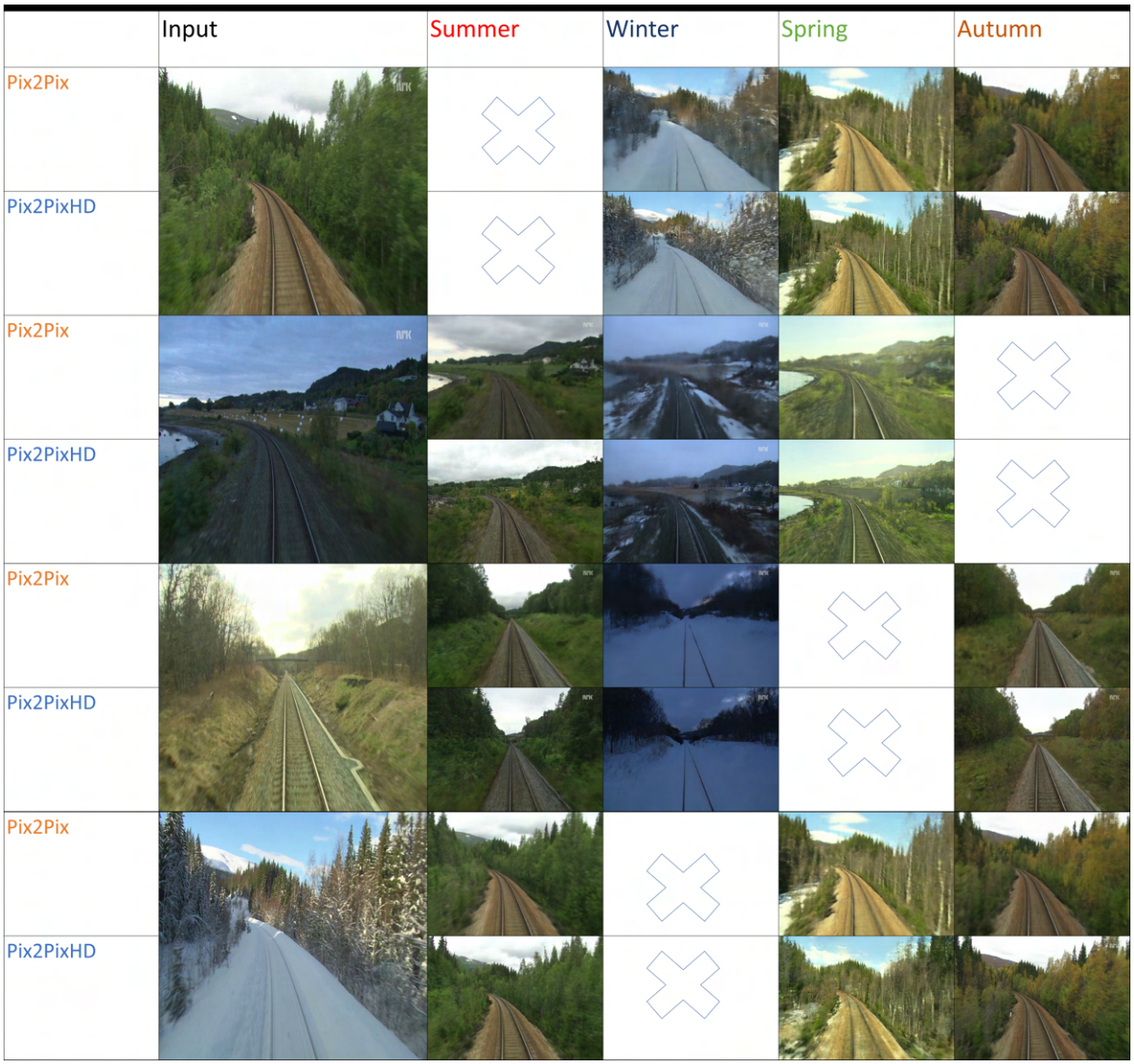

Figure 4.2.9: Comparing the output of Pix2Pix with Pix2PixHD

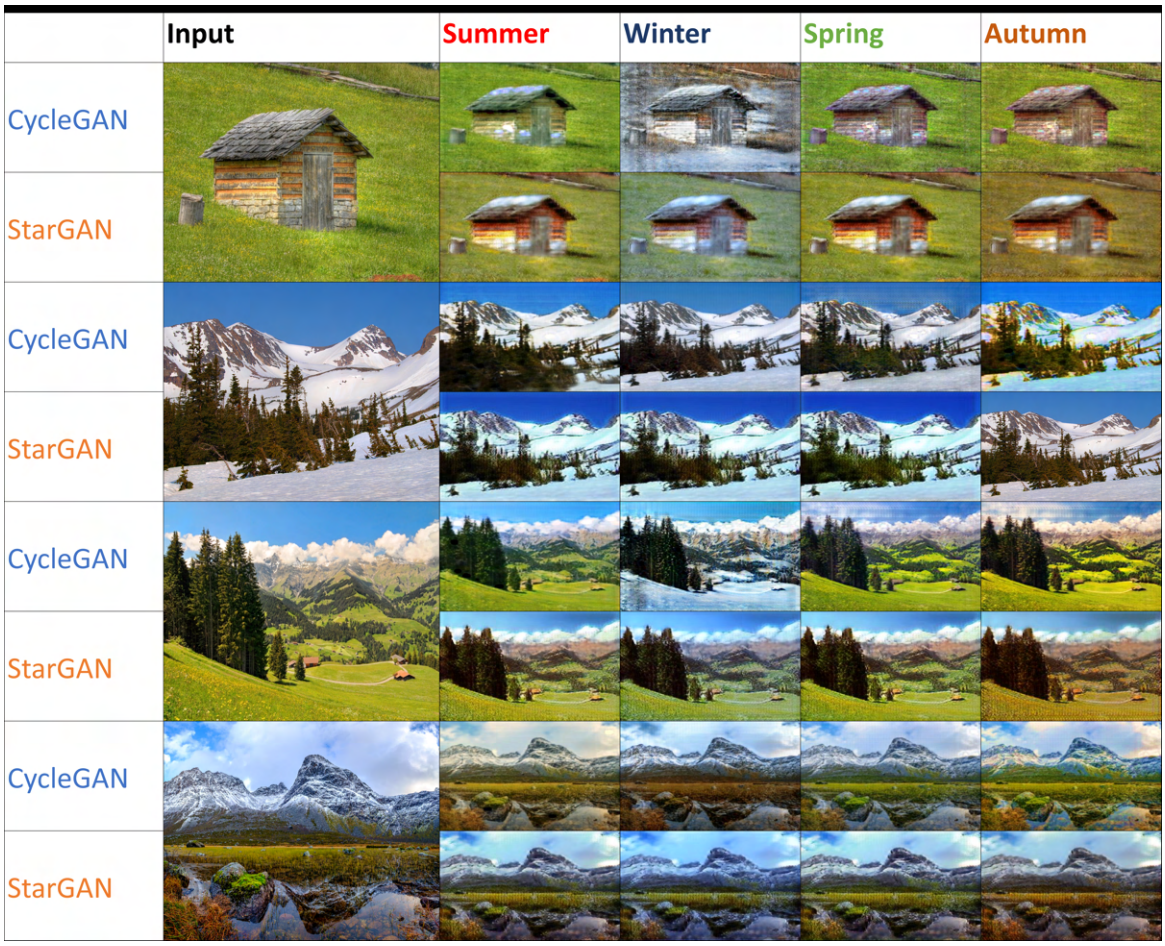

Figure 4.2.10: Comparing the output of CycleGAN with StarGAN

4.3 COMPARING THE NETWORKS

We conducted a comparison of the networks to determine which one is most effective for generating specific seasons. Since we do not have ground truth images to compare with the generated images, particularly for the unpaired dataset, we cannot use mathematical models to evaluate the results. Therefore, we conducted a survey [17] using a paid subscription called SurveyHero in order to compare the images.

4.3.1 *About the Survey*

The survey was conducted using SurveyHero [17], an online platform for creating and sharing surveys. It included 48 images, with 12 generated photos from each of the following models: Pix2Pix, Pix2PixHD, CycleGAN, and StarGAN. Of these 12 images per model, 3 images were from Summer, 3 from Winter, 3 from Spring, and 3 from Autumn. The survey contained a total of 96 questions, with two questions asked for each image. Participants were asked to **"Rate the quality of the image"** on a scale of 1 to 5 stars, with 1 star being the lowest and 5 stars being the highest.

They were also asked to answer **"How well does the Picture represent the Season"** (summer, winter, spring, or autumn) on a scale of very poor, poor, fair, good, or very good. An example of one of the questions from the survey is shown in figure (4.3.1).

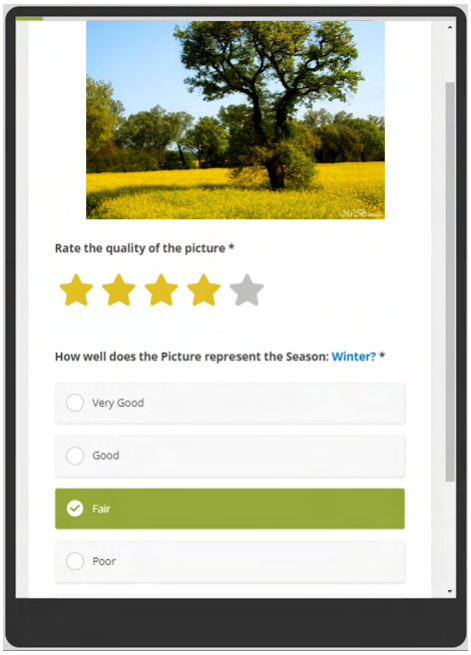

Figure 4.3.1: Survey Example

The survey was distributed through various platforms and received a total of 291 views by 20/12/2022. Of these views, 146 people participated in the survey, with 105 completions and 41 left it incomplete as shown in figure (4.3.2). The participation rate was 50% and the completion rate was approximately 72%. For the purposes of analysis, only the completed surveys were considered, and the incomplete surveys were ignored. The results of the completed surveys were imported into a CSV file and analyzed using Microsoft Excel, with the data being transformed into graphs for easier interpretation.

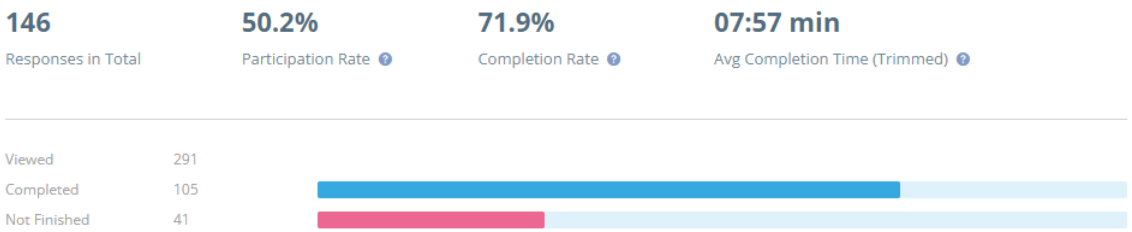

Figure 4.3.2: Survey Stats

4.3.2 *Interpreting the results from survey*

As previously discussed in the survey section (4.3.1), users were shown three images per season, totaling 12 images per model, and asked to rate their quality and desired season attributes. To reduce bias, we took the average of the ratings for each season and model. We first analyzed the results for the paired models, Pix2Pix and Pix2PixHD, and then examined the results for the unpaired models, CycleGAN and StarGAN. Finally, we compared the overall ratings for all models to get a general sense of how the participants rated them. To better understand the results, we created bar graphs with different colors representing each season: red for summer, blue for winter, green for spring, and orange for autumn. In the next sections, we will see two types of bar graphs: the graphs with a frame represent the image quality score and the graphs without a frame represents the desired attribute score.

Paired Models Pix2Pix and Pix2PixHD

The bar graph in figure (4.3.3) shows the overall average rating of image quality on a scale of 1 to 5, with 1 being the lowest rating and 5 being the highest, for the paired models, based on the responses from the participants in the survey.

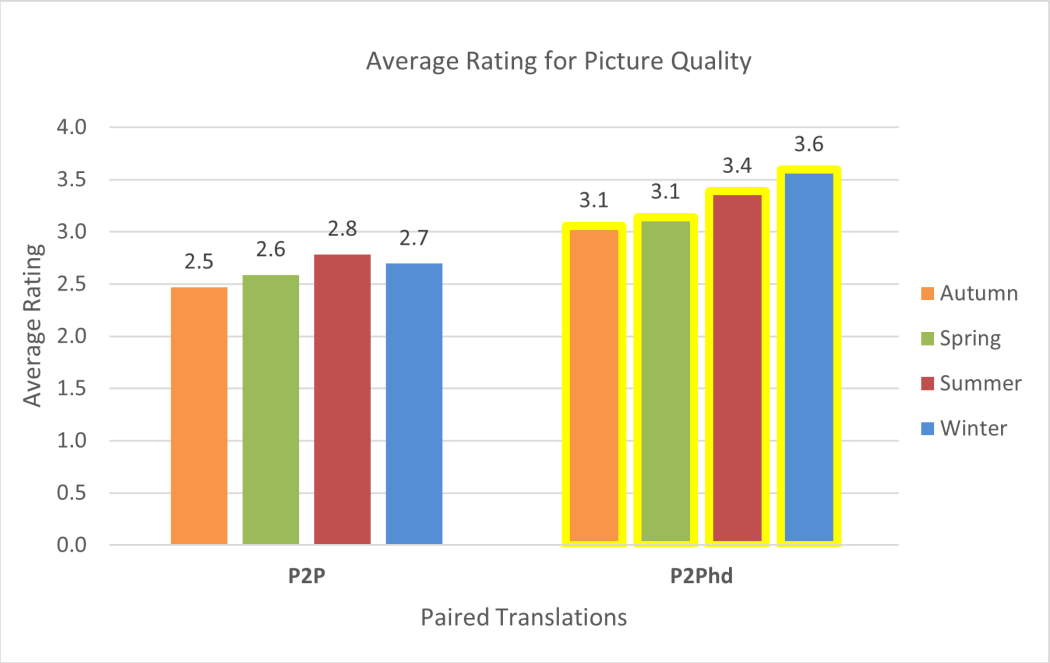

Figure 4.3.3: Survey results: Paired Quality Score

Based on the responses, Pix2PixHD (figure 4.3.3) demonstrated significantly better performance than the standard Pix2Pix model across all seasons. The best image quality was for the winter season images generated by Pix2PixHD, with an average of 3.6 stars out of 5. Pix2PixHD outperformed Pix2Pix in terms of image quality for all seasons, with all averages ratings higher than 3, while the pix2pix averages for all seasons were below 3, with the highest rating achieved by Pix2Pix of 2.8 for the summer images. Overall, Pix2PixHD had a higher image quality than the standard Pix2Pix network.

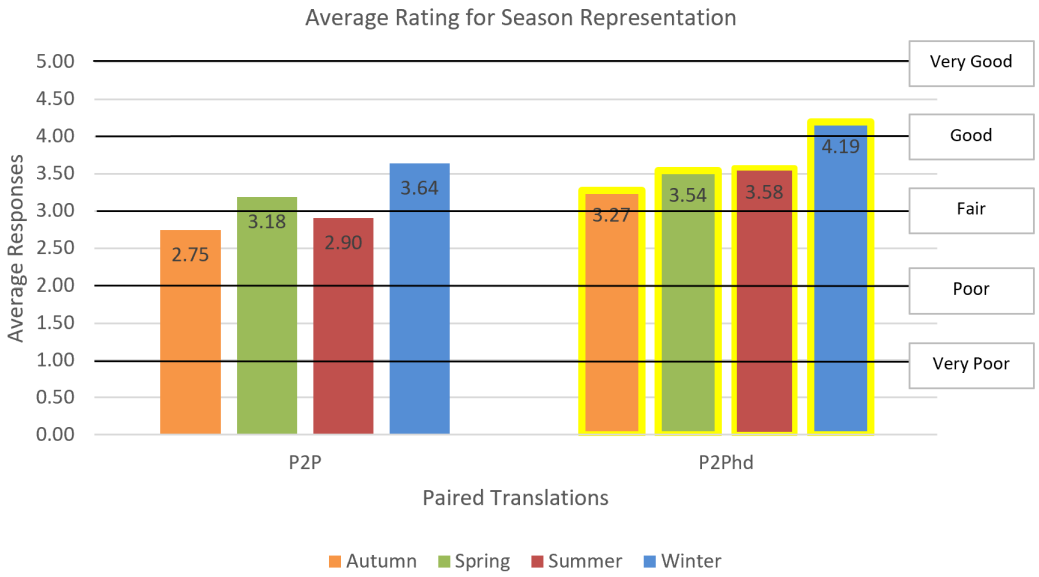

Figure 4.3.4: Survey results: Paired Desired Attribute Score

The second bar graph (figure 4.3.4) shows the ratings given by participants for how well the generated images represented the desired season attribute, for the paired models. Participants were asked to choose from five options: very good, good, fair, poor, and very poor. These ratings were scaled from 1 to 5 for computation purposes, with 5 being very good and 1 being very poor.

As shown in figure (4.3.4), the Pix2PixHD model performed significantly better than the standard Pix2Pix model. The images generated by both models were generally rated as fair to good, the only exception being the autumn and summer images generated by Pix2Pix which were rated as poor. As can be seen, the highest values in both models are marked with a yellow border. We can see that participants consistently rated the Pix2PixHD model as producing better representations of the seasons compared to the simple Pix2Pix model. The winter images generated by Pix2PixHD received the highest score.

Overall, it can be concluded that Pix2PixHD is a superior network to standard Pix2Pix in terms of image quality and the ability to generate images with desired attributes.

Unpaired Models CycleGAN and StarGAN

The bar graph in figure (4.3.5) shows the ratings of image quality on a scale of 1 to 5, with 1 being the lowest rating and 5 being the highest, for the unpaired models, based on the responses from the participants in the survey.

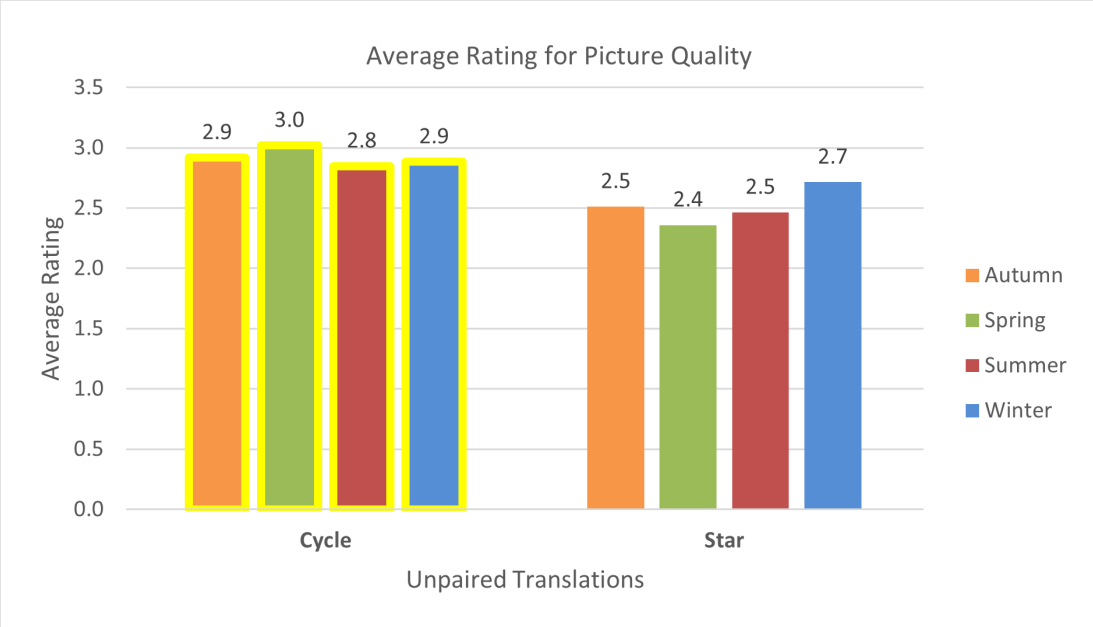

Figure 4.3.5: Survey results: Paired Quality Score

According to the responses, CycleGAN (figure 4.3.5) performed better than StarGAN in terms of image quality across all seasons. The Spring images generated by CycleGAN were rated the highest. CycleGAN averages for all seasons were above 2.8, whereas StarGan averages for all seasons were below 2.8. These results indicate that CycleGAN is a superior network to StarGAN in terms of image quality.

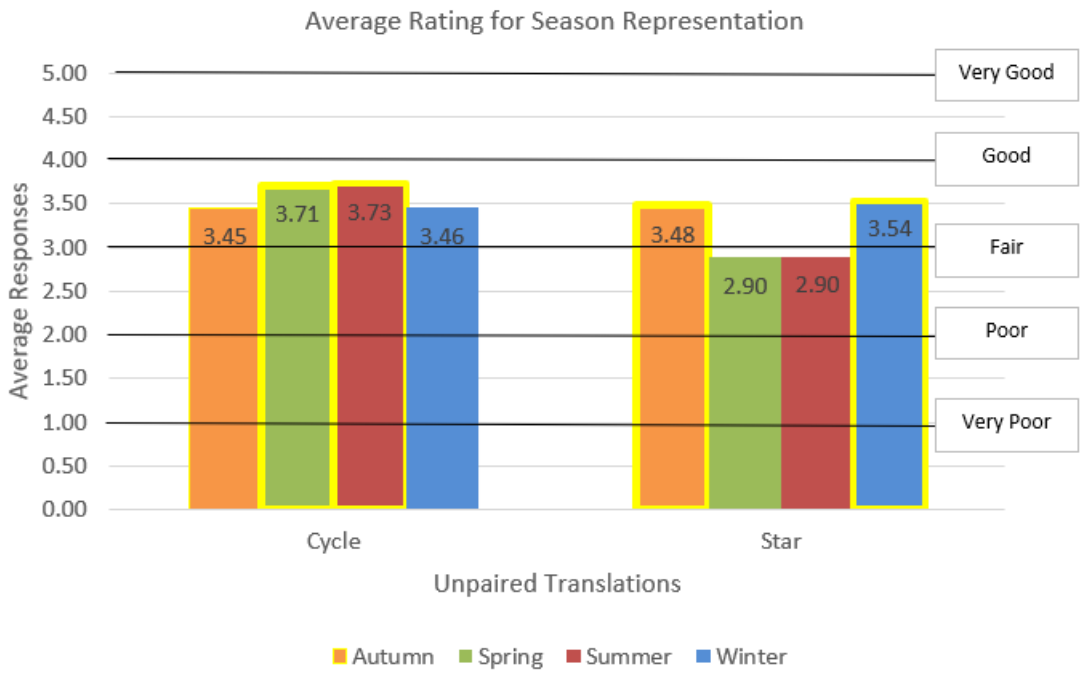

Figure 4.3.6: Survey results: Paired Desired Attribute Score

In the second bar graph (4.3.6), we asked participants to rate how well the generated image represented the desired season attribute, for the unpaired models. They were given five options: Very Good, Good, Fair, Poor, and Very Poor, which we scaled from 1 to 5 for computation purposes, with 5 being Very Good and 1 being Very Poor.

Based on the responses we received, the participants had mixed opinions on the ability of the CycleGAN and StarGAN models to generate desired season attributes. While more participants thought that CycleGAN performed better for summer and spring, the majority opinion showed that StarGAN generated better autumn and winter images. As shown in the graph, CycleGAN received the highest ratings for spring and summer, with average values of 3.71 and 3.73, respectively. On the other hand, StarGAN received slightly higher ratings for autumn and winter, with values of 3.48 and 3.54, respectively. However, overall, it appears that CycleGAN performed better across all seasons, with all ratings falling in the 'fair' range. In contrast, StarGAN received 'poor' ratings for spring and summer.

Therefore, based on these results, we can conclude that CycleGAN is a superior network compared to StarGAN in terms of image quality and the generation of desired attributes.

Paired And Unpaired Models

Despite training both paired and unpaired models on different datasets we will still be comparing all four models together and will see according to our survey which network was most voted by our participants. We will start with Image quality and see for which seasons which model works best, we will then see the results of desired attributes, and finally, we will sum up all the responses and see which model worked best.

As shown in figure (4.3.7), we compare the performance of all four networks in terms of image quality, as rated by participants in our survey. It can be seen that Pix2PixHD received the highest averages of above 3 stars across all seasons, performing better than the other models, with an overall average of 3.3. Pix2PixHD was the only model to have averages for all seasons above 3 and to have an overall average above 3. The picture quality of Pix2PixHD was voted the best, followed by CycleGAN. Pix2Pix and StarGAN had similar picture quality ratings, according to the opinions of our participants.

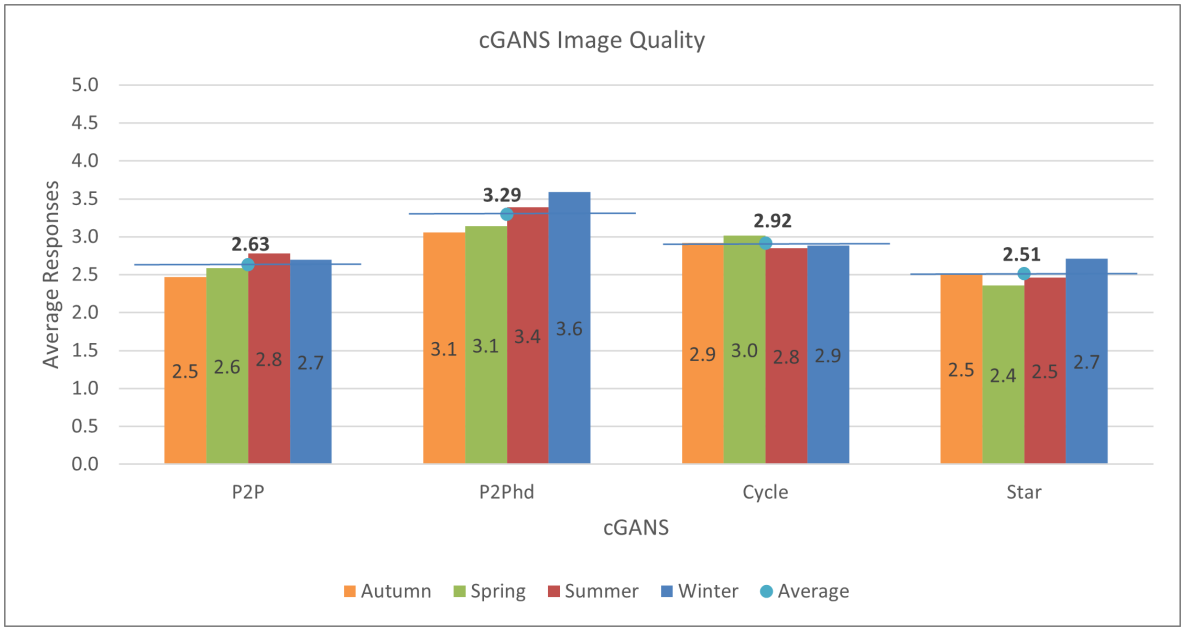

Figure 4.3.7: Survey results: Combined Picture Quality Score

In terms of the desired season translation, figure (4.3.8) shows the overall ratings given by participants in our survey. Overall Pix2PixHd had the highest average of 3.7, followed closely by CycleGAN, 3.6. For the winter season, Pix2PixHD outperformed all seasons and models. Whereas for the summer season, CycleGAN performed better than others. Therefore, the results for desired attributes are mixed among the models, with all models doing well with an overall average above 3. However, if we need to make a decision based on the overall performance, we can conclude that Pix2PixHD is the best model for generating desired attributes.

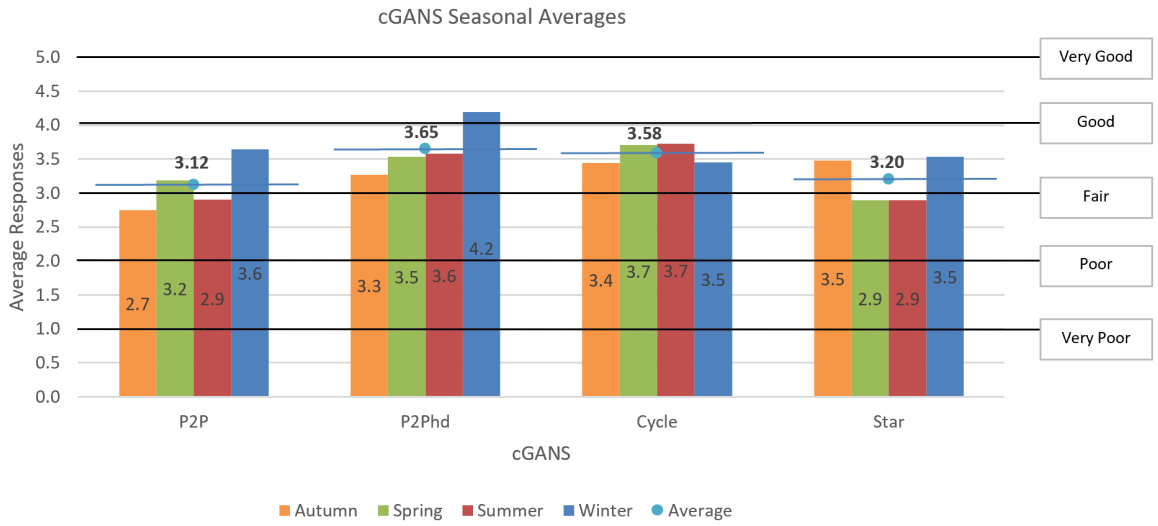

Figure 4.3.8: Survey results: Combined Desired Attribute Score

We have evaluated the performance of all four models for each season. Now, we will combine the results from all seasons and examine the overall performance of each model in terms of image quality and desired attributes. This will allow us to determine which model performs the best overall.

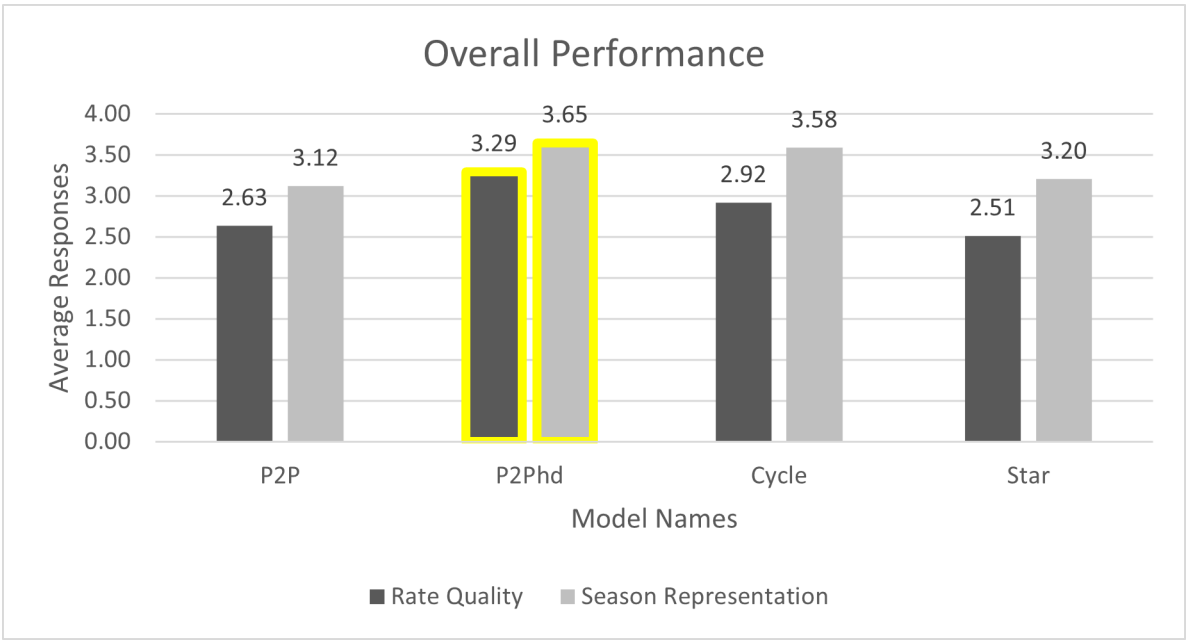

Figure 4.3.9: Survey results Overall

The results in Figure (4.3.9) indicate that Pix2PixHD and CycleGAN are the top models in terms of image quality and generating desired seasonal attributes. To understand the results we take an

average of rate quality and desired season representation to see the overall average. Both models have similar scores, with Pix2PixHD having a slightly higher average rating of 3.47 compared to CycleGAN's 3.25. Therefore, it is difficult to determine a clear winner between the two models. The results of Pix2Pix and StarGAN were similar, with both models receiving comparable ratings for image quality and desired attributes. The overall average for Pix2Pix is 2.88 and for StarGAN it is 2.86.

Based on these results, it is inconclusive as to whether Pix2PixHD is better than CycleGAN or vice versa. To make a decision, it is necessary to consider factors such as ease of training and requirements. In this regard, it can be determined that CycleGAN is a better option as it is an unsupervised network that does not require a paired dataset, but can also be trained on paired data.

5 CONCLUSION

In this project, we evaluated four papers dealing with image-to-image translation problems: Pix2Pix (3.1), Pix2PixHD (3.2), CycleGAN (3.3), and StarGAN (3.4).

We aimed to determine which model would be the best choice for image-to-image translation tasks for different seasons. We trained 37 models and tested them on unseen data. We selected the best-generated images from each model and conducted a public survey. The aim of a public survey is to use human perception to rate which network produces the best quality image and which network produces the best desired seasonal attributes.

The survey results were analyzed and we found that Pix2PixHD and CycleGAN are tied as both have similar scores when considering both image quality and desired attributes. We cannot conclude which network is better as both are trained on different datasets. It is unclear which network is the winner, therefore, we recommend training both CycleGAN and Pix2PixHD on the same paired dataset for all seasons, and then conducting another survey to determine which is the best. Additionally, using mathematical metrics such as the FCN score can also assist in determining which network is superior.

In order to declare a winner, we would consider CycleGAN as the superior network due to its unsupervised nature and ability to be used for a wide range of tasks, whereas Pix2PixHD requires paired datasets and is limited in its capabilities. This project serves as a preliminary comparison of networks and aims to determine the best model based on human perception.

BIBLIOGRAPHY

[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[2] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[3] 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*, pages 2223–2232, 2017.

[4] Hajar Emami, Majid Moradi Aliabadi, Ming Dong, and Ratna Babu Chinnam. Spa-gan: Spatial attention gan for image-to-image translation. *IEEE Transactions on Multimedia*, 23:391–401, 2020.

[5] Feng Xiong, Qianqian Wang, and Quanxue Gao. Consistent embedded gan for image-to-image translation. *IEEE Access*, 7:126651–126661, 2019.

[6] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017.

[7] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018.

[8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[9] MATLAB. Train Conditional Generative Adversarial Network (CGAN) . https://de.mathworks.com/help/deeplearning/ug/train-conditional-generative-adversarial-network.html#:~:text=A%20conditional%20generative%20adversarial%20network%20(CGAN)%20is%20a%20type%20of,corresponding%20to%20the%20same%20label. [Online; accessed 22-April-2022].

[10] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[12] Ugur Demir and Gozde Unal. Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1803.07422*, 2018.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] V Kurama. A review of popular deep learning architectures: Resnet, inceptionv3, and squeezenet. *Consult. August*, 30, 2020.

[15] Daniel Olid, José M Fácil, and Javier Civera. Single-view place recognition under seasonal changes. *arXiv preprint arXiv:1808.06516*, 2018.

[16] Yunjey. StarGAN - Official PyTorch Implementation. https://github.com/yunjey/stargan. [Online; accessed 10-October-2022].

[17] Adeel Ahmed. Image Translation Using GAN Survey. https://surveyhero.com/c/mwcvk3h7. [Online; accessed 22-December-2022].