Imperial College London

MEng Individual Project Interim Report

Imperial College London

Department of Computing

Counterfactuals: The impact of image properties on the quality of generated explanations in XAI

Supervisor:
Dr Ahmed Fetit

Author:
Daniel Nguyen

Co-Supervisor:
Kanwal Bhatia

Second Marker:
Prof Paul Kelly

June 17, 2024

Contents

Chapter 1

Introduction

1.1 Motivation

The increasingly rapid development and success of artificial intelligence (AI) has various industries with the healthcare sector in particular keen in adopting such technology. Capable of reading scans and making diagnoses, paired with the national shortage in staff numbers [1], AI appears to be the perfect answer - saving significant amounts of time and manpower. While initial attempts by researchers to build Computer-aided Detection (CAD) systems to aid tasks such as mammography screening in the 1980s have been shown to have had overall negative impact [2], modern architecture built with deep learning techniques have shown promising results with increasing usage of such systems over the past few years in radiology and oncology [3, 4].

Integrating machine learning systems into clinical workflow does not come without risks - an incorrect diagnosis can have devastating consequences. The possibility of such grave repercussions results in a lack of trust by adopters and many challenges and barriers that must first be overcome [5]. Fostering trust is not an easy nor a quick process, especially for black box classification models - models that just give you a prediction without any particular apparent explanation. Can one unquestionably trust a seemingly correct prediction solely based on its high performance? Most contemporary deep learning architecture models which consists of many complex layers and parameters are black box models. While extremely powerful, these algorithms can be hard to explain, especially to clinicians and adopters who most likely do not have the relevant background experience in machine learning.

There are many methods that have been developed that attempt to explain an AI's "thought process". There are methods that are model specific and require knowledge of the implementation of the AI such as Gradient-weighted Class Activation Mapping (GradCAM) [6]. However, when you do not have access to the AI's architecture which most often occurs when third party AI products are made available for clinical use - only the predictions, post-hoc methods which are model agnostic are required to evaluate these AI. One such example is Local interpretable model-agnostic explanations (LIME) [7]. LIME is a method that perturbs various parts of the images while keeping track of the resultant impact on the classifier's prediction. The areas which successfully flip the prediction are then overlayed and thresholded

to build a saliency map which effectively highlights the regions which the AI think are relevant to the task.

Another example is using counterfactual explanations. Counterfactual explanations represent a methodology aimed at elucidating the rationale behind an artificial intelligence's decision-making process. This approach involves constructing an artificial image that closely resembles the original, with minimal alterations strategically applied resulting in a different decision from the original made by the AI [8]. For example, the most minimal changes made to a cat image that would cause the classifier to instead predict tiger. One effective technique for generating such counterfactual images involves the utilization of cycle generative adversarial networks (GANs), each specialized in training on distinct classes [9]. Each GAN is tasked with the translation of an image from one domain to another, or, in the context of the classifier, from one prediction to an alternative. The visual disparities between the original image and its counterfactual counterpart serve as the explanation for understanding the classifier's decision-making dynamics.

While these methods have been explored and developed in the realm of 2D images [10], it is important to note that the extension of such techniques into the 3D domain is still an emerging area with ongoing exploration and development. Applying these methods to 3D medical images, which involve factors like slice thickness, is not as straightforward as with 2D data. The added dimensionality introduces complexities, and the curse of dimensionality makes implementing these techniques in 3D more challenging.

1.2 Contributions

The goal of this project is to investigate and reason how image properties of dataset can affect the quality of generated counterfactuals. Furthermore, we also develop and extend existing state of the art counterfactual methods to support and operate on 3D images.

We illustrate our findings using ophthalmic and artificial datasets, demonstrating that both the classification model's architecture and the images' textural and shape properties strongly impact the quality of the generated counterfactuals. We also develop and implement an extension of the existing 2D counterfactual image generation by Zhu et al.[9] methods to work with 3D images on a different dataset and showcase some results produced by our new extended architecture. This work has been accepted for publication for Medical Image Understanding and Analysis (MIUA) 2024 which has been peer reviewed and will be published as an e-book in Frontiers in Medical Technology.

Chapter 2

Background

In this chapter, we will:

- cover the basic background knowledge on relevant machine learning terminology and knowledge such as Convolutional Neural Networks, Generative Adversarial Networks, and more to have some degree of understanding on how counterfactuals are generated.
- provide a comprehensive literature review on some existing XAI methods as well as AI problems and architectures that these methods rely on and exploit.

2.1 Image Classification

Image classification is a specific task within the realm of computer vision, wherein the computer categorizes images into predefined labels. This process is commonly accomplished through the utilization of deep learning algorithms. By providing the model with ground truth labels and a set of training images labeled accordingly, the objective is to train a model capable of accurately labeling similar, unseen images. Datasets containing only two classes are referred to as binary classification, while those with more than two are designated as multi-class classification. Models trained exclusively on two types of images are known as binary image classifiers.

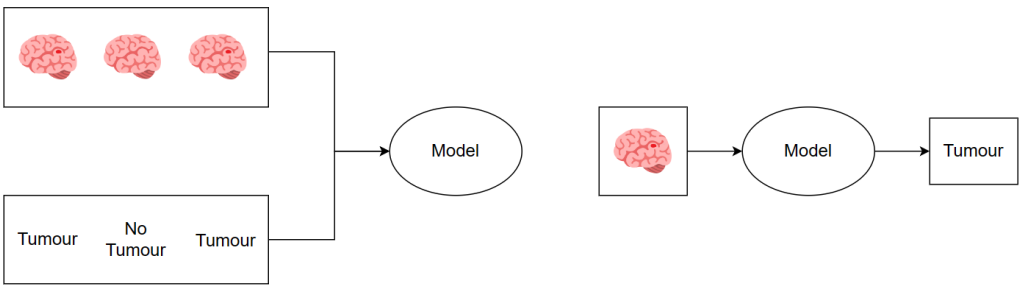


Figure 2.1: An example of a binary image classifier that identifies brain tumours

Image classification utilises neural network structures, like Convolutional Neural Networks (CNNs), to unravel intricate details within visual data. The model undergoes a learning process, mapping input images X to a set of predefined labels

Y by iteratively adjusting a set of parameters θ. Through the sequential processing of images via convolutional layers, pooling, and activation functions, the model captures relevant features crucial for accurate classification. The training objective revolves around minimizing the loss (L) and fine-tuning of parameters θ to enhance the model's accuracy in assigning labels to images. Given a perfect model that perfectly maps the correct set of images to labels, we have that:

$$\forall x \in X \, C(x;\theta) = X \quad \text{and} \quad \forall y \in Y \, C(y;\theta) = Y$$

where in our figure 2.1 example, x would be the images belonging to the Tumour class, and given a tumour image, the classifier C would give us the label Tumour and likewise for the normal images y. The goal of the model, C, is to learn the set of parameters θ such that we learn the mapping above for any given image. How θ is most typically learnt is through the usage of deep learning architecture such as Convolutional Neural Networks.

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) have become the main type of models used for image classification. Compared to traditional Artificial Neural Networks (ANN), CNNs excel at capturing image features through the usage of convolution, pooling, and fully-connected layers [11].

The structure of CNNs can be typically broken down into:

- The **input layer**. This is the first layer of the network. The layer that receives the input to process. For images, this is the raw pixel values.
- A series of **convolutional layers**. These convolutional layers applies filters (or kernels) - parameters which aim to learn and extract certain features. Different convolutional layers can learn different features. Between each of these layers, an activation function (such as Rectified Linear Unit (ReLu) or Sigmoid) is applied to introduce non-linearity.
- The **pooling layers**. These layers downscale the dimensions of the input data from previous layers which helps reduce the number of parameters and consequently speeds up computation.
- The **fully-connected layer**. It comes after the convolutional and pooling layers and has neurons equal to the number of classes in our dataset. This layer plays a role in combining features from earlier layers to make the final classification decision.

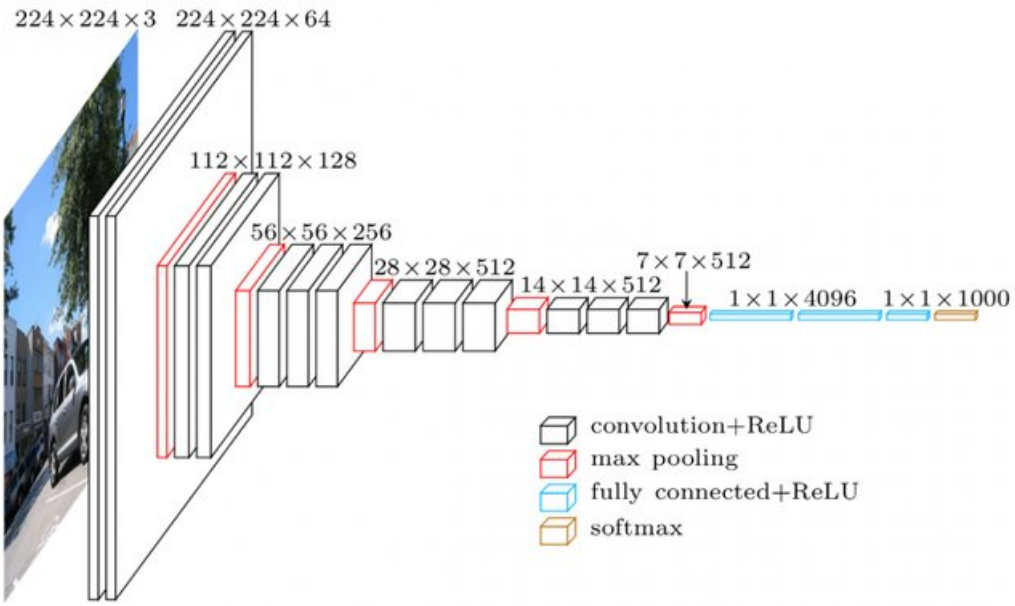


Figure 2.2: Architecture of the VGG-16 neural network. Note the usage of convolutional layers (white) that increase the number of feature maps produced and pooling layers (red) which downscale the input to half resolution [12]

2.3 Image-to-Image translation

Image-to-image translation is another task within the realm of computer vision where the aim is to learn a mapping function from one visual domain to another. Given a set of paired data (X, Y) where the input domains are aligned, the goal is to learn the optimal parameters θ for the function $F : X \to Y$ such that $\hat{y}$ is indistinguishable from y.

$$F(x; \theta) = \hat{y}$$

However, this can also be achieved with unpaired data with a set of images X and a set of images Y. As unpaired data is highly unconstrained with the lack of aligned input domains, an additional inverse mapping $G : Y \to X$ to enforce $F(G(X) \approx X)$ (and vice versa) [9].

$$F(x; \theta_x) = \hat{y}$$
$$G(\hat{y}; \theta_y) \approx x$$

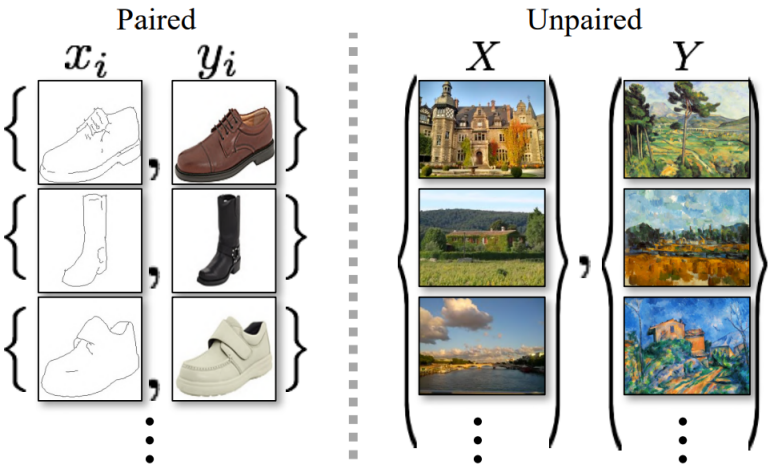


Figure 2.3: An example of paired and unpaired datasets. Unlike paired datasets, unpaired datasets are less constrained and therefore have many possible mappings between the two datasets. [9]

Image-to-image translation is typically achieved with the usage of deep learning Generative Adversarial Models (GAN) which consists of 2 models, a generative model G and a discriminator D which train with each other in a competitive manner. The goal of the Generator is to make convincing samples that would fool the discriminator, whereas the goal of the Discriminator is to differentiate between real data and samples generated by the Generator. The training continues until the Generator is capable of generating indistinguishable samples which can be then used to generate new data that resembles the training data [13].

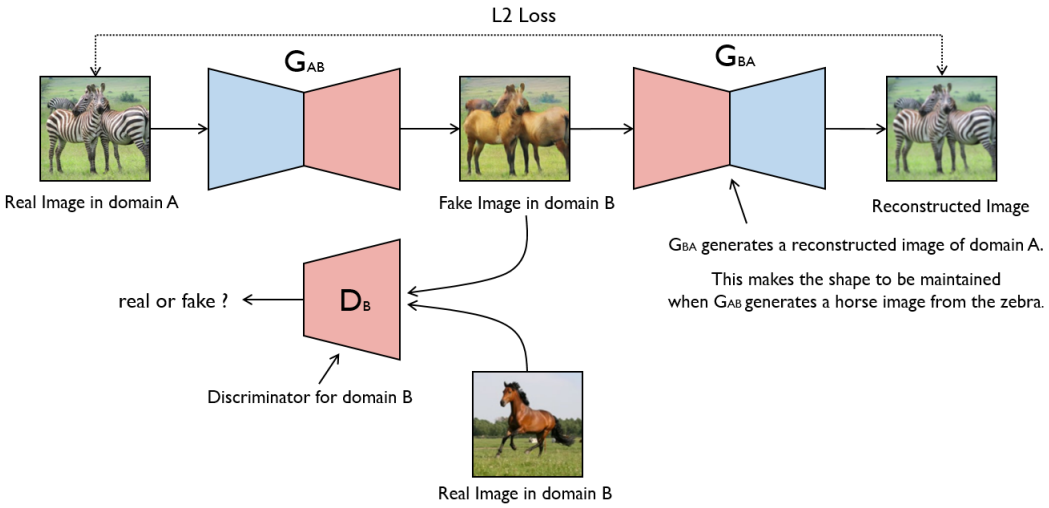


Figure 2.4: Unpaired Image to Image translation using a CycleGAN. The generator G_{AB} learns to generates a fake horse given an image of a zebra whereas the discriminator D_s learns to identify the origins of a given horse image. [14]

2.4 Post-hoc explanations for Image Classifiers

As seen by the complicated structures of CNNs, it is not easy to infer or explain to someone the reasoning behind the architecture design choice and how it correlates to an imaging classifier's decision. Furthermore, given that access to the model's architecture is often limited or not given at all, the black-box nature of these models requires methods of explanations that do not require any knowledge or access to the model at all other than the predictions. These methods are known as post-hoc methods and are model agnostic.

There are many types of post-hoc explainability methods but they can all be classified into two types of methods, local explainability or global explainability. Local explainability aims to explain the AI's behaviour on a lower level - why a certain predictions were given for a specific data subset, what features contributed to this decision. Global explainability on the other hand, aims to explain the overall behaviour of the model for the dataset.[15].

2.5 Related Work

The issue of explainability in AI has been a focal point of extensive research within the academic community.[16]. Numerous studies have delved into methods and techniques aimed at enhancing the transparency and interpretability of AI systems. While methods for explainable AI have been researched and developed for 2D images [10], the extension of such techniques to 3D images is extremely limited and remains an ongoing area of exploration.

2.5.1 Randomized Input Sampling for Explanation of Black-box Models

Randomized Input Sampling for Explanation of Black-box Models (RISE) is a local explainability method that perturbs a given image by randomly masking out parts of the image. This masked out image is then fed to the classifier where its prediction is recorded. We repeat this process a numerous amount of times and record the areas that were masked out that caused the classifier to change its prediction as the masked out area most likely contained features deemed important to the classifier. Finally, by overlaying these masked out areas and applying a threshold, we can create a saliency map highlighting the regions with the most importance to the classifier.

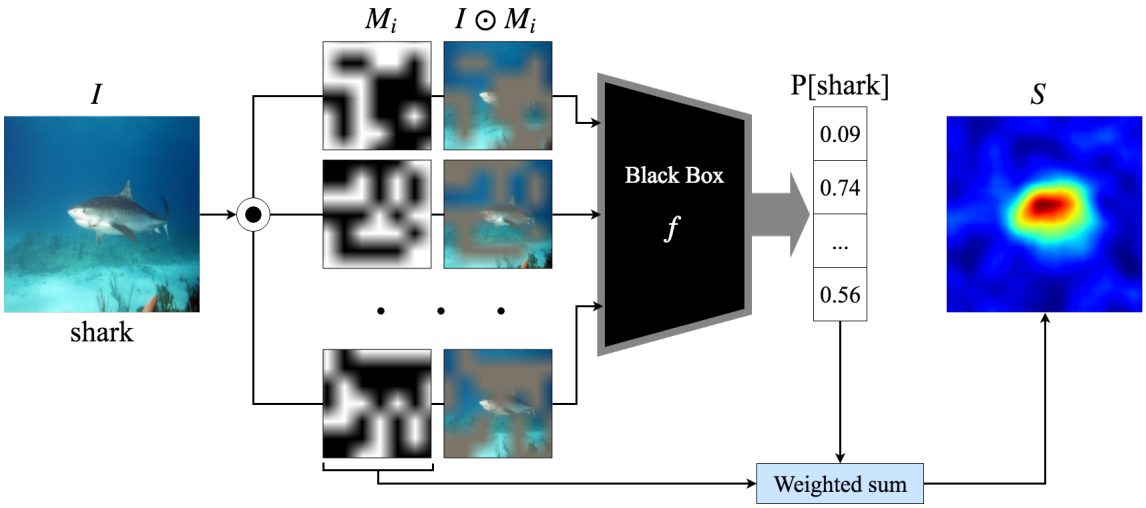


Figure 2.5: How RISE works to generate a saliency map[17]

There are also alternative local explainability methods which can also generate saliency maps such as LIME that do not rely on randomly obscuring the image but instead directly creating a new dataset through perturbations.

2.5.2 Local Interpretable Model-agnostic Explanations

Local Interpretable Model-agnostic Explanations (LIME) is a type of local explainability method that perturbs various parts of the images [7]. By perturbing random parts of the image to create a new dataset, the model is then trained on this new dataset. How close certain features in the new dataset resemble the original dataset determines the weighting and impact a particular feature has on the classifier's decision. From this, a saliency map can be created showing which of the images had the greatest impact.

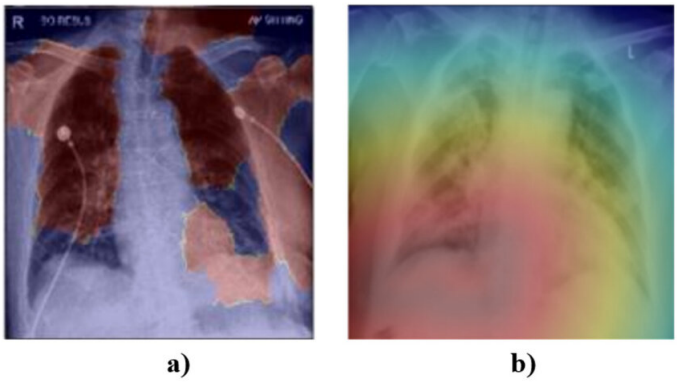


Figure 2.6: An example of LIME being applied to a Covid classifier [18]

While LIME excels at highlighting regions of the classifier's interest, when there are multiple features located within the region, it can be sometimes hard to tell which feature(s) that the classifier is specifically looking at. As a result, alternate

XAI methods have been explored and developed which address this problem and one is counterfactuals. Counterfactuals aim to directly modify the image and introduce/remove the relevant features that define class of images and this is done by using Generative Adversarial Networks and re-interpreting the problem as an Image-to-Image translation problem.

2.5.3 Generative Adversarial Networks

The concept of GANs was first introduced by Ian Goodfellow [13] in 2014. Unlike other networks, the GAN consists of two models, the generator and the discriminator. The two models would play a two player mini-max game, in which the generator would aim to generate convincing images that could feasibly exist in the distribution of the real dataset, and the discriminator would aim to accurately differentiate the origins of the image. Since the introduction of the GAN, many different types of GANs have been made with the aim to improve the training process or outputs of the GAN.

2.5.4 Image-to-Image translation

Image-to-Image translation (I2I) is a problem typically solved using GANs in the computer vision field with growing traction and attention due to its numerous applications in the real world ranging from image synthesis and segmentation to style transfer or restoration [19].

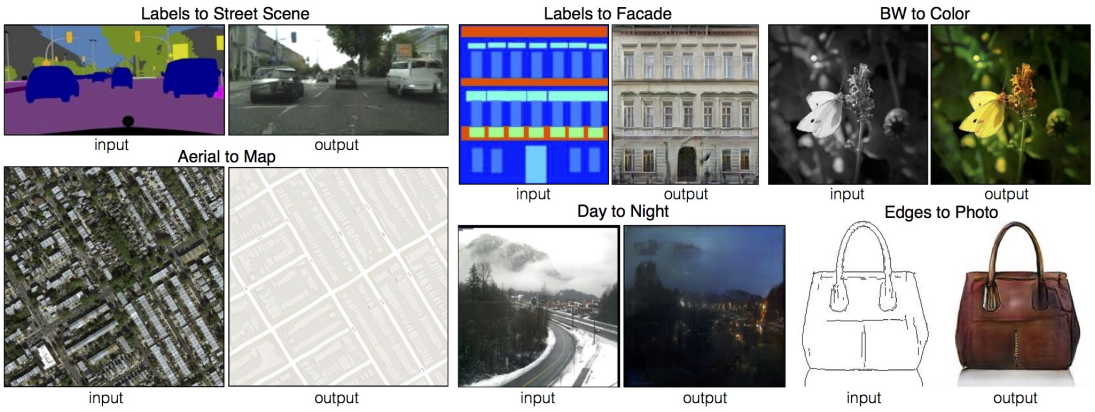


Figure 2.7: Example applications of image-to-image translations[20]

An interesting application of Image-to-Image translation is using it to generate counterfactuals. While the uses of counterfactuals have been explored by Wang et al. [21] in 2020 to improve the mammogram classifier, Mertes et al. [22] instead explored the usage of counterfactuals as a means of XAI. They note that the criteria that define a counterfactual similarly define the problems and goals of an image-to-image translation problem, that is the counterfactual should belong to the domain of the other image, and that the counterfactual should resemble as closely to the original image as possible.

2.5.5 Counterfactual Instances

Counterfactual instances is an alternative local explainability method. Unlike map based methods such as LIME which require the user to make a direct judgement of which features within the highlighted regions are relevant, counterfactual instances aim to directly remove the relevant features. The problem of generating counterfactuals can be interpreted as a form of image-to-image translation - Given a set of images X and a set of images Y, we can train a GAN that learns the mapping between X and Y. Using the GAN to generate a counterfactual for an image, we can use the differences of features between the original image and the counterfactual instance as an explanation of which features drove the AI to make a particular decision.

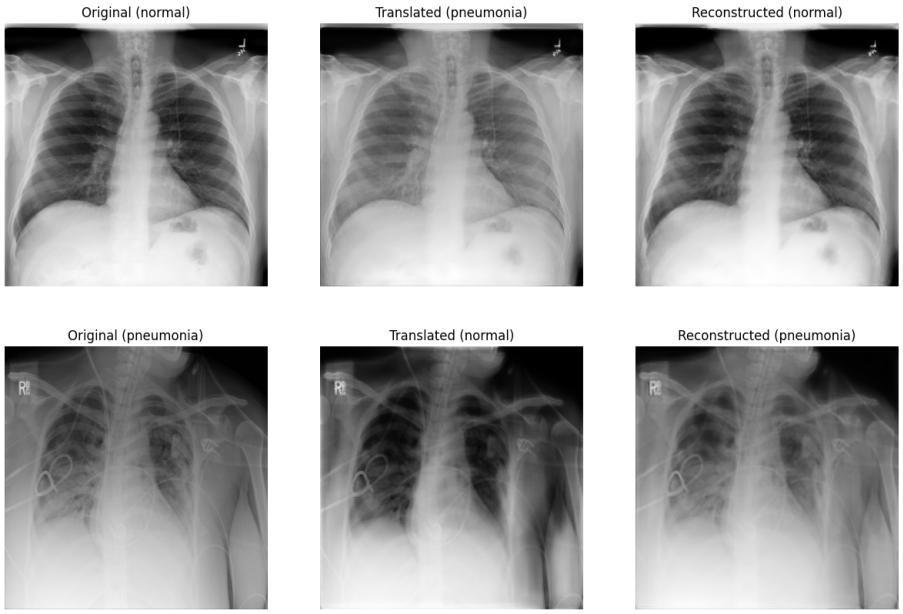


Figure 2.8: An example of counterfactual for a pneumonia dataset using the RSNA Pneumonia 2018 dataset [23]. We can see that the opacity present in the lungs is the feature the AI thinks is an indication of pneumonia as indicated by the fact that the original image (left column) had no opacity for the normal class (top row) or some opacity present for the pneumonia class (bottom row). The generated counterfactuals (middle column) then introduced and removed the opacity for the translated normal and pneumonia images respectively.

Zhu et al.[9] first proposed the algorithm for generating images with unpaired datasets by using 2 GANs in a cycle called a cycleGAN, where each GAN is responsible for learning a mapping from one image domain to another whilst enforcing cycle consistency - $F(G(X)) \approx X$ in the form of a cycle loss to further constrain the possible mappings it can learn due to the nature of unpaired datasets having less constraints and more mappings. There have been various works that use these

techniques to create counterfactual images. Wang et al.[21] has published an algorithm for counterfactual image generation for mammography classification using breast images in order to improve their algorithm but additionally rely on the fact that healthy human breasts should look symmetrical to guide their approach thus limiting generalisability.

One issue with using CycleGANs to generate counterfactuals is that the generator is generating images for the discriminator to differentiate which Mertes et al. argues that the counterfactuals generated can only be seen as an explanation for the GAN rather than the classifier we are trying to explain [22]. They improve on this and presented a new algorithm, which introduces an additional counterfactual in addition to the cycle loss, and evaluated this on an RSNA pneumonia dataset and AlexNet classifier and evaluated how this method performs compared to traditional cycleGANs as well as other explainable methods. They conclude that the new modifications to the original cycleGAN architecture resulted in a positive impact on results as well as surveys showing the participants preference to counterfactual explanations over other map based explanations such as LIME and Layer-wise Relevance Propagation (LRP). They note that their approach should be used where raw spatial information provided by LIME and LRP is not enough. Counterfactuals can fill in the weakness provided by other XAI methods such as LIME and LRP but as we will see, counterfactuals also have their own weaknesses and inability to handle certain datasets.

Chapter 3

Methods

In this chapter, we cover the methods used on a technical level, how we implemented the GANterfactual CycleGAN and the models that were used throughout the project which the CycleGAN will try to generate explanations for. We also cover how the models were trained and if any preprocessing steps were required.

3.1 Using CycleGANs to generate counterfactuals

There are many methods to generate counterfactual images using generative models such as diffusion models [24] and GANs [25]. We will be focusing on using GANs, specifically the GANterfactual CycleGAN method proposed by Mertes et al. [22], to generate the counterfactuals for the dataset.

3.1.1 GANs

The original GAN first proposed by Goodfellow et al. [13] generates convincing images that follow the same probability distribution as the training dataset by having the generator network G, as well as the discriminator network D, play a two player mini-max game. The generator, as the name implies, is responsible for generating images that could appear to originate from the original training dataset whereas the discriminator is responsible for identifying the origin of the given image. The prediction given by the discriminator is then used to improve both networks in an adversarial manner. We can define the objective function for the GAN as follows:

$$\min_G \max_D E_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{3.1}$$

where x are the images from the real dataset and z are randomly generated latent variables. We can see that the generator wants to minimise this objective function as this indicates that the discriminator is incapable of distinguishing fake from real images whereas the discriminator wants to maximise this to indicate that the generator needs to perform better.

3.1.2 CycleGANs

Since the inception of the GAN architecture, many new additions and modifications have been made to suit various needs and purposes and one of them was the CycleGAN architecture proposed by Zhu et Al [9]. to solve unpaired Image-to-Image translation. A CycleGAN consists of two GANs where each GAN is responsible for learning to transform an image from a domain X to the other domain Y and the other GAN learns the reverse. To define the objective function for the cycleGAN, we must first define some other loss terms first. We can depict the full objective loss function as a diagram:

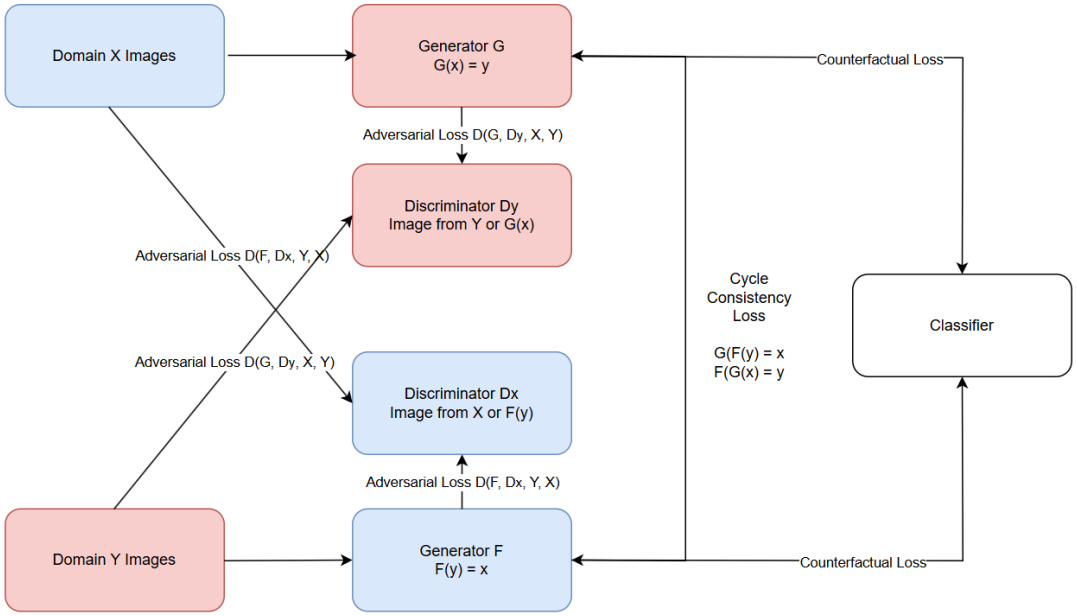


Figure 3.1: A simplified diagram of the CycleGAN and how each loss component is calculated. We omit the identity loss from this diagram for clarity.

Adversarial loss

We define the adversarial loss for the generator G and the discriminator D_Y as:

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = E_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + E_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))] \quad (3.2)$$

Where G and D_Y are the generator and discriminator for the GAN responsible for learning to transform an image from domain X to Y. Similarly, we can define the adversarial loss for the other generator F and the discriminator D_X as:

$$\mathcal{L}_{\text{GAN}}(F, D_X, Y, X) = E_{x \sim p_{\text{data}}(x)}[\log D_X(x)] + E_{y \sim p_{\text{data}}(y)}[\log(1 - D_X(F(y)))] \quad (3.3)$$

Where F and D_X make up the other GAN which is responsible for learning to transform images from domain Y to X.

Cycle-Consistency Loss

In addition to the adversarial losses of the two GANs, the cycle-consistency loss enforces that an image translated to the other domain and then back to the original domain is similar to the original image. It is defined as:

$$\mathcal{L}_{\text{cycle}}(G, F) = E_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + E_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1] \quad (3.4)$$

Identity Loss

While this loss is optional, we add this loss to encourage the generators to preserve the colour composition and structure between the input and the output. It is defined as:

$$\mathcal{L}_{\text{identity}}(G, F) = E_{x \sim p_{\text{data}}(x)}[\|G(y) - y\|_1] + E_{y \sim p_{\text{data}}(y)}[\|F(x) - x\|_1] \quad (3.5)$$

Counterfactual Loss

In addition to all the other losses, Mertes et al. proposed an additional loss called the Counterfactual loss [22] as they argue that generating counterfactuals without the input of the classifier you are generating the counterfactuals for cannot be seen as anything other than a counterfactual of the discriminators of the cycleGAN. They define the loss as:

$$\mathcal{L}_{\text{counter}}(G, F, C) = E_{x \sim p_{\text{data}}(x)}[\|C_2(G(x)) - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \|_2^2] + E_{y \sim p_{\text{data}}(y)}[\|C_2(F(y)) - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \|_2^2] \quad (3.6)$$

Where C_2 is the softmax output of the classifier for the two classes given an input translated image $G(x)$ or $F(y)$

Full Objective Function:

The full objective function combines these losses. The weights λ_{cycle}, $\lambda_{\text{identity}}$ and λ_{counter} control the importance of the cycle-consistency, identity and counterfactual losses, respectively.

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) &= \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ &+ \lambda_{\text{cycle}}\mathcal{L}_{\text{cycle}}(G, F) + \lambda_{\text{identity}}\mathcal{L}_{\text{identity}}(G, F) + \lambda_{\text{counter}}\mathcal{L}_{\text{counter}}(G, F, C) \quad (3.7)\end{aligned}$$

3.2 Model training and preparation

Before we can use the CycleGAN method to explain a classifier's decision making for a given dataset, we must first train the models on the dataset. We aim to train all the models to the same levels of accuracy of at least 85% on the test subset of data and 70% for the more difficult datasets. Where available, we apply transfer learning using pre-trained weights for the complex datasets.

3.2.1 Dataset preparation

All datasets (Section 4) were split into training, validation, and testing subsets with a 7/2/1 split respectively. We use the testing subset to evaluate the accuracy of our models for both the regular dataset as well as when generating counterfactuals. The images were also normalised to be in the range of [-1, 1] for all models to achieve the best results when working with Tanh layer that the GANs commonly use as the Tanh activation function maps all values to [-1, 1] whereas pixel values range from [0, 255].

3.2.2 AlexNet

We use the same provided AlexNet model as provided in the paper [26]. This was the main model we used to evaluate the method as it was used. The AlexNet architecture was first invented in 2012 by Alex et al and won the ImageNet competition [27]. It was only after this moment that interest and development into building deeper and better neural networks such as VGG and ResNet. AlexNet takes in images of 227 x 277 x 3 and consisted of a 5 convolutional layers, 3 pooling layers, and 3 dense layers. Compared to the LeNet-5 [28], one of the first CNNs, the massive increase in depth showed a correspondingly increase in performance.

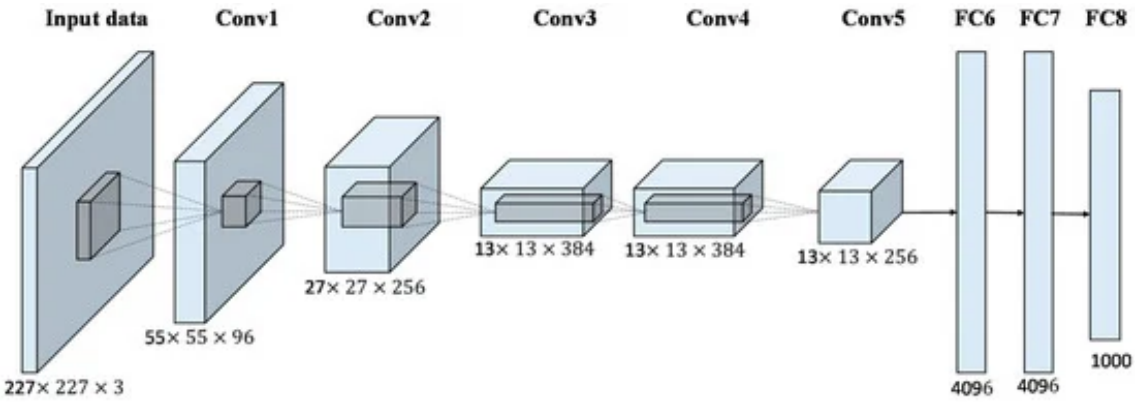


Figure 3.2: AlexNet Architecture Diagram[29]

3.2.3 VGG-19

We use the provided VGG models in the Keras library [30] to train our models. AlexNet while using more convolutional layers than LeNet-5 hit a limit in resources due to the large size of convolutions it was performing. VGG proposed by Simonyan et al. [31] in 2014 aimed to make deeper networks by using a series of smaller convolutions rather than 1 big convolution. This enabled them to make networks out of VGG blocks, consisting of 1-3 3x3 convolution layers followed by a max pooling layer, and the VGG-19 uses 19 of these blocks.

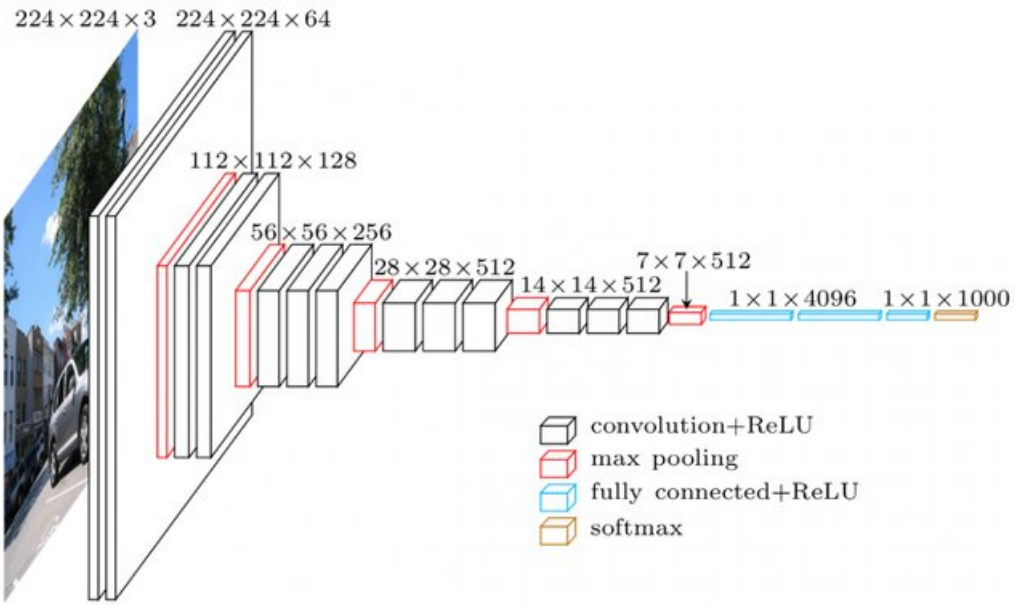


Figure 3.3: Architecture of the VGG-16 neural network. The VGG-19 model has the same architecture but with an additional three convolutional layers. [12]

3.2.4 ResNet-50

We similarly use the provided ResNet-50 models in the Keras library [30] in our project. The ResNet architecture is similar to the VGG architecture in that it consists of many residual blocks. These residual blocks however also include residual connections between the shallow and deeper layers preventing the vanishing gradient problem as the deeper you go, the further you back propagate, and the smaller the gradients become until they disappear. As the number implies, ResNet-50 consists of 50 of these residual blocks which is a huge increase compared to 19 from VGG.

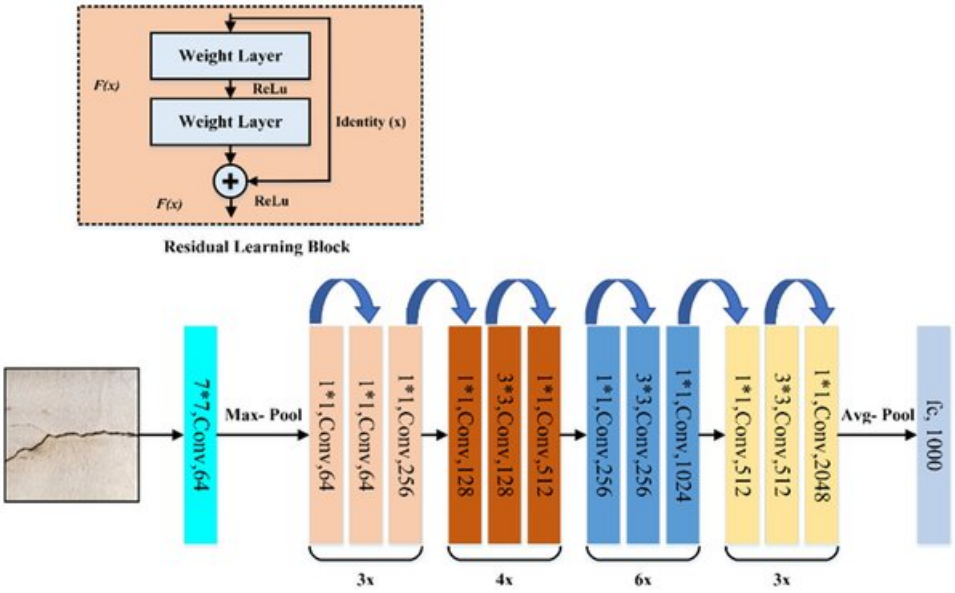


Figure 3.4: Architecture of the ResNet-50 model. [32]

3.2.5 3D-CNN

To test our 3D implementation, we used the model proposed in the MICCAI'2020 PRIME workshop paper [33] with some minor modifications. The 3D-CNN is a 17 layer model with a series of Convolutional, Pooling, and Batch Norm 3D layers. The architecture doesn't differ much compared to 2D CNN other than the usage of 3D layers.

3.3 Counterfactual Evaluation

Once the CycleGAN has trained for a sufficient amount of epochs (20 for large datasets with more than 10,000 images, and 40 epochs for smaller datasets) whilst saving the generators for each epoch, we generate the translated counterfactuals as well as reconstructed images for the entire test subset and feed in these images in addition to the original. We then calculate the accuracy for each of the subset of images where we define a successful translated prediction to be the class that is opposite of the original class, and the reconstructed class to be the same as the original class. That is given an image from class 0, we would want the classifier to predict this image as class 0, the translated counterfactual image as class 1, and the reconstructed image as class 0 again and vice versa for the other class. In addition, we compare the quality of reconstructed images across models using the Mean Absolute Error (MAE) values calculated based on the differences between the original and reconstructed images which is defined as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where y_i and $\hat{y}_i$ are the i^{th} pixel value in each image.

Chapter 4

Datasets

This chapter explores the datasets with different image properties that were used throughout the project to investigate and evaluate the GANterfactual method.

We define each image property and describe the dataset and their image properties. Using these datasets, we investigate how these image properties such as shape and texture can impact the quality of the generated counterfactuals as well as reason as to why this is the case.

4.1 Requirements

To evaluate the GANterfactual method fairly, a wide range of datasets with varying levels of complexity and image properties was used. Notably, we aim to expand the existing experiments by evaluating the method on datasets where relevant information was stored not just as texture but in other forms of information as well such as structure and spatial information. Furthermore, as this method was proposed and evaluated with a medical dataset, we primarily used publicly available medical datasets. As counterfactuals aim to explain an image classifier trained with supervised learning, we also require these datasets to have discrete labels.

While this method works with multi-class classification, for the purposes of simplification, we aim to use binary class and where necessary, preprocess some multi-class datasets into several binary class datasets.

4.2 Image Properties

We categorise our datasets on how they store their relevant information into three distinct categories. A dataset can express information in more than one category.

4.2.1 Textural Information

We say a dataset stores its information texturally if it we can express the image as two distinct layers where the first layer is the original image, and the second layer is the transformation that the CycleGAN applies to the image. Specifically, the transformation does not require to modify the structure or features of the first layer pertaining the original image.

Figure 4.1: An example of how pneumonia stores its information texturally

4.2.2 Spatial Information

A dataset stores its relevant information spatially if we can express the transformation as additional embedding of small *physical* features or objects in the image which can but not necessarily cause some minor modifications to the original layer.

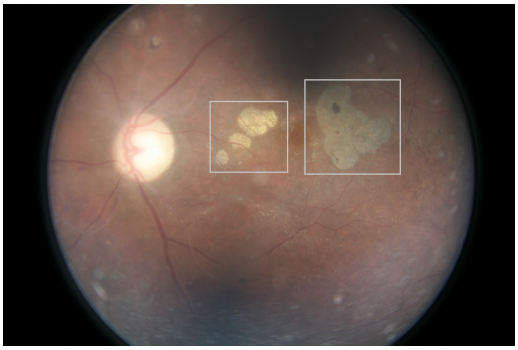


Figure 4.2: Diabetic Retinopathy manifesting as patches in two particularly distinct areas. [34]

4.2.3 Structural Information

For structural information, the relevant information is expressed, not as new features or objects, but rather requires the modification of existing features that are contained in the original image.

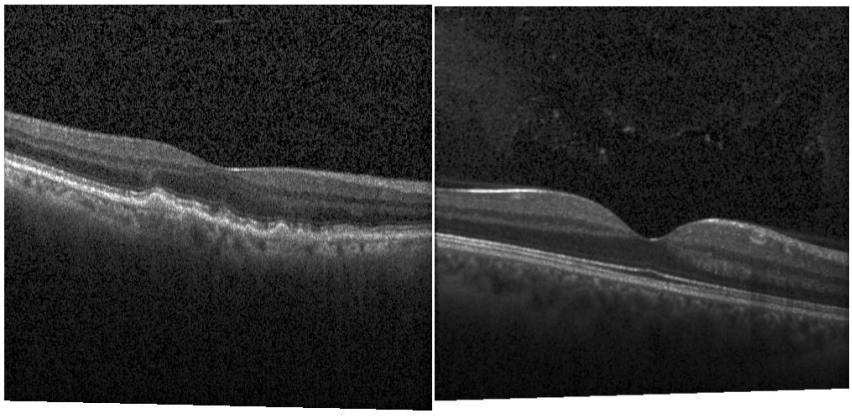
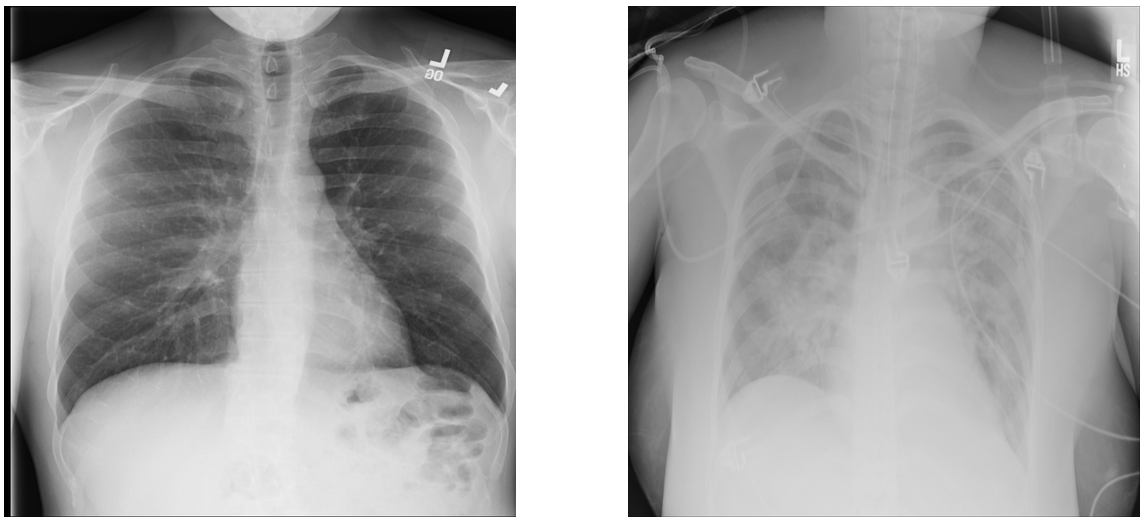


Figure 4.3: The membrane existing in both images with Drusen manifesting as bumps in the left image. [35]

4.3 RSNA Pneumonia Detection Challenge

The RSNA Pneumonia dataset, published in 2018 on Kaggle [23], was the dataset used by Mertes et Al. [22] to evaluate their proposed method. Consisting of chest x-rays of the chest of both normal chests and pneumonia patients stored in DICOM format, the paper provides a preprocesser that converts these DICOMs into PNGs which were then used to train both the AlexNet Model as well as the CycleGAN models. Pneumonia typically manifests as an area of increased opacity in the chest.



(a) A normal chest x-ray

(b) A patient with pneumonia

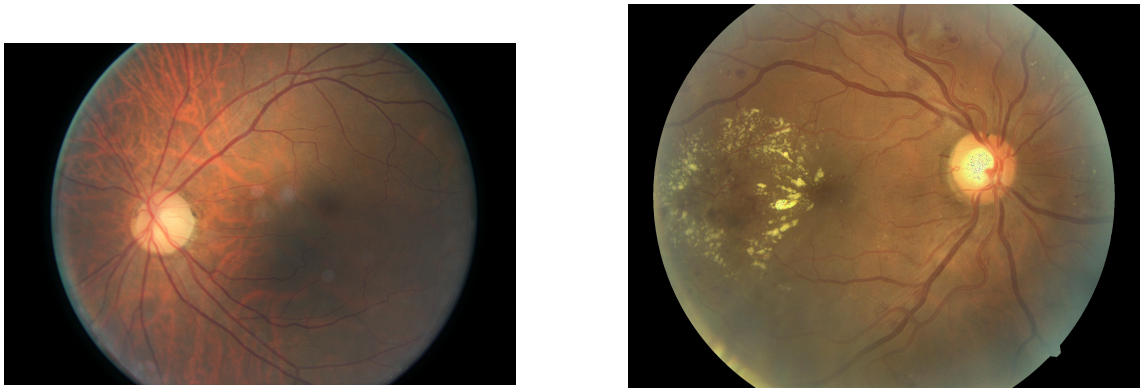
Figure 4.4: An example pair of images from the Pneumonia dataset [23]

This dataset exhibits relatively low feature complexity due to its greyscale nature as well as the consistency of the features across the entire dataset. The chest is always positioned at the center of the image with a consistent black background.

The features that differentiate the two classes are also visually apparent and cover the majority of the image. We note that this dataset stores relevant information as textural information.

4.4 Diabetic Retinopathy

Diabetic Retinopathy (DR) is an eye disease that occurs with long-term diabetes. The DR dataset, published in 2015 on Kaggle [34], consists of various coloured fundus images stored as PNGs of various severity. It is a multi-class dataset labelled by clinicians according to a scale of 0 (No DR present) to 4 (Proliferate DR present). This dataset was reprocessed into a binary class task by combining classes 3 (Severe DR) and 4 (Proliferate DR) to form a new class Severe/Proliferate where as class 0 was used as the normal class. We omit classes 1 and 2 to make the classes visually distinct.



(a) An image of a normal retina

(b) An image of a retina with extreme DR symptoms

Figure 4.5: An example pair of images from the DR dataset [34]

This dataset is the hardest dataset that was used to test the GANterfactual method due to its extremely complex features. In addition to being RGB images, the images also vary a lot in many ways such as:

- Symptoms of DR - Unlike pneumonia which has a consistent transformation between the 2 classes, DR symptoms can be identified by the presence of red spots (microaneurysms) and yellow lesions (exudates).
- Colour - Due to how fundus images are taken, a filter or dye is applied resulting in varying colours.
- Features of the retina - Features of the eyes such as the optical disc or the veins, while present in every image, vary in positioning or quantity and visual clarity.
- Artifacts - Bright crescents may sometimes exist at the edge of the retina due to an illuminated iris

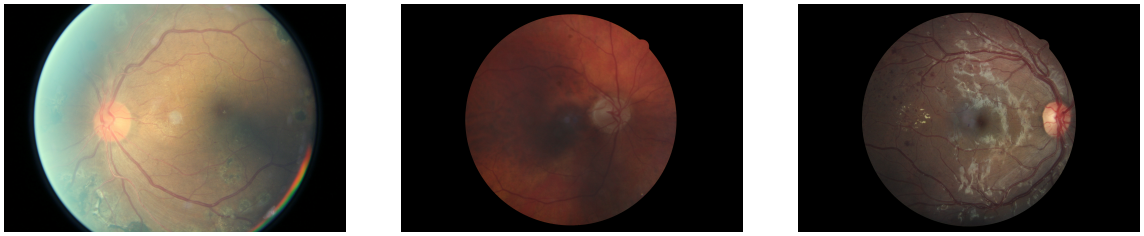


Figure 4.6: Examples of how DR image can vary
[34]

4.5 Retinal OCT

Retinal OCT dataset, published in 2018 [35] consists of high resolution cross section x-rays of the retina of 4 categories: Normal, Drusen, Diabetic Macular Edema (DME), and choroidal neovascularization (CNV).

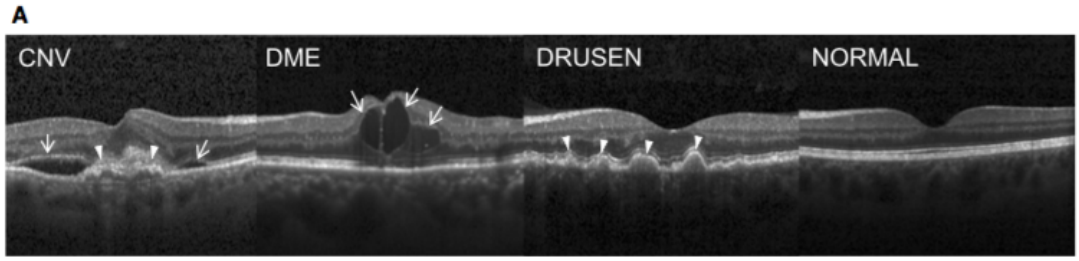


Figure 4.7: Example OCT images of each class [36]

The dataset was repurposed into 2 binary datasets consisting of only DME and Drusen. These 2 classes were chosen as each class only has a singular additional feature. In terms of complexity, these datasets rank slightly higher than the pneumonia dataset. While both the Retinal OCT and Pneumonia dataset are greyscale with simple transformations between the classes, DME and Drusen have extra constraints. Drusen is between the Bruch's membrane and the retinal pigment epithelium of the eye where as DME is typically formed in the macular area.

4.6 Synthetic datasets

The datasets mentioned above are all extremely complex and therefore it is difficult to disambiguate the behaviour of the CycleGAN. Therefore, in addition to some medical datasets, some synthetic datasets were constructed by us, either from scratch or other existing datasets in order to identify which image properties influence the CycleGAN. For synthetic datasets that were made from other existing datasets, the abnormal class was directly derived by applying some transformation to images from the normal class.

4.6.1 Synthetic Box dataset

This dataset consists of 2 extremely simple classes made from scratch. Both images are a 512 x 512 black image with the normal class having a randomly placed grey box. The abnormal class similarly has a randomly placed grey box but with an additional smaller white box inside.

Figure 4.8: An example pair of images from the Synthetic Box dataset. The normal class image (left) consists only of a grey box whereas the not normal class (right) has an additional white box located in the center of the grey box.

This is one of the most simple synthetic datasets created with a singular consistent transformation between the classes. The simplicity of this dataset allows for clear demonstrations of "perfect" generated counterfactuals.

4.6.2 Box DR dataset

Similarly to the synthetic box dataset, this dataset uses boxes to differentiate between the normal and abnormal classes. However, this dataset was constructed by us using the normal class from the DR dataset. The normal class remained the same whereas the abnormal class was created by masking out a random part of the eye by applying a white box in the normal images.

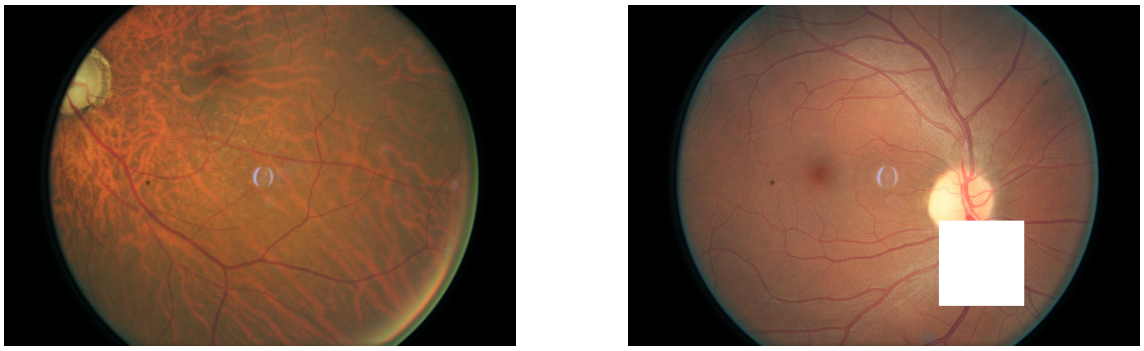


Figure 4.9: An example pair of images from the DR box dataset

This dataset while more complicated due to its RGB nature as well as additional features in the background but still remains as a simple transformation from one class to another.

4.6.3 Synthetic Drusen dataset

This synthetic dataset was similarly created as a proof of concept mocking the features of the Retinal OCT Drusen dataset but to a simplified degree to investigate how the CycleGAN can deal with simple structural information. The normal class is classified as a simple straight line whereas the not normal class has a bump located at the center to mimic the Drusen bumps.

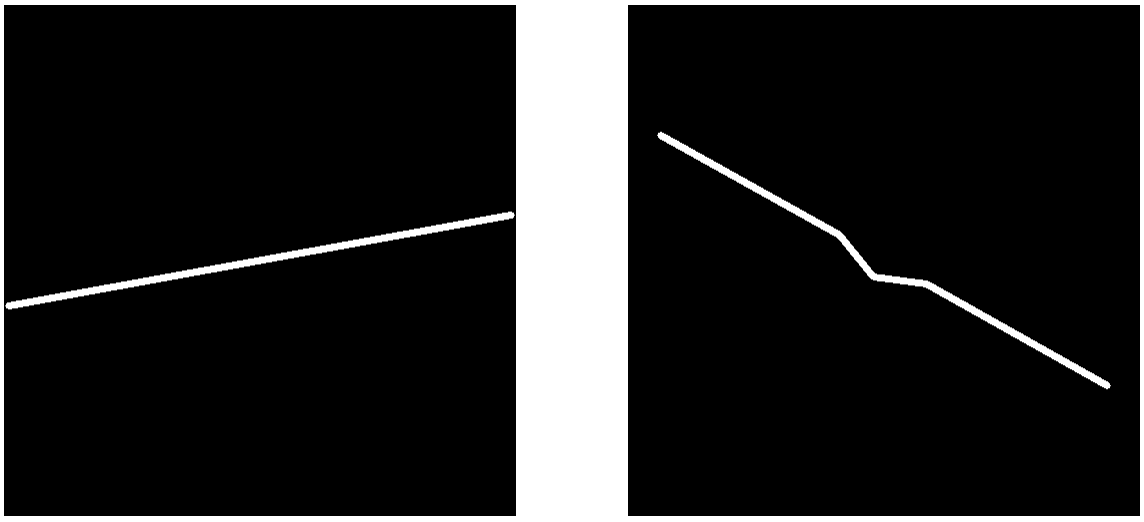


Figure 4.10: An example pair of images from the Synthetic Drusen dataset. The normal class (left) is simply a straight line whereas the not normal class (right) has an additional bump in the center.

4.6.4 Synthetic Box 3D

This dataset is an extension to the existing Synthetic Box dataset made for the 3D version of the CycleGAN method. Stored as a 64 x 64 x 64 numpy files, the normal class consists of a 64 x 64 x 64 black box with a random 16 x 16 x 16 grey box randomly placed. The not normal class consisted the same, but with a smaller random 8 x 8 x 8 white box placed randomly throughout.

Chapter 5

Results & Analysis

In this chapter, we describe the experiments with the previous datasets (Section 4.3 - 4.6) and models mentioned (Section 3.2). We present the generated counterfactuals and evaluate them according to the methods described earlier (Section 3.3).

5.1 RSNA Pneumonia

Starting with the Pneumonia dataset to first recreate the experiment and results by Mertes et al. The generated counterfactuals from the AlexNet Model was in similar quality to the ones presented by in their paper.

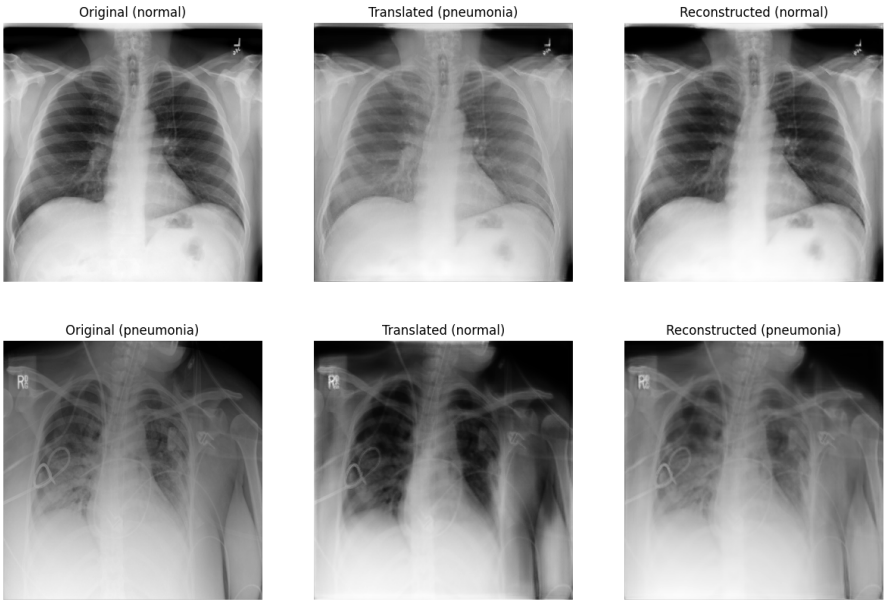


Figure 5.1: The generated counterfactual for the Pneumonia dataset. The left column consists of the original images, the center column the generated counterfactuals by the CycleGAN, and the right column, the reconstructed images.

The generated counterfactuals look visually convincing and similar to the Pneumonia images. As the authors noted, they attribute the great success for this method to the fact that the relevant information is stored texturally. Furthermore, as the images themselves have consistent properties (e.g. chest is always centered of the image on a black background, images are greyscale and therefore don't vary in colour) paired with the fact that the transformation is a simple global transformation that is applying opacity to the center of the image, these conditions allow for the CycleGAN to have easily generate the counterfactuals.

5.2 Diabetic Retinopathy (DR)

To further investigate how image properties can affect the performance of the CycleGAN, we decided to test the DR dataset next. It is a medical dataset with visually apparent features for the severe DR classes. Furthermore, unlike the Pneumonia dataset, it expresses the relevant information in the form of spatial information with local objects occurring within the image. To keep things more simple so we can perform a more fair comparison in terms of number of features and feature complexity to the Pneumonia dataset, we opted to only use the green channel of the image to keep it single channel.

Figure 5.2: Each of the colour channels extracted and colour coded from a DR image.

For this dataset, we used the VGG model to train and attempt to explain. The CycleGAN failed to generate any meaningful explanations for the model.

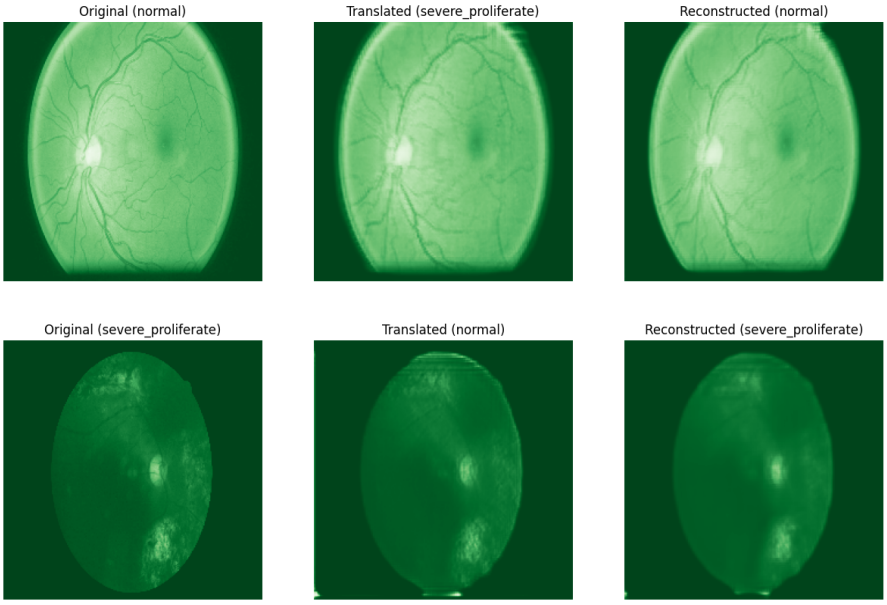


Figure 5.3: The generated counterfactual for the DR dataset. We use a green colour map to reflect the fact that only the green channel of the image was used.

For both classes, the CycleGAN learnt to simply blur the images for the DR images and for the normal images adding some weird blurring artifacts as well. This is most likely due to the fact that the CycleGAN has learnt that blurring the images changes the VGG's models prediction as it obscures the DR features for the DR images and introduces new abnormal features for the normal images.

Given that the relevant information was not expressed texturally, paired with the fact that the spatial information had no consistency (e.g. placement, size, shape), this meant that the CycleGAN was unable to generalise and generate any relevant meaningful features and instead simply blurred the image, or exaggerated the edges of the eye (both of which are global consistent transformations). Using the test set of 159 images for each class, we get the following results:

	Original Class	Translated	Reconstructed
Normal	81.1%	100%	98.7%
Severe Proliferate	56.6%	99.4%	89.3%

Table 5.1: The accuracy of the VGG classifier when given original, translated, and reconstructed images

	Normal	Not Normal
VGG	0.041	0.028

Table 5.2: The mean reconstruction loss for each class

Despite the CycleGAN not generating any meaningful counterfactuals, the classifier still had a good accuracy when given the translated counterfactual images. This could be due to the fact that the classifier performance on the original dataset was relatively subpar and therefore does not know how to classify the given translated images correctly. Given the classifier responded positively to the counterfactuals, the CycleGANs simply continued to blur the images and not learn the correct features that represent DR. Similarly as the reconstructed images were blurrier than the original images, they also scored a higher accuracy when fed to the model.

5.3 Synthetic Box

To test if the CycleGAN was capable of learning spatial transformations had the dataset been more generalisable and less complex, a simple dataset that exhibited consistent spatial transformations. For this dataset, successful counterfactuals were generated for the VGG-19 and AlexNet models. Due to the simplistic nature of this dataset, the more modern and powerful models such as Resnet-50 would overfit too easily even with data augmentation techniques. Both VGG-19 and AlexNet were able to generate successful counterfactuals for both classes.

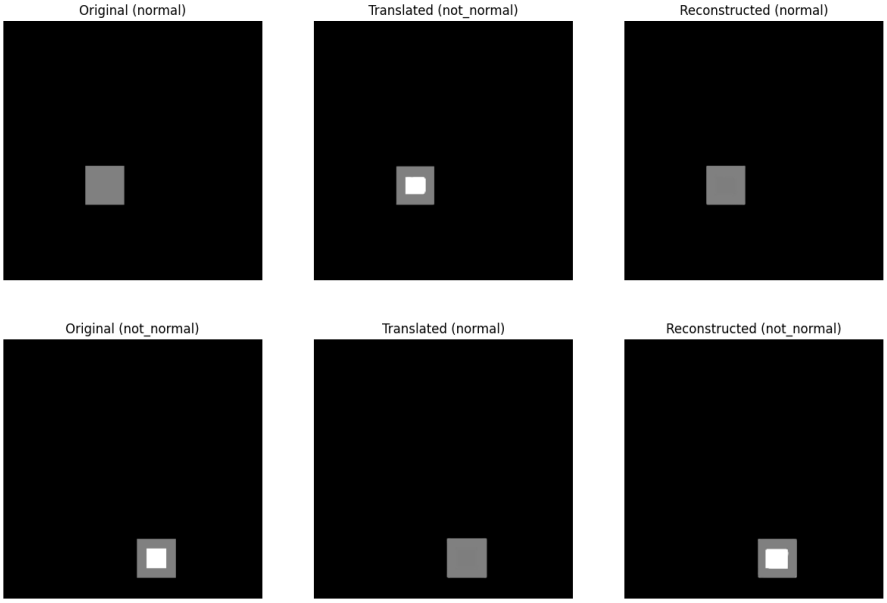


Figure 5.4: A counterfactual generated by the AlexNet Model

As we can see, the CycleGAN was capable of learning the feature that differentiates between the two classes with the CycleGAN even preserving the inner white box shape suggesting that the models does not just learn the classes by the presence of white but furthermore the shape. An interesting result is that the CycleGAN is also capable of learning general transformations that apply on a much smaller localised area signifying that the CycleGAN can learn some localised transformations that also vary in location as long as the placements of the transformed features are well defined.

Given the simplistic nature of the dataset with limited features, data expressed as textural information, as well as a simple transformation between the classes, it was expected that this dataset would achieve the most convincing results. Using the test subset of 200 images for each class to generate counterfactuals and evaluate the two classifiers, we observe the following results:

	Original Class	Translated	Reconstructed
Normal	100%	81%	100%
Not Normal	76%	100%	79.5%

Table 5.3: The accuracy of the AlexNet classifier when given original, translated, and reconstructed images

	Original Class	Translated	Reconstructed
Normal	100%	100%	100%
Not Normal	99%	100%	100%

Table 5.4: The accuracy of the VGG classifier when given original, translated, and reconstructed images

	Normal	Not Normal
AlexNet	0.00011	0.0015
VGG	0.00047	0.0030

Table 5.5: The mean reconstruction loss for each class and model

We can see that the classifiers responded extremely positively to the counterfactuals with the VGG classifier in particular performing the best which is expected based off the quality of the counterfactuals generated for each classifier.

5.4 Synthetic DR Box

Following the success of the Synthetic Box dataset, we then wanted to test this on a slightly more complex level while keeping the transformation the same to see if the same level of success could be replicated. For the Synthetic DR Box dataset, relatively successful looking counterfactuals were generated by the AlexNet model. Unlike the simple Synthetic Box dataset, there were additional features present such as the optical disc and artifacts such as the light bleeding that exists on some images. Furthermore, as the placement of the box was applied randomly rather than

following certain features such as the Synthetic Box dataset, the counterfactuals produced has some interesting placements.

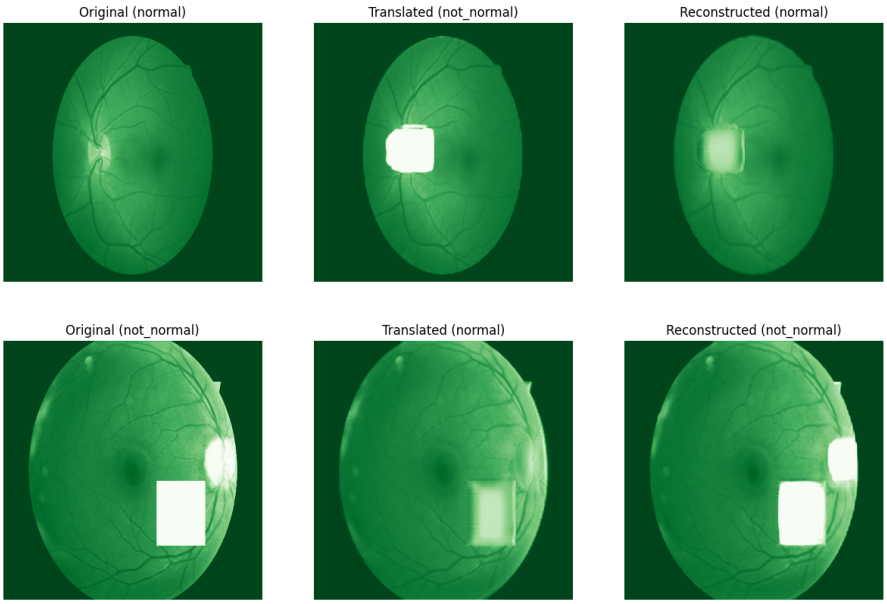


Figure 5.5: A counterfactual generated for the VGG Model. A green colour map was used to reflect the fact that these images were trained and generated solely from the green channel. The generated counterfactuals (middle) created/filled in the box for the normal and not normal class respectively.

From these images, we can see that the CycleGAN, while capable of learning general transformations, are incapable of learning anything more complex. For example, we can see that the CycleGAN has learnt the properties and features of the box, but nothing beyond that as it is incapable of restoring the features that were omitted due to the box masking. We can interpret the presence of the whiteness within the filled box for the translated not normal class as an attempt to recreate the optical disc as we can see in the reverse case, the CycleGAN has learnt to apply the white box at the location of the optical disc.

As this dataset similarly reflects the properties of the synthetic dataset, it is also expected for reasonable counterfactuals to be generated. Using the testing subset of 50 images of each class to generate counterfactuals, the classifiers give us the following results:

	Original Class	Translated	Reconstructed
Normal	98%	96%	100%
Not Normal	84%	100%	84%

Table 5.6: The accuracy of the AlexNet classifier when given original, translated, and reconstructed images

	Original Class	Translated	Reconstructed
Normal	100%	98%	98%
Not Normal	100%	100%	100%

Table 5.7: The accuracy of the VGG classifier when given original, translated, and reconstructed images

	Normal	Not Normal
AlexNet	0.033	0.026
VGG	0.027	0.023

Table 5.8: The mean reconstruction loss for each class and model

The results show that both models responded positively to the counterfactuals. The counterfactuals generated for the AlexNet model did not recreate the white box but instead exaggerated any areas of whiteness present in the image (see Appendix A.3). The counterfactuals generated for the VGG model on the other hand was able to reproduce the white box which suggests that the shape of the 'white box' generated by the CycleGAN plays a significantly smaller role in what the AlexNet classifier defines the classes to be.

5.5 Retina OCT - DME

To finalise the testing of spatial information, we opted for another real medical dataset that exhibits spatial information but with a lower level of complexity and better generalisability compared to DR. The DME dataset from the Retinal OCT dataset had some promising results. While being a dataset that uses spatial information like DR, this dataset on the other hand is a lot more simpler and consistent which allows for better generalisation across the entire dataset. Unlike previous datasets where all the models were able to succeed to some degree, the CycleGAN was only able to produce explanations for AlexNet. This is most likely due to the fact that AlexNet was the least powerful models of all the trained models and therefore had to generalise more which allowed for more generic counterfactuals to influence the classifier's decision.

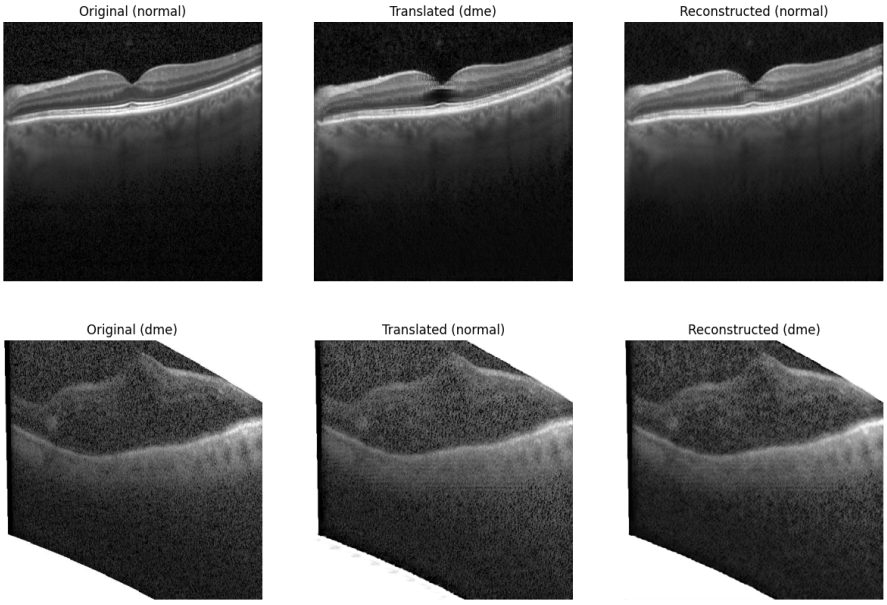


Figure 5.6: A counterfactual generated by the AlexNet Model. The CycleGAN created a tear in the retina for the counterfactual of the normal image (middle)

We can see that a tearing has been made in the retina for the translated normal class which mostly accurately reflects the features and properties of DME. The tearing added by the CycleGAN is always added to the center of the eye with a singular tear because the CycleGAN is only capable of learning general global transformations and as a result, is unable to add any variety to the tearing such as quantity, size or location. Generating counterfactuals on the test subset of 242 images for each class does not yield any surprising results.

	Original Class	Translated	Reconstructed
Normal	93.4%	0.83%	75.6%
Not Normal	93.8%	0%	100%

Table 5.9: The accuracy of the AlexNet classifier when given original, translated, and reconstructed images

	Original Class	Translated	Reconstructed
Normal	100%	0%	100%
Not Normal	92.9%	4.55%	95.9%

Table 5.10: The accuracy of the VGG classifier when given original, translated, and reconstructed images

	Original Class	Translated	Reconstructed
Normal	98.8%	0%	94%
Not Normal	95.9%	0%	97.5%

Table 5.11: The accuracy of the ResNet classifier when given original, translated, and reconstructed images

	Normal	Not Normal
AlexNet	0.037	0.041
VGG	0.092	0.039
ResNet	0.034	0.044

Table 5.12: The mean reconstruction loss for each class and model

We can see that the results are what we expect with poor accuracy on the translated images as we the translated images are mostly if not identical to the regular images. We can also see that the AlexNet translated images did poorly despite generating the most convincing counterfactuals which suggests that just because the counterfactual could visually explain some features of DME, if it doesn't perfectly recreate all the features the classifiers will not give the prediction we are looking for. This can be seen with the convincing tear in the generated counterfactual for the AlexNet model yet poor accuracy when given these counterfactuals suggesting that it was insufficient to influence and change the classifier's decision.

5.6 Retina OCT - Drusen

Investigating the third kind of information, structural information, we decided to test and train a classifier on the OCT Drusen dataset. Unlike the previous dataset, the relevant information is expressed using already existing features, being the membrane of the retina, but with some modifications to its structure. A ResNet model was trained and the CycleGAN was unable to produce any meaningful counterfactuals.

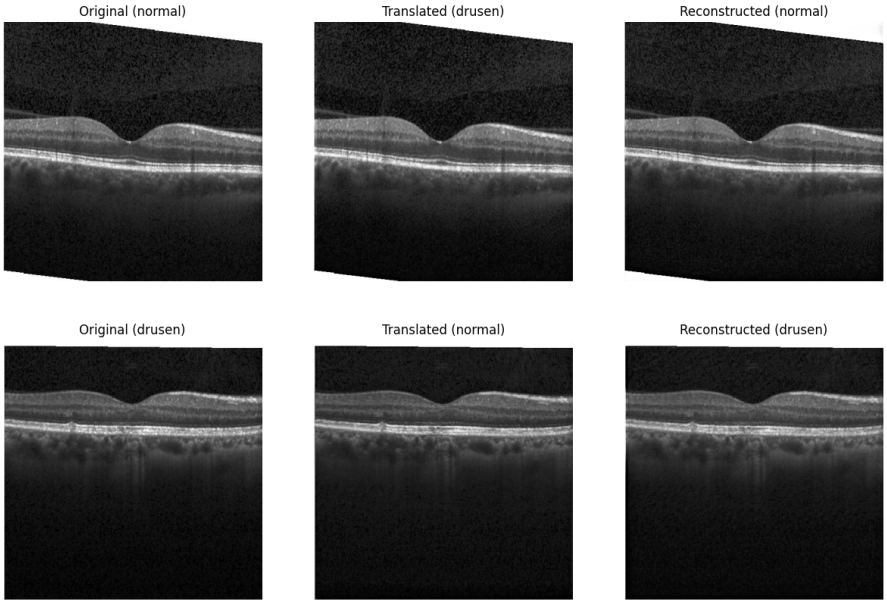


Figure 5.7: A failed counterfactual generated by the ResNet Model

Unlike the other failed counterfactuals from the DR dataset, the CycleGAN essentially did nothing here. Other than some extremely minor modifications made to the background noise, the CycleGAN has learnt nothing. This was to be expected as the CycleGAN is incapable of learning transformations involving structural information other than in the most simple cases. The inconsistent placement of the retina within the image such as the retina of the image (e.g. top row of images are placed center unlike the bottom images) paired with varying angles of the retina (see Appendix A) making the dataset more deceptively easy than it appears to be. Not only that, the placement of the bumps of the Drusen are inconsistent and random, while still having more constraints on where these bumps can be, makes this problem similar to the DR dataset. These combinations makes the generalisation of what Drusen features extremely hard to generalise to a simple transformation and therefore render the CycleGAN incapable of learning complex transformations that involve structural information. If we limit the variability of the features of the dataset and limit the complexity of the dataset, the CycleGAN is capable of learning simple structural transformations as we will see with the Synthetic Drusen dataset. Generating counterfactuals for the test set of 242 images for each class yielded the following results:

	Original Class	Translated	Reconstructed
Normal	100%	0%	99.6%
Not Normal	83.5%	8.26%	85.1%

Table 5.13: The accuracy of the AlexNet classifier when given original, translated, and reconstructed images

	Original Class	Translated	Reconstructed
Normal	90.1%	0%	95.5%
Not Normal	100%	0%	100%

Table 5.14: The accuracy of the VGG classifier when given original, translated, and reconstructed images

	Original Class	Translated	Reconstructed
Normal	100%	0%	100%
Not Normal	89.2%	0.4%	97.9%

Table 5.15: The accuracy of the ResNet classifier when given original, translated, and reconstructed images

	Normal	Not Normal
AlexNet	0.036	0.045
VGG	0.033	0.043
ResNet	0.028	0.028

Table 5.16: The mean reconstruction loss for each class and model

Similar to the DME dataset, the CycleGAN was completely incapable of generating any meaningful counterfactuals but for all 3 models. The models predict the same for most if not all translated images which is to be expected which is supported by the metrics above.

5.7 Synthetic Drusen

To see if the CycleGAN are incapable of learning all types of structural transformations, a new synthetic dataset mimicking the features of Drusen was created whilst simplifying all the other features. The CycleGAN was capable of creating decent explanations for the VGG model.

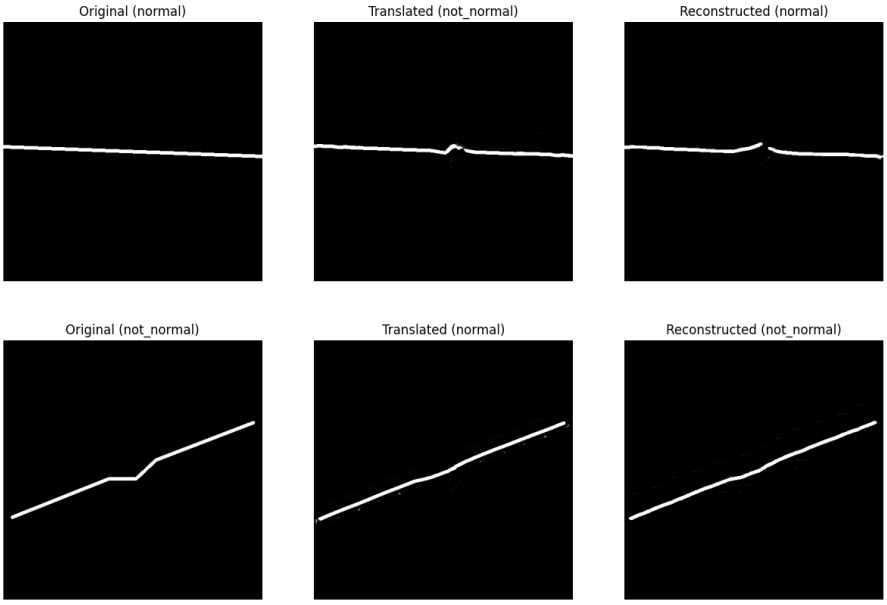


Figure 5.8: One of the best counterfactuals generated by the VGG Model

We can see that while neither explanations are perfect, the CycleGANs do transform the images in a way that closely resembles the translated class well enough. The reasoning for why the CycleGAN can perform well on this specific dataset is most likely due to the fact that we have turned the Drusen transformation from a random local transformation to a consistent global transformation - for all images, the bump is always located at the exact center of the image. Using the test set of 100 images for each class, we observed the following results:

	Original Class	Translated	Reconstructed
Normal	100%	46%	100%
Not Normal	87%	100%	65%

Table 5.17: The accuracy of the AlexNet classifier when given original, translated, and reconstructed images

	Original Class	Translated	Reconstructed
Normal	100%	97%	94%
Not Normal	100%	100%	85%

Table 5.18: The accuracy of the VGG classifier when given original, translated, and reconstructed images

	Normal	Not Normal
AlexNet	0.0049	0.0067
VGG	0.0067	0.012

Table 5.19: The mean reconstruction loss for each class and model

We can see that the VGG model responded the most positive despite the fact that not all of the transformations made by the CycleGAN were what defined the not normal class, a singular bump in the center, (see appendix A.1) which suggests that what the VGG defined the not normal class features to be any bumps regardless of location and quantity.

The AlexNet model on the other hand performed worse for the translated normal class which was to be expected as the counterfactual instances for the AlexNet classifier were of worse quality compared to VGG's. The difference in performance between these models is due to the cycleGAN not working as well for AlexNet rather than the cycleGAN training process failing as show by the similar mean reconstruction loss for the two models.

5.8 Summary

From our various testing across all types of datasets expressing all types of information - texturally, structurally, and spatially, we can observe that the CycleGAN not only has affinity toward some types of information more than others, but additionally the features and properties of the datasets and the models used can also influence and affect the quality of the generated counterfactuals.

The CycleGAN struggles the most with structural information for any dataset that has any moderate level of complexity in terms of features as seen with the Drusen (Section 5.3) but with a simple enough set of features can still do some structural modification to a certain degree as seen by the Synthetic Drusen (Section 5.4).

This similarly applies to spatial information as seen with DR (Section 5.2) where the CycleGAN was incapable of learning any transformation for this particular dataset due to its large complex number of features and variation in how the transformation could be applied. In contrast, the Synthetic DR box dataset shows that simplifying the transformation enough allows the CycleGAN to learn the transformation and even recreate some features albeit to a minimal degree. Not only that, it is capable of learning transformations that rely on local information and apply them to specific relevant regions meaning this method is also capable of working with transformations that take place on a smaller scale (rather than the global transformation on the whole image as was the Pneumonia dataset). We see this best with the Synthetic Box dataset (Section 5.6), where with strict set of features and transformations were defined, the CycleGAN was capable of not only applying the correct transformation but even preserving other details such as the shape and size of the box in addition to the colour.

There can also be a compromise between not learning any transformations and learning a perfect transformation, and instead have an imperfect, but understandable transformation as we could see with DME (Section 5.5). We can see that with DME that the CycleGAN has learnt a transformation that partially reflects the nature of DME as the CycleGAN can only learn a generalised transformation across the entire dataset. Notably, only the AlexNet model was consistently explainable, where the visual tearing produced for most if not all images unlike the other two models where very occasionally faint tearing can be seen. This is most likely due to the weaker capabilities of the AlexNet model having to generalise more and therefore having more meaningful feedback and influence on the training process of the CycleGAN compared to the other two models. Note that despite an imperfect but visible explanation was created for the AlexNet model, the AlexNet model still responded poorly to the generated counterfactuals which show that while imperfect explanations may be satisfactory for humans, this will not be the case for the classifier.

From the results generated for the Pneumonia and Synthetic Box dataset (Section 5.1, 5.6), the CycleGAN was most capable of reconstructing the relevant features as they were stored in a textural format which the CycleGAN operates best with.

Chapter 6

3D Extension

In this chapter, we go into depth explaining our 3D implementation and how this differs from the original method. We also share some results generated from our 3D implementation for our 3D Synthetic Box dataset.

6.1 3D CycleGAN Architecture

For the 3D extension, the CycleGAN must first be extended to support 3D images with an additional dimension (Depth), images with dimension, *(Height, Width, Depth, Channels)*. To support this, we can preserve the architecture structure and simply modify the layers to take an additional dimension and this is done by replacing instances of 2D specific layers, such as Pooling layers, with their 3D counterpart. We also need to modify the input size as well.

Some other modifications that are required to support 3D images is in the nature of how 3D images are stored compared to 2D. While existing ML libraries such as Keras [30] have lots of support for directly loading 2D images with common formats such as PNGs or JPEGs, 3D images are stored in unconventional formats such as Numpy (npy), NIFTI or DICOM which require additional implementation to use these image formats. As a result, a custom dataloader is needed that supports npy dataset that will be used to train the CycleGAN and classifier.

6.2 3D Classifier

In addition to extending the original CycleGAN architecture to support 3D images, we also need a 3D classifier to not only generate counterfactuals for but to also incorporate in the training process of the CycleGAN. 3D Classifiers have already been extensively researched and implemented meaning implementing one is fairly simple. For, this experiment, we implemented the 3D-CNN classifier proposed in the MICCAI'2020 PRIME workshop paper [33] with minor modifications to the output layer to output prediction probability for each class rather than just the class as the outputs were needed for the CycleGAN's training process.

6.3 3D Image Loader

Unlike 2D images where we can view the entirety of the image in one go and do a direct comparison, we could only do a comparison of the 3D images on a slice by slice (layer) basis. A 3D image comparison tool was built using Matplotlib [37] which allowed to load 3 3D images at once (the original, translated, and reconstructed image) with synchronised slice viewing to allow for direct comparisons.

6.4 3D Synthetic Box Dataset

To test our 3D Extension, we decided to recreate the Synthetic Box dataset as a 3D dataset as we wanted to see if the concept of a CycleGAN would also extend and work for 3D images. To test this proof of concept, a simple dataset with a simple global visually obvious transformation would be best as a baseline as if this method does not work for a simple dataset, then the implementation will also unlikely work for anything more complex.

6.5 Results

We evaluate the generated counterfactuals in a similar manner as defined previously (Section 3.3).

The 3D counterfactuals generated by the 3D GANterfactual did not perform as well as the 2D dataset. While the correct feature (white box) was able to created in the correct location (inside the grey box), the other properties of the correct feature (size of box) as well as other features of the dataset such as keeping the outer grey box was not correctly preserved. In addition, the reconstructed images failed to resemble the original images as well. For the not normal class, while successful in getting rid of the feature that defines the not normal class (the white box) the CycleGAN failed to generate counterfactuals with any resemblance to the target class.

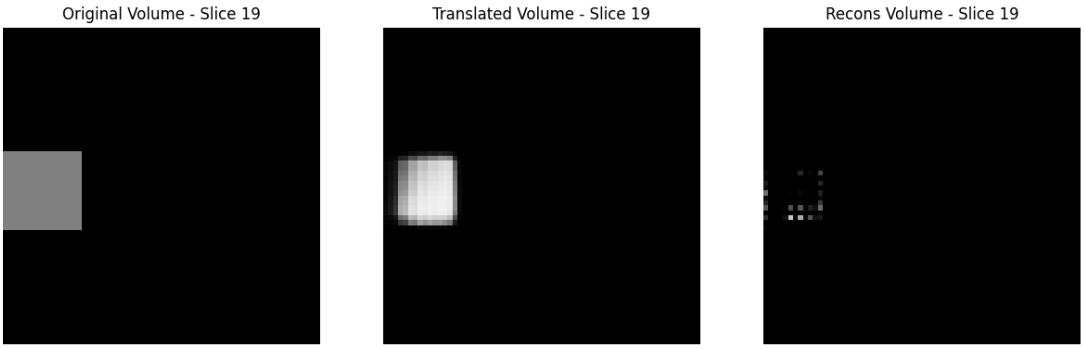


Figure 6.1: An example counterfactual generated for normal image for the 3D-CNN Model

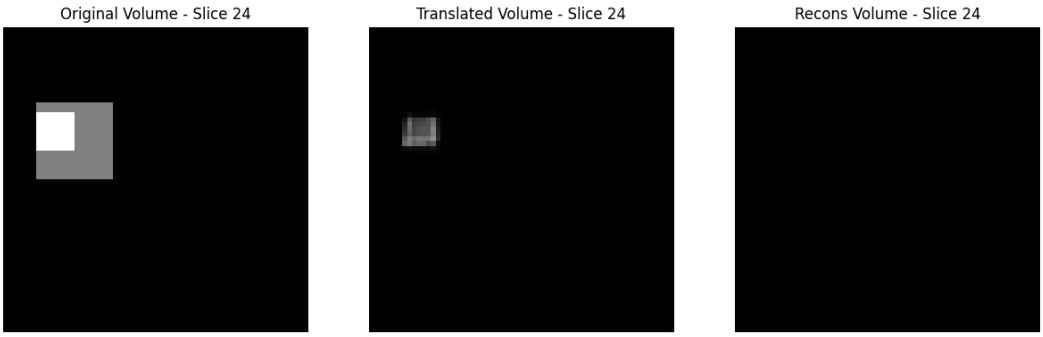


Figure 6.2: An example counterfactual generated for not normal image for the 3D-CNN Model

Using the test subset of 25 images to generate counterfactuals and feed to our classifier, we observe the following results:

	Original Class	Translated	Reconstructed
Normal	100%	100%	0%
Not Normal	100%	96%	100%

Table 6.1: The accuracy of the 3D-CNN classifier when given original, translated, and reconstructed images

	Normal	Not Normal
3D-CNN	0.016	0.018

Table 6.2: The mean reconstruction loss for each class

We can see that the normal reconstruction was poor which was expected as the reconstructions did not resemble the original class at all. We also can see that the translated image accuracy was high for both classes which indicates that the classifier uses the presence/absence of the white box when predicting the class irrespective of the shape and size.

Chapter 7

Discussion

This chapter briefly recounts our findings and contributions (Section 5, 6) as well as discusses possible avenues for future work. We also cover some possible ethical concerns due to the nature of the work involving counterfactuals.

7.1 Summary

We have examined how the properties of a dataset and a model's architecture can impact the quality of its generated outputs. For datasets that express relevant information texturally or spatially as simple general transformations (i.e. the counterfactual class can be represented as two layers consisting of the original image as one layer, and the transformation that the cycleGAN applies as the second layer), the cycleGAN produces meaningful results. This can be seen in the Synthetic Box, Pneumonia and DME datasets (Section 5.1, 5.5, 5,6). The differences in performance between these datasets is likely due to being able to construct the Pneumonia and Synthetic Box examples through one simple consistent transformation across all images, whereas DME can be described as a singular but varying transformation. The DR and Drusen datasets both require several complex transformations to be applied or the modification of the original first layer, respectively, which the cycleGAN fails to learn. However, we have also demonstrated that if the dataset and transformation is simple enough, the CycleGAN is capable of learning structural transformations as shown with the Synthetic Drusen dataset (Section 5.4).

Furthermore, we extend this state of the art CycleGAN method to support 3D images and demonstrate its functionality on the 3D Synthetic Box dataset (Section 6)

7.2 Future Work

7.2.1 Improving the original CycleGAN method

We note that one of the shortcomings of this method is that the CycleGAN method struggles for datasets that have large variety of feature placements and local rather than global transformations. However, as seen by the Synthetic Box dataset and Synthetic Drusen dataset, if we reduce the complexity and variation of the dataset enough, the CycleGAN is capable of learning transformations that are not global

and textural. It would be interesting to see if either modifications to the CycleGAN architecture, or other methods such as preprocessing can help it perform better on these kind of datasets.

7.2.2 Investigating Real 3D Datasets

Our project has limited investigation into the realm of 3D. In medical settings, the use case of 3D images is much more common and widespread. As such, testing on real 3D datasets to see if the 3D extension has similar capabilities would be next. Furthermore, should this be the case, investigating if image properties that affect counterfactual quality for 2D will still similarly hold for 3D images.

7.3 Ethical Consideration

As this research was conducted on publicly available datasets, this project is exempt from ethical approval. All datasets procured and used have already been processed and anonymised appropriately by the companies who submit to open source websites such as Kaggle as this is required by Kaggle.

This project focuses on the work of Generative AI and specifically counterfactuals. While counterfactuals are used here as a form of XAI in order to explain and reason a classifier's decision to foster trust in the users and adopters, it is possible to use counterfactuals for more nefarious purposes such as creating fake and misleading content.

Appendix A

Extra Counterfactual Images

This chapter contains some extra counterfactual images generated that couldn't fit into the results chapter without obscuring the results. We include some interesting and other types of imperfect counterfactuals that were generated by the CycleGAN.

A.1 Synthetic Drusen

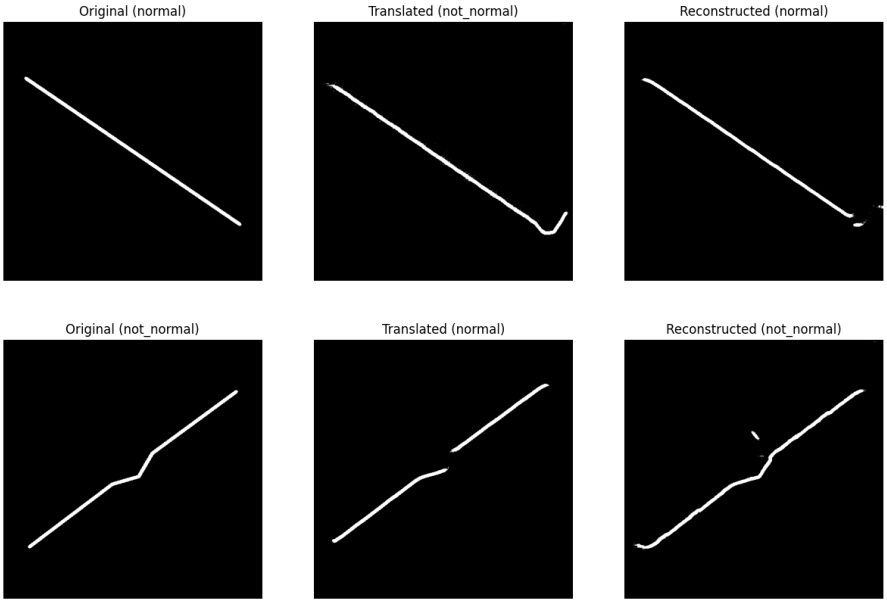


Figure A.1: An example of an imperfect generated counterfactual (middle) for the AlexNet model. For the not normal counterfactuals, removing the bump is enough to change the model's prediction without having to fill in the missing gap.

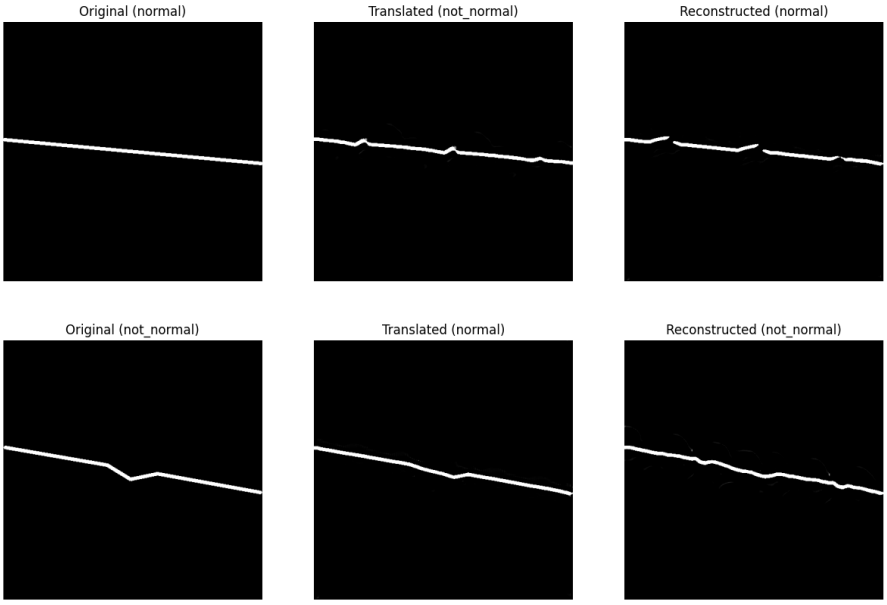


Figure A.2: A different transformation used when generating a counterfactual for the VGG model. Note how the translated images (middle column) do not perfectly replicate the features from the original image (left column) but is sufficient to change the VGG model's prediction indicating that some features such as positioning or quantity of the bumps do not impact the classifier's decision

A.2 Drusen

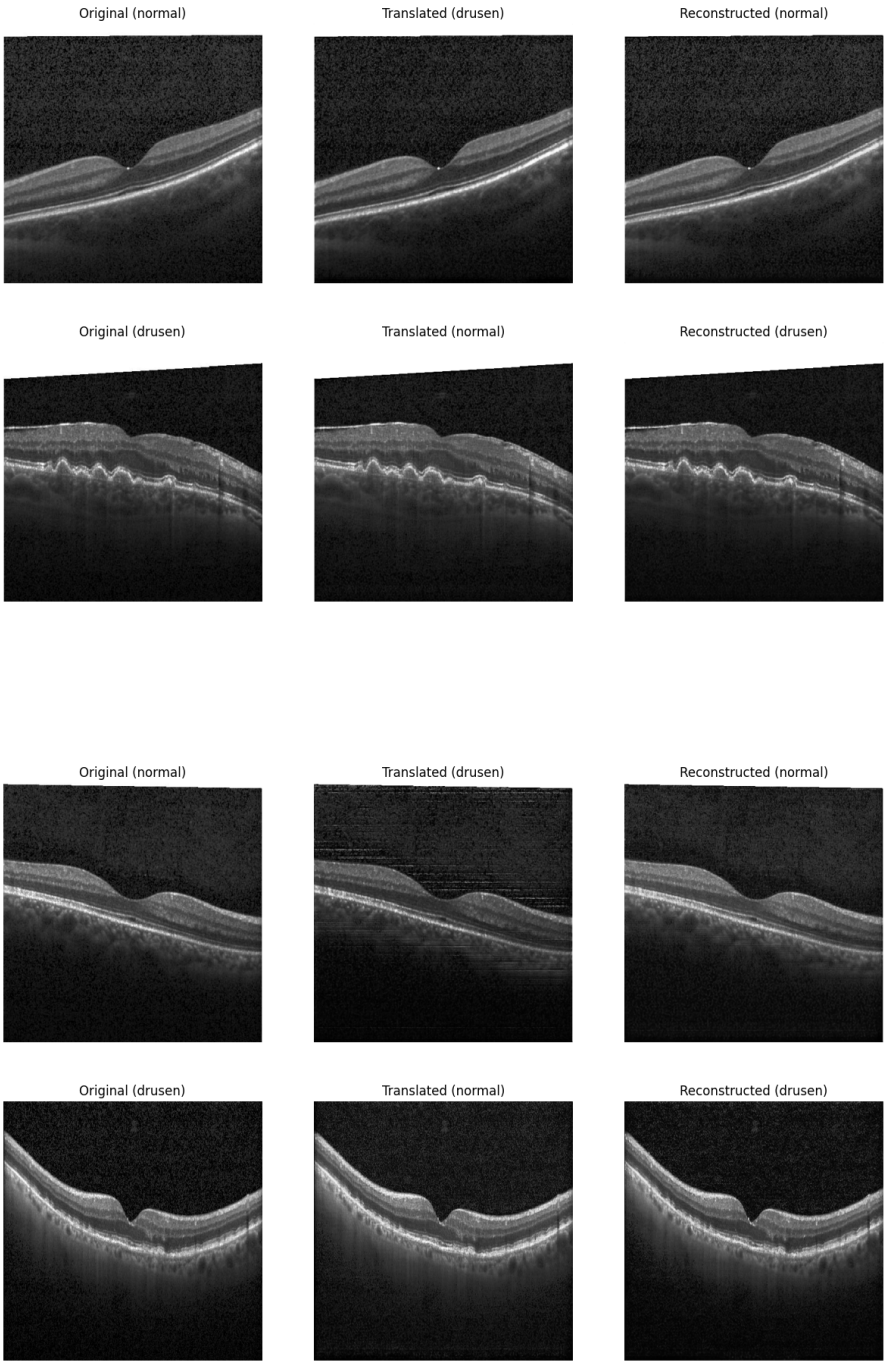


Figure A.3: Some examples of Drusen images. Note how the image properties such as positioning, orientation, quantity of Drusen bumps vary a lot resulting in the CycleGAN struggling to generate any meaningful counterfactuals (middle).

A.3 DR BOX

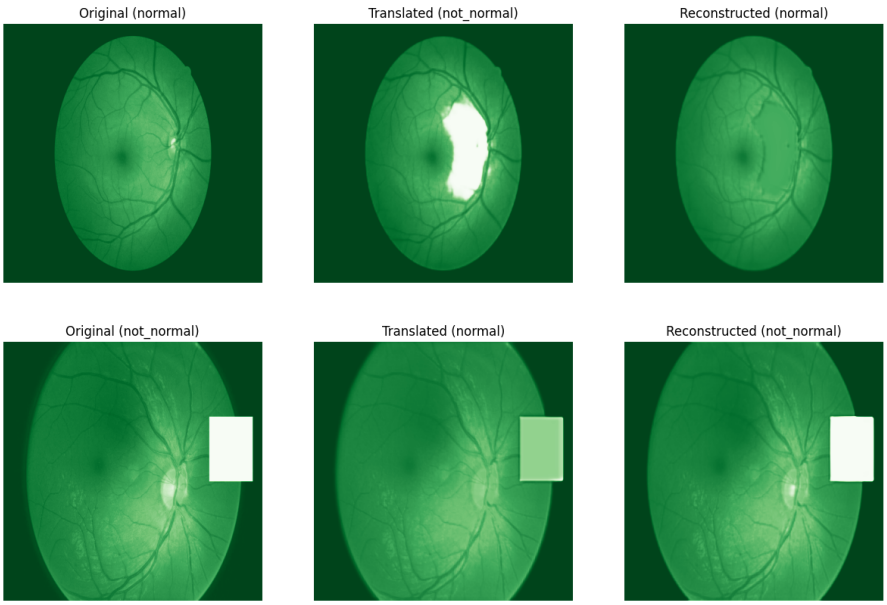


Figure A.4: A counterfactual generated for the AlexNet model on the DR Box dataset. Unlike the VGG counterfactual, no white box was introduced but this was sufficient to change the AlexNet model's prediction.

Bibliography

[1] Relić D Džakula A. "Health workforce shortage – doing the right things or doing things right?" In: *Croatian Medical Journal* 63.2 (Apr. 2022), pp. 107–9. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9086817/.

[2] Constance D. Lehman et al. "Diagnostic Accuracy of Digital Screening Mammography With and Without Computer-Aided Detection". In: *JAMA Internal Medicine* 175.11 (Nov. 2015), p. 1828.

[3] Ardila D et al. "End-to-End Lung Cancer Screening with Three-Dimensional Deep Learning on Low-Dose Chest Computed Tomography". In: *Nature Medicine* 25.6 (2019), pp. 954–961. URL: https://www.nature.com/articles/s41591-019-0447-x.

[4] Tang X. "The Role of Artificial Intelligence in Medical Imaging Research". In: *BJR|Open* 2.1 (Nov. 2020), p. 20190031. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7594889/.

[5] Singh RP et al. "Current Challenges and Barriers to Real-World Artificial Intelligence Adoption for the Healthcare System, Provider, and the Patient". In: *Translational Vision Science & Technology* 9.2 (Jan. 2020), pp. 45–55. URL: https://tvst.arvojournals.org/article.aspx?articleid=2770632.

[6] Ramprasaath R. Selvaraju et al. *Grad-cam: Visual explanations from deep networks via gradient-based localization*. Dec. 2019. URL: https://arxiv.org/abs/1610.02391.

[7] Ribeiro MT, Singh S, and Guestrin C. ""Why Should I Trust You?"". In: (2016). DOI: 10.1145/2939672.2939778.

[8] Ruth M. Byrne. "Counterfactuals in explainable artificial intelligence (XAI): Evidence from human reasoning". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence* (2019). DOI: 10.24963/ijcai.2019/876.

[9] Jun-Yan Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *arXiv.org* (Aug. 2020), pp. 1–2. URL: https://arxiv.org/abs/1703.10593.

[10] Ahmad Chaddad et al. *Survey of explainable AI techniques in Healthcare*. Jan. 2023. URL: https://doi.org/10.3390/s23020634.

[11] Keiron O'Shea and Ryan Nash. "An introduction to Convolutional Neural Networks". In: *arXiv.org* (Dec. 2015), pp. 2–5. URL: https://arxiv.org/abs/1511.08458.

[12] Manolis Loukadakis, José Cano, and Michael O'Boyle. "Accelerating Deep Neural Networks on Low Power Heterogeneous Architectures". In: Jan. 2018.

[13] Ian J. Goodfellow et al. *Generative Adversarial Networks*. June 2014. URL: https://arxiv.org/abs/1406.2661.

[14] Erik Linder-Norén. *Eriklindernoren/Pytorch-Gan: Pytorch implementations of generative adversarial networks*. URL: https://github.com/eriklindernoren/PyTorch-GAN.

[15] Cristian Munoz et al. *Local and global explainability metrics for machine learning predictions*. Feb. 2023. URL: https://arxiv.org/abs/2302.12094.

[16] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. *Explainable AI: A Review of Machine Learning Interpretability Methods*. Dec. 2020. URL: https://www.mdpi.com/1099-4300/23/1/18.

[17] Sarp S;Catak FO;Kuzlu M;Cali U;Kusetogullari H;Zhao Y;Ates G;Guler O; *An XAI approach for COVID-19 detection using transfer learning with X-ray images*. Apr. 2023. URL: https://pubmed.ncbi.nlm.nih.gov/37041935/.

[18] Vitali Petsiuk, Abir Das, and Kate Saenko. *Rise: Randomized input sampling for explanation of black-box models*. Sept. 2018. URL: https://arxiv.org/abs/1806.07421.

[19] Yingxue Pang et al. *Image-to-image translation: Methods and applications*. July 2021. URL: https://arxiv.org/abs/2101.08629.

[20] Phillip Isola et al. *Image-to-image translation with conditional adversarial networks*. Nov. 2018. URL: https://arxiv.org/abs/1611.07004.

[21] Chu-ran Wang et al. *Bilateral asymmetry guided Counterfactual Generating Network for Mammogram Classification*. Sept. 2020. URL: https://arxiv.org/abs/2009.14406.

[22] Silvan Mertes et al. *Ganterfactual-counterfactual explanations for medical non-experts using generative adversarial learning*. Mar. 2022. URL: https://www.frontiersin.org/articles/10.3389/frai.2022.825565/full.

[23] 2018. URL: https://www.rsna.org/rsnai/ai-image-challenge/rsna-pneumonia-detection-challenge-2018.

[24] Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. *Diffusion models for counterfactual explanations*. Mar. 2022. URL: https://arxiv.org/abs/2203.15636.

[25] Daniel Nemirovsky et al. *Countergan: Generating realistic counterfactuals with residual generative adversarial nets*. May 2021. URL: https://arxiv.org/abs/2009.05199.

[26] Silvan Mertes. *HCMLAB/Ganterfactual: Generating counterfactual explanation images through generative adversarial learning*. 2020. URL: https://github.com/hcmlab/GANterfactual.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[28] Yann LeCun et al. Nov. 1998. URL: http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf.

[29] Xiaobing Han et al. "Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification". In: *Remote Sensing* 9.8 (2017). ISSN: 2072-4292. DOI: 10.3390/rs9080848. URL: https://www.mdpi.com/2072-4292/9/8/848.

[30] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[31] Karen Simonyan and Andrew Zisserman. *Very deep convolutional networks for large-scale image recognition*. Apr. 2015. URL: https://arxiv.org/abs/1409.1556.

[32] Luqman Ali et al. "Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures". In: *Sensors* 21 (Mar. 2021), p. 1688. DOI: 10.3390/s21051688.

[33] Hasib Zunair et al. "Uniformizing Techniques to Process CT Scans with 3D CNNs for Tuberculosis Prediction". In: *International Workshop on PRedictive Intelligence In MEdicine*. Springer. 2020, pp. 156–168.

[34] Emma Dugas et al. *Diabetic retinopathy detection*. 2015. URL: https://kaggle.com/competitions/diabetic-retinopathy-detection.

[35] Daniel Kermany, Michael Goldbaum, and Kang Zhang. *Labeled optical coherence tomography (OCT) and chest X-ray images for classification*. Jan. 2018. URL: https://data.mendeley.com/datasets/rscbjbr9sj/2.

[36] Daniel S. Kermany et al. "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning". In: *Cell* 172.5 (2018), 1122–1131.e9. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2018.02.010. URL: https://www.sciencedirect.com/science/article/pii/S0092867418301545.

[37] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.