Universiteit Leiden

Master Computer Science

f-AnoGAN with Transformers and Vector Quantisation for unsupervised anomaly detection in head CTs

Name:	Vasiliki Kogia
Student ID:	s2954699
Date:	13/01/2024
Specialisation:	Artificial Intelligence
1st supervisor:	Niki van Stein
2nd supervisor:	Anna Kononova
External adviser:	Alexander Zeiser

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Universiteit Leiden

Contents

1 Introduction **2**

2 Related Work **5**

3 Background Knowledge **9**

3.1 Generative Modelling 9

3.2 Generative Adversarial Network 10

3.3 Anomaly Detection 11

3.4 f-AnoGAN 12

3.5 Learning Vector Quantisation 15

3.6 Transformer 19

3.7 Evaluation metrics 20

4 Methodology **22**

5 Data Set **24**

6 Experiments **25**

6.1 Data Set 25

6.2 Final Experiments 30

7 Results **35**

8 Discussion **41**

9 Conclusion **43**

References **45**

Abstract

Constant technological development and need for anomaly detection in vast amounts of images demand the usage of more than just human input. Machine Learning and, specifically, Generative Adversarial Networks (GANs) have been a standard and effective way to encounter this problem. There are some state of the art methods that have been introduced to computer vision tasks in the recent years and enable GANs to achieve even better results when combined with each other. This study focuses on high quality images and utilising such methods - Vector Quantisation (VQ) and Transformers - in combination with f-AnoGAN for anomaly detection in three different experiments. Synthetic images (generated by GANs and by manual image analysis) are used for the training process. The experiments that involve VQ produced unstable results with average evaluation metric values (accuracy, precision, recall and F1-score) of 45%, whereas the experiment involving solely a Transformer method achieved an average of approximately 85%. Further research needs to be made to improve the performance of the VQ experiments and even provide better results by focusing on its implementation's weaker points.

1 Introduction

The constant advancements in technology and data acquisition in various fields, including medicine, astronomy and mass production, often require human scrutiny to detect anomalies and relevant patterns in a vast number of images [1]. However, this manual process proves to be inefficient due to its time-consuming nature and the substantial allocation of resources it demands. These challenges arise a pressing need for innovative solutions that can consolidate and accelerate the image testing process while maintaining accuracy and reliability. Such solutions include image analysis technologies [2] and generative algorithms [3] which, combined, can provide scientists and industries with opportunities to revolutionise the way visual data are examined and interpreted.

Generative adversarial networks (GANs) were first introduced in 2014 [4]. They comprise a combination of generative modelling [3] and deep learning [5] methods, such as Convolutional Neural Networks (CNNs) [6]. The core idea of a GAN is based on the "indirect" training through the discriminator, another neural network that can tell how "realistic" the input seems, which itself is also being updated dynamically. This means that the generator - a neural network that aims at creating realistic synthetic images - is not trained to min-

imise the distance to a specific image and create an extra realistic output, but rather to fool the discriminator. This enables the model to learn in an unsupervised manner. Generative modelling is an unsupervised learning task of machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original data set. In other words, a generative model describes how a data set is generated in terms of a probabilistic model and new data is generated by sampling from the model.

GANs are a smart way of training a generative model by framing a supervised learning problem with two sub-models: the generator model that is trained to generate new examples, and the discriminator model that tries to classify examples as either real (from the domain) or imitated (generated). The two models are trained together in a zero-sum game adversarially until the discriminator model is deceived multiple times, meaning the generator model is generating plausible examples. GANs are an exciting and rapidly changing field, delivering on the promise of generative models in their ability to generate realistic examples across a range of problem domains, most notably in image-to-image translation tasks [7] such as translating photos of summer to winter or day to night, and in generating photo-realistic photos [8] of objects, scenes, and people that even humans cannot tell are imitations.

As a form of unsupervised learning algorithm, GANs have been widely used in anomaly detection [9, 10, 11, 12]. Anomaly detection involves recognising uncommon elements, occurrences or patterns within a data set that differ from the usual or expected characteristics. It can be useful to solve many problems including fraud detection [13], medical diagnosis [14], potential risks [15], control failures [16], business opportunities [17], etc. Machine learning methods allow to automate anomaly detection and make it more efficient, especially when large data sets are involved. One of the crucial aspects of GANs for computer vision lies in their potential application and synergy with anomaly detection. By treating anomaly detection as an unsupervised learning problem utilising GANs, questions emerge regarding the possibility of generating high-quality images and their compatibility with state of the art techniques. The experiments conducted for this study include such methods to achieve the desired goals. The main focus of this study will be answering the following questions:

- Is it possible to use and generate high quality images using GAN models for anomaly detection?
- Could state of the art models such as Transformers [18] and Vector Quantised (VQ) technology [19] be useful in combination with GANs?

2 Related Work

The methods mentioned above are relatively recent. GANs have been discussed and studied extensively and important and effective generative models have been created [20]. One of them is f-AnoGAN [21], a GAN-based unsupervised learning approach capable of identifying anomalous images and image segments, that outperforms alternative approaches and yields high anomaly detection accuracy. The study proposes a framework that combines a GAN-generated model with a distance-based anomaly scoring method that measures the dissimilarity between a generated image and the corresponding real image in a feature space, enabling efficient anomaly identification across different data sets. While the method showcases effective anomaly detection without necessitating labelled training data, its performance might be influenced by intricate data distributions and optimal parameter configuration. Further investigations could focus on exploring its versatility across various data sets and assessing its resilience against diverse anomaly patterns.

Additionally, there have been a lot of studies working on various aspects of GANs and state of the art models [22, 23]. One of them is a review paper on GAN-based anomaly detection [10] that includes multiple new models that have not been thoroughly studied on GANs before. It provides a comprehensive exploration of anomaly detection methods utilising GANs, while also addressing the potential of transformers and vector quantisation. The study delivers an insightful overview of the landscape, highlighting the diverse methodologies and techniques that harness GANs for anomaly detection tasks. This review underscores the substantial promise that GANs hold in advancing the field of anomaly detection across a range of domains. However, the effectiveness of GAN-based anomaly detection approaches, including those incorporating transformers and VQ, might be influenced by variables such as data distribution and model architecture. Moving forward, research could focus on comprehensive evaluations of these combined approaches across diverse data sets, refining their implementations to achieve heightened robustness and precision in anomaly detection outcomes.

Another paper that is insightful for the purpose of this study focuses on anomaly detection, learned adversarially [24]. In this paper, a novel anomaly detection technique is presented, utilising an adversarial learning framework. The authors propose a Transformer model (tra-

ditionally used in Natural Language Processing (NLP)) that is applied directly to sequences of image patches and substitutes the utilisation of CNNs as a classification method. This approach proves to be proficient in identifying anomalies while concurrently acquiring valuable data representations. Future investigations could concentrate on fortifying the method's ability to handle diverse anomaly characteristics and evaluating its adaptability to extensive and intricate data sets.

Learning Vector Quantisation is an algorithm used for classification and clustering tasks that refines a group of codebook vectors over iterations to categorise data by assigning them to the nearest codebook vector, combining aspects of supervised and unsupervised learning to group data based on their attributes - explained thoroughly in Section 3.5. LVQ is a method that is also used in Computer Vision tasks, especially when interpretability, simplicity, and efficiency are crucial. For specialised use cases, or when dealing with limited data and resources, LVQ can be a valuable and viable choice for solving image classification and pattern recognition problems. An application on images in the medical domain is included in the paper [25]. The paper introduces a novel method aimed at distinguishing facial and fingerprint images by employing a LVQ-centred approach to generate template keys. The study innovatively employs LVQ to create unique template keys for individual identification. The method showcases promising results in achieving distinct recognition for both facial and fingerprint data, thereby augmenting biometric security measures. Nevertheless, the effectiveness of the technique might be influenced by variables such as data set composition and algorithmic configurations.

Even though the Transformer architecture is rather new, it has been mostly used in NLP. Nevertheless, since recently there have been studies utilising Transformers in computer vision as well, without solely depending on CNN structures that have been one of the main components of image-related tasks, as mentioned in the paper [26]. The authors apply a Transformer network directly on images instead of utilising any CNN model. The study unveils a novel approach that employs transformers to analyse images in a grid-based manner, enabling the capture of intricate visual context. This technique showcases promising capabilities in achieving high levels of accuracy in large-scale image recognition tasks. However, the suitability of the approach may depend on variables like the variability of data sets and computational demands.

Finally, an important addition is a work that gives emphasis to the need of more extensive studying of high quality generated images [27], a useful asset to accurate anomaly detection. In this paper, a significant exploration is presented regarding the integration of deep generative models into the realm of medical imaging. The study initiates an open challenge that encourages the practical application of these models for authentic medical image analysis. The research underscores the substantial potential of deep generative models in augmenting medical diagnostics and imaging procedures. Nevertheless, the effectiveness of this approach could hinge on variables such as the heterogeneity of medical data sets and the robustness of the models. Future undertakings might involve assessing the outcomes across diverse medical imaging techniques and enhancing the models to achieve heightened performance and dependability.

Regarding the above findings and limitations, it is obvious that the main model used for this study (f-AnoGAN) could benefit from its combination with other methods in order to achieve more accurate, diverse, and high-quality image generation and better anomaly detection results. Transformers can improve contextual understanding and control over generated content, while VQ can enhance data representation and stability during training. Also, concerning image quality, VQ aids in creating a more diverse and structured latent space, while Transformers excel in capturing intricate contextual relationships, collectively enabling the GAN to produce images with enhanced realism, finer details, and improved coherence. Integrating these techniques offers exciting opportunities to push the boundaries of GAN-based image synthesis and anomaly detection.

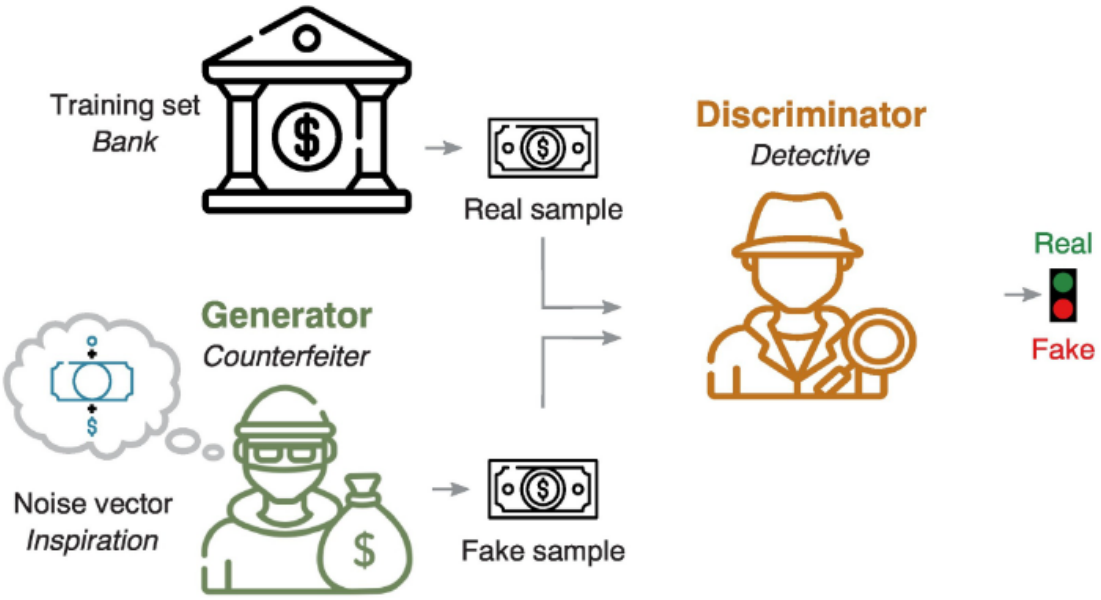


Figure 1: A real-life example of a GAN [28] where the Discriminator (detective) tries to distinguish real data (bank notes) from fake data, provided by the Generator (counterfeiter).

3 Background Knowledge

Although the main techniques previously used by researchers for anomaly detection in images are generative models - especially GANs -, autoencoders [29], CNNs and statistical methods such as Z-score [30], the focus of this study is the f-AnoGAN, a GAN model.

3.1 Generative Modelling

Neural network-based generative modelling was introduced in the 1980s [31] when the objective was to get insights from data without the need for supervision, which could have implications for conventional classification tasks. The appeal of unsupervised learning lies in its efficiency and cost-effectiveness for gathering training data, as it does not require labelled data. Yet, generative models hold substantial potential for diverse applications, given the abundance of unexploited information they offer.

The central idea of generative modelling stems from training a generative model whose samples $\tilde{x} \sim p_\theta(\tilde{x})$ come from the same distribution as the training data distribution, $x \sim p_\theta(x)$. In the initial development of neural generative models, energy-based models [32] accomplished this by establishing an energy function for data points, which was directly linked to their likelihood. Nevertheless, these models faced scalability issues when dealing with complex, high-dimensional data sets like natural images, and they necessitated Markov Chain Monte Carlo (MCMC) [33] sampling in both the training and inference phases, resulting in a time-consuming iterative process.

In recent years, there has been a recovery of interest in generative models due to the availability of large, freely accessible data sets and advancements in both general deep learning architectures and generative models. These developments have pushed the boundaries regarding how accurately and rapidly these models can generate visuals. Many of these improvements have come from working with latent variables z - hidden variables in the model that represent significant characteristics of the data - which are easy to sample or compute density from, rather than learning the joint distribution of x and z (denoted as $p(x, z)$). However, dealing with latent variables that require marginalisation presents a computational challenge. Consequently, generative models often need to compromise on execution time, architectural choices or optimising proxy functions.

3.2 Generative Adversarial Network

Generative adversarial networks (GANs) [4] operate within the framework of a game and involve two machine learning models, often implemented using neural networks. One of these models, known as the generator, defines the probability distribution $p_{model}(x)$. It is important to note that the generator is not always required to explicitly calculate the density function p_{model}. In certain GAN variations, it is possible for the generator to estimate this density function, but it is not a strict necessity. Instead, the generator's primary function is to generate samples from the distribution $p_{model}(x)$. The generator is characterised by a distribution $p(z)$, which represents a distribution over a vector z. This vector serves as input to the generator function $G(z;\theta^{(G)})$, where $\theta^{(G)}$ represents a set of learnable parameters that define the strategy of the generator in the adversarial game. The input vector z can be considered as a source of randomness within an otherwise deterministic system, similar to the seed used in a pseudorandom number generator. The distribution $p(z)$ typically takes the form of a relatively unstructured distribution, such as a high-dimensional Gaussian distribution or a uniform distribution over a hypercube. Samples drawn from this distribution, denoted as z, essentially represent noise. The primary objective of the generator is to learn the function $G(z)$ that transforms this unstructured noise z into realistic data samples.

The other participant in this game is called the discriminator. It assesses samples denoted as x and provides an evaluation, represented as $D(x;\theta^{(D)})$, regarding whether x is genuine (originating from the training dataset) or artificial (generated by the generator from p_{model}). In the original GAN formulation, this evaluation typically takes the form of a probability, indicating the likelihood that the input is real rather than fake, assuming that both real and fake inputs are equally sampled. While there are other variations and formulations of GANs, the discriminator's role is to decide about whether the input is real or fake.

In the GAN framework, both the generator and the discriminator face costs: $J^{(G)}(\theta^{(G)},\theta^{(D)})$ for the generator and $J^{(D)}(\theta^{(G)},\theta^{(D)})$ for the discriminator. Each player strives to minimise its own cost. More simply, the discriminator's cost encourages it to accurately distinguish between real and fake data, while the generator's cost motivates it to generate samples that the discriminator wrongly identifies as real. In the original GAN version, $J^{(D)}$ was defined as the negative log-likelihood assigned by the discriminator to the real-versus-fake labels

given the input. Essentially, the discriminator is trained like a typical binary classifier. The original GAN work proposed two variations for the generator's cost. One, known as minimax GAN (M-GAN), sets $J^{(G)} = -J^{(D}$, creating a minimax game that can be theoretically analysed straightforwardly. In M-GAN, the generator's cost is defined by reversing the sign of the discriminator's cost. Another approach, non-saturating GAN (NS-GAN), defines the generator's cost by reversing the discriminator's labels. In other words, the generator aims to minimise the negative log-likelihood assigned by the discriminator to incorrect labels. This approach helps prevent the problem of gradient saturation during model training.

3.3 Anomaly Detection

Anomaly detection [9] is the task of identifying irregular patterns within data that deviate from the expected behaviour. These atypical patterns - often referred to as "anomalies" - are the most commonly used in the context of anomaly detection. The identification of anomalies within data has a long history, dating back to the 19^{th} century in the field of statistics [34]. Throughout the years, various anomaly detection methods have been created by different research communities. Some of these techniques are tailored for specific application domains, while others have a more general applicability. Its range of applications includes tasks such as detecting fraud in credit cards, insurance, and healthcare, enhancing cyber-security through intrusion detection, distinguishing faults in critical safety systems and monitoring military activities to identify potential enemy actions.

Anomaly detection methods for images can be categorised into several approaches:

- Statistical Techniques: These techniques make use of statistical measures to establish a cutoff point above which data values are regarded as anomalous. Z-scores, percentiles, and histograms are typical methods.
- Supervised Learning: A model is trained on labelled data, where abnormalities are clearly indicated in supervised anomaly detection. Based on the given labels, the model learns to distinguish between typical and anomalous events.
- Unsupervised Learning: In the case of unsupervised anomaly detection, the algorithm picks up on the intrinsic patterns of typical data without the aid of labelled abnormal-

ities. The data points that significantly differ from these discovered patterns are then found.

- Deep Learning Approaches: Deep neural networks, such as autoencoders and GANs, are increasingly being used for anomaly detection due to their ability to capture complex patterns in data.

In the domain of medical imaging, anomaly detection takes on a paramount role due to the extensive volumes of images generated for diagnostic and research purposes. The rapid advancements in medical imaging technology have led to the accumulation of vast data sets containing a multitude of visual information, ranging from X-rays and MRIs to CT scans and histopathological slides. However, the sheer volume of medical images makes manual anomaly detection an impractical endeavour, prompting the need for intelligent automated solutions. Anomaly detection techniques in this context are crucial for identifying subtle variations, irregularities, or potential pathologies that might elude human scrutiny. By leveraging data-driven approaches, machine learning algorithms, and deep learning models such as GANs, medical professionals and researchers can streamline the process of identifying anomalies within medical images. These techniques not only enhance the accuracy, speed and efficiency of diagnosis but also pave the way for early disease detection, personalised treatment planning and the discovery of novel medical insights that have the potential to transform patient care and medical research.

3.4 f-AnoGAN

During GAN training, a generator denoted as $G(z) = z \rightarrow x$ maps from a space called Z to another space called X. However, for the purpose of anomaly detection, the inverse mapping from X to Z is required. This inverse mapping, $E(x) = x \rightarrow z$, is learned by training a deep encoder network, denoted as E. There are two fundamental architectures for training this encoder; z-image-z (abbreviated as ziz) encoder training and image-z-image (abbreviated as izi) encoder training. In both cases, a convolutional autoencoder (AE) architecture is used. This architecture consists of a trainable encoder, denoted as E, which maps from an image to the z-space. The generator, which serves as the decoder, maps from z to the image space using fixed weights obtained from the WGAN training. The key distinction between the two encoder training approaches lies in the order in which the encoder and the decoder (the

trained generator) are utilised. During encoder training, only the parameters of the encoder are optimised, while the parameters of the generator are constant.

Reversing the order of encoder and decoder utilisation results in the ziz architecture within a standard AE. During training, a random sample is drawn from the z-space and mapped to the image space using the fixed generator G. The encoder, denoted as E, is trained to map this image back to the z-space. It is important to note that ziz encoder training does not require actual image data. This architecture essentially resembles a z-to-z AE, where the mapping from z to an image, G, remains fixed. In the training process, we aim to minimise the mean squared error (MSE) between input z-samples, z, and their reconstructed counterparts, E(G(z)), expressed as $L_{ziz} = \frac{1}{d}||z - E(G(z))||^2$, where 'd' represents the dimensionality of the z-space. Unlike the izi architecture, ziz provides known target z locations. However, one limitation is that the encoder only encounters generated images and does not receive real input images.

The izi architecture adheres to a conventional AE setup, where an encoder is followed by a decoder, which is essentially the generator. During the training process, the encoder, which is a trainable component, performs the mapping from real images to their latent encodings denoted as 'z'. On the other hand, the mapping from 'z' back to the image space is carried out using the fixed generator G. This configuration essentially resembles an image-to-image AE. The training objective is to minimise the MSE residual loss between input images, 'x', and their reconstructed counterparts, which are generated by passing them through G after encoding with E. This is expressed as $L_{izi}(x) = \frac{1}{n}||x - G(E(x))||^2$, where $||\cdot||^2$ signifies the sum of squared pixel-wise differences in grey values, and 'n' represents the number of pixels in an image. It is worth noting that the izi encoder is trained using the same data set that was employed for WGAN training, which typically consists of normal images. However, this approach has a notable limitation. Since the true target location in the z-space for a given query image is unknown, the accuracy of the image-to-z mapping can only be assessed indirectly by reversing the process back to the image space and evaluating the image-to-image differences.

The training objective of izi emphasises achieving similarity in the image space. When mapping new images, it is possible for them to end up in regions of the latent space that

were not sampled as much during training, and such positions might not effectively deceive the discriminator when translated back to the image space. Consequently, solely minimising pixel-wise differences can sometimes generate images that do not resemble typical examples of normal images, but can still exhibit small residuals, even for anomalous images. This implies that relying solely on the residual in the image space may not be a dependable indicator of anomalies. To deal with this limitation, incorporating the residual in the feature space, which is populated by the discriminator, proves to be a robust basis for identifying anomalous images. This insight has led to the development of the izi_f architecture, where the image statistics for both the real image and the reconstructed image are additionally computed. The loss function for discriminator-guided izi encoder training, abbreviated as izi_f, is as follows:

$$L_{izi_f}(x) = \frac{1}{n} \cdot ||x - G(E(x))||^2 + \frac{\kappa}{n_d} \cdot ||f(x) - f(G(E(x)))||^2$$

, where discriminator features, denoted as $f(\cdot)$, extracted from an intermediate layer, are utilised to serve as statistics for a given input. Here, n_d represents the dimensionality of this intermediate feature representation, and κ is a weighting factor. It is important to note that the parameters of the discriminator remain fixed and are the ones learned during WGAN training. Given that the izi_f architecture effectively guides encoder training both in the image space and the latent space concurrently, the f-AnoGAN creators opt for the izif approach as the preferred encoder training architecture within the f-AnoGAN framework.

In the process of anomaly detection at the image level, we assess how much the query images differ from their corresponding reconstructions. All the elements required for generating these image reconstructions and conducting anomaly quantification are trained through both the WGAN training phase and the encoder training phase. The way we formulate the quantification of anomalies directly follows the specific definition of the loss used during encoder training. In the context of the f-AnoGAN model the authors ([21]) propose, which incorporates the discriminator-guided izi_f encoder training, the ultimate anomaly score denoted as $A(x)$ for a new image x is determined as $A(x) = A_R(x) + \kappa \cdot A_D(x)$, where

$$\left\{ \begin{array}{l} A_R(x) = \frac{1}{n} \cdot ||x - G(E(x))||^2 \\ \\ A_D(x) = \frac{1}{n_d} \cdot ||f(x) - f(G(E(x)))||^2 \end{array} \right\}$$

and the parameter κ serves as a weighting factor. For encoder training architectures that do not involve a term based on the discriminator, namely the izi and ziz architectures, the definition of the anomaly score simplifies to $A(x) = A_R(x) = \frac{1}{n} \cdot ||x - G(E(x))||^2$. In general, both formulations tend to yield high anomaly scores for anomalous images and low anomaly scores for typical input images. Since the model is exclusively trained on normal images, it can only reconstruct images that visually resemble the input image and fall within the normal image manifold, denoted as X. The model's ability to reconstruct visually similar images is inversely related to the degree of anomaly. Normal query images result in small deviations, while anomalous images are associated with reconstructions that exhibit substantial deviations. The absolute value of pixel-wise residuals, denoted as $\dot{A}_R(x)$ and defined as $|x - G(E(x))|$, is employed for pinpointing anomalies at the pixel level.

3.5 Learning Vector Quantisation

The Learning Vector Quantisation (LVQ) [35] algorithm shares similarities with the k-Nearest Neighbors (kNN) method [36], as it predicts outcomes by finding the closest match among a set of learned patterns. However, LVQ stands out by adopting a more structured approach to building the pattern library, known as codebook vectors, which adds a layer of adaptability and fine-tuning to the prediction process.

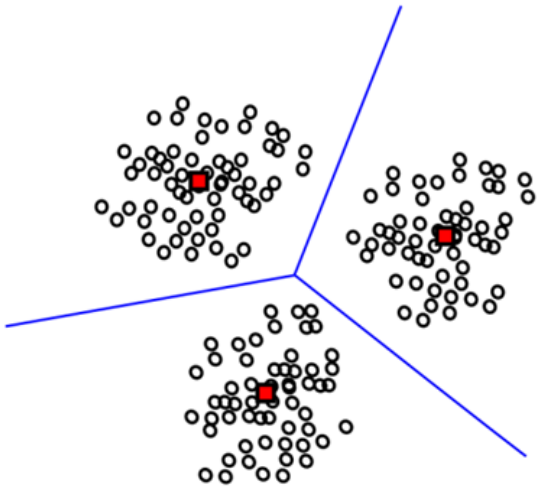


Figure 2: The result of Vector Quantization.

In LVQ, the codebook vectors serve as representatives of distinct classes in the dataset, and each pattern in the library is referred to as a codebook. Unlike kNN, which uses the raw training patterns as is, LVQ dynamically learns these codebook vectors from the training data. To start, the algorithm initialises the codebook vectors by randomly selecting patterns from the training data set. Subsequently, over several training iterations known as epochs, these codebook vectors are optimised and adapted to best encapsulate the underlying structure of the training data. This process allows the codebook vectors to effectively capture the key characteristics of each class in the data set.

During the learning process, LVQ takes one training record at a time and evaluates which codebook vector (representing a class) is the best match for the input record. It then updates the position of the chosen codebook vector, moving it closer to the training record if they share the same class, or pushing it further away if they belong to different classes. This iterative adjustment of the codebook vectors based on class similarity helps fine-tune the model to better represent the underlying distribution of the training data.

Once the codebook vectors are adequately prepared, LVQ employs a kNN mechanism, with k set to 1, for making predictions on new, unseen data points. The prediction process involves finding the closest codebook vector to the input data point and assigning it the class label associated with that codebook vector. An example of how the LVQ method works on an image is observed below:

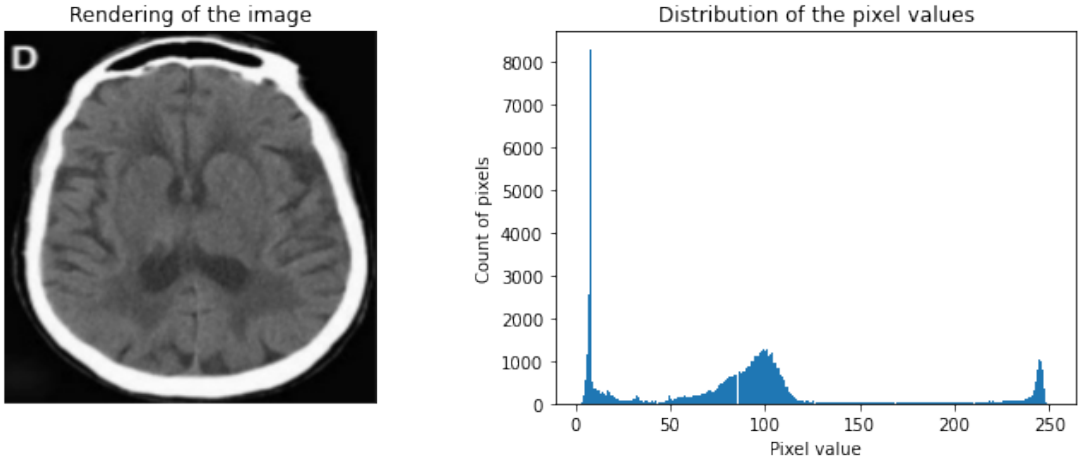


Figure 3: A typical healthy brain CT and its histogram. Nearly all values in the interval [0, 255] are occupied.

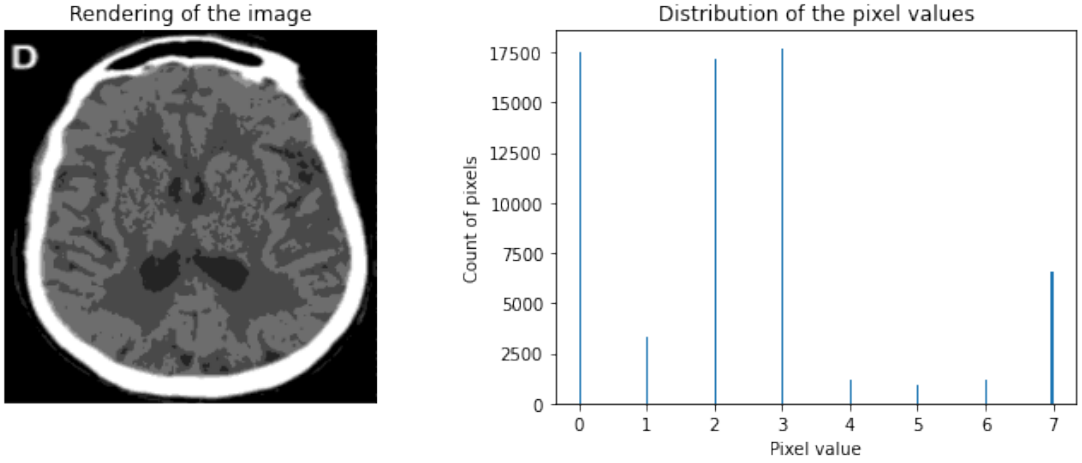


Figure 4: The Figure 3 brain CT in 3 bits and its histogram. The pixel values have been grouped in 2^3=8 bins of stable width and then discretised to 8 values.

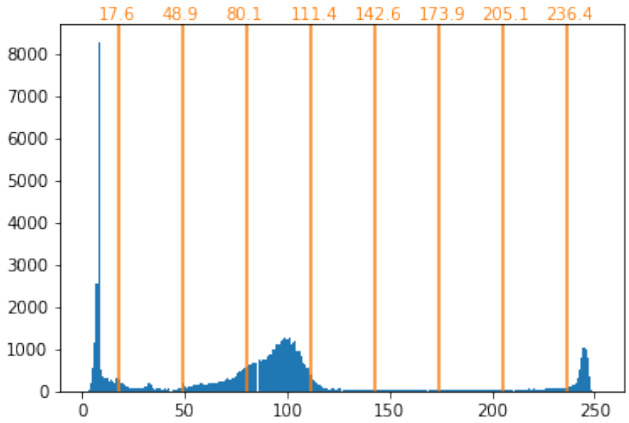


Figure 5: The values of the histogram chosen from the 8 bins.

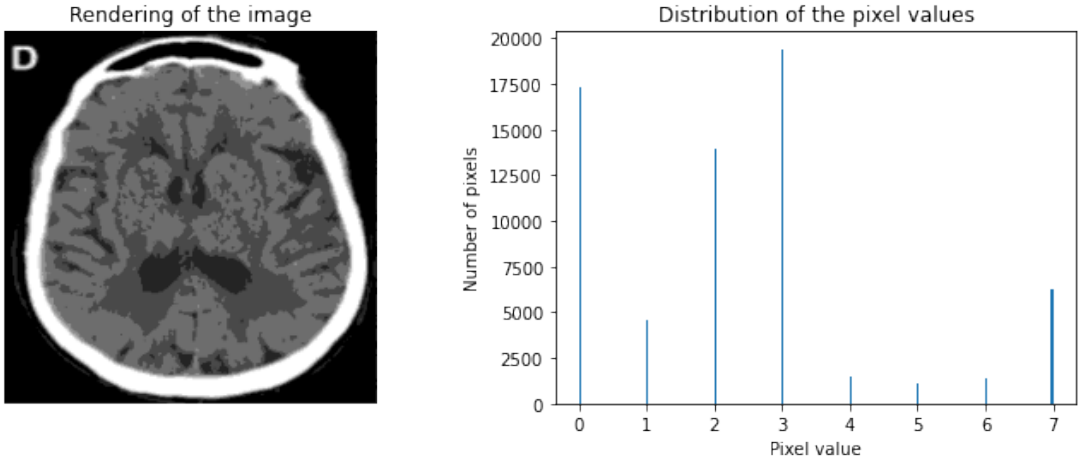


Figure 6: The Figure 3 brain CT in 3 bits and its histogram. The pixel values have been grouped in 8 bins of stable width and then discretised to 8 values.

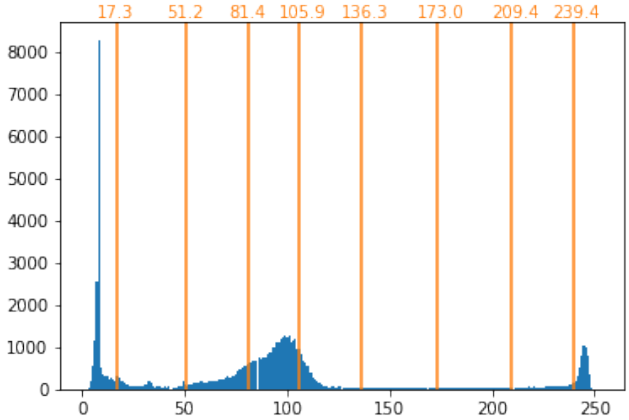


Figure 7: The values of the histogram chosen from the 8 bins. Here, these values are chosen so that they are closer to higher frequencies between the pixel values, but still in the same bins.

Originally developed for classification tasks, LVQ can be adapted and extended for regression problems as well, making it a versatile algorithm suitable for both predictive modelling scenarios. While not as widely used as more advanced deep learning models, LVQ remains relevant in specific situations, especially when interpretability and simplicity are desired or when computational resources are limited. Its ability to learn meaningful codebook vectors from relatively small to medium-sized data sets can provide valuable insights and context for classification decisions.

3.6 Transformer

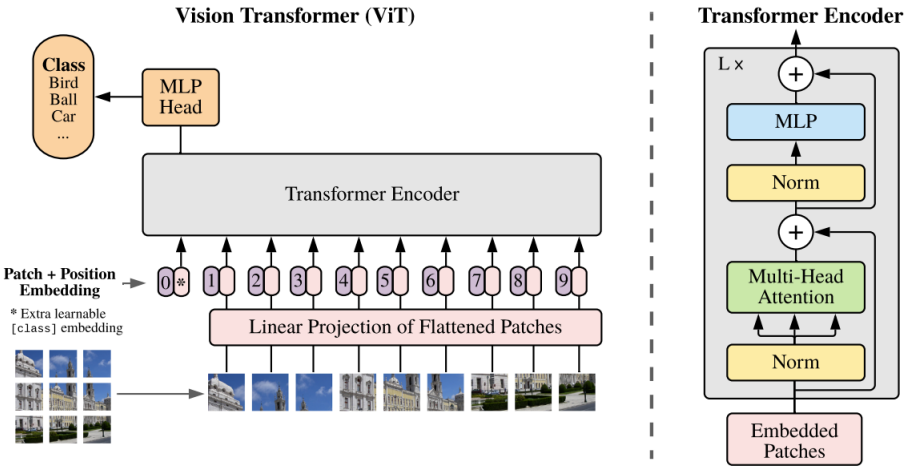


Figure 8: An image is split into patches of the same size. Next, each of them is linearly embedded, position embeddings are added and the produced sequence is the input of the Transformer encoder. linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder.

The authors of the paper [26] designed the Vision Transformer (ViT) model inspired by the Original Transformer [37]. ViT brings the power of the transformer architecture, originally designed for natural language processing, into the domain of computer vision.

The key idea behind ViT is to treat images as sequences of patches and use a transformer model to process these patches. The main goal of ViT is image classification, albeit this study will only use the encoder part of ViT to extract the most important information of images. The architecture of the model is included in the following steps:

- Image Patching: The first step is to break down the input image into smaller fixed-size patches. Each patch is considered as a token and is treated as a word in the context of natural language processing. This patching process allows ViT to handle images of arbitrary sizes and turns the 2D image into a 1D sequence of tokens.
- Token Embeddings: Each patch is linearly projected to a lower-dimensional representation, known as the token embeddings. The token embeddings retain the spatial information of the image patches and form the input sequence for the transformer model.
- Positional Embeddings: In the transformer architecture, positional information is cru-

cial to provide context about the order of tokens in the sequence. For language tasks, this positional information is typically represented as positional encodings. In ViT, positional embeddings are used to encode the 2D spatial location of the original image patches. The positional embeddings are added to the token embeddings to form the final input embeddings for the transformer.

- Transformer Encoder: The main part of ViT is the transformer encoder, which processes the input embeddings and learns the relevant features. The transformer encoder consists of multiple stacked transformer blocks. Each block has a multi-head self-attention mechanism (allows the model to weigh the importance of different parts of the input sequence concerning each other and helps the model capture global context and relationships between tokens) and position-wise feedforward neural networks (after self-attention, the outputs pass through position-wise feedforward neural networks, which introduce non-linearities and further process the information).

ViT has shown impressive performance on various image classification benchmarks, demonstrating that transformers can be highly effective in processing visual information when appropriately adapted to the image domain.

3.7 Evaluation metrics

Evaluation metrics are crucial tools in assessing the performance of machine learning models. They help quantify the effectiveness and reliability of a model's predictions. Four commonly used evaluation metrics are accuracy, precision, recall, and F1-score.

1. Accuracy: Accuracy measures the portion of correctly predicted instances out of all the instances in the data set. It is a straightforward metric and works well when the classes are balanced. However, accuracy can be misleading when dealing with imbalanced data sets, where one class significantly outnumbers the other. In such cases, a high accuracy might not reflect the model's actual performance.

2. Precision: Precision is the number of true positive predictions over the number of all positive predictions (true positives + false positives). Precision focuses on the accuracy of positive predictions. It is particularly useful when the cost of false positives is high. For instance, in medical diagnosis, false positives could lead to unnecessary treatments.

3. Recall: Recall is the ratio of true positive predictions to the total number of actual positive instances (true positives + false negatives). Recall calculates the model's ability to detect all positive instances. It is important when the cost of false negatives is high. For example, in fraud detection, missing a fraudulent transaction can have severe consequences.

4. F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall, making it a suitable metric when there's a trade-off between false positives and false negatives. The F1-score is particularly useful when dealing with imbalanced data sets, as it considers both false positives and false negatives in its calculation.

An easy way to obtain the true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) in order to calculate these metrics is by creating a confusion matrix about each of the experiments.

4 Methodology

In this phase of the project, the focus is around enhancing anomaly detection in images within the framework of GANs. The existing literature has explored the utilisation of GANs for anomaly detection, with some projects using methods such as LVQ and Transformers. However, a novel approach is studied in this work, combining these established techniques with GANs in a new way.

The primary insight leading to this methodology is the integration of LVQ and Transformers within the architecture of the GAN's generator network. Traditionally, GANs consist of an encoder-decoder structure, where the encoder compresses the input data into a latent space and the decoder reconstructs it back into the original form. This proposed methodology places LVQ and Transformers in this generator's encoding-decoding pipeline, aiming to optimise the size of the encoded images while maintaining or even enhancing the anomaly detection capability.

LVQ, a technique well-established in the field of machine learning, aims to be a valuable addition to the GAN framework. Placed between the encoder and decoder networks, the LVQ module serves as an inbetween layer responsible for compressing the encoded images' representations. By using the functionality of LVQ, the encoded features go under a refinement process that emphasises important information, ensuring more efficient storage and succeeding reconstruction.

The inclusion of Transformers aims to enhance the encoding process as well, providing a robust mechanism for capturing main characteristics within the encoded data. Transformers excel in modelling relationships between different parts of an image, making them an appropriate candidate for image compression tasks. Placed alternately and concurrently with LVQ, Transformers contribute to the reduction of encoded image size, while preserving the essential characteristics necessary for accurate anomaly detection.

A distinctive feature of this methodology is the intentional gradation of the compression process. In the initial phase, the encoder operates with an extended range, as the images are not altered or compressed. As the process advances, the role of both LVQ and Trans-

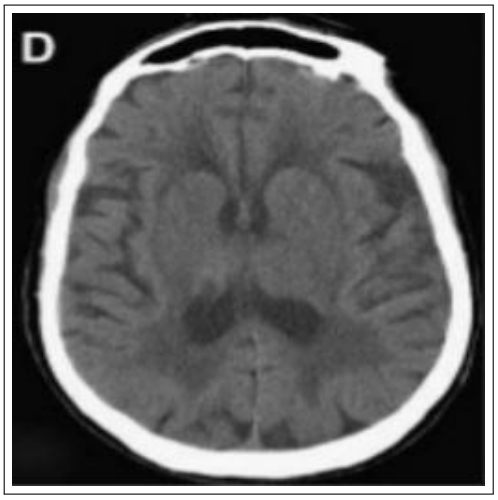
formers evolves, gradually narrowing the sizes of the images in the latent space and keeping the most valuable features. This intentional reduction aims to achieve a balance between compact representation and anomaly detection accuracy. By intertwining LVQ and Transformers within the GAN's generator architecture, this methodology seeks to optimise the encoding-decoding process for improved anomaly detection in images. The organised collaboration of these techniques addresses the challenge of reducing encoded image size without compromising the precision required for accurate anomaly identification.

In Section 6, the study delves into the implementation details and empirical evaluations to validate the effectiveness of this innovative approach in the context of anomaly detection using GANs.

5 Data Set

As technology advances, the need for anomaly detection in a great number of radiographical scans of various modalities increases. The problem that stems from this situation is that an estimated percentage of 10-15% of such scans may be delayed, missed, or incorrectly diagnosed [38]. As the focus of this study is to check if f-AnoGAN can be combined with LVQ and ViT methods successfully to perform anomaly detection on such data sets, a relevant data set should be used as a supplement.

The data set that is utilised to apply the models on can be found on Kaggle [39]. More specifically, the images that are used regard 90 head CTs of healthy brains (e.g. Figure 9a) and 50 masks of the same dimensions - 256x256. These masks comprise black and white images that contain black pixels in places where healthy tissue would be observed on a head CT and white pixels represent the existence of cancerous parts.



(a) Healthy brain CT.

(b) Tumorous brain CT.

Figure 9: Two different outcomes of a CT scan: no tumour vs tumour.

6 Experiments

6.1 Data Set

Deep Convolutional Generative Adversarial Network: Deep Convolutional Generative Adversarial Network (DCGAN) [40] is a specific variant of a GAN model that uses CNNs in both the Generator and Discriminator. It is a model that has been frequently used for image generation. There are two differences between GANs and DCGANs:

1. In contrary to the lenient structure of GANs, DCGAN uses a specific architectural pattern based on convolutional and transpose convolutional layers in the generator and discriminator, as will be analysed below.

2. GANs are applied to various types of data generation tasks, while DCGAN is designed for image generation.

The DCGAN Generator architecture consists of an input layer, dense layers that map their input to higher-dimensional spaces, reshape layers that reshape the output of a dense layer into a small spatial volume, several transpose convolutional layers that progressively upsample the spatial dimensions of the data, transforming them into higher-resolution images (each transpose convolutional layer is typically followed by batch normalisation and ReLU activation to stabilise and improve the training process) and an output layer. The final layer often uses a transpose convolution with a sigmoid or tanh activation function to produce the generated image.

The Discriminator architecture also starts with the input layer, followed by several convolutional layers that process the input image, progressively downsampling its spatial dimensions. Also, a LeakyReLU activation function is used to introduce a small negative slope, which helps prevent the vanishing gradient problem and allows for better training. Next, dropout layers are optional to regularise the discriminator and prevent overfitting and the feature maps are flattened into a 1D vector by using a Flatten layer. Finally, one or more dense layers are used for further processing and producing the final output and the output layer is typically a single neuron with a sigmoid activation function, providing a probability score indicating whether the input is real or fake.

Morphological opening: Morphological opening [41] is an operation that smooths the image, breaks down the bridge and eliminates small objects. It is a process that applies erosion and is followed by dilation on the input image in the following fashion:

$$Opening = IM \ominus SE \oplus SE \tag{1}$$

where IM = input image, SE = structuring element and $\ominus$ and $\oplus$ denote erosion and dilation respectively. The structuring element is an important subject in morphological-image processing, as the characteristics of the structuring element can affect the opening and closing processes. The pixel value of one on the structuring element is set as the foreground, while the pixel value of zero is set as the background.

The erosion process shrinks the foreground of the image by increasing the background area. The dilation process enlarges the foreground of the image by increasing the foreground area. Both erosion and dilation processes use the same structuring element.

Train and Validation set: Since the number of healthy brain CTs is very low, the first step to be taken for the train set is to create more 256x256 images in order to have a sufficient amount of images for the training phase of the final model. In order to achieve the vast amount of images, a DCGAN model is used and 45,500 images (40,500 and 5,000) are generated. This number is chosen as the number of test set's images should be 9,500 (see at "Test set" in current subsection 6.1) and the size of the train set should be greater than the test set. More specifically, ∼80% of the whole amount of images which is 40,500 images represents the train set (will be split into 70% train and 10% validation) and ∼20% the test set. Nine random samples can be seen in Figure 10.

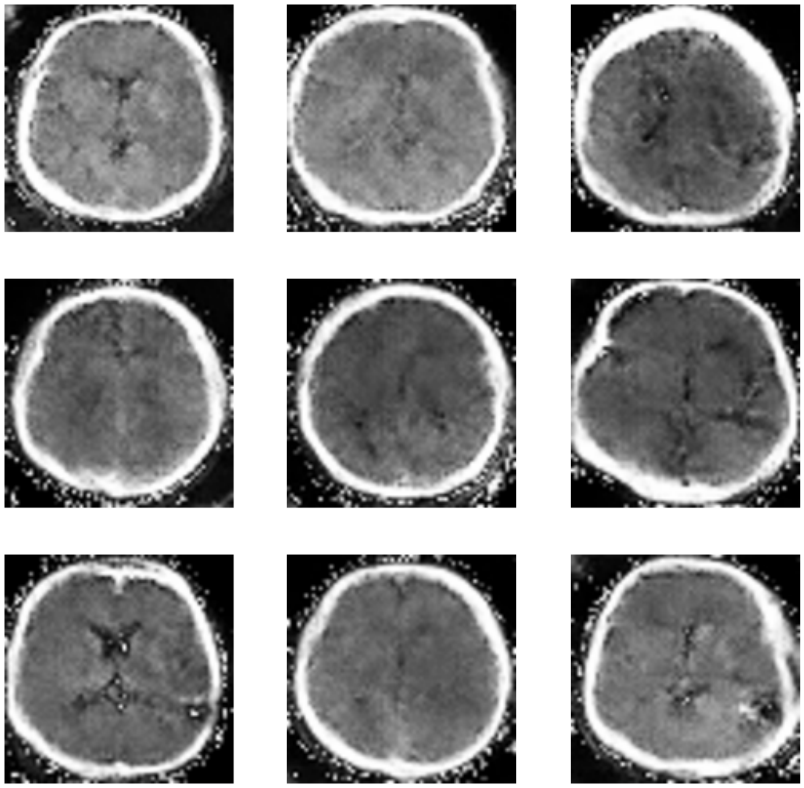
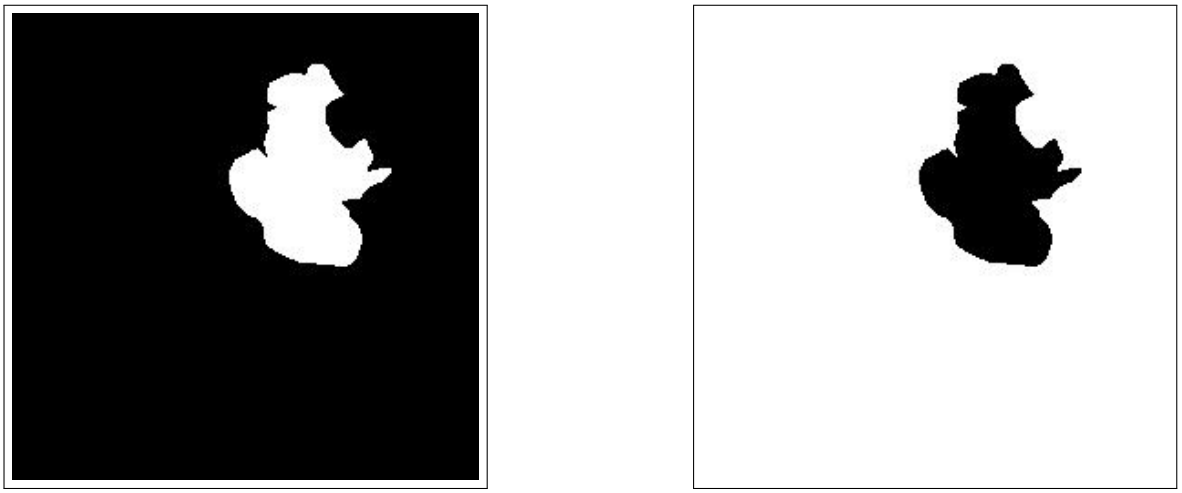


Figure 10: Nine samples of the generated healthy brain CT images.

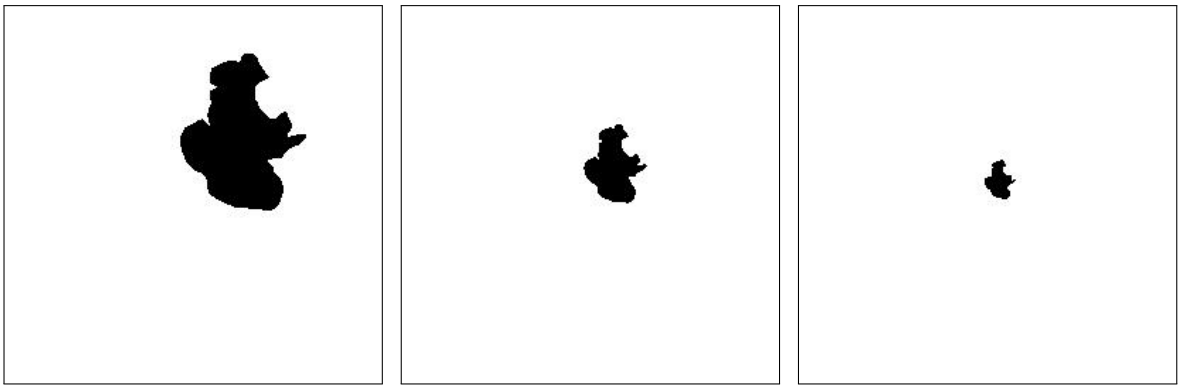
Test set: The masks are placed over the healthy brain images as a top layer to create 90x50 = 4,500 images of brains with tumours. The placing happens bit-wise, meaning that for each pixel of the healthy brain CT image, if the corresponding mask pixel is 0 (black) then the healthy image pixel is converted to 0 whilst it is kept as it is in case the mask pixel is 255 (white). The tumours in the masks are represented by white and the background by black pixels (Figure 11a). Due to the functionality of the bit-wise AND, the object in the mask (tumour) needs to be black and the background white in order to be replaced in the healthy brain image and leave the rest of it intact, so the colours are inverted by setting newPixelColour = 255 - currentPixelColour (Figure 11b). These 4,500 images are combined with the 5,000 generated healthy images in order to have both classes - healthy and tumorous - relatively balanced in the test set and be able to calculate the evaluation metrics.



(a) Original mask. (b) Inverted mask.

Figure 11: A mask before and after inverting its colours.

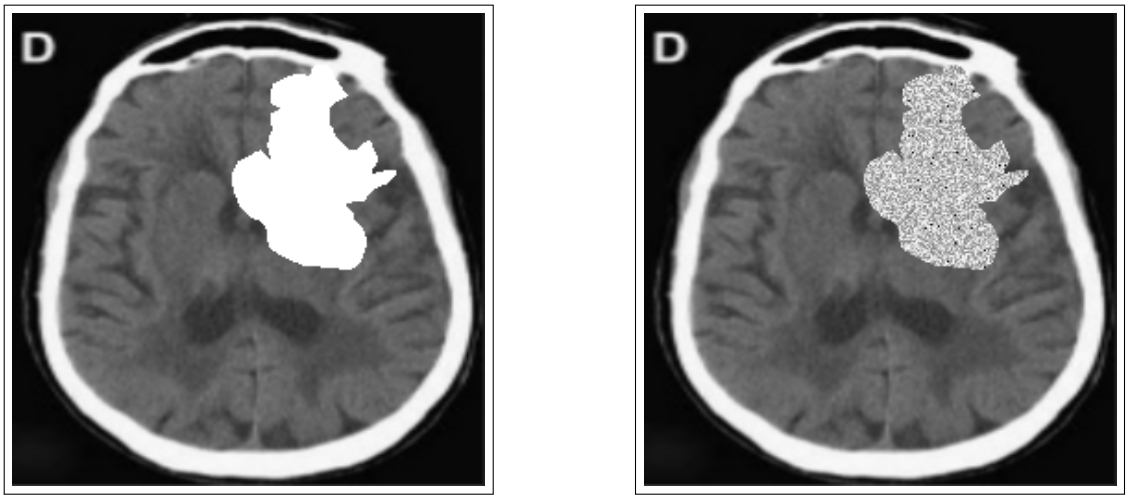
Most of the masks represent possible tumours of big size. This is not always the case, as there also exist tumours that are much smaller and not immediately recognisable. Hence, the data set needs to include a variety of shapes and sizes. In order to achieve such an outcome, the size of the object of the mask is halved and then halved again, resulting to masks with a ratio of 1/2 and 1/4 of the original size. The shape of the object is maintained and the object itself remains in a relative position to the original in the mask. Since this modification adds 2 more states of the size of the tumour, the amount of test set images is 4,500 x 3 = 13,500. In order to have two balanced classes for the test set - healthy and tumorous - these 13,500 images are combined with a portion of 13,500 healthy brain CTs and create a 27,000-image set. The different sizes are displayed in Figure 23.



(a) Original size (1/1). (b) Halved size (1/2). (c) Twice halved size (1/4).

Figure 12: The three different sizes used for each mask.

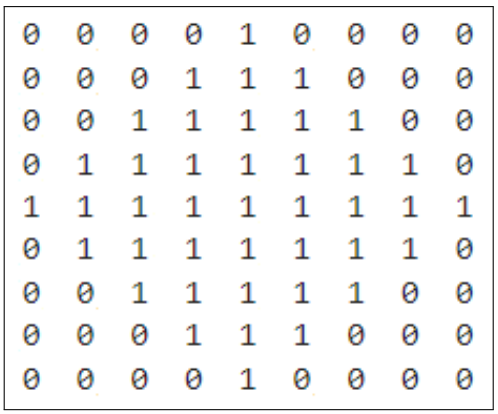
The next step is to apply the mask to the healthy brain CT and check how it looks. Looking into Figure 13a, it is fairly obvious that the anomalous part of the image - the tumour - is not realistic enough, since it is a plain blank part in the image. Checking how a real CT scan of a brain with cancer looks like in Figure 9b, some random noise needs to be added to the mask in order to blend in more smoothly but still be noticeable (Figure 13b). The object's pixels are coloured randomly with a number in the interval [128,256) and such numbers of the highest half of the black and white spectrum are used to keep the anomaly in a light shade and be differentiated from the rest of the brain.



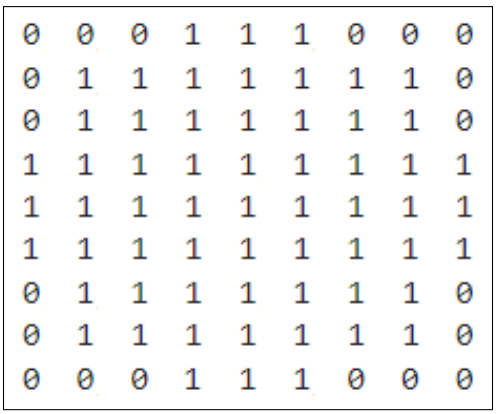
(a) Plain mask. (b) Noisy mask.

Figure 13: Before and after adding noise to the mask's object.

Another aspect of the images that needs to be fixed is noise in the sense that not only are there lines around the head that need to be removed but also there is a big letter on the upper left corner of the image that has no purpose for anomaly detection. A technique that is commonly used in this case is Morphological Opening, analysed in Section **??**. The structuring element utilised to perform the opening operation is an array with ones in a shape of a diamond, as seen in Figure 14a.



(a) Diamond-shaped kernel.



(b) Disc-shaped kernel.

Figure 14: The two types of 9x9 kernels used as structuring elements for the opening operation.

6.2 Final Experiments

The experimental setup goes as follows:

- Use the DCGAN model to produce 40,500 train set images.
- Train the f-AnoGAN model on the train set in order to learn the features of healthy brain CT images.
- Create the 27,000 test set images with various applied masks.
- Test the trained f-AnoGAN model on the test set and distinguish anomalous images.

The Generator of f-AnoGAN is modified in order to apply the LVQ and Transformer methods. Specifically, the encoder of the Generator is combined with each of the methods seperately and at the same time. The three experiments that occur are:

1. Encoder $\longrightarrow$ LVQ $\longrightarrow$ Decoder
2. Encoder $\longrightarrow$ Transformer encoder $\longrightarrow$ Decoder
3. Encoder $\longrightarrow$ LVQ $\longrightarrow$ Transformer encoder $\longrightarrow$ Decoder

As discussed previously, the DCGAN model consists of two main parts: the Generator network that produces fake images based on the distribution of the images that it is trained on and the Discriminator network that tries to distinguish real from fake images. Moreover, the structures of the WGAN's Generator and Critic networks and Encoder network that comprise

the f-AnoGAN model are analysed in detail. The architectures of the two components of the DCGAN model and the three components of the f-AnoGAN model can be examined below:

```
Model: "Generator"
_________________________________________________________________
 Layer (type)                 Output Shape
=================================================================
 Generator-Hidden-Layer-1 (D  (None, 131072)
 ense)

 Generator-Hidden-Layer-Resh  (None, 32, 32, 128)
 ape-1 (Reshape)

 Generator-Hidden-Layer-2 (C  (None, 64, 64, 128)
 onv2DTranspose)

 Generator-Hidden-Layer-Acti  (None, 64, 64, 128)
 vation-2 (ReLU)

 Generator-Hidden-Layer-3 (C  (None, 128, 128, 256)
 onv2DTranspose)

 Generator-Hidden-Layer-Acti  (None, 128, 128, 256)
 vation-3 (ReLU)

 Generator-Hidden-Layer-4 (C  (None, 256, 256, 512)
 onv2DTranspose)

 Generator-Hidden-Layer-Acti  (None, 256, 256, 512)
 vation-4 (ReLU)

 Generator-Output-Layer (Con  (None, 256, 256, 1)
 v2D)

=================================================================
```

Figure 15: DCGAN Generator network.

```
Model: "Discriminator"
_________________________________________________________________
 Layer (type)                 Output Shape
=================================================================
 Discriminator-Hidden-Layer-  (None, 128, 128, 64)
 1 (Conv2D)

 Discriminator-Hidden-Layer-  (None, 128, 128, 64)
 Activation-1 (LeakyReLU)

 Discriminator-Hidden-Layer-  (None, 64, 64, 128)
 2 (Conv2D)

 Discriminator-Hidden-Layer-  (None, 64, 64, 128)
 Activation-2 (LeakyReLU)

 Discriminator-Hidden-Layer-  (None, 32, 32, 128)
 3 (Conv2D)

 Discriminator-Hidden-Layer-  (None, 32, 32, 128)
 Activation-3 (LeakyReLU)

 Discriminator-Flatten-Layer  (None, 131072)
  (Flatten)

 Discriminator-Flatten-Layer  (None, 131072)
 -Dropout (Dropout)

 Discriminator-Output-Layer   (None, 1)
 (Dense)

=================================================================
```

Figure 16: DCGAN Discriminator network.

Layer Type	Output Shape
Dense	(None, 8192)
BatchNormalization	(None, 8192)
ReLU	(None, 8192)
Reshape	(None, 4, 4, 512)
LVQ and/or ViT	(None, 4, 4, 512)
UpSampling2D	(None, 8, 8, 512)
Conv2D	(None, 8, 8, 256)
BatchNormalization	(None, 8, 8, 256)
ReLU	(None, 8, 8, 256)
UpSampling2D	(None, 16, 16, 256)
Conv2D	(None, 16, 16, 128)
BatchNormalization	(None, 16, 16, 128)
ReLU	(None, 16, 16, 128)
UpSampling2D	(None, 256, 256, 128)
Conv2D	(None, 256, 256, 1)

Table 1: f-AnoGAN Generator network.

Layer Type	Output Shape
Conv2D	(None, 128, 128, 128)
LeakyReLU	(None, 128, 128, 128)
Conv2D	(None, 64, 64, 256)
LeakyReLU	(None, 64, 64, 256)
Conv2D	(None, 32, 32, 512)
LeakyReLU	(None, 32, 32, 512)
Flatten	(None, 524288)
Dense	(None, 1)

Table 2: f-AnoGAN Critic network.

Layer Type	Output Shape
Conv2D	(None, 256, 256, 8)
Conv2D	(None, 128, 128, 16)
Conv2D	(None, 64, 64, 128)
GlobalAveragePooling2D	(None, 128)

Table 3: f-AnoGAN Encoder network.

7 Results

Each of the three experiments and the original model are run ten times in order to get their average results - the results closer to the average are depicted in this section. Also, the focus of this study is to examine the potential success of combining the LVQ and ViT methods with f-AnoGAN, so the hyperparameter optimisation of the overall models is not performed as it deviates from the main goal. The focus is solely on the losses of the networks and the evaluation metrics.

The training process of the f-AnoGAN on the train set of 40,500 healthy brain CT images produces the following results:

Experiment 1:

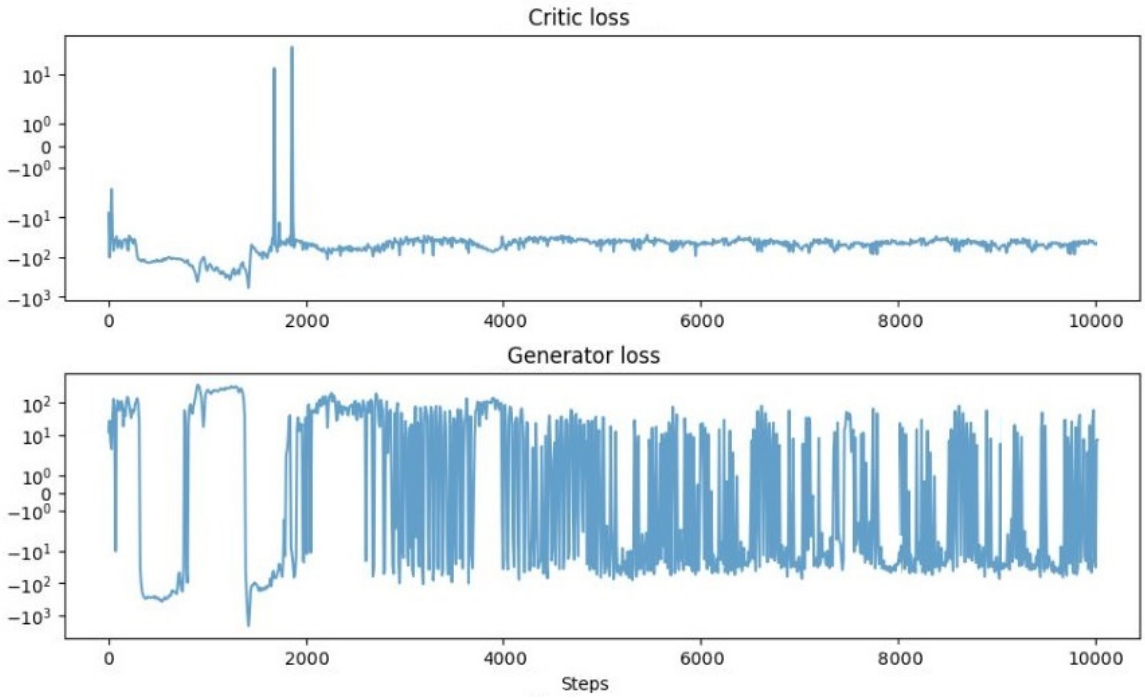


Figure 17: Losses of the networks on the training set for Experiment 1.

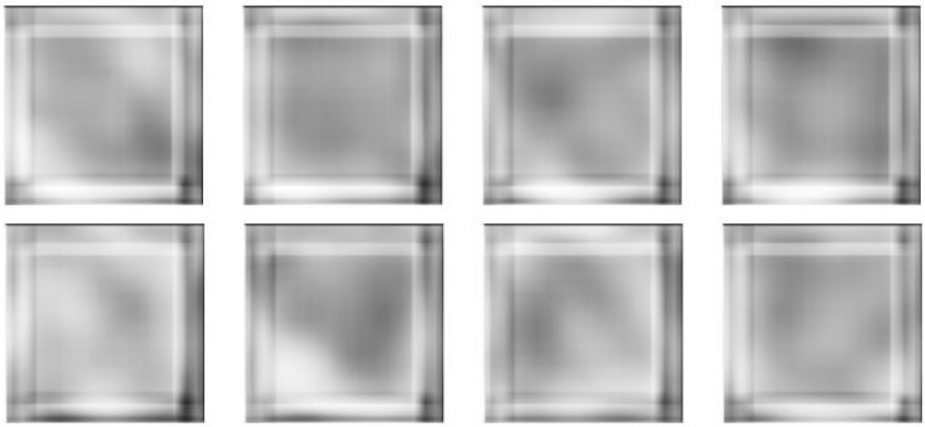


Figure 18: Structure of healthy brain CT images captured by the model for Experiment 1.

What is observed by studying Figure 17, is that the critic loss fluctuates constantly at negative values, with two exceptions just before step 2,000, where its value becomes positive. The generator loss fluctuates even more, with a slower rate before approximately epoch 2,300 and a faster rate afterwards. The final loss of the critic at epoch 10,000 is -23.51, while the generator loss terminates at 9.73.

The eight samples of the structure of healthy brain CT images captured by the model are included in Figure 18. The shapes they include are inconsistent, whereas the images themselves do not depict clearly the shape of a brain CT.

Experiment 2:

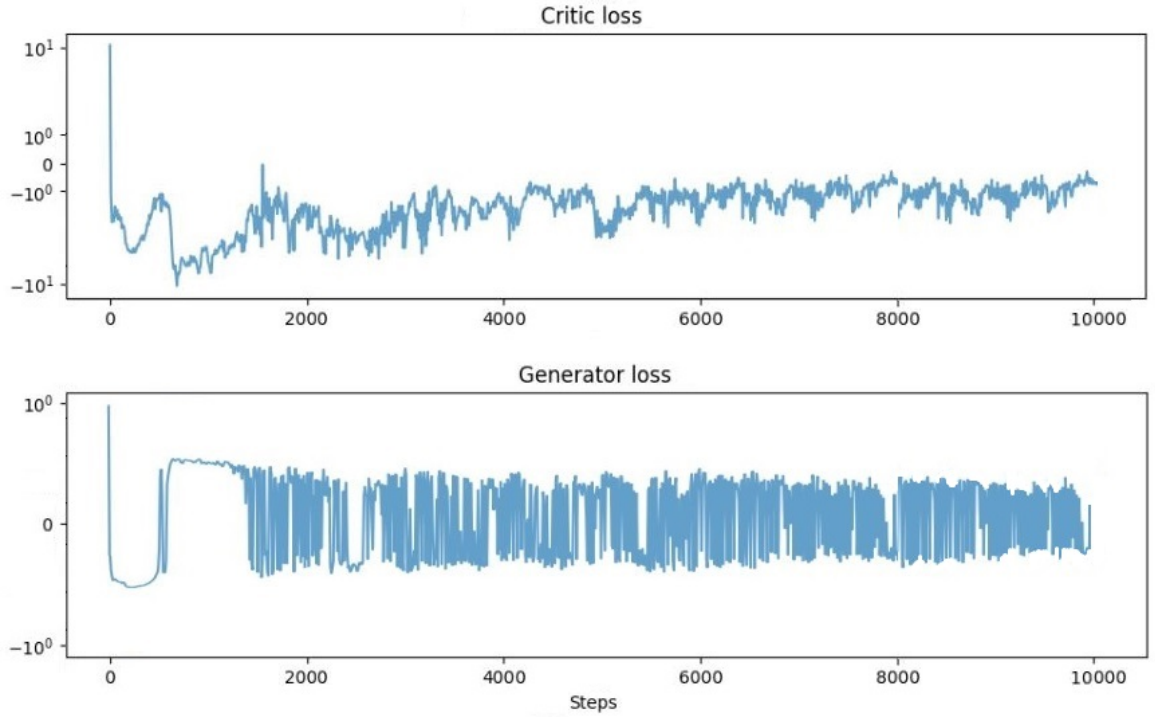


Figure 19: Losses of the networks on the training set for Experiment 2.

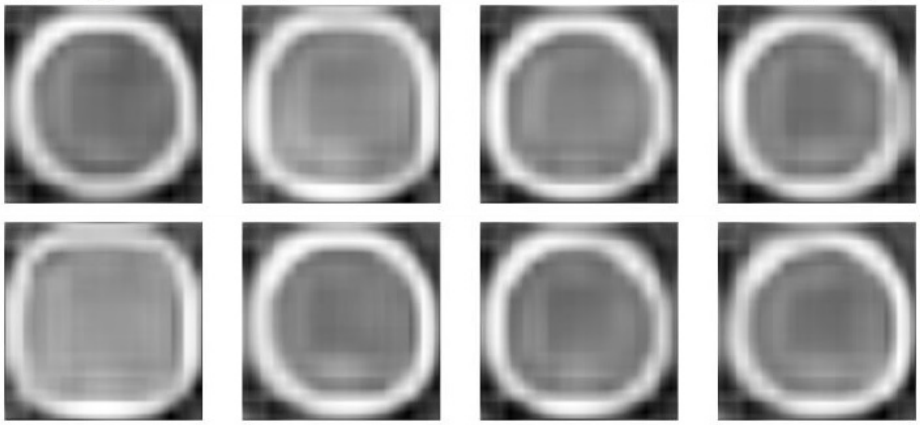


Figure 20: Structure of healthy brain CT images captured by the model for Experiment 2.

Figure 19 shows that the results of Experiment 2 follow a similar fashion. The critic and generator losses fluctuate in this case as well, although not in the same scale as Experiment 1. At approximately epoch 1,800 the losses start to stabilise and converge gradually - the critic loss rises, while the generator loss reduces its distance from 0. Finally, the losses stabilise at 0.23 and -0.5 for the generator and the critic network respectively.

Regarding the healthy brain image structure that f-AnoGAN managed to learn from the train set, the eight images of Figure 20 clearly resemble the shape of a healthy brain CT. The colours are captured correctly both in the brain and out, as well as the lighter colour of the area surrounding the brain and representing the skull.

Experiment 3:

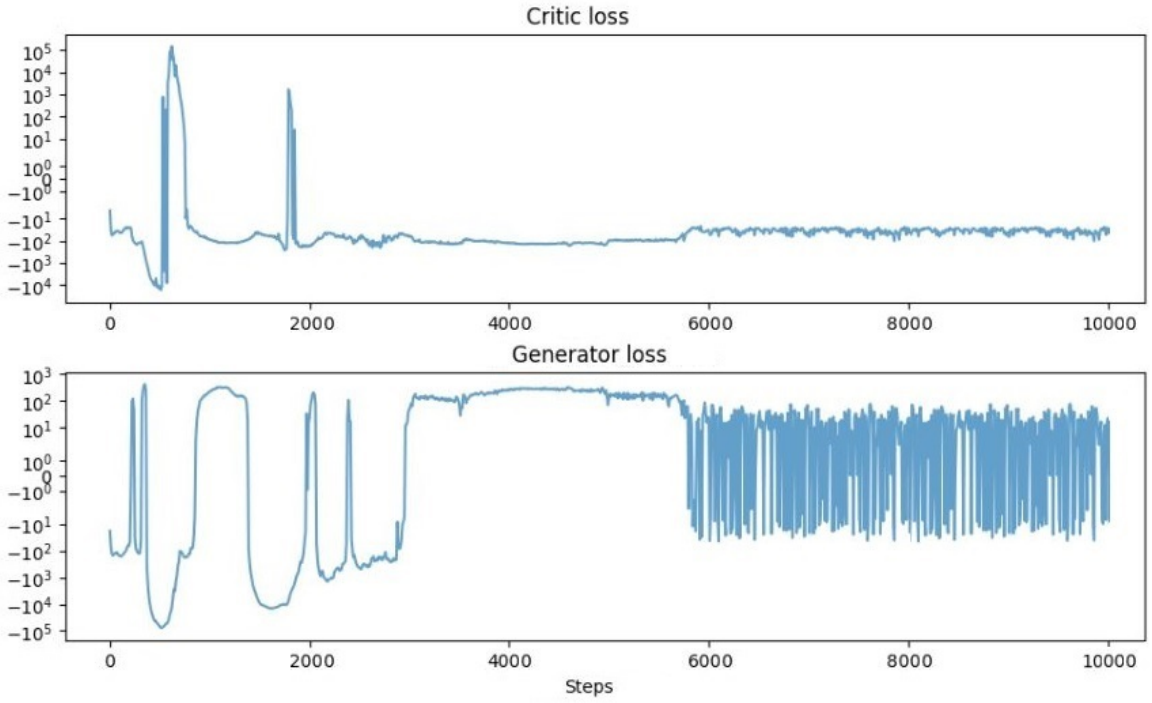


Figure 21: Losses of the networks on the training set for Experiment 3.

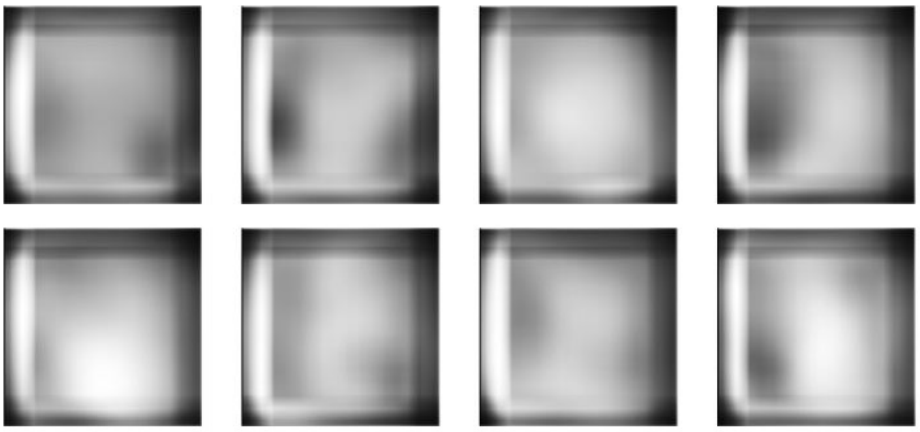


Figure 22: Structure of healthy brain CT images captured by the model for Experiment 3.

For Experiment 3 in Figure 21, the generator loss has two very big drops of scale -25,000 and

-20,000 before and after epoch 1,000, keeps rather steady from epoch 3,000 to approximately 5,700 and then fluctuates until the last epoch. Something similar happens to the critic loss. There are two sudden rises from negative to positive values at approximately epochs 500 and 1,800. Later on and until epoch 5,700 the critic loss also keeps rather steady at -100 and then fluctuates around -25. Lastly, the critic loss and the generator loss are -23.17 and 11 at epoch 10,000.

Final loss values:

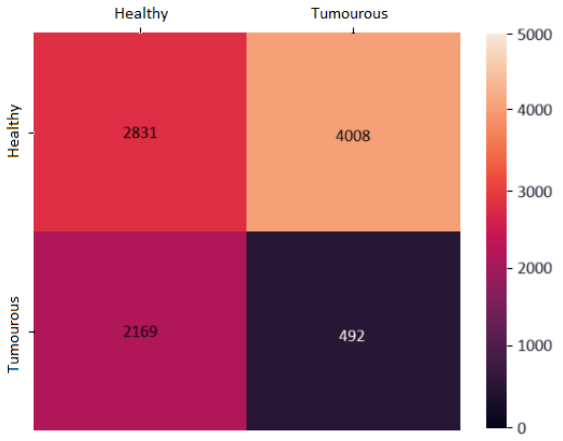
The results depicted in the images of Figure 22 vaguely resemble brain CTs, as they tend to keep the colour scheme of darker content inside of a lighter border, and the grayscale shades vary a lot so there is not enough consistence. The circular shape of the skull is distorted and, instead, the shapes are closer to squares.

Experiment	Critic Loss	Generator Loss
1 (VQ)	-23.51	9.73
2 (ViT)	-0.5	0.23
3 (VQ & ViT)	-23.17	11

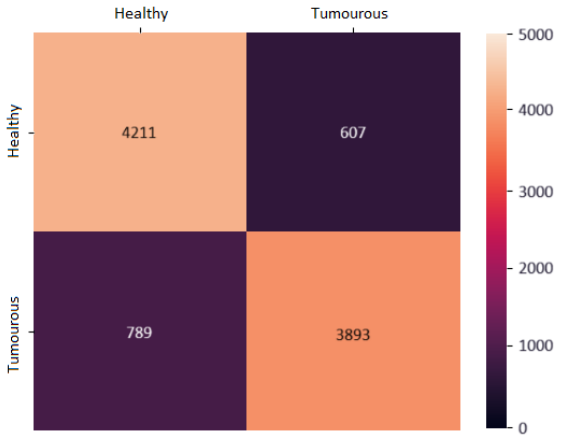
Table 4: Generator and Critic losses of the f-AnoGAN model after the training process.

Test results:

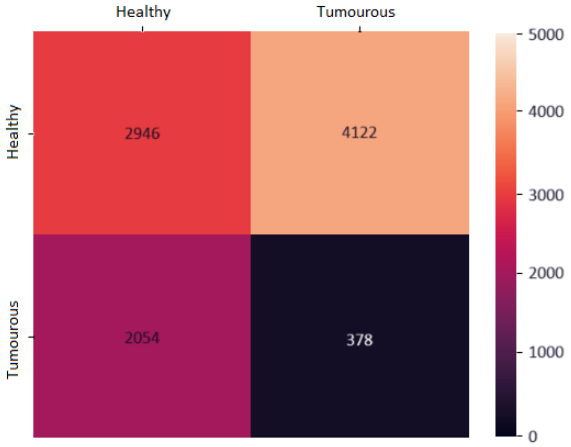
Regarding the testing of the model, 9,500 images of healthy and tumorous brain CTs are utilised as mentioned previously. In order to look at the efficiency of the experiments, three confusion matrices are created and it is easy to see the amount of each of the classes - healthy and tumorous - that is classified as the correct or wrong class.



(a) Experiment 1.



(b) Experiment 2.



(c) Experiment 3.

Figure 23: Confusion matrices for the testing results of the three experiments.

Obtaining the TP, FP, FN and TN values from the three confusion matrices, the following metrics are calculated:

Experiment	Accuracy	Precision	Recall	F1-score
1 (VQ)	0.388	0.414	0.566	0.481
2 (ViT)	0.809	**0.874**	0.842	**0.857**
3 (VQ & ViT)	0.344	0.417	0.589	0.486
Original	**0.811**	0.863	**0.848**	0.854

Table 5: Evaluation metrics for the baseline and the three experiments.

8 Discussion

Experiment 1 showed that applying VQ makes the training unstable and this is easily noticeable by its evaluation metrics. These metrics point toward a model that is not performing as well as desired. The accuracy score of approximately 38.8% suggests that the model's predictions are largely incorrect, indicating potential issues in its predictive capabilities. The low precision score of about 41.4% highlights a high rate of false positive predictions, meaning that the model incorrectly identifies instances as positive. The recall score of around 56.6% indicates that the model captures a moderate number of actual positive instances, but it is not as sensitive as desired. The F1-score of about 48.1% reflects an imbalanced trade-off between precision and recall, indicating room for improvement. In Experiment 1, there is clear scope for enhancing the model's predictive accuracy and balance.

In Experiment 2, the model achieved relatively high values across all evaluation metrics. The accuracy score of approximately 80.9% suggests that the model's predictions are correct for a significant portion of instances. The high precision score of around 87.4% indicates that when the model predicts a positive class, it is generally correct. Moreover, the recall score of about 84.2% demonstrates the model's capability to identify a substantial number of actual positive instances. The F1-score of approximately 85.7% shows that the model strikes a balanced trade-off between minimising false positives and false negatives. Overall, Experiment 2 appears to yield a well-performing model with a strong ability to classify instances correctly.

Regarding Experiment 3, the model's performance leans towards the first experiment's results. The accuracy score of approximately 34.4% suggests that the model's predictions are correct for a moderate to low proportion of instances. The precision score of around 41.7% indicates a moderate rate of false positive predictions, while the recall score of about 58.9% signifies that the model is relatively effective at capturing actual positive instances. The F1-score of approximately 48.6% reflects an imbalanced trade-off between precision and recall. Experiment 3 shows a slight potential for improvement compared to Experiment 1. However, there is still room for enhancing overall predictive accuracy.

One of the deductions from these results and their interpretation is that the way vector quantisation is used is not effective at all. The reason behind this is utilising VQ after its

input is encoded because, instead of discretising pixel values, the method descritises the captured image features. So, these are basically some numbers that do not follow any specific distribution. In a way, this can be considered as randomly placing numbers into bins and, thus, the resulting captured structure of the images is vague and random, not outlining the shape of a brain.

In contrast, the use of a Transformer with encoded input is efficient. The Generator network of f-AnoGAN is trained on encoding the already encoded input and so the significant features and information of the images are not lost. Nevertheless, the measures of Experiment 2 with values averaging at approximately 84% did not exceed the slightly higher success of the plain f-AnoGAN model, as the threshold is equal or higher that 85% for most of the measures. It is important to notice at this point that the size of the images used for the authors' experiments is significantly smaller as they used 64x64 images, while this study experiments on 256x256 grayscale images. This renders the comparison between the measures' values under question.

Finally, combining the significantly good performance of the Transformer method with the unsuccessful employment of the VQ method is expected to produce mediocre, but still non-satisfying results. This is also proved by the actual metrics' values, ranging from approximately 35% to 60%. Such numbers can also be achieved by randomly predicting the class of the images, a fact that renders the experiment as a failed one.

9 Conclusion

Advancements in technology have led to a demand for efficient image anomaly detection and pattern recognition. Image analysis technologies and generative algorithms are potential solutions, allowing accelerated testing while maintaining accuracy. GANs excel in generating realistic examples across domains like image-to-image translation and they have been widely applied to unsupervised anomaly detection and automating tasks such as fraud detection and medical diagnosis. The aim of this study is to give an answer to the two research questions of the Section 1 regarding the possibility of high quality images being utilised and generated for anomaly detection using GAN models and the contribution of the combination of GANs with state of the art techniques such as Vector Quantisation and Transformers.

During the experiments, 256x256 highly detailed images are generated in vast amounts to be used as input data for the main f-AnoGAN model. Three experiments are conducted where methods are added between the encoder and decoder of the Generator network of the model as follows: Learning Vector Quantisation for the first experiment, Vision in Transformers for the second experiment and both for the third and final experiment, in the order they are mentioned. The first experiment is not effective as it produces low evaluation metric values, - approximately 45% on average - due to Vector Quantisation's inability to discretised the captured characteristics of the data in the latent space. On contrary, the second experiment produces very high evaluation metric values, just under 90%. This leads to the belief that the encoder part of the Transformer is able to correctly encapsulate the characteristics of the latent space data. Lastly, the third experiment lies somewhere in the middle, closer to the results of the first one with average metric values of 50%, as the combination of an unsuccessful and a successful method could not prove to succeed.

Answering the research questions, it is indeed possible to use and generate high quality images using GAN models for anomaly detection and state of the art models such as Transformers and Vector Quantised technology can be useful in combination with GANs. First, with a lot of time and resources, even higher quality images (512x512, 1024x1024, etc.) can be generated and be used for a more accurate and detailed anomaly detection. Second, utilising the two models inside of f-AnoGAN's Generator network is an innovative method not broadly used before. The perfection of such an experiment can provide with an even

faster and more effective model that can prove to be useful in detecting irregularities in vast amounts of images, especially in domain such as the medical or astronomy.

Although better results can be acquired, there are some limitations. The resources needed are too many, as mentioned above, and this is limiting in many cases and one should consider the scale of their data - size and amount - and their goal in order to decide if this the right methodology they would like to use. Also, Vector Quantised technology may not be able to be combined in such a way with f-AnoGAN in order to be effective and produce decent results.

Left for future work, exploring the nature of the limitations and checking if any alterations to the way the methods are implemented may be helpful are necessary. Also, experimenting with various image sizes and the hyperparameters of the model and methods could produce even better results and burnish the aim of the study. Finally, another step could be to combine other state of the art methods with f-AnoGAN - e.g. Explainable AI - and take the experiments a step further.

References

[1] Muhammad Zaigham Zaheer, Arif Mahmood, M Haris Khan, Marcella Astrid, and Seung-Ik Lee. An anomaly detection system via moving surveillance robots with human collaboration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2595–2601, 2021.

[2] Lior Shamir, John D Delaney, Nikita Orlov, D Mark Eckley, and Ilya G Goldberg. Pattern recognition software and techniques for biological image analysis. *PLoS computational biology*, 6(11):e1000974, 2010.

[3] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *arXiv preprint arXiv:2103.04922*, 2021.

[4] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[6] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[7] 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.

[8] Daniel Sáez Trigueros, Li Meng, and Margaret Hartnett. Generating photo-realistic training data to improve face recognition accuracy. *Neural Networks*, 134:86–94, 2021.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[10] Xuan Xia, Xizhou Pan, Nan Li, Xing He, Lin Ma, Xiaoguang Zhang, and Ning Ding. Gan-based anomaly detection: A review. *Neurocomputing*, 2022.

[11] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.

[12] Xu Han, Xiaohui Chen, and Li-Ping Liu. Gan ensemble for anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4090–4097, 2021.

[13] Tahereh Pourhabibi, Kok-Leong Ong, Booi H Kam, and Yee Ling Boo. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, 133:113303, 2020.

[14] Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Deep learning for medical anomaly detection–a survey. *ACM Computing Surveys (CSUR)*, 54(7):1–37, 2021.

[15] Mahsa Salehi and Lida Rashidi. A survey on anomaly detection in evolving data: [with application to forest fire risk prediction]. *ACM SIGKDD Explorations Newsletter*, 20(1):13–23, 2018.

[16] John Sipple. Interpretable, multidimensional, multimodal anomaly detection with negative sampling for detection of device failure. In *International Conference on Machine Learning*, pages 9016–9025. PMLR, 2020.

[17] Ming C Hao, Daniel A Keim, Umeshwar Dayal, and Jörn Schneidewind. Business process impact visualization and anomaly detection. *Information Visualization*, 5(1):15–27, 2006.

[18] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

[19] Shuying Xu, Chin-Chen Chang, and Yanjun Liu. A novel image compression technology based on vector quantisation and linear regression prediction. *Connection Science*, 33(2):219–236, 2021.

[20] M Durgadevi et al. Generative adversarial network (gan): a general review on different variants of gan and applications. In *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, pages 1–8. IEEE, 2021.

[21] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54:30–44, 2019.

[22] Yingxue Pang, Jianxin Lin, Tao Qin, and Zhibo Chen. Image-to-image translation: Methods and applications. *IEEE Transactions on Multimedia*, 24:3859–3881, 2021.

[23] Marc W Chale. Generative methods, meta-learning, and meta-heuristics for robust cyber defense. 2022.

[24] Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar. Adversarially learned anomaly detection. In *2018 IEEE International conference on data mining (ICDM)*, pages 727–736. IEEE, 2018.

[25] V Arulkumar and P Vivekanandan. An intelligent technique for uniquely recognising face and finger image using learning vector quantisation (lvq)-based template key generation. *International Journal of Biomedical Engineering and Technology*, 26(3-4):237–249, 2018.

[26] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[27] Xiaoran Chen, Nick Pawlowski, Martin Rajchl, Ben Glocker, and Ender Konukoglu. Deep generative models in the real-world: an open challenge from medical imaging. *arXiv preprint arXiv:1806.05452*, 2018.

[28] Lore Goetschalckx, Alex Andonian, and Johan Wagemans. Generative adversarial networks unlock new methods for cognitive science. *Trends in Cognitive Sciences*, 25(9):788–801, 2021.

[29] Walter Hugo Lopez Pinaya, Sandra Vieira, Rafael Garcia-Dias, and Andrea Mechelli. Autoencoders. In *Machine learning*, pages 193–208. Elsevier, 2020.

[30] Alexander E Curtis, Tanya A Smith, Bulat A Ziganshin, and John A Elefteriades. The mystery of the z-score. *Aorta*, 4(04):124–130, 2016.

[31] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning internal representations by error propagation, 1985.

[32] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[33] Stephen Brooks. Markov chain monte carlo method and its application. *Journal of the royal statistical society: series D (the Statistician)*, 47(1):69–100, 1998.

[34] Francis Ysidro Edgeworth. Xli. on discordant observations. *The london, edinburgh, and dublin philosophical magazine and journal of science*, 23(143):364–375, 1887.

[35] Teuvo Kohonen and Teuvo Kohonen. Learning vector quantization. *Self-organizing maps*, pages 175–189, 1995.

[36] Oliver Kramer and Oliver Kramer. K-nearest neighbors. *Dimensionality reduction with unsupervised nearest neighbors*, pages 13–23, 2013.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[38] Michele Guerra, Abhijeet Parida, Daniel Rueckert, Mehmet Yigitsoy, and Shadi Albarqouni. Ctfsh: Full head ct anomaly detection with unsupervised learning. 2021.

[39] Arktis2022. Medical anomaly detection. Kaggle, 2022. https://www.kaggle.com/datasets/arktis2022/medical-anomaly-detection.

[40] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[41] Khairul Anuar Mat Said, Asral Bahari Jambek, and Nasri Sulaiman. A study of image processing using morphological opening and closing processes. *International Journal of Control Theory and Applications*, 9(31):15–21, 2016.