TUHH
Hamburg
University of
Technology

Nagalikhitha Reddipalli

Evaluation of State-Of-The-Art image generation models in Unsupervised Anomaly Detection using Brain MRI

Institute of Medical Technology
and Intelligent Systems
Building E | 21073 Hamburg
www.tuhh.de/mtec

Institute of Medical Technology and Intelligent Systems

A project paper written at the Institute of Medical Technology
and Intelligent Systems and submitted in partial fulfillment of the requirements for the degree Master of Science.

Author: Nagalikhitha Reddipalli

Title: Evaluation of State-Of-The-Art image generation models in Unsupervised Anomaly Detection using Brain MRI

Date: October 9, 2022

Supervisors: Finn Behrendt (MSc.)
Alexander Schlaefer (Dr.-Ing.)

Referees: Prof. Dr.-Ing. Alexander Schlaefer

Declaration

I hereby certify that this report has been composed by me and is based on my own work, unless stated otherwise. No other persons work has been used without due acknowledgment in this report.

Date: 09.10.2022 R. Nagalikhitha .....
(Signature)

Contents

List of Figures

List of Tables

Abstract

Detection of lesions in Medical imaging is time-critical and error-prone. One of the demanding tasks is the detection of abnormalities in Magnetic resonance imaging (MRI), which radiology experts often interpret. A decision assistance tool for segmenting and detecting lesions would be required to reduce the efforts of radiologists. Unsupervised anomaly detection is extensively studied in the field of Deep Learning, as this method uses healthy images for training rather than unhealthy ones, which are difficult to obtain. There have been studies on Variational Autoencoders (VAE), which utilize 2D brain slices. In this work, we have studied VAE, Vector quantized Variational Autoencoders (VQVAE), and Vector quantized Generative Adversarial Networks (VQGAN) in sample-wise, slice-wise, and pixel-wise fashion, along with the effect of latent sizes on each model. Models' effectiveness on imagery other than brain imaging is evaluated.

1 Introduction

Magnetic Resonance Imaging of the brain is the most frequently employed high-resolution image modality in radiology for diagnosing and treating various neurological ailments [1]. This process is time-critical and cumbersome and prone to errors [2]. As per studies, about 5-10% of pathologies have been unrecognized [3]. With the increase in the necessity of MRI, radiologists have to interpret more images simultaneously. Thus, providing an additional support tool for automatic detection and segmentation would ease the burden on experts and aid inexperienced radiologists.

Recently, with the advancement in machine learning, various supervised learning models have shown outstanding performance in disease detection in imaging modalities. Supervised learning algorithms are more constraining in comparison with unsupervised learning, as the former needs labeled data, which is costly to obtain as human experts perform it [4]. Heavy data imbalance in the medical sector can also affect the efficiency of supervised learning [5]. In contrast to supervised learning, Unsupervised Anomaly Detection (UAD) learning utilizes readily available healthy images to train networks and differentiate anomalous data from healthy data [6], thus being the point of interest for today's medical imaging field [7]. This approach is similar to how domain experts detect the abnormalities in scans without being explicitly trained in pixel-level anomaly detection. These unsupervised methods are classified into reconstruction-based [8, 9, 10] and clustering-based methods [11].

Even though supervised learning outperforms unsupervised learning, the latter has two significant advantages over the former. Firstly, learning of latent distribution doesn't require annotated data, which is very costly and even error-prone while segmenting. Secondly, the nature of detecting outliers of these methods can detect any kind of outlier independent of prior knowledge about their appearance and features [12]. There are wide range of architectures for unsupervised learning, ranging from Auto encoders to Generative Adversarial Networks. Variational Auto Encoders (VAE) have shown remarkable performance in detecting outliers in 2D brain slices [12, 13]. AE and VAE use continuous latent space representation. This latent space fails to capture local information of the images, which is highly important for differentiating regions of brains in T1 weighted images [14]. To tackle this problem, [15] has developed Vector Quantized Variational Auto Encoders (VQVAE), which used discrete latent space representation. Furthermore, [16] has developed Vector Quantized Generative Adversarial Networks (VQGAN), which combines the concepts of VQVAE, GAN, and transformers.

In this study, we compare the three significant architectures: Variational Auto Encoders (VAE), Vector Quantized Variational Auto Encoders (VQVAE), and Vector Quantized Generative Adversarial Networks (VQGAN) for the 2D slices of MRI scans. Further, the effect of different hyper parameters in each model is studied.In this project, we focus on T1-weighted MRI scans, contrary to [8, 9, 17, 18], as they are widely available. 1856

T1-weighted healthy images are used to train the models and tested on publicly available Brain Tumor Segmentation (BraTS) and Anatomical Tracings of Lesions After Stroke (ATLAS) data.

This project work is structured as follows: In chapter 2, we explain the basic principle of Magnetic Resonance Imaging and various theoretical concepts of machine learning. Further, we describe various state-of-the-art models in chapter 3. Later on, in chapter 4, the methodology and data sets used are elucidated. We present the results of the current work and discuss them in Chapters 5 and 6, respectively. We conclude and wind up this project with an outlook.

2 Background

We describe the foundational theories that are employed in this current work, beginning with the principle of the core concept, Magnetic resonance Imaging. Later, explaining the concepts of machine learning including artificial neural networks, convolutional neural networks and deep learning.

2.1 Magnetic Resonance Imaging

MRI is one of the non-invasive imaging modalities for the human brain and central nervous system[2]. Magnetic resonance imaging is based on Nuclear Magnetic Resonance (NMR) principle. The primary NMR deals with the interaction of certain atomic nuclei, radio frequency, and strong magnetic field [19]

MR images are made up of a series of voxels or volume elements. Each square on the image corresponds to the volume of tissue on the body. The MR machine is designed to measure NMR signals from each of these volumes, localize them in 3d space, and make us visual pictures [20, 21]. Water molecules make up a major portion of the body. Each molecule of water has two nuclei or protons. MRI uses magnetic properties of the hydrogen atom and its high prevalence in the human body. Usually, all the protons in the body are randomly oriented and cancel each other's magnetic power resulting in zero net magnetic momemt. When we place ourselves in a strong magnetic field, all the protons will line up parallel to the magnetic field as shown in 2.1. Protons in low energy state will be in the direction of the magnetic field, whereas high energy state protons line up in the exact opposite direction leading to precession motion. The rate of rotation, Larmor frequency f, is directly proportional to the strength of the local magnetic field B_0, as defined by Larmor precession.

$$f = \gamma.B_0 \tag{2.1}$$

where γ is Gyromagnetic Ratio. The more protons in the direction of the strong magnetic field will result in longitudinal magnetization like 2.2. When the radio frequency is applied to the protons, some protons will shift from low to high energy, leading to zero longitudinal magnetization (M_Z). Furthermore, radio frequency makes protons synchronize and spin together, leading to transverse magnetization (M_{XY}) similar to 2.2, perpendicular to M_Z. As the protons precess, the transverse magnetization will produce small measurable currents. After removing the radio frequency signal, protons will be relaxed and move to their original positions. Initially, there will be T2-relaxation or spin-spin relaxation, where transverse magnetization is zero. Furthermore, the protons in high energy state will fall back to low energy states, dissipating heat to the surrounding tissues, resulting in T1-relaxation or Spin-Lattice relaxation. To improve the contrast between particular tissue types, MRI sequences take advantage of various proton spin relaxations. T1 and T2 relaxation times vary with the input of radio frequency signals

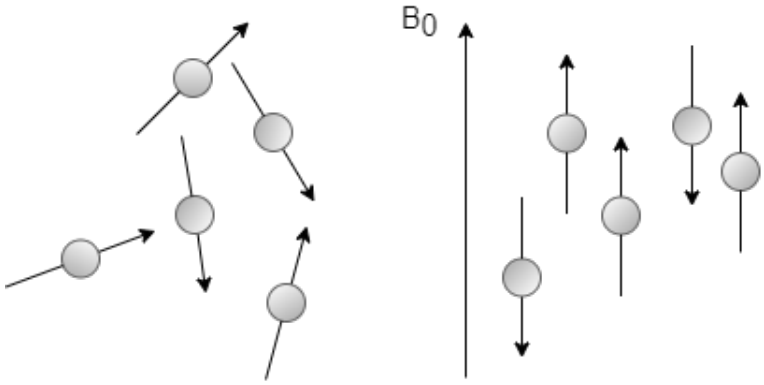


Fig. 2.1: Schematic diagram of protons without (left) and with (right) magnetic field B_0 adapted from [22].

or the repetition time (T_R) or echo time (T_E) for water molecules and different tissues in the body. T2 weighted image will have long repetition and echo time, whereas the T1 weighted image has less repetition and echo time, reducing the effect of M_Z. Fat tissue relaxes very quickly, which causes it to realign with B_0 quickly, appearing bright ("hyper-intense") on T1-weighted images. In contrast, water realigns slowly leading to low intensity ("hypo-intense") in images. In T2 relaxation, fat and water tissue signal intensities are reversed, in contrast to T1, making fat tissue seem dark and the signal intensity of the water appear high.

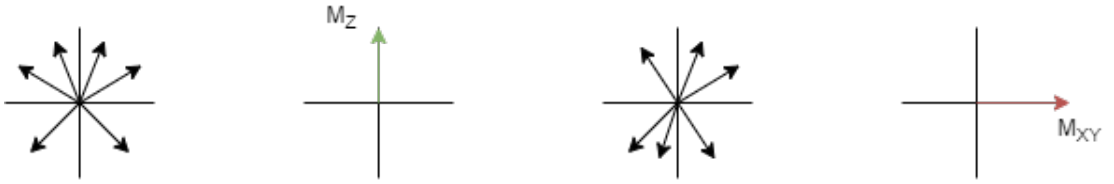


Fig. 2.2: Spin-Lattice(left) and Spin-Spin(right) relaxation adapted from [23].

2.2 Machine Learning

An area of artificial intelligence, machine learning, employs routinely applying algorithms to synthesize the underlying connections between data and information. Arthur Samuel [24] puts Machine learning as a “field of study that gives computers the ability to learn without being explicitly programmed.” In this section, we briefly describe the fundamentals of Machine learning, including supervised and unsupervised learning, and later move on with the description of Multilayer perceptron, Convolutional neural networks (CNN) taken from books [25, 26].

Supervised and Unsupervised learning

Supervised learning methods extract associations between data and a designated label. These algorithms utilize a training dataset associated with corresponding annotated labels

to develop a machine learning model for predicting labels of new datasets. Supervised learning requires healthy and unhealthy samples in the medical image analysis setting. Each pixel of the unhealthy image has to be annotated and given labels. In contrast to supervised learning, images in unsupervised or self-supervised learning doesn't need to be annotated for training. This method depends entirely on features of each sample in dataset and thus is advantageous in disease detection, removing the tedious label annotation process. Unsupervised learning tried to obtain meaningful features without the usage of labels. In this work, we explore various unsupervised learning methods and draw comparisons between them.

Hyperparameters

Parameters that govern the learning process and determine model parameter values are referred to as Hyperparameters. These are input parameters we choose before the training of the model and directly influence the learning model. These are often termed as external parameters, as they cannot be changed during training. There are various hyperparameters such as optimization algorithms, choice of activation functions, number of epochs to be trained, dropouts, embedding vectors, latent size dimensions, and many more.In our work, we study the effect of some of the hyperparameters, latent sizes in VAE, embedding dimensions and vectors in VQVAE, and in VQGAN.

2.3 Deep Learning

Deep learning is the branch of machine learning, a sub field of Artificial Intelligence. This section describes starting from elementary component Artificial Neuron to advanced CNN.

2.3.1 Artificial Neuron

An artificial neuron is the primary entity of computation in a neural network. This neuron resembles the biological neuron, which gets fired when information is received and transmits the data to the connecting neurons. Even though the vast portion of the human brain is not explored yet, AN can be analogous to human neurons [27].

$$a = f(n) = f(w \cdot p + b) \tag{2.2}$$

As shown in the 2.3, each neuron receives information from either external sources or neighboring nodes, and these m inputs $p \in \mathbb{R}^m$ are associated with weights ($w \in \mathbb{R}^m$). To the weighted sum of these inputs, an additional bias term b is added which becomes n. An activation function is applied to n giving a as output as in equation 2.2. In most modern-day networks, Rectified Linear Unit (ReLU) activation function $f(n) = max(0, n)$ is employed.

2.3.2 Multi Layer Perceptron

Multilayer perceptron stacks many layers of combinations of neurons. The first layer is known as the input layer, as it receives input, and the last layer is named the output layer, as it gives output. The layers in between are called hidden layers. Every layer

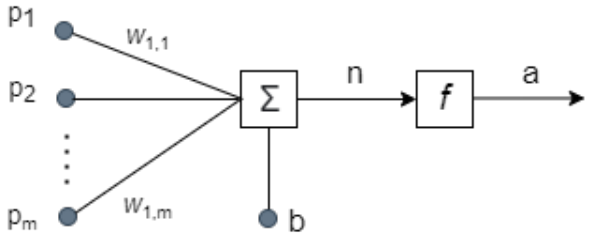


Fig. 2.3: Schematic diagram of Single Neuron with activation function adapted from [27].

has its weight matrix W, net inputs,bias, and outputs, as shown in the 2.4 and weights calculated according to 2.3.

$$y = f^2(W^2 f^1(W^1 \cdot p + b^1) + b^2) \tag{2.3}$$

Artificial neurons and MLP are the heart of Deep learning. With today's advancement

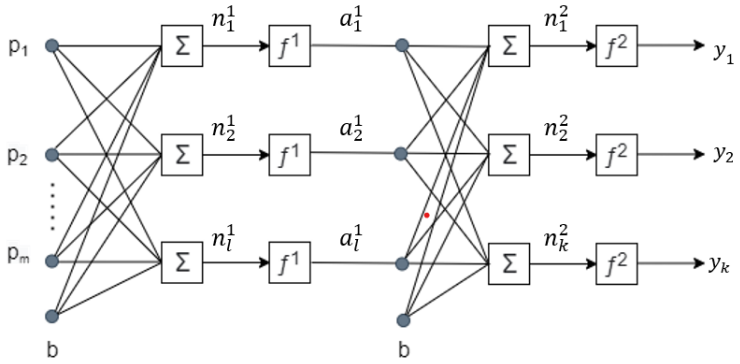


Fig. 2.4: Schematic diagram of Multi-layer Perceptron adapted from [27].

in computation power, there are several layers of neurons in almost every architecture published. In the following section, we further discuss another important concept, CNN, in deep learning.

2.3.3 Convolutional Neural Network

Convolutional Neural Networks are the basis of various computer vision algorithms like image recognition, detection, etc. As defined by Ian Goodfellow in his book [25], CNN is a type of neural network that process grid-like topological data. Neural networks employ mathematical operation called convolution. The sophisticated technology and extensive research in deep learning make it difficult to recommend the best architecture as one or the other is published frequently. Nevertheless, all the architectures lie on the below-explained building blocks of CNN. Convolution takes advantage of sparse interactions, parameter sharing, and equivalent representations. We have two terms for CNN, one input matrix and another feature detector or kernel. Various kernel sizes, which are helpful in obtaining features, can be employed depending on the architectures. In our current work, we utilize kernel sizes of 1×1, 3×3, 5×5. The kernel, which leads

to sparse interactions, will slide over the image grid depending on the step size called stride and performs convolution operation, and outputs fewer values than the original size. The feature maps or output layer corresponds to edges and textures in the input

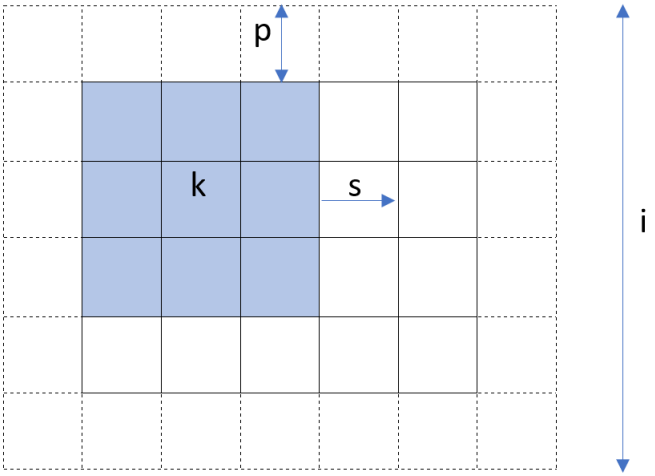


Fig. 2.5: Representation of parameters in Convolutional Layers.

image. Depending on the number of layers employed, feature extraction varies from generic to complex features. Parameter sharing is implemented in the form of kernels, as every element doesn't need to have specific weights, thus reducing memory storage. Equivariance to translation is nothing but change in input leads to change in output. The formula 2.4 determines the convolution layer output for a specific input value.

$$o = \frac{i + 2p - k}{s} + 1 \tag{2.4}$$

where o is output, i is input image dimensions, k is a kernel, s is stride, and p is padding as illustrated in 2.5. Padding helps achieve the desired output, thus allowing the kernel to even look at edges. The process, as mentioned earlier, is usually used for downsampling, reducing the image dimensions, and extracting features. For obtaining images from the features, upsampling or transpose convolution is implemented. We further discuss these in the upcoming chapters.

3 State-of-the-Art

In this chapter, we demonstrate the principle of working of UAD and shed light on standard models ranging from Auto encoders to Vector Quantized Generative Adversarial Networks. Having described the state-of-art models, we briefly discuss the related work.

3.1 Unsupervised Anomaly Detection

With recent advancements in DL, Unsupervised Anomaly detection has become one of the most researched topics in medical image analysis, as it reduces the load of annotations and uses vastly available normal images [2]. In contrast to Supervised learning, where training requires an ample amount of segmented data [28] for a specific pathology and can only detect the pathologies that have been trained on [2], UAD detects the outliers irrespective of pathologies, as they are trained on healthy data. This section explains the basic principle of UAD using autoencoders.

Principle of UAD

Even without much training, humans can properly segment abnormal-looking areas and diagnose most lesions instantaneously. The focus of UAD is to mimic this human behavior, to identify the abnormalities without being explicitly trained on them. Contrary

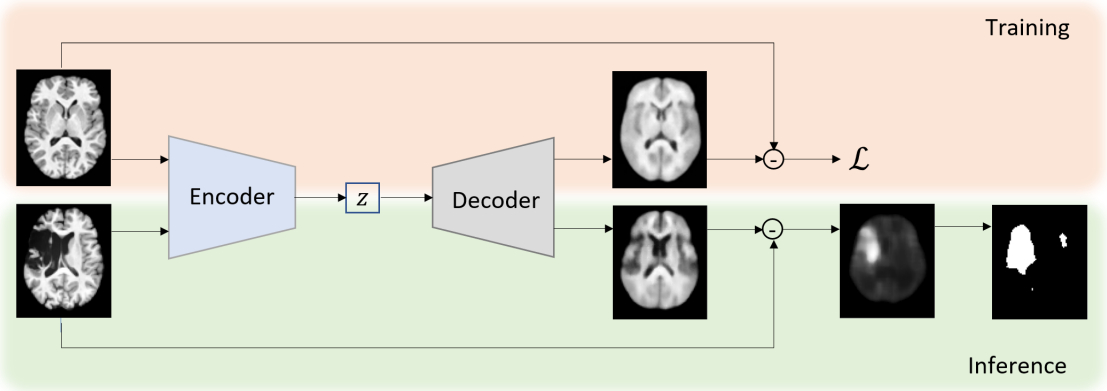


Fig. 3.1: Principle of UAD in our study adapted from [12].

to well-established supervised methods, we feed healthy data to CNN models and provide them with required prior information to let models detect the lesions. As shown in the figure 3.1, we divide our network into two phases, Training and Inference. In the training phase, we obtain latent space or hidden representation from the encoders, and the sample from latent space is provided to the decoder network to reconstruct images. We further obtain reconstruction loss ($\mathcal{L}$) which is used for updating latent spaces. Depending on the different architectures, we consider additional loss terms along with

our base reconstruction loss for training. We demonstrate the latent spaces, VAE, and other models in the forthcoming sections. In the evaluation phase, the model tries to reconstruct the images based on health data, thus leading to higher reconstruction errors. We leverage this notion and further do post-processing steps to acquire the lesion. We measure our output segmentations sample-wise and pixel-wise, which we describe in upcoming chapters.

3.2 Latent Space

Most of the higher dimensional data can often be represented as lower dimensions, which are sufficient for the generation of images. These hidden representations of raw input data distribution are titled Latent spaces. Most architectures use continuous representations, while recently, research has focused on discrete hidden spaces. Section 3.2 briefly explains the differences between the two.

Continuous Space

Consider the data $x \in \mathbb{R}^n$, is produced from the lower dimension $z \in \mathbb{R}^m (m < n)$, by the equation

$$x = A \cdot z + v \tag{3.1}$$

3.1, where v is independent Gaussian distribution of n dimensions. We get to access only raw data but not the lower-dimensional representation. The classical unsupervised algorithm, Principal Component Analysis, tries to find this hidden representation. While PCA [29] is concentrated more on linear representations, our latent spaces are designed to determine non-linear representations. These hidden representations can also be considered as compressed information from a huge data pool. Apart from the continuous latent space explored in AE [30] and its successors, we exploit discrete latent space in VQVAE and VQGAN.

Discrete Spaces

So far in 3.2, we have handled latent representations in the form of continuous distributions. As described in [15], most of the real data is in discrete representations, thus motivating to shift hidden representations from continuous to discrete space. Transformers, a recent breakthrough in natural language processing, also leverages this concept. [15] has tried to implement this idea by replacing continuous sample space with discrete codebooks, compressing the information bottleneck, and enforcing regularization effects. This compressed data can effectively retain more semantical information and discard sample-specific information. We further establish the application of the codebook in the following section.

3.3 Network Architectures

Having explained basic principle of UAD and the types of hidden representations, in this section we demonstrate state-of-the-art architectures which employs continuous and discrete latent representations.

Auto Encoders

An Autoencoder is the simplest version of neural networks employed in unsupervised anomaly detection that finds non-linear hidden representations for given input data [30]. It has two divisions, encoder networks, $z = f_\theta(x)$ with θ parameters, and a decoder network $\hat{x} = g_\phi(z)$ with ϕ parameters, where x is input, z is latent vector representation and $\hat{x}$ is reconstruction from compressed representation. The reconstruction loss $\mathcal{L}$ is obtained from mean square error (MSE) from input x and reconstructions $\hat{x}$. The entire autoencoder network is put in the following formula 3.2,

$$arg \min_{\theta,\phi} \mathcal{L}_{Rec}(x, \hat{x}) = arg \min_{\theta,\phi} \mathcal{L}_{Rec}(x,\ g_\phi(f_\theta(x))) \tag{3.2}$$

The error can be mathematically represented as $|| \ x - \hat{x} \ ||_2^2$ or $log(p(x|z))$. The hidden part in the autoencoder can also be termed the bottleneck, as it squeezes the entire data into lower dimensions. z should always be less than x, forcing the encoder to find the essential attributes.

Variational Auto Encoders

Variational autoencoders introduced by Kingma et al. in [31] are one of the standard approaches in the field of unsupervised learning. The significant difference between auto encoders and VAE is in the structure of bottleneck representation. Semantically related data points should be clustered in our latent space, while semantically unrelated data points should be spread apart. Preferably, the majority of the distribution of the data should not reach infinity and instead take up minimal space in hidden space. The limitation of autoencoders is latent space that can extend up to infinity, making networks memorize the data. Thus researchers have come up with a model with applied prior distribution to the network. VAE has both neural network and probabilistic

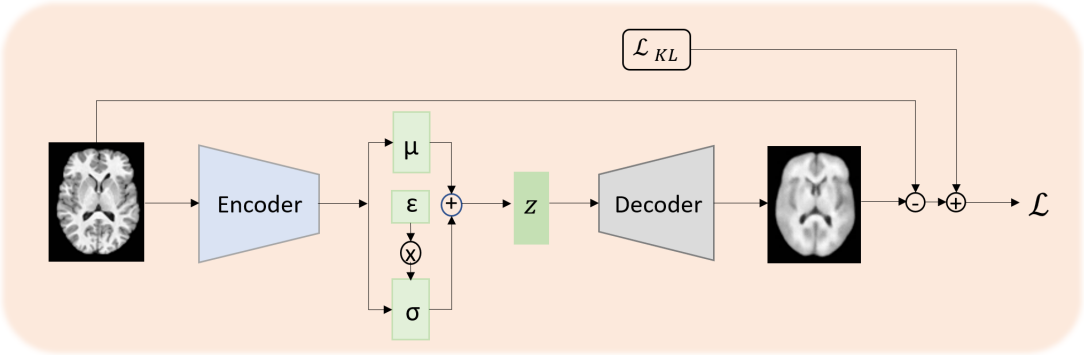


Fig. 3.2: Schematic diagram of VAE.

models perspective. We will dive into each viewpoint briefly. VAE is almost entirely similar to the auto encoders as shown in 3.2, with an encoder $q_\theta(z|x)$ that outputs Gaussian probability distribution, decoder $p_\phi(x|z)$ that takes sampled z which follows reparameterization trick with the formula 3.3, and latent spaces z.

$$z = \mu + \epsilon\sigma \tag{3.3}$$

In the equation 3.3, μ is mean, σ is standard deviation and ϵ is a random value drawn from prior distribution, $Normal(0, 1)$. We calculate reconstruction loss with log likelihood $log\ p_\phi(x|z)$. We can split loss function l_i of every data point x_i as 3.4.

$$l_i = -\mathbb{E}_{z\sim q_\theta(z|x_i)}[log\ p_\phi\,(x_i|z)]\ + KL[q_\theta(z|x_i)||p(z)] \tag{3.4}$$

$$\mathcal{L}_{VAE} = \mathcal{L}_{Rec} + \ \mathcal{L}_{KL} \tag{3.5}$$

The first term in equations 3.4, 3.5, $\mathcal{L}_{Rec}$ is reconstruction loss, which helps the network to learn for better reconstructions. The higher the reconstruction error, the poorer the reconstructions. Hence, the model learns to reduce the reconstruction error. The second term is Kullback-Leibler divergence between distribution of encoder, $q_\theta\,(z|x)$ and prior distribution g(z) is $\mathcal{N}(0, 1)$, with mean zero and standard deviation one. If the encoder produces the z, that are not consistent with standard Gaussian distribution, the loss function acquires a penalty in the form of KLD. This $\mathcal{L}_{KL}$ term helps to keep different semantical information diverse while keeping similar information sufficiently together. From the probabilistic method approach, we can say latent variables z are taken from prior distribution $p(z)$ and concerning to z, data x have a conditional probability or likelihood $p(x|z)$. By the above two terms, we can define a joint distribution $p(x, z) = p(x|z) \cdot p(z)$. The goal is to determine posterior distribution $p(z|x)$, which follows the Bayes theorem 3.6,

$$p(z|x) = \frac{p(x|z) \cdot p(z)}{p(x)} \tag{3.6}$$

As we cannot calculate the evidence term $p(x)$ directly, approximate it by posterior $q_\lambda(z|x)$. KLD term says how close approximation is to true posterior.

$$\mathbb{KL}\ (q_\lambda(z|x))\ ||\ q(z|x)) = E_q\,[log\ q_\lambda\ p(z|x)] - E_q\,[log\ p(x, z)] + log\ p(x) \tag{3.7}$$

As KLD is always greater than 0, maximizing ELBO, instead of minimizing Kullback Divergence, which are equivalent, will make the entire equation computationally tractable.

$$log\ p(x) = ELBO(\lambda) + \mathbb{KL}\ ((q_\lambda(z|x))\ ||\ q(z|x)) \tag{3.8}$$

$$ELBO(\lambda) = \mathbb{E}_{q\lambda}\,(z|x)\ [log\ p(x|z)] - \mathbb{KL}\ (q_\lambda(z|x))\ ||\ q(z|x)) \tag{3.9}$$

By further modifications as mentioned [31], we obtain the equation 3.8, which is similar to the equation 3.9. Thus, we can draw the significance of Kullback Divergence in the variational Autoencoders.

Generative Adversarial Networks

Generative Adversarial Networks [32] are one of the standard generative deep learning models. To train a generative model with GANs, we conveniently frame the task as a supervised learning problem with two sub-models: the generator model, which we train to create new examples, and the discriminator model, which tries to categorize examples as either real or fake. Here generator and discriminator play a strictly competitive game against each other.

As shown in the figure 3.3, a generator model G takes a sample from latent dimensions

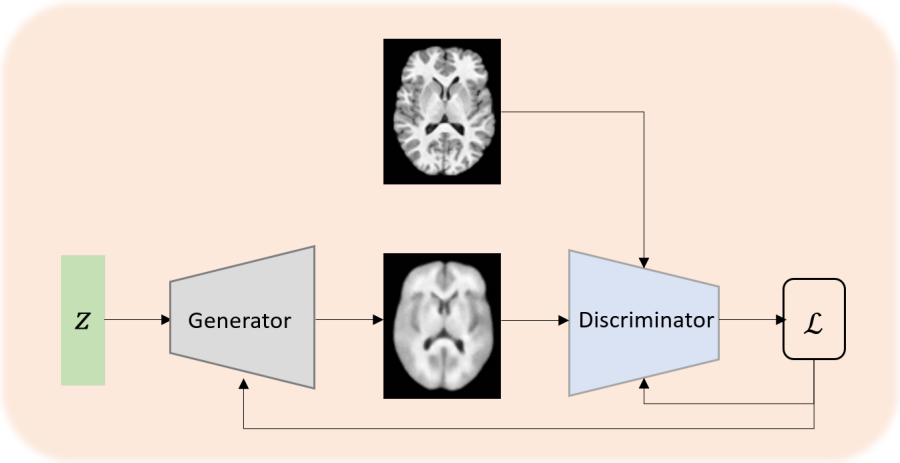


Fig. 3.3: Schematic diagram of GAN.

z and reconstructs the image, $x = G(z)$. This image and the actual image are fed to discriminator network D, which distinguishes between generated and original images. The objective of the discriminator is to maximize, whereas the generator tries to minimize the entire equation 3.10.

$$\mathcal{L}_{GAN} = \min_{G} \max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[log\ D(x)] + \mathbb{E}_{z\sim p_z(z)}[log\ (1 - D(G(z)))] \tag{3.10}$$

The first term interprets the discriminator's output on real images, and the latter represents prediction on generated images. Even though optimization of the equation is complicated and computationally expensive, GANs showed promising results for image generation. As GANs do not reconstruct but generate the images from latent space, we employ this model along with other models in our pipeline as described in AnoGAN [33].

VQVAE

VQ-VAE [15] is an extension of the standard variational auto encoders with the additional discrete codebook component. The major difference between VQVAE and standard VAE is that continuous space is replaced with discrete and prior value is learned instead of being constant [15].

As presented in the figure 3.4, the encoder generates a latent vector from an input image, then compares it to each embedding vector in the codebook to determine closest vector, depending on the euclidean distance, $z_q = argmin||z_e(x) - e_i||_2$, where $z_e(x)$ is output vector of encoder, e_i is ith embedding vector in codebook. For image reconstruction, the decoder receives the corresponding quantized codebook vector z_q. As argmin operation is not differentiable, we copy the decoder gradient $\nabla_z L$, directly to the encoder for training purpose, which is set to 1 concerning the encoder and resulting codebook vector and zero regarding other vectors.

$$\mathcal{L}_{VQ} = \mathcal{L}_{Rec} + \mathcal{L}_{alignment} + \mathcal{L}_{commitment} \tag{3.11}$$

$$log(p(x|q(x))) + ||sg[z_e(x)] - e||_2^2 + \beta||z_e(x) - sg[e]||_2^2 \tag{3.12}$$

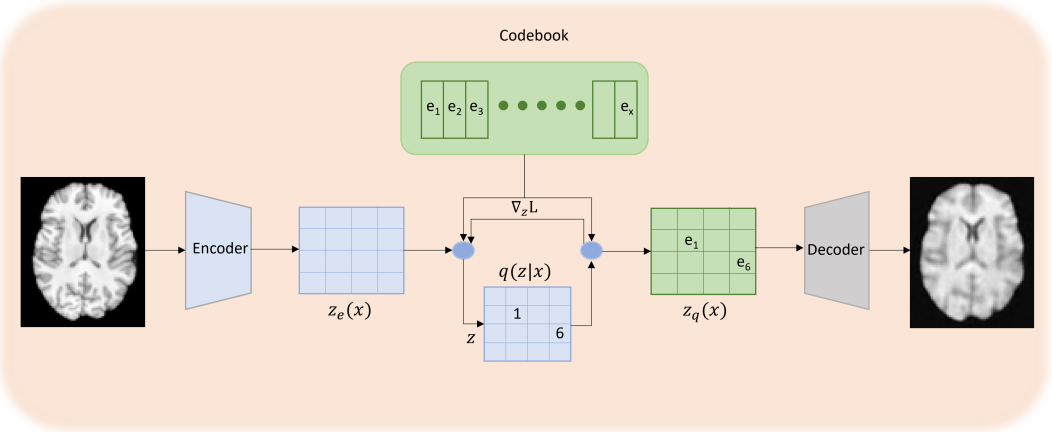


Fig. 3.4: Schematic diagram of VQVAE adapted from [15].

Similar to the encoder and decoder, the codebook learns using gradient descent. Learning codebook vectors that align to the encoder output is bidirectional. The loss function consists of three terms, reconstruction loss, codebook alignment loss, and codebook commitment loss as in 3.11, 3.12. Reconstruction loss is calculated by the mean square error of the original and reconstruction images. Codebook alignment loss helps bring the selected embedding vector close to the output of the encoder, setting a stop gradient on the encoder output. Conversely, codebook commitment loss places a stop gradient on the codebook vector to get the encoder output to commit to the nearest codebook vector. A hyperparameter β scales the importance of commitment loss. We employ VQVAE in our current work, and compare the outputs for various hyperparameters, embedding vectors and embedding dimenions.

VQGAN

Vector Quantized Generative Adversarial Network (VQGAN) [16] extends VQVAE by adding a discriminator network, which tries to identify real and reconstructed images. We use VQGAN without transformer model from the paper [16] in our work, as transformers are computationally expensive and require extended training periods.

VQGAN training is similar to [15] up to the reconstructed images from the decoder, which are later fed to the discriminator along with normal healthy data. In VQGAN, the adversarial loss of the discriminator component is added, and the $\mathcal{L}_{rec}$ of the 3.12 is substituted with the perceptual loss for higher visual quality. The loss function $\mathcal{L}_{VQGAN}$ is formulated as follows 3.13,

$$\mathcal{L}_{VQGAN} = \underset{E,G,Z}{arg\,min}\ \underset{D}{max}\ \mathbb{E}_{x\sim p(x)}[\mathcal{L}_{VQ}(E,G,Z) + \lambda\mathcal{L}_{GAN}(\{E,G,Z\},D)] \tag{3.13}$$

where E, Z, G, D are encoder, codebook, decoder and discriminator respectively. The λ, that scales the discriminator loss term is estimated by the formula 3.14 using perceptual

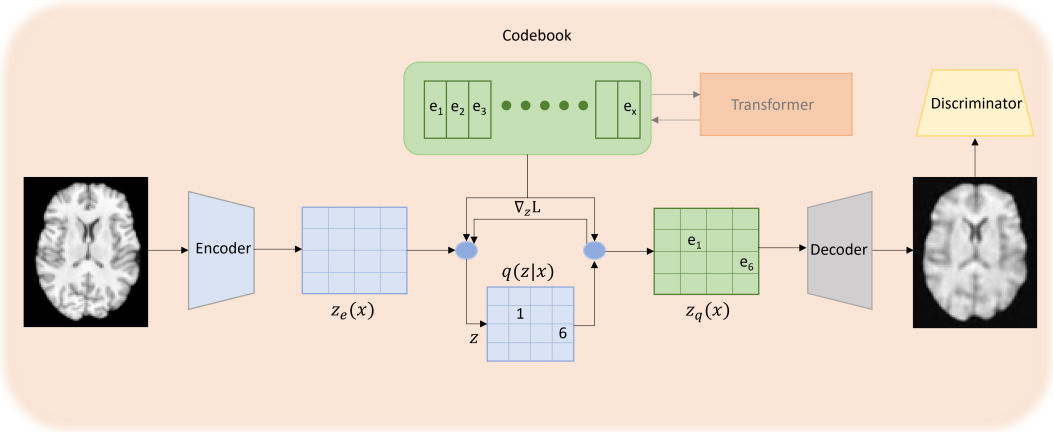


Fig. 3.5: Schematic diagram of VQGAN.

reconstruction loss and gradient with respect to the final layer.

$$\lambda = \frac{\nabla_{GL}\left[\mathcal{L}_{Rec}\right]}{\nabla_{GL}\left[\mathcal{L}_{GAN}\right] + \delta} \tag{3.14}$$

3.4 Related Work on 2D Slice-based UAD

This section discusses the related work on unsupervised anomaly detection of 2D slices of Brain MRI images using VAE and its successors.

Christoph Baur et al. have drawn a comparison on basic auto encoders and VAE to their extended versions of AE-GAN and AnoVAEGAN for Multiple Sclerosis (MS) segmentations [8]. They chose inhouse data with FLAIR and T1 images using the CurvatureFlow preprocessing method to denoise and later normalize the samples the samples after skull stripping using the tool ROBEX. Twenty axial slices from the middle regions with the size of 256×256 are selected and conducted experiments for varying latent size dimensions. Further post-processing of reconstruction is performed using median filter and thresholding. Dice score is reported for comparison on various procedures. They concluded dense latent space of AE and VAE models had underperformed their counter spatial latent space.

In [28], Xiaoran Chen et al. have compared VAE, adversarial AE, and Bayesian AE with supervised techniques Gaussian Mixture models, in which they declared supervised learning models performed way better than unsupervised and proposed a vast scope for development. In this work, they have trained on Cambridge for Ageing and Neuroscience (Cam-CAN) [34] and tested on brain tumors (BraTS) and stroke lesions (ATLAS) with sample sizes 256×256 and 128×128. They have used Area Under Curve and maximum Dice score as the mode of metrics. Similar to [9] , they also declared spatial latent spaces are better than dense spaces. Further, T2- weighted images for tumors have shown significant detection in comparison with T1-weighted images.

Zimmerer et al. have investigated anomaly detection in different levels of brain structure perspective [35]. They exploited the limitations of VAE for constructing and learn-

ing high-level features by using two groups of encoding and decoding parts, which is motivated by the works VQVAE-2 and Principal Component Analysis (PCA). The reconstructions are divided into granulated and more refined brain structures. The network is trained on 800 images of the HCP dataset and tested on 200 HCP images for reconstructions and on BraTS for abnormality identification along with the previously mentioned data. Then the entire network is tested for two sectors, one for reconstructions and the other for outlier detections, which used the MSE score and AUROC, AP score, respectively. These metrics are later used to compare VQVAE-2, low VAE, high VAE, and original pchVAE.

Baur et al., in their other publication, have attempted to draw an appropriate comparison on various state-of-the-art models [12]. They have used the same datasets, single architecture and resolution, and similar pre and post-process techniques for better comparisons. VAE has proven to be the best performer among AE and constrained AE in constraining effects on the hidden manifold. For most datasets, dense latent space networks have outperformed their spatial counterparts in AE and GMVAE models. They also suggested that GMVAE, to a large extent, depends on the type of datasets. In f-AnoGAN and AnoVAEGAN, the former does not produce clear images but maintains anatomical segments, unlike the latter. In experiments conducted, the Monto Carlo method does not perform well as expected. Restoration-based approaches surpassed reconstruction-based procedures. Amongst all, VAE restoration has excelled, and in reconstruction processes, f-AnoGAN is declared to be better in terms of dice score, AUPRC, AUROC, and other estimators.

Xiaoran Chen et al., in their other work, have compared VAE with AAE by tuning the effect of the divergence parameter [9]. Training of model is done using T2-weighted Human Connectome Project (HCP) data and evaluated on BraTS dataset. All the images are histogram normalized and standardized to zero mean and variance one. Further bias correction is also applied to the BraTS dataset. This publication has considered significantly reduced dimensions 32×32 for training and evaluation, and AUC has been considered to be the metric value. They demonstrated AAE which has higher divergence factor has shown the better results.

In the work [13], the authors experimented with anomaly detection using two autoencoder models. The first encoder-decoder is trained on normal images, whereas the second network is on both healthy and unhealthy subjects. Later during the evaluation period, images are reconstructed using the former network, then the distance between the hidden vectors of the original and reconstructed is calculated with the help of a second VAE. This relies on the concept that a network trained on healthy images cannot reconstruct abnormal images, leading to a higher distance in encoding latent spaces. These models have used publicly available HCP and BraTS data, which have undergone preprocessing techniques later. The F1 score, accuracy, and AUC are compared with the basic VAE model.

In the [36] publication, Walter Pinaya et al. have proposed Anomaly detection with a combination of VQ-VAE and Transformers. The hidden space of VQ-VAE is combined with autoregressive transformers to achieve better performances. Discrete hidden space learned after the VQVAE training phase is reshaped to a sequence and can be used to

feed the transformer to calculate each latent vector's probabilities. The low likelihood of abnormal values is replaced with transformer samples, which are later used to reconstruct healed images. The authors once more take advantage of the probability densities of the spatial latent space vectors to avoid the issue of high-frequency areas mislabelling as a result of AEs' hazy reconstructions. An upsampled version of the mask, with possibly abnormal places to image resolution and the reconstruction, is multiplied together. They have conducted various experiments on artificial and real data. Dice scores, AUC, and AUPRC, are reported comparing models like AE, VAE, VQ-VAE, and VQ-VAE with transformers, maskings, etc. For the real datasets segmentation, low anomalous volumes of 15000 images of the UK Biobank dataset are used in VQ-VAE training, which is later evaluated on FLAIR images of BraTS, White Matter Hyperintensities Segmentation Challenge (WHM), and Multiple Sclerosis dataset (MSLUB). The proposed method showed remarkable results when combined with preprocessed methods of registering images to the MNI space mentioned in the paper. Instead of complete healthy images, VQVAE is trained on low infarct volume images for the avoidance of correction of the encoder, if performed, leads to the transformer's inability to find detections. All the experiments were performed on a few selected axial slices in the mid-range cropped to size 224×224.

Sergio Naval and Giacomo [37] have investigated the performance of VAE and VQVAE, including autoregressive models in the architecture. They considered restoration-based methods instead of reconstruction methods. The high loss latent vectors are restored with the prior given by autoregressive models, which allows models to identify abnormalities more accurately. The vector with restored samples is subsequently used to feed the decoder for image generation. They opted for the sum of cross-entropy of all thresholded samples from prior probability as an anomaly score for sample-wise evaluation. The weighted mean of absolute difference of several images generated and their original image from restored latent vectors are used for pixel-wise testing to reduce variance. Training and testing are conducted on Brain MRI and abdominal CT images resized to 160×160 from the dataset of the MOOD challenge. Similar to other works [9, 12], these are also normalized to zero mean and unit variance along with multiple data augmentation methods. In this work, VQVAE has embedding dimensions of 256 with 128 embedding vectors, and the prior model is similar to PixelSNAIL [38]. Area Under Receiver Operating Curve, Average Precision, and Dice score is recorded for analyzing models. For both pixel-wise and sample-wise, [37] has outperformed VAE models.

Changhee Han et al. have used GAN for image reconstructions by taking several simultaneous 2d slices of Brain MRI [39]. They exploited GAN for Alzheimer's and other neurological diseases in their work. They formulated the structure into two stages. The first stage is reconstruction, where they train the GAN model to reconstruct the three image slices from the previous three, leveraging Wasserstein loss and l1 loss, which is the absolute difference between healthy and reconstructed images. In the second stage, the testing - model is made to reconstruct the three images from the original images, and l2 loss is calculated between the reconstructed and original. Apart from l2 loss, they also determined AUC and ROC for evaluation. T1 and contrast-enhanced T1-weighted images resized to 176×256 are used to train and evaluate different diseases in different stages. As CE data is less, they have shown poor reconstructions. They have drawn comparisons of MADGAN [39] and several variants of it.

In the publication [40], authors Halima Hamid et al. have proposed a setup BrainGAN for generating and validating brain MRI images. This model used 400 actual scans with both tumor and healthy images. These real images are used to create 1400 synthetic data by leveraging generative models DCGAN and basic GAN, with 700 images of each class. Later these images are used to train three different models, general CNN, ResNet152V2, and MobileNetV2, and tested on real data. Several metrics like accuracy, precision, recall, loss, and AUC are employed as the performance measure. Dataset generated using DCGAN have outperformed other results when combined with ResNet152V2 architecture. This publication has paved the way for artificial data generation, which can be further utilized for model learnings.

In work [41], Chatterjee et al. have developed compact Context encoding VAE, an improvised version of original ceVAE [42]. All the MOOD challenge, T1w, and T2w IXI healthy images, and BRaTS test images have undergone several preprocessing methods. Input images are first segmented into four parts: grey matter, white matter, Cerebrospinal fluid, and background employing the FSL tool [43], which automatically does bias field correction. Later likewise [37, 9, 12], images are normalized to unit variance and zero mean, then split into 2d slices of size 256×256, fed to the network. This network uses fewer encoding and decoding layers than the original CEVAE, reducing overfitting for preprocessed data used in this model, unlike other unprocessed data. During training, data is augmented using multiple augmentation techniques like flipping, rotating, Gaussian noise, illumination artifacts, etc. The residual of reconstructed and original images is later subjected to replacing negative intensities to zero, which is later thresholded using the Otsu method. Synthetic data is also created for testing along with the above-mentioned test data. This work is later compared against three base models, CEVAE, GMVAE, and Skip AE using dice score as the metric.

4 Methods and Materials

Having explained start-of-art models and the current scenario, in this chapter, we shed light on the data sets and architectures employed in our work. We further explain pre-processing and post-processing steps performed during the training, along with several types of evaluation strategies in our work.

4.1 Datasets

This section describes the properties and characteristics of training and test data and shows exemplary images in figure 4.1 from each dataset.

IXI

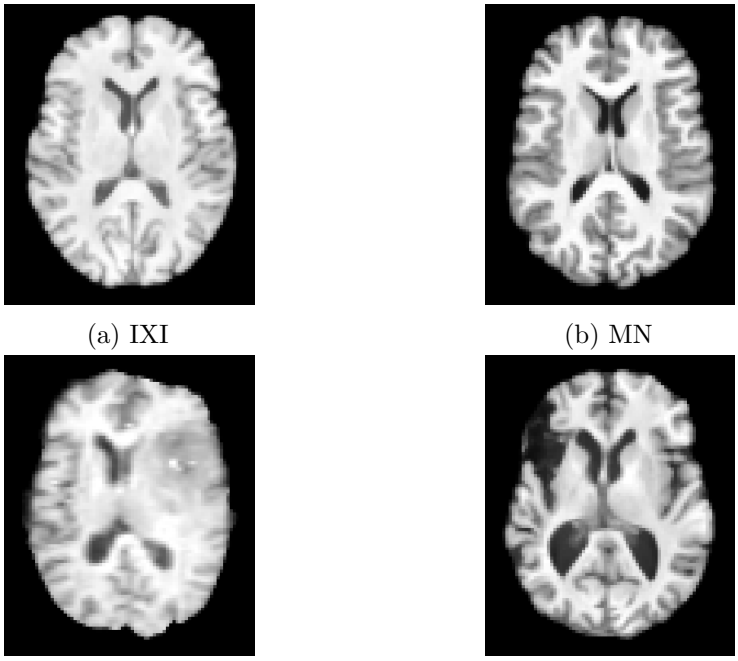
The data set Information eXtraction from Images (IXI) contains 577 healthy images[44]. These scans are an amalgamation of three different scanners, of which two are from Philips Medical Systems, namely, Gyroscan Intera 1.5T and Intera 3T. The other is GE 1.5T scanner. The scans are taken from the age group of 20 to 86 years old. In this work, we have utilized only T1- weighted images from the IXI dataset.

MixedNormals

Mixed Normals data is acquired from Jung diagnostics which has T1-weighted MRI scans from 1971 subjects. All of them have been examined as healthy images, which are used for training. The data is collected from 22 different devices with 68 varied configurations with parameters ranging from Repetition time, echo time, etc. In 1971 scans, 1.5 T field strength was used for most of the scans, whereas 3.0 T field strength was used in 435 images, and the remaining with a 1.0 T field strength. The slice thickness of the acquired images was between 0.90 to 2.40 mm. Images are spanned from age 6 to 90, with the majority around mid-to-late-forties.

Tab. 4.1: Overview of datasets with type of pathology and number of samples used for training, validation and testing.

Name of dataset	Samples	Type	Train	Val	Test
IXI	577	Normal	376	44	157
MN	1969	Normal	1483	166	320
BraTS	334	Tumor	0	111	223
ATLAS	304	Stroke	0	101	202



(a) IXI (b) MN

(c) BraTS (d) Stroke

Fig. 4.1: Images from Datasets.

BraTS

The Multimodal Brain Tumor Segmentation (BraTS) [45, 46] challenge dataset is publicly accessible for gliomas detection. The provided scans have an aggressive brain tumor, glioblastoma (HGG), and the slowing growing, lower-grade glioma (LGG), which frequently appear in younger patients. The dataset has 335 scans taken from different protocols and scanners manually annotated and approved by radiology experts. The equipped images are registered on similar anatomical structures and are skull stripped. The scans in the entire dataset are interpolated to $1 \times 1 \times 1$ mm resolution. The data contain T1, T2 weighted, T1-ce, T2- Fluid Attenuated Inversion Recovery (FLAIR), among which we utilized T1-weighted in our work.

ATLAS

Anatomical Tracings of Lesions After Stroke (ATLAS) is also publicly available data with 304 T1-weighted samples. More than half of scans have one lesion, rest with multiple lesions. Lesions are roughly distributed equally on the left and right hemispheres, with fewer than 8% on the cerebellum or brain stem. Seven of ten lesions are subcortical, and the rest are classified as cortical lesions, with a total of 512 lesions on all scans. Most images are isotropic resolution from a 3T Scanner, with the remaining $0.9 \times 0.9 \times 3$ mm from a 1.5T scanner [47].

4.1.1 Sampling

For the training process, we combine MixedNormals and IXI datasets for generalizing capability. Overall, of 2596 images, we divide data in the proportion of 7:1:2, for training,

validation and testing. Healthy test data is used for distinguishing healthy and abnormal slices. The test sets, BraTS and ATLAS, are divided into one-third and two-thirds for validation and testing. The composition of train, validation, and test datasets is spread out in the table 4.1.

4.1.2 Pre-Processing steps

Before supplying the samples to the neural networks, each scan went through a number of preprocessing procedures. To begin, the CurvatureFlow filter is used to smooth images while maintaining edges [12]. For uniform input, all the volumes are zero-padded to $140 \times 190 \times 158$ ($D \times H \times W$) size, for which later percentile clipping is applied. Intensity at the 1 % percentile is used to replace pixel values below that percentile, and intensity at the 99 % percentile is used to replace values above that percentile. Further, images are rescaled to $70 \times 95 \times 79$ with a rescaling factor of 0.5 and anti-aliasing effect. Afterward, the normalization of data is conducted.

4.2 Implementations

UAD has extensive development in contemporary times with the implementations of Autoencoders and its inheritors. As shown in the section 3.4, VAE has shown comparative results with GANs. Most works have investigated continuous embedding space with few discrete hidden space publications. We use vector quantization fused with VAE and further extend with GANs in our work apart from the explicit VAE model.

4.2.1 Architectures

This section explains the architectures VAE, VQVAE, and VQGAN we have employed in our current work, later continuing with describing hyperparameter tuning.

VAE

As mentioned in related work, VAE is one of the conventionally used architectures and is considered a benchmark for comparisons in most recent publications. Similar to all other approaches, we adopted VAE as the baseline model for our work. Less data requirements and fewer hyperparameters tuning make VAE a better choice as an exemplary model.

In this section, we demonstrate the implementation of VAE for 2d slices of MRI images inspired from [17] in 4.2. The resized image is provided to the encoder network, which has three convolutional layers with kernel size 5×5, stride 2, and padding 2. Each layer is then integrated with batch normalization and a Leaky Rectified Linear Unit (ReLU) as an activation function to form a single downsampling block. After these downsampling operations, the image dimensions are reduced from 95×79 to 12×10, with feature maps increasing from 1 to 128. Next is a convolutional layer with k = 1, s = 1, p= 1, reducing features from 128 to 16, maintaining spatial dimensions with batch normalization and Leaky ReLU. Later vector is flattened and is fed to a fully connected (FC) layer to obtain scalar outputs mean μ and variance σ of latent dimension 128. Then a sample is obtained using a reparameterized trick, which is supplied to the FC layer to get a dimension of 12×10. Now for reconstructing the images to their original

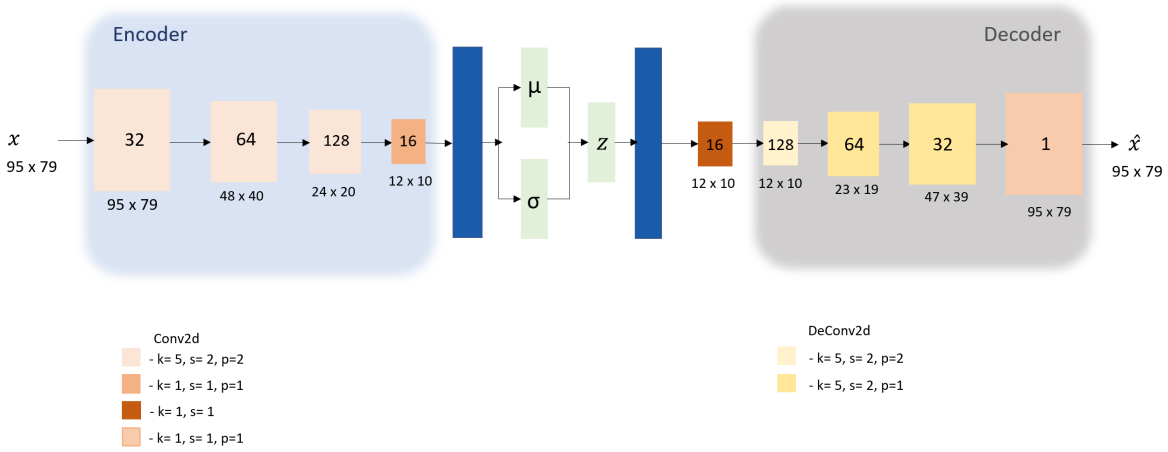


Fig. 4.2: Implementation of VAE Architecture.

size, a convolutional block with k = 1 and s = 1 is used. Next, an up-sampling block with kernel size 5, stride 2, and padding 2 continued with two up-sampling blocks with different padding p = 1. Other operations are similar to downsampling blocks. Finally, the reconstructed image is fed into the convolution block with value 1 for kernel, stride, and padding.

VQVAE

In VQVAE similar to [15], the encoder has three convolutional layers, each having a kernel size of 5, stride of 2, and padding of 1, with feature maps changing from 1 to 32, 64, and 128. Then the dimensionally reduced image is fed to another convolutional layer with k = 3×3, s= 1, and p = 1. Later a residual stack with residual blocks is employed. The residual block contains a ReLU activation function followed by a convolutional layer (k = 3×3, s= 1, p= 1), ReLU and again a conv2d layer as depicted in the picture. Then there is a pre-vq-convolutional layer, with filter size 1 and stride 1, for maintaining the dimensions, before inputting data to the codebook.

In the codebook, the vectors are reshaped to format batch, height, width, and channel, which are later flattened to obtain mapping with the closest codebook vector. The obtained codebook vector is then fed to a single strided post-vq-convolution layer with filter size 1×1 and padding 1 for reducing feature maps to 128. Then two residual blocks, further accompanied by three transposed convolution layers mirrored symmetrically to convolutions in encoder, are operated on vector, leading to original image sizes which is demonstarted in 4.3.

VQGAN

VQGAN architecture is chosen from the base paper [16] published by Esser et al. The network has many convolution layers, residual and non-local blocks, as illustrated in the figure 4.4. The first block has a single stride and padding convolution layer with window size 3×3, followed by two residual blocks. Then there are four down sampling blocks, each with a convolution layer of filter size 3×3 and stride 2 accompanied by two

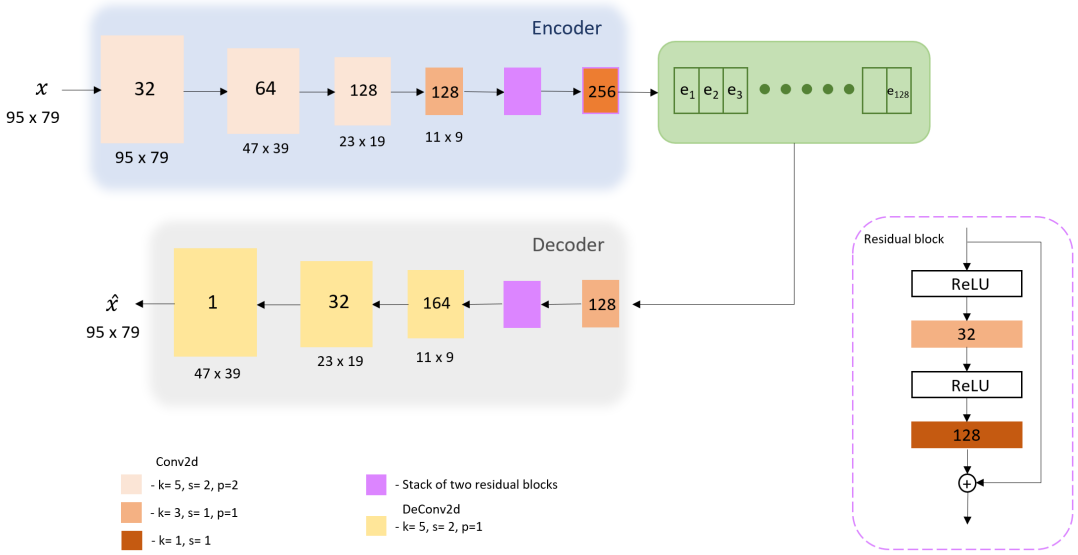


Fig. 4.3: Implementation of VQVAE Architecture.

residual blocks. Later, a combination of Non-local and residual blocks is repeated thrice before feeding the image to a pre-vq-convolution layer with k = 1×1 and s = 2. Then tensor is fed to the codebook, and the process is similar to VQVAE.

After quantization, there is another single stride post-quantization convolutional layer with window size 3×3 and padding 1, followed by a non-local block sandwiched between two residual blocks. The residual block consists of a Group Norm, a normalization layer where channels are divided into groups and features are normalized in each group. It is followed by a Swish activation layer $f(x) = x \cdot sigmoid(x)$ and a 2d convolutional layer with kernel size 3×3 and single padding and stride value. The above three layers are repeated in the same order, forming a complete residual block. A non-local block has a Group Norm layer followed by four convolutional layers with 1×1 kernel size and stride 1. Later, the combination of residual and non-local blocks is implemented six times. Further, there are four upsampling blocks, each having a conv2d layer (k = 3×3, s = 1, p = 1), and three residual blocks. After upsampling, images are interpolated from size 5×4 to 80×64. Further dimension increase is done with three transpose convolutional single stride layers with window size 5×5, followed by twice padded, single stride, 4×4 kernel. Later the reconstructed image is provided to the discriminator for further classification into real and fake images, with the mixture of convolution layers, activation functions, and batch normalizations as shown in the table 4.2.

4.2.2 Hyperparameter Tuning

Our work also delineates the effect of certain hyper parameters on the anomaly scores. Latent sizes affect the quality of reconstruction images, thus leading to better or worse lesion detections.

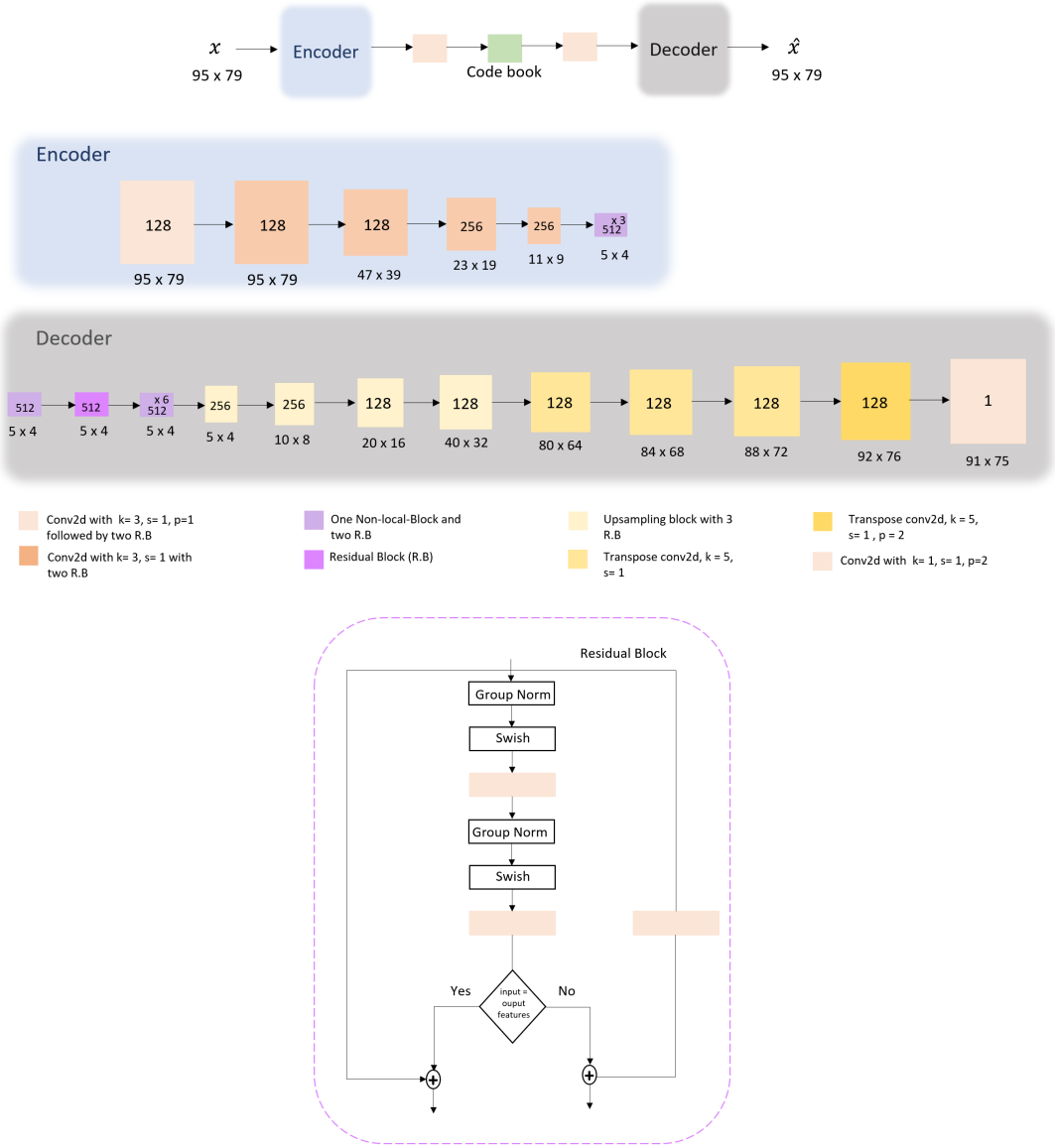


Fig. 4.4: Implementation of VQGAN Architecture.

Tab. 4.2: Layers of Discriminator having 2d Convolutional Layers, LeakyReLU activation function and Batch Normalization.

Type of Layer	Kernel	stride	padding
Conv2d	4×4	2	1
LeakyReLU	-	-	-
Conv2d	4×4	2	1
BatchNorm2d	-	-	-
LeakyReLU	-	-	-
Conv2d	4×4	2	1
BatchNorm2d	-	-	-
LeakyReLU	-	-	-
Conv2d	4×4	1	1
BatchNorm2d	-	-	-
LeakyReLU	-	-	-
Conv2d	4×4	1	1

VAE

In most publications, the dimensions of the latent manifold are 128. In our work, we vary latent sizes from 32 to 256 and observed the results later presented in the results section. The entire architecture is identical for all latent dimensions.

VQVAE

The reconstructions in VQVAE depend highly on the choice of codebook parameters, embedding dimensions, and vectors. We adapted the embedding vectors initially as per the publication [15] of van den Oord et al. Further, we varied embedding vectors and dimensions varying from 32 to 256 each.

VQGAN

Similar to the previous network, we have considered codebook vectors and dimensions. Due to computational constraints, parameters are not modified as extensively as VQVAE. We choose 64, 128, 256 vectors with dimensions 64, 128, 256.

4.2.3 Training

For training the networks, we combine IXI and MN healthy data leading to 1859 images. As mentioned in the section, we vary the parameters for further experiments. For VQVAE and VQGAN, commitment cost, the effect of encoder output, is set to 0.25. We trained all the networks with an ADAM optimizer and a learning rate of 0.0001 for 1200 epochs with batch size 16.

This section presents the number of trainable parameters of each architecture we implemented. In table 4.3, the number of trainable parameters of each architecture we implemented is depicted. The parameters are presented for a few variations of embedding vectors and dimensions. The rest of the experiments have similar parameters count for the specific architecture.

Tab. 4.3: Summary of trainable parameters of models.

Name of model	Latent size	Latent Vector	# Parameters
VAE	64	-	1 423 393
VAE	128	-	1 915 041
VQVAE	128	128	1 005 825
VQVAE	128	256	1 022 209
VQGAN	128	128	73 445 826
VQGAN	128	256	73 462 210

4.2.4 Evaluation

Having explained the training parameters, we elucidate post-processing steps and different strategies applied for evaluations based on slices, samples and pixels.

Post-processing steps

After training, the reconstructions are interpolated to dimensions of padding, which further underwent a few post-processing steps. We obtain the positive difference images from the original and reconstructions, which are later multiplied with an eroded brain mask to discard prominent edges at peripheries. Then a median filter of 2 dimensions with kernel size 9×9 is applied, which helps to smoothen edges and eliminate aberrations. As a final step, regions smaller than seven pixels are eliminated from the obtained image. Few experiments are conducted by changing median filter size to 5×5.

Metrics

Our work calculates performance metrics for evaluating the models from the residual images. We calculate reconstruction error, on which models are primarily dependent using absolute different (l1 loss) and mean squared error (l2 loss) terms for both anomalous and healthy regions. We use the given ground-truth segmentation and conduct a greedy search to identify the specific Operating Point on the Precision-Recall Curve that produces the highest dice score for each dataset. Along with Area under the precision-recall curve, AUPRC, we also calculate Area Under receiver operating characteristics curve, which is drawn against true positive and false positive rates, though it is not so prominent in imbalance datasets as it has weightage on samples with higher negative labels. Dice score, a well-chosen metric for medical image segmentation, gives more information on the overlap of detected and original segmentation is estimated. We later make use of the confusion matrix and calculate various metrics, as shown below.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Sensitivity = Recall = \frac{TP}{TP + FN} \tag{4.2}$$

$$Dicescore = \frac{2.Precision.Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{4.3}$$

Here TP is True Positive - anomalous samples classified as anomalous, FP is False Positive - healthy samples classified as abnormal, and FN is False Negative - anomalous samples classified as healthy. The precision determines model accuracy in calculating positive samples. Sensitivity or Recall gives how well unhealthy samples are classified.

Sample-wise evaluation

We try to identify subjects with lesions in the healthy and unhealthy data pool. As we employ the 2d models, reconstruction error for every slice in the image is calculated and averaged to get an anomaly score for a sample. Along with evaluating unhealthy samples, we also calculate for the hold-out healthy dataset. These are all stacked together to differentiate abnormal subjects from regular scans. We chose AUPRC and AUC as metrics for assessing the model performance.

Slice-wise evaluation

In slice-wise evaluation, we identify anomalous scans in each brain scan. Every MRI scan has healthy and unhealthy slices labeled as 0 and 1 based on ground truth segmentations and reconstructions separately. Our goal is to discriminate between these slices. We present AUC and AUROC for metrics for assessing the model performance.

Pixel-wise evaluation

In the above methods, we discriminated between healthy and unhealthy slices and samples. In pixel-wise evaluation, anomaly lesions are more exactly localized by further conveying the shape of lesions. For this strategy, residual volume is to be binarized. We employed a greedy search algorithm similar to [30], depending on the Dice score of the validation set samples for each type of dataset. In this case, the dice score is calculated for thresholds at the top and bottom quartiles on intensity range on the validation set, which is later used to cut the interval range to either top or bottom half, and the search is continued. This is repeated 10 times to obtain the best Dice score. The threshold related to the best dice score is stored for the calculation of Dice scores on the test data set. The dice score is calculated for every 2d image and further piled together to obtain the total dice score of the model. We also mention subject-wise dice score, which is more relevant in this setting. Along with Sørensen–Dice coefficient, AUC and AUPRC are calculated for entire images and also patient-wise.

5 Results

In this section, we present the results of our study, starting with a sample-wise evaluation and later slice-wise and pixel-wise evaluations for VAE, VQVAE, and VQGAN models. We show the effect of various latent size dimensions employed in each evaluation strategy for all the models. AUROC and AUPRC metrics are chosen for sample-wise and slice-wise evaluation. For pixel-wise, we indicate the Total Dice score DICE_T of all samples and the DICE_S, average of patient-wise samples. Later, a small experiment is conducted on pre-trained models for image reconstructions of non-brain images to challenge and verify the core assumptions behind UAD methods.

5.1 Sample-wise evaluation

Sample-wise evaluation detects the anomalous sample from the whole dataset combined with healthy and unhealthy subjects. The ratio of healthy samples to unhealthy samples (BraTS and ATLAS) is approximately in the ratio 7:3. We initially present the comparison of the three models and later demonstrate the embedding space effect on each model. To compare the models, we consider the total dice score for various dimensions and present the top two versions of it in the table 5.1. Overall, AUROC, AUPRC values for BraTS19 is less compared to ATLAS. VQGAN model has outperfomed VAE and VQVAE, with VQVAE being least performing model. Sample-wise evaluation graphs for all the models is detailedly plotted in 7.

VAE

In this section, the model is evaluated sample-wise on various latent dimensions starting from 32 to 256 increasing powers of 2. The AUROC and AUPRC information for different latent space for two datasets utilized in our study. Model performance on BraTS19

Tab. 5.1: Comparison of models on sample-wise anomaly detection based on Reconstruction errors. The AUROC and AUPRC values are present for best two versions of each model on test data. Here ED is Embedding Dimensions and EV is Embedding vectors. All metrics are represented in percentage.

			BraTS19		ATLAS(Stroke)	
Model	ED	EV	AUROC	AUPRC	AUROC	AUPRC
VAE	128	-	45.43	30.36	89.67	78.35
VAE	256	-	37.09	26.67	88.82	77.84
VQVAE	256	128	19.09	20.11	84.54	66.53
VQVAE	256	64	21.74	20.82	84.49	66.93
VQGAN	64	128	75.64	59.84	89.69	77.41
VQGAN	64	64	**81.42**	**68.88**	**90.80**	**79.94**

Tab. 5.2: Slice-wise Comparison of models based on Reconstruction errors. The AUROC and AUPRC values are present for best two versions of each model on test data. Here ED is Embedding Dimensions and EV is Embedding vectors. All metrics are represented in percentage.

			BraTS19		ATLAS(Stroke)	
Model	ED	EV	AUROC	AUPRC	AUROC	AUPRC
VAE	128	-	84.08	82.77	86.27	72.89
VAE	256	-	**84.32**	**83.16**	**86.42**	**73.74**
VQVAE	256	128	82.78	79.55	84.62	66.41
VQVAE	256	64	82.55	79.36	85.02	68.31
VQGAN	64	128	83.54	81.77	85.10	68.14
VQGAN	64	64	83.66	82.42	86.01	70.61

dataset based on reconstruction loss is much less than stroke data. The values for Stroke almost remained similar on all sizes of latent manifold, whereas there is a nominal jump of almost 7% in AUROC and AUPRC for latent size 64 from 32.

VQVAE

As VQVAE has discrete latent space expressed in terms of embedding vectors with dimensions, we have two parameters to be tuned. For our study, we keep the latent vector constant and increase the embedding sizes. Late, we change the latent vectors and repeat the process.

The stroke dataset is better separated from healthy datasets in comparison to BraTS. The best achieved AUROC and AUPRC values for both datasets are at latent dimension 256 with latent vectors 64 except for Stroke's AUROC at dimension 64 with 32 codebook vectors with 0.4 difference to the above values. Overall there are minor variations in all the varieties. For sample-wise detection, a combination of 64 vectors with manifold size 256 can be used.

VQGAN

Similar to VQVAE, we have two parameters that has to be varied. Limited experiments on latent sizes are conducted due to computational power and longer training periods. In the experiments conducted, ROC curve and AUPRC is better for dimension 64 and codebook vectors 64 for BraTS19 and stroke. For a particular combination of dimenions and vectors, the high performance is achieved, while for all others, there is decline in performance.

5.2 Slice-wise evaluation

We report slice-wise evaluation metrics for Tumor and Stroke datasets utilized for all three models with varied parameters. We further elaborate influence of bottleneck dimensions of each model on each dataset. All the values are considered based on reconstruction errors. In slice-wise, we attempt to discriminate anomalous slices from healthy counterparts in an image. The best performance is achieved on selected metrics

by VAE in comparison to the other two. Overall, all the values are approximately similar to each other, with very minute percentage changes. Good AUROC and AUPRC for BraTS and Stroke are obtained with VAE-256. Stroke has better AUROC values, whereas BraTS has better precision-recall curves for all models.

VAE

All the experiments conducted are in the similar fashion like sample-wise anomaly detection. There is almost similar performance on all the latent dimensions indicating no influence. An insignificant difference of less than 1 % is observed, which is negligible.

VQVAE

AUPRC values are higher in BraTS19 and vice-versa for AUROC values. Performance metrics for Stroke are better at codebook dimension 64 at 32 vectors. For BraTS, the model performed well at 32 embedding vectors with dimensions 256 for AUROC and 128 for AUPRC. Overall, we observed good metrics at lower embedding vector 32. There is a difference of about 1-3 % between all the combinations.

VQGAN

For Brats, model performed best on 128 embedding dimensions with 64 vectors, while for stroke it is best when embedding vectors and dimensions are increased to 128 and 256 respectively. Similar to all others models, there are only slight variations in performance metrics. Slice-wise detection performance metrics for both the datasets for all three models is further illustrated with graphs in 7.

5.3 Pixel-wise evaluation

Pixel-wise evaluation is considered to be widely used strategy and it presents the dice-scores which localizes and segments the lesions. We evaluate pixel-wise metrics for all slices in whole data and also calculate average on each sample. We also report AUPRC values for all the models. Like other evaluation strategies, we do not represent AUROC as it has huge effect on imbalance datasets.

For BraTS, VQGAN model outperformed other models, while VAE performed well for Stroke. In VQGAN, for the latent dimension 256, more the code book vectors, better the performance. Overall, segmentation is better on Stroke data rather than BraTS. VQVAE is far behind both the models with almost less than 12% for Stroke as shown in table 5.3. The figure 5.1 shows reconstructions on training data. VQVAE gives best reconstructions maintaining all the curves of brain, while VAE has blurry reconstructions. VQGAN is missing out the texture of brain. As illustrated in figure 5.2, VAE has shown blurry reconstruction but better segmentation. VQVAE reconstruction is almost similar to the original data, thus leading to segmentation of normal pixels also as anomalous regions, whereas for VQGAN reconstructions are little crisp but not as good as VQVAE.

We further compared DICE_T and DICE_S for best performing model in our experiments by changing the kernel size in the median filter from 9×9 to 5×5, which reduced the

Tab. 5.3: Comparison of models on pixel-wise detection using dice scores and AUPRC values. $DICE_T$ represents total dice score of all slices of all samples and $DICE_S$ represents average of samples in a specific dataset. All values are reported in percentages.

			BraTS19		
Model	ED	EV	$DICE_T$	$DICE_S$	AUPRC
VAE	128	-	24.94	21.72±12.58	17.54
VAE	256	-	24.34	20.88±12.97	17.34
VQVAE	256	128	18.24	17.71±9.48	10.38
VQVAE	256	64	17.92	17.43±9.35	11.65
VQGAN	64	128	**27.30**	**23.76±13.59**	**20.52**
VQGAN	64	64	25.87	22.36±13.19	18.61

			ATLAS(Stroke)		
Model	ED	EV	$DICE_T$	$DICE_S$	AUPRC
VAE	128	-	31.00	**13.68±18.41**	21.32
VAE	256	-	31.43	13.08±18.29	21.70
VQVAE	256	128	7.45	4.70±6.72	3.85
VQVAE	256	64	9.88	5.45±8.43	5.25
VQGAN	64	128	**31.80**	12.05±18.22	**22.34**
VQGAN	64	64	30.98	11.54±17.89	21.06

performance. There is a drastic difference for the Stroke dataset when a lower kernel size is used in VAE and VQGAN model, which is noticed in the table 5.4.

VAE

Both Dice scores and precision-recall curves has increased with the increase in latent dimensions for both datasets. So, for pixel-wise higher the latent manifold, better the segmentations. Graphs analyzing the performance of models are indicated in 7.

VQVAE

VQVAE has the lowest performance in all the models with the highest dice score being 18.24% and 9.88% for BraTS and Stroke respectively. Similar style is resembled on AUPRC values also. For stroke, latent vector 256 with size 128 has shown significant performance in both the performance metrics. Overall, performances are better at 128 code book vectors. Dice scores are better at higher dimensions.

VQGAN

VQGAN has achieved the better performances for both the datasets in every aspect of pixel-wise detection except for Subject wise Stroke Dice score. The best achieved performances are achieved when codebook vectors are 128. Moving in either direction of vector values led to decrease in performance for the models in any of latent dimensions.

Tab. 5.4: Comparison of models on pixel-wise detection using dice scores. 9×9 and 5×5 are kernel sizes for median filter. Here DICE_T represents total dice score of all slices of all samples and DICE_S represents average of samples in a specific dataset. All values are reported in percentages.

			BraTS19			
			9×9		5×5	
Model	ED	EV	DICE_T	DICE_S	DICE_T	DICE_S
VAE	256	-	24.83	20.88±12.97	21.27	18.65±10.89
VQVAE	256	64	17.91	17.43±9.34	17.53	17.08±9.15
VQGAN	64	128	27.29	23.76±13.59	23.19	20.74±10.87

			ATLAS(Stroke)			
			9×9		5×5	
Model	ED	EV	DICE_T	DICE_S	DICE_T	DICE_S
VAE	256	-	31.42	13.08±18.29	22.09	10.32±14.59
VQVAE	256	64	9.88	5.44±8.42	7.72	4.88±6.44
VQGAN	64	128	31.79	12.05±18.22	21.19	9.09±14.04

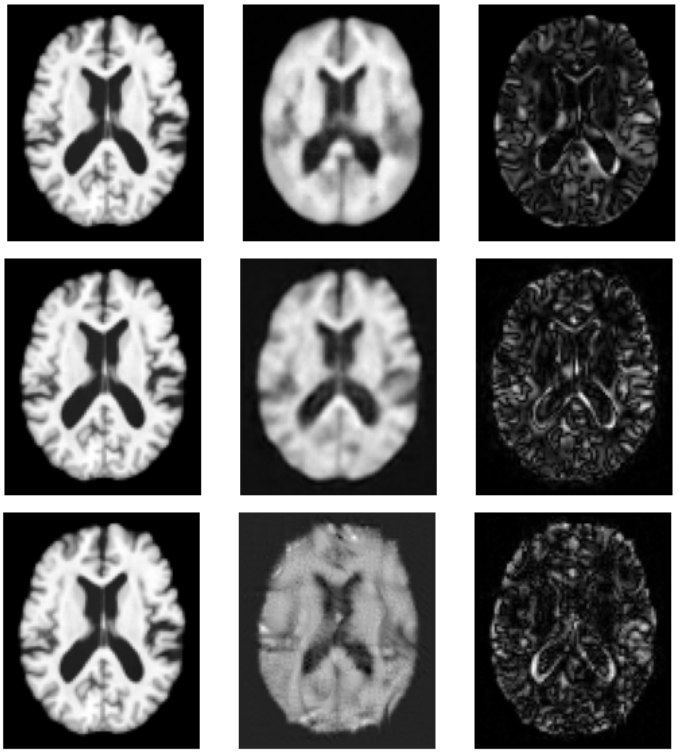


Fig. 5.1: Reconstructions of healthy data on models. From left to right: Original Image, Reconstruction, difference volume. All images presented here are down scaled images. From top to bottom : Reconstructions based on VAE, VQVAE, VQGAN.

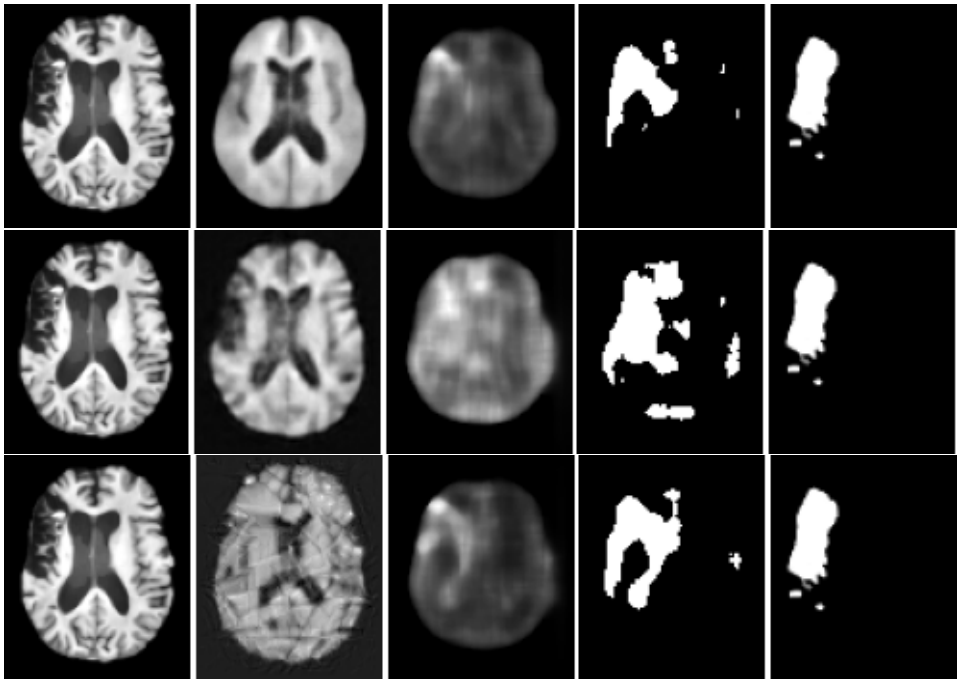
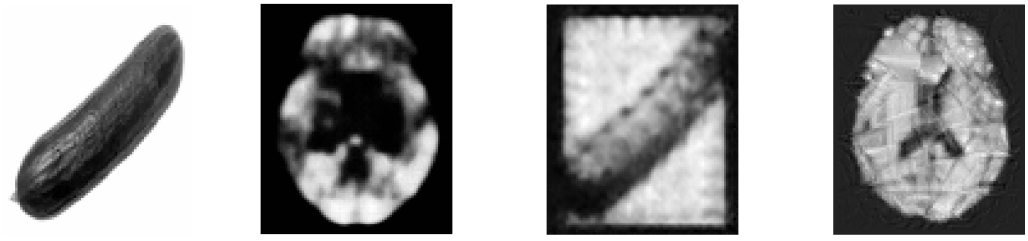


Fig. 5.2: Reconstructions of Stroke on models. From left to right: Original Image, Reconstruction, difference volume after eroding brain mask and application of median filtering, thresholded image used for metrics calculation, Ground-truth. All images presented here are down scaled images. From top to bottom : Reconstructions based on VAE, VQVAE, VQGAN.

5.4 Assessment of model performance on non-brain images

Reconstruction experiments are conducted on the non-brain image for the best performance model on each variant. Here the best performance is considered to be the model that has the highest total dice score.For VAE, the model with latent size 128 is adapted. Similarly, for VQVAE and VQGAN, models with embedding size 256 and vectors 64,128, respectively, are utilized. The figure 5.3 shows the reconstructions obtained.



(a) Input (b) VAE (c) VQVAE (d) VQGAN

Fig. 5.3: Model performance on Non brain images from left to right original Image, reconstructions obtained from VAE, VQVAE, VQGAN.

Here input is given in the size 95×79 without any pre-processing methods. The image

chosen is a picture of cucumber which is entirely different both in shape and texture from the trained brain images. The results of reconstructions are further discussed in the next section briefly.

6 Discussion

In this section, we analyze results obtained from the various models and their variants with changes in the input parameter. We mainly discuss the performances of models later, moving to a concise argument on latent sizes and concluding the section concerning datasets used.

6.1 Comparison of model performances

We compare the performance of three models, VAE, VQVAE, and VQGAN, concerning their abilities to differentiate anomalous samples from a set of both healthy and unhealthy data, discriminate unhealthy slices within a subject, and finally, the model's ability to segment the lesions in the slices respectively. The section 5.4 results show models' reconstruction capability on the different types of data other than trained sets. VQGAN tried to capture brain textures and reconstructed images similar to brain. It has learned global features well compared to other models because of perceptual loss, in which, instead of finding loss between pixels, it calculates the high-level differences like context and style between the images. And discriminator component further helps to train the codebook vectors more precisely on a particular training dataset by generating labels for fake and real data. Next, the VAE reconstructions slightly look like brain data. In contrast to other models, VQVAE has shown worse performance in latent dimension learning, which makes it obvious not to consider VQVAE in provided setting. VQVAE tends to reconstruct the image similar to input data irrespective of trained latent vectors. Contrary to the proposed model [15], the discrete vectors does not help in capturing the features. Overall reconstruction errors are very low for VQVAE model as opposed to the assumptions, that models trained on healthy data will poorly reconstruct unhealthy parts. The copying of the input image in reconstructions of VQVAE models, indicates the latent vectors are not trained well to hold the characteristics of images they are trained. Further, adding a discriminator model to VQVAE showed a significant performance jump, indicating that discriminator plays a major role in codebook learning.

Concerning the tables 5.1 and 5.2, VQGAN has better performance in sample and VAE in slice-wise. We noticed the difference in performances, making it difficult to propose one model for all the types of testing strategies and datasets. As there is a high data imbalance in the sample wise, with 70 percent of images being healthy, it is challenging to draw conclusions. For BraTS, AUPRC is higher than Stroke, indicating the effect of lesion sizes. Larger anomaly sizes make the model predict slices in each sample correctly as anomalous. As seen in the experiment 5.3, it is evident that VQVAE tends to reconstruct unhealthy images as they lead to lesser reconstruction errors, which is justified by the values we obtained in our experiments.

Contemplating pixel-wise evaluation, there is a substantial drop in AUPRC values, indicating the challenges in segmentation tasks compared to classification tasks. VQGAN

and VAE have shown similar performances in pixel-wise detection. Various factors like different number of lesions and their sizes in each sample and the data obtained through multiple machines pose difficulty for detection. Thresholding used to segment is data specific, which is one of the tasks that have to be further simplified in a way to consider the same threshold for all the types of classes, making models truly unsupervised.

The training time of VQGAN is very large in compared to VAE which is not proportionate to the performance. It is evident from the table 4.3, where VQGAN has large number of trainable parameters taking significant training time. So, VAE can be used for anomaly detection with limited computational power.

6.2 Effect of latent sizes

Having discussed the model performances, we briefly elucidate the influence of latent sizes. For the VAE model, an increase in latent size increased the performance of the model. The overall l2 reconstruction errors are lower for lower dimensions. Our assumption more the reconstruction error for unhealthy scans should lead to better performances is satisfied. In VQVAE, an increase in vectors has increased the overall reconstruction errors, thus indicating higher codebook vectors are better for VQVAE. The reconstruction errors for the VQVAE model are very low compared to other models, thus conveying model is able to reconstruct unhealthy parts along with healthy parts. The reconstruction errors for the VQGAN model are higher than the other two, implying better anomalous areas detecting models on unhealthy scans which are trained on healthy scans. The overall mean reconstruction error is lower at codebook vectors 128. On the whole, we extrapolate that the effect of latent dimensions is better observed at VAE than in others. For the remaining models, the change of sizes and vectors does not have a considerable impact, thus indicating the random choice of latent space.

6.3 Performance on Datasets

In this section, results for diverse pathologies utilized in this study are analyzed. In sample-wise, ATLAS has better performance than BraTS, whereas BraTS data has better performance in slice-wise analysis. Better discrimination between the slices in BraTS is might be because of large sizes of tumor, while there are many smaller lesions in Stroke which might have been ignored. ATLAS has many smaller lesions including few larger lesions. More than 40% of scans have multiple lesions making it easier in discrimination of anomalous samples from healthy in sample-wise detection strategy. The size of lesions is assumed to be influence in lower AUPRC values for stroke data in Pixel-wise than other strategies.

In pixel-wise total dice score, which is the average of all the slices of all samples, ATLAS has better performance, whereas, in subject-wise dice score, BraTS has outperformed the former dataset. This performance drop in ATLAS for $DICE_S$ can be due to different contrast levels in the tumor region and surrounding tissues. Difference images with brighter areas are computed, leading to higher dice scores, and the size of lesions may also affect the scores. The size of tumors in BraTS is more extensive compared to stroke samples. In pixel-wise evaluation, $DICE_S$ is more relevant in the clinical setting than the

overall dice score across all samples. This significant difference in model performance on datasets further put forth a complex task of deciding various pre-processing and post-processing for specific data.

7 Conclusion

Unsupervised Anomaly Detection in the medical sector is significantly increasing day by day. The ratio of patients to radiologists is lowered, suggesting the benefits of supportive tools based on Artificial Intelligence. Though supervised learning is superior in segmenting anomalous regions, it always comes with two major problems: annotating a large number of samples by expert domain and it is pathology specific and cannot be generalized. We research UAD in order to identify the existence and position of abnormalities, which fixes the issues with supervised learning.

Several architectures starting from VAE to Generative Adversarial Networks and their variants have been implemented for detecting various types of pathologies in Brain MRI images. There as significant works on 2D VAE but very few on recently developed VQVAE and VQGAN models.

A combination of two healthy datasets is used for training and evaluated on BraTS and ATLAS datasets. Firstly, a baseline VAE model is implemented using 2D brain slices from 3D scans which use continuous spaces. We further exploited discrete spaces by implementing VQVAE, which uses discrete latent spaces in the form of codebook vectors with specific dimensions. Finally, we extend VQVAE to VQGAN by adding a discriminator component and changing the loss component. By using these architectures and changing the embedding space parameters, we analyzed the influence of latent dimensions. Further, an experiment is conducted for non-brain images using the models trained on MRI data to assess the capability of the models.

The model performances are investigated based on sample-wise, slice-wise, and pixel-wise anomaly detections. Pixel-wise is the hardest among strategies, as it locates the position of anomaly, unlike the discrimination task done by the other two. Pixel-wise detection, is more meaningful for hospital settings, where anomaly regions can be predicted. In this context, VQGAN outperformed other models. In sample and slice-wise detection techniques, VAE has achieved good performance. A notable effect on latent sizes is observed on VAE and VQGAN. The experiment on assessing models clearly shown that VQGAN learnings are better than the two. VQVAE is seen to not have learned anything, indicating it is a very bad choice for anomaly detection.

In conclusion, we can put forth that VQGAN is better for segmenting various lesions, while VAE is better for discriminating between unhealthy and healthy segments. The extension of VQGAN with the transformer that might potentially increase the detections could be one of the future works. Furthermore, the choice of T2-weighted MRI scans can be investigated as lesions are higher contrast than T1-weighted. The study on restoration-based methods can also be included in the future.

Bibliography

[1] Katie L McMahon, Gary Cowin, and Graham Galloway. Magnetic resonance imaging: the underlying principles. *journal of orthopaedic & sports physical therapy*, 41(11):806–819, 2011.

[2] Christoph Baur. *Anomaly Detection in Brain MRI: From Supervised to Unsupervised Deep Learning*. PhD thesis, Technische Universität München, 2021.

[3] Michael A Bruno, Eric A Walker, and Hani H Abujudeh. Understanding and confronting our mistakes: the epidemiology of error in radiology and strategies for error reduction. *Radiographics*, 35(6):1668–1676, 2015.

[4] Rémi Domingues, Maurizio Filippone, Pietro Michiardi, and Jihane Zouaoui. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern recognition*, 74:406–421, 2018.

[5] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[6] Karthik Seetharam, Nobuyuki Kagiyama, and Partho P Sengupta. Application of mobile health, telemedicine and artificial intelligence to echocardiography. *Echo Research and Practice*, 6(2):R41–R52, 2019.

[7] Byungjai Kim, Kinam Kwon, Changheun Oh, and Hyunwook Park. Unsupervised anomaly detection in mr images using multicontrast information. *Medical Physics*, 48(11):7346–7359, 2021.

[8] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI brainlesion workshop*, pages 161–169. Springer, 2018.

[9] Xiaoran Chen and Ender Konukoglu. Unsupervised detection of lesions in brain mri using constrained adversarial auto-encoders. *arXiv preprint arXiv:1806.04972*, 2018.

[10] Philipp Seeböck, José Ignacio Orlando, Thomas Schlegl, Sebastian M Waldstein, Hrvoje Bogunović, Sophie Klimscha, Georg Langs, and Ursula Schmidt-Erfurth. Exploiting epistemic uncertainty of anatomy segmentation for anomaly detection in retinal oct. *IEEE transactions on medical imaging*, 39(1):87–98, 2019.

[11] Philipp Seeböck, Sebastian M Waldstein, Sophie Klimscha, Hrvoje Bogunovic, Thomas Schlegl, Bianca S Gerendas, René Donner, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised identification of disease marker candidates in retinal oct imaging data. *IEEE transactions on medical imaging*, 38(4):1037–1047, 2018.

[12] Christoph Baur, Stefan Denner, Benedikt Wiestler, Nassir Navab, and Shadi Albarqouni. Autoencoders for unsupervised anomaly segmentation in brain mr images: a comparative study. *Medical Image Analysis*, 69:101952, 2021.

[13] Alexandra Albu, Alina Enescu, and Luigi Malagò. Tumor detection in brain mris by computing dissimilarities in the latent space of a variational autoencoder. In *Proceedings of the Northern Lights Deep Learning Workshop*, volume 1, pages 6–6, 2020.

[14] Rohan Kapre, Mentored By, Jiahong Ouyang, and Qingyu Zhao. Using discrete vaes on t1-weighted mri data to embed local brain regions.

[15] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[16] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[17] Christoph Baur, Robert Graf, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Steganomaly: Inhibiting cyclegan steganography for unsupervised anomaly detection in brain mri. In *International conference on medical image computing and computer-assisted intervention*, pages 718–727. Springer, 2020.

[18] David Zimmerer, Fabian Isensee, Jens Petersen, Simon Kohl, and Klaus Maier-Hein. Unsupervised anomaly localization using variational auto-encoders. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 289–297. Springer, 2019.

[19] Judy Illes, Patricia Lau, and JT Giacino. Brain imaging: Understanding the basics. *The University of British Columbia. URL: http://www. acrm. org/pdf/BrainImagingFAQ. pdf [Accessed: 11-February-2013]*, 2008.

[20] Ray Hashman Hashemi, William G Bradley, and Christopher J Lisanti. *MRI: the basics: The Basics*. Lippincott Williams & Wilkins, 2012.

[21] Wu-Chung Shen. Basics of interpretation of brain ct and mri. In *Diagnostic Neuroradiology*, pages 19–54. Springer, 2021.

[22] Daphne Caligari Conti. Magnetic resonance imaging. 03 2016.

[23] Stuart Currie, Nigel Hoggard, Ian J Craven, Marios Hadjivassiliou, and Iain D Wilkinson. Understanding mri: basic mr physics for physicians. *Poraduate medical journal*, 89(1050):209–223, 2013.

[24] Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, 44:206–227, 1959.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[26] Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[27] Howard B Demuth, Mark H Beale, Orlando De Jess, and Martin T Hagan. *Neural network design*. Martin Hagan, 2014.

[28] 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.

[29] Takio Kurita. Principal component analysis (pca). *Computer Vision: A Reference Guide*, pages 1–4, 2019.

[30] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *arXiv preprint arXiv:2003.05991*, 2020.

[31] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[33] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

[34] Jason R Taylor, Nitin Williams, Rhodri Cusack, Tibor Auer, Meredith A Shafto, Marie Dixon, Lorraine K Tyler, Richard N Henson, et al. The cambridge centre for ageing and neuroscience (cam-can) data repository: Structural and functional mri, meg, and cognitive data from a cross-sectional adult lifespan sample. *neuroimage*, 144:262–269, 2017.

[35] David Zimmerer, Jens Petersen, and Klaus Maier-Hein. High-and low-level image component decomposition using vaes for improved reconstruction and anomaly detection. *arXiv preprint arXiv:1911.12161*, 2019.

[36] Walter Hugo Lopez Pinaya, Petru-Daniel Tudosiu, Robert Gray, Geraint Rees, Parashkev Nachev, Sébastien Ourselin, and M Jorge Cardoso. Unsupervised brain anomaly detection and segmentation with transformers. *arXiv preprint arXiv:2102.11650*, 2021.

[37] Sergio Naval Marimont and Giacomo Tarroni. Anomaly detection through latent space restoration using vector quantized variational autoencoders. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1764–1767. IEEE, 2021.

[38] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model. In *International Conference on Machine Learning*, pages 864–872. PMLR, 2018.

[39] Changhee Han, Leonardo Rundo, Kohei Murao, Tomoyuki Noguchi, Yuki Shimahara, Zoltán Ádám Milacski, Saori Koshino, Evis Sala, Hideki Nakayama, and Shin'ichi Satoh. Madgan: Unsupervised medical anomaly detection gan using multiple adjacent brain mri slice reconstruction. *BMC bioinformatics*, 22(2):1–20, 2021.

[40] Halima Hamid N Alrashedy, Atheer Fahad Almansour, Dina M Ibrahim, and Mohammad Ali A Hammoudeh. Braingan: Brain mri image generation and classification framework using gan architectures and cnn models. *Sensors*, 22(11):4297, 2022.

[41] Soumick Chatterjee, Alessandro Sciarra, Max Dünnwald, Pavan Tummala, Shubham Kumar Agrawal, Aishwarya Jauhari, Aman Kalra, Steffen Oeltze-Jafra, Oliver Speck, and Andreas Nürnberger. Strega: Unsupervised anomaly detection in brain mris using a compact context-encoding variational autoencoder. *arXiv preprint arXiv:2201.13271*, 2022.

[42] David Zimmerer, Simon AA Kohl, Jens Petersen, Fabian Isensee, and Klaus H Maier-Hein. Context-encoding variational autoencoder for unsupervised anomaly detection. *arXiv preprint arXiv:1812.05941*, 2018.

[43] Mark Jenkinson, Christian F Beckmann, Timothy EJ Behrens, Mark W Woolrich, and Stephen M Smith. Fsl. *Neuroimage*, 62(2):782–790, 2012.

[44] Ixi – information extraction from images. `https://brain-development.org,`. accessed: 2021-06-01.

[45] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[46] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4(1):1–13, 2017.

[47] Sook-Lei Liew, Julia M Anglin, Nick W Banks, Matt Sondag, Kaori L Ito, Hosung Kim, Jennifer Chan, Joyce Ito, Connie Jung, Nima Khoshab, et al. A large, open source dataset of stroke anatomical brain images and manual lesion segmentations. *Scientific data*, 5(1):1–11, 2018.

A Appendix

Sample-wise Evaluation Graphs

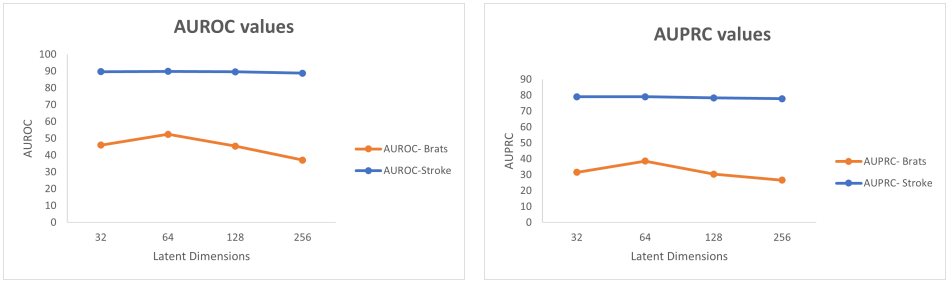


Fig. A.1: Comparison of sample-wise evaluation of VAE on various latent dimensions for AUROC and AUPRC on BraTS19 and Stroke data.

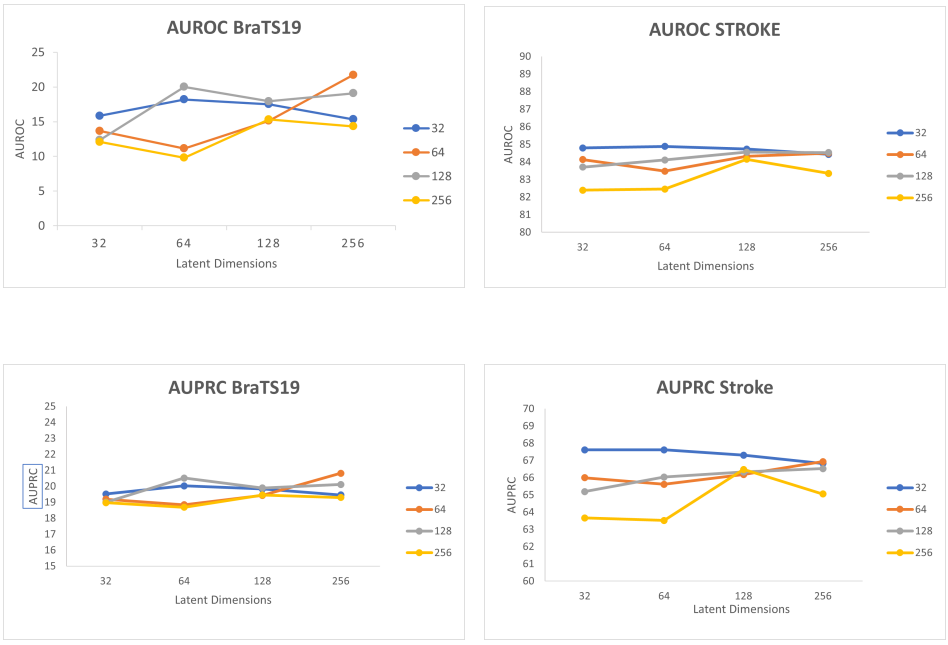


Fig. A.2: Comparison of sample-wise evaluation of VQVAE model with varying Embedding vectors and embedding dimensions.

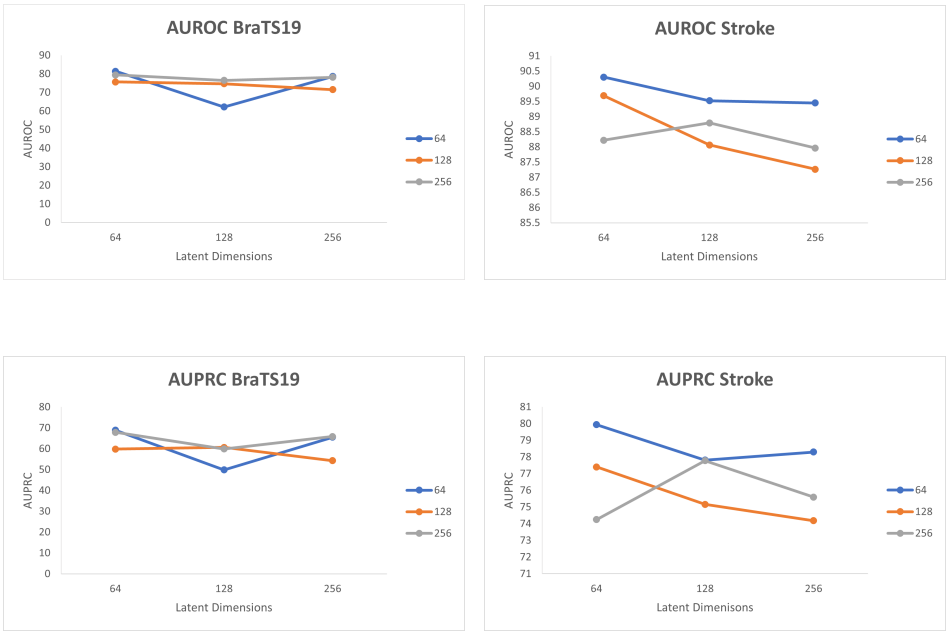


Fig. A.3: Comparison of sample-wise evaluation of VQGAN model with varying Embedding vectors and embedding dimensions.

B Appendix

Slice-wise Evaluation Graphs

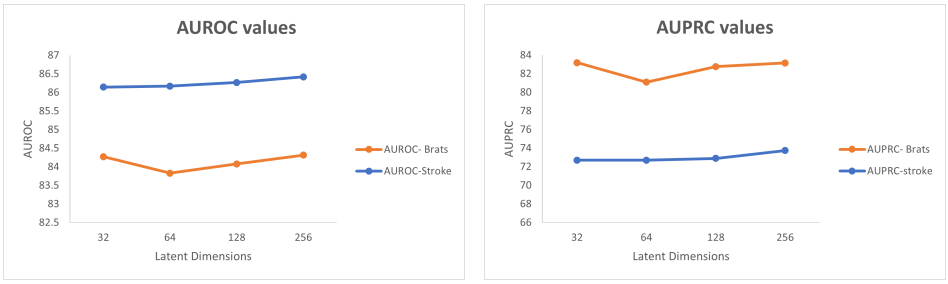


Fig. B.1: Comparison of slice-wise anomaly detection on VAE with various latent manifold dimensions.

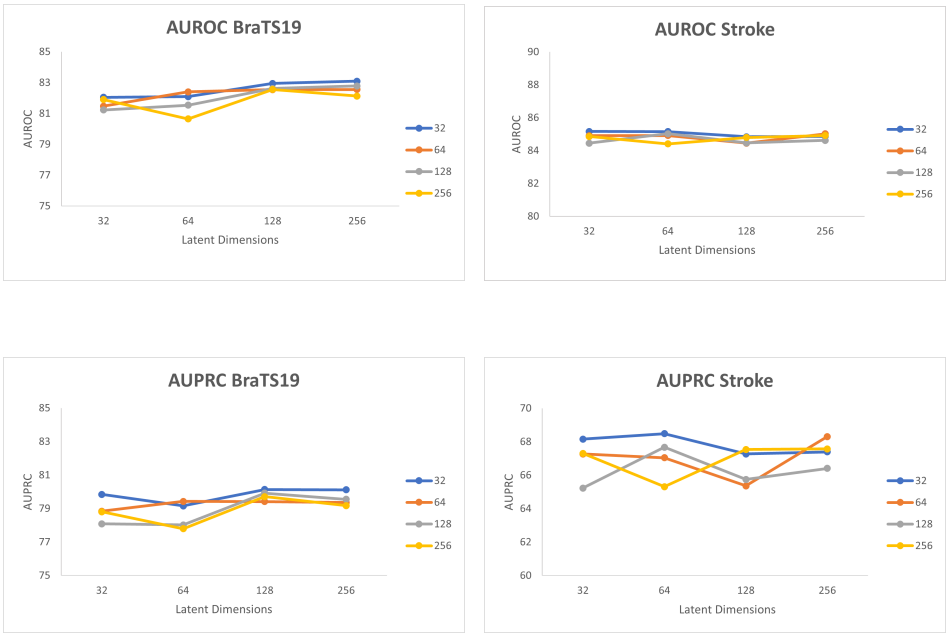


Fig. B.2: Comparison of slice-wise detection of VQVAE with varied parameters codebook vectors and codebook dimensions.

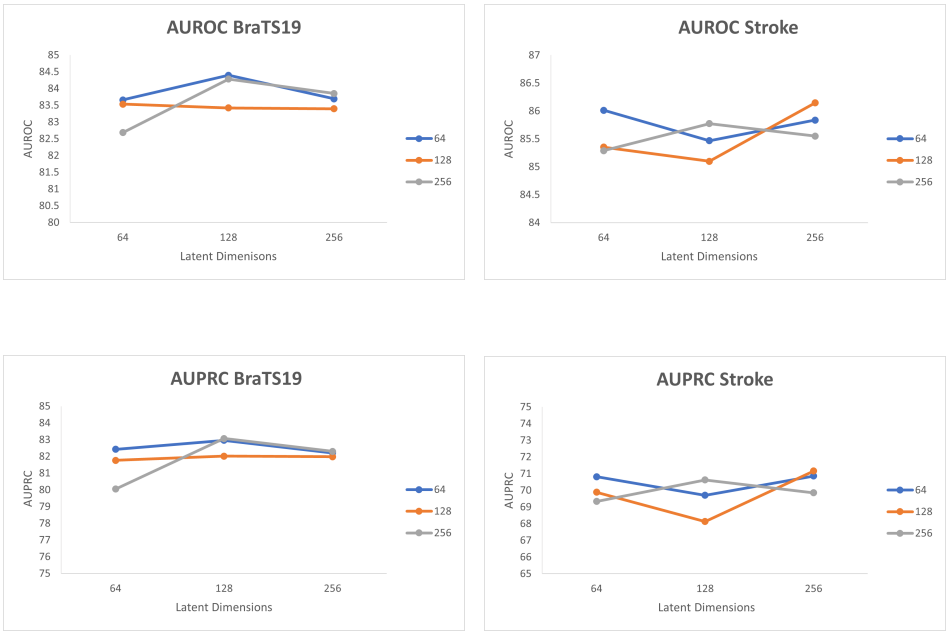


Fig. B.3: Comparison of slice-wise detection of VQGAN with varied parameters codebook vectors and codebook dimensions.

C Appendix

Pixel-wise Evaluation Graphs

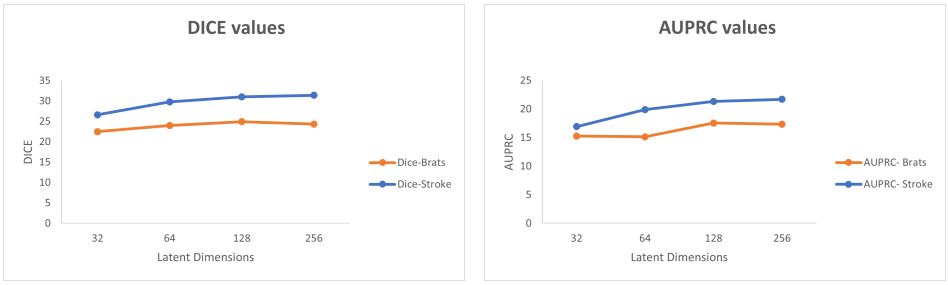


Fig. C.1: Comparison of pixel-wise evaluation of VAE for different dimensions.

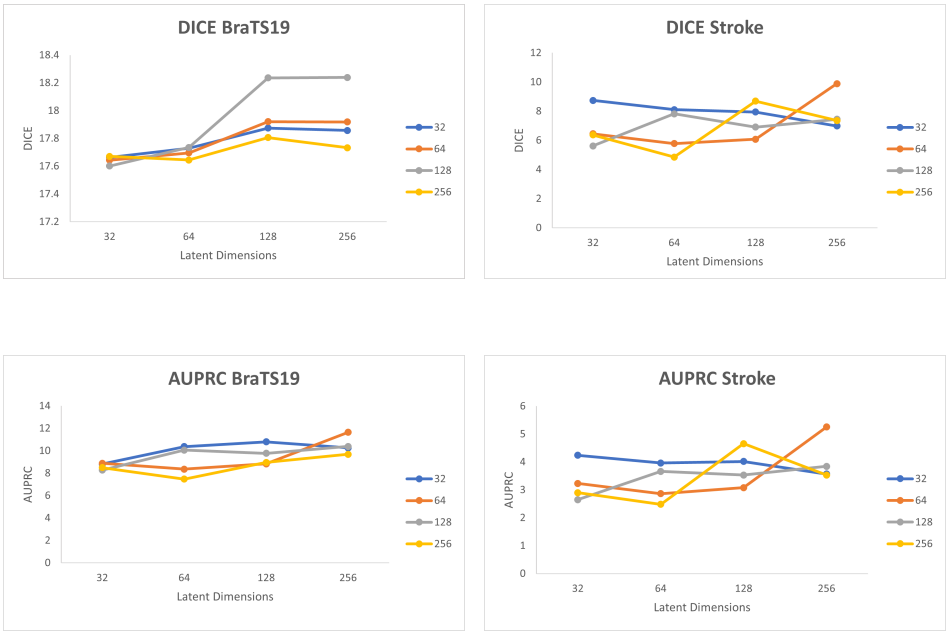


Fig. C.2: Pixel-wise Anomaly detection for VQVAE with latent sizes and vectors from 32 to 256.

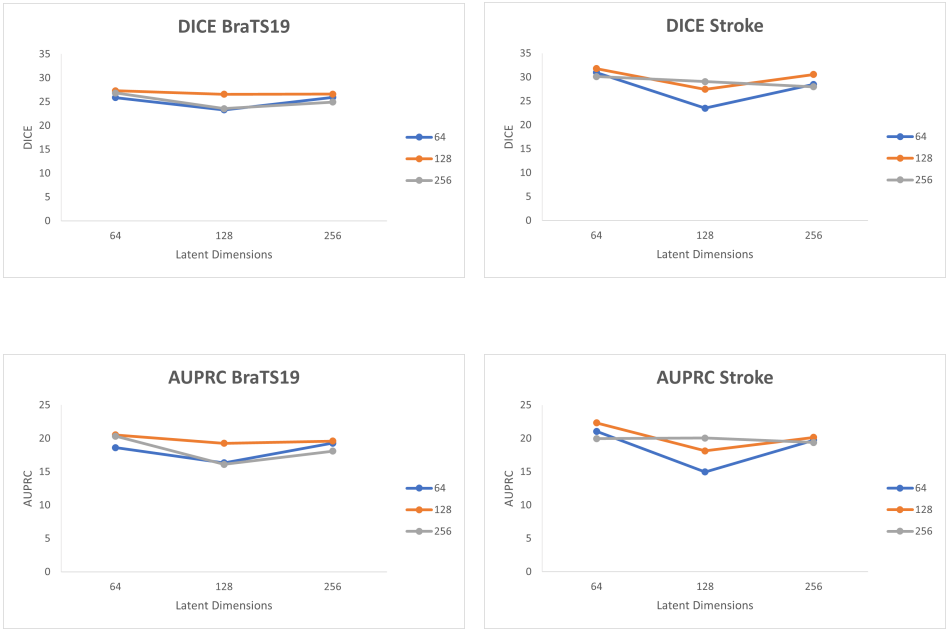


Fig. C.3: Pixel-wise Anomaly detection performance on VQGAN model.