Computational Science and Engineering
(International Master´s Program)

Technische Universität München

Master´s Thesis

Self Supervised Out of Distribution detection for Medical applications

Abinav Ravi Venkatakrishnan

Computational Science and Engineering (International Master's Program)

Technische Universität München

Master's Thesis

Self Supervised Out of Distribution detection for Medical applications

Author:	Abinav Ravi Venkatakrishnan
Director:	Prof. Dr. Nassir Navab
Supervisor:	Dr. Kim Seong Tae
Submission Date:	August 13, 2020

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

August 13, 2020 Abinav Ravi Venkatakrishnan

Acknowledgments

There are numerous people whose effort involved in keeping me motivated towards finishing this thesis. First I would like to thank Dr.Kim Seong Tae and Prof.Nassir Navab for their guidance throughout the thesis period. I am eternally thankful for the enlightening discussions and support of Dr.Seong Tae through the period of global pandemic keeping me motivated to complete the thesis.
Next I would like to thank my employers at Deepc gmbh Dr.Franz Pfister ,Julia Moosbauer, Paul Mayer and Michael Meyerhoff for providing the data and Computation power for doing this thesis.
I would like to thank my friends Abhijeet, Sricharan, Vishnu and Naveen whose encouragement and support helped me a lot. I would next like to thank my parents who have been a pillar in this phase of life standing by me while I keep aiming for my dreams.
At last I would like to thank lord almighty for keeping me in good health amidst a global pandemic and bestowing grace upon me to complete my thesis.

Abstract

Deep learning has been widely used for computer-aided diagnosis based on the large size of training data and advanced techniques for training large models. However, it highly depends on training data annotated by medical experts which is a time-consuming and expensive task. Moreover, even if the annotation is available, supervised learning-based approaches are not only limited to already known bio-markers but also suffer highly imbalanced data. In other words, most of the scans collected in the hospital consists of normal subjects or major pathologies. If the model is trained with the class-imbalanced data, it is not easy to detect various abnormal cases which should be treated in golden time. Recently, autoencoder-based unsupervised anomaly detection methods have been explored to handle this issue by training the model with a large size of normal scans and detect abnormal cases by calculating the reconstruction error. To use the autoencoder framework for the anomaly detection purpose, it is important to fully understand the normal scans. In this thesis, to address this issue, a novel self-supervised learning method has been proposed. The proposed method largely consists of two parts. The first part is a variational autoencoder framework that is trained to reconstruct the input image from latent features. To exploit the normal scans for the ability of the variational autoencoder, we pretrain the model with a self-supervised learning scheme that uses context restoration. The pretraining with context restoration encourages the model to better learn semantic image features. The second part is to train a model to predict geometry transformations. By training the model to predict geometric transformations, the model could effectively learn the image features and the distribution of normal scans. Finally, in the test phase, the anomaly scores measured from the reconstruction part and the geometric transformation prediction part are aggregated to improve the performance of the anomaly detection task. We validate the proposed method in brain CT data. By comprehensive and comparative experiments, the effectiveness of the proposed method is verified for brain anomaly detection.

Contents

Part I.

Introduction and Background Theory

1. Introduction

1.1. Motivation

Deep learning currently works on human level for tasks on image classification by [28] and this is possible only when the distribution of input data and the distribution of the output data are similar i.e. if the classifier is being trained on classes of liver, spleen, and stomach as input then the classifier does a very human-like job in identifying only these classes but when a gall bladder belonging to same region is shown the predictions tend to be overconfident and they predict the gall bladder as one of the three classes that it knows. These overconfident predictions can be very dangerous in applications that require high precision such as Autonomous driving or medical imaging. The cost of such overconfident predictions may be very expensive in the above-mentioned application and hence the classifier should be able to detect things that are out of its input distribution thereby having a distinction between what it knows and what it doesn't.

In the medical field, the cost of obtaining labels for the task of segmentation and classification is a very expensive one and hence training deep learning models in supervised method is very expensive. Moreover even the labels that are obtained also belong to major classes such as bleeds and minor anomalies the number of data points is less.This leads to very high class imbalance in training of the model. To avoid using supervised learning due to above mentioned limitations and collection of labels which are expensive the problem can be formulated in the following way. We can learn the representation of only the healthy class of subjects, the data of which are in abundance and easier to get. The model can then flag the cases that do not conform to this representation of healthy as anomalous samples. This is known as unsupervised anomaly detection.

Previous work in the field of unsupervised anomaly detection such as [40], [3], [45], and [46] use a combination of well-established architectures such as Autoencoders, Variational Autoencoders, Generative Adversarial Networks(GANs), etc to learn the distribution of healthy images and then use the residual from the reconstruction of the original image by these architectures to predict whether the sample is anomalous or not.

1.2. Problem statement

Deep Neural Network can learn non-linear features that do the task with precision. Deep learning has been very successful in medical applications. The goal of this thesis is

- To explore Self supervised learning to learn the distribution of the images of CT scan

- Develop anomaly detection with a new framework of Out of Distribution detection
- Provide a Model that can detect the anomaly as a complete volume and localize the location of anomaly.

1.3. Contribution

The main contribution of this thesis are the following

1. A novel architecture for unsupervised anomaly detection in brain CT scans. This architecture contains two phases of training. The first phase is pretraining the Autoencoder network for context restoration based tasks. The second phase involves the addition of classification head to the encoder and then training the network to do both self supervised classification and reconstruction. This ensures that we can classify whether a sample is healthy or anomalous and get the corresponding segmentation from it.
2. A novel anomaly detection score that combines reconstruction score from the Variational autoencoder and the classification head of the architecture.
3. Novel training loss for the training of the unsupervised anomaly detection scheme that combines cross entropy loss and the reconstruction based loss.

2. Related Work

2.1. Out of Distribution detection

The first work out of distribution(OOD) detection is by [21] where they use softmax score as the out of distribution detector. The idea is that in-distribution samples will have a higher softmax score compared to Out of Distribution samples. In [29] they make use of ensembles of deep neural networks as a means to detect the OOD samples. [31] uses temperature scaling proposed in [19] to increase the difference in distribution between the In and OOD samples. It also makes use of Input pre-processing to make the classifier more robust to adversarial examples. All these methods use Supervised learning methods to determine whether the sample belongs to the in-distribution of trained samples or is OOD. In [22] attempt to use OOD detection for detecting anomalies by exposing the classifier to Out of distribution samples during the training phase itself. This is the first work that attempts Out of distribution detection for Unsupervised learning. In the paper [23] use self-supervision inspired from [15] to construct an OOD detector that can classify samples as In or Out of distribution. From the paper, we can infer that self-supervision overtakes the state of the art method from supervision approaches. Hence this work is inspired by the approach from this paper.

2.2. Anomaly Detection in Medical Imaging

Unsupervised Anomaly detection has been a very long sought after problem in the field of medical image analysis. A review of traditional methods such as content-based retrieval, clustering, and Outlier detection for CT scans has been done in [42]. With the development of Autoencoders first proposed by [37] which brought on the abilities to learn nonlinear transformations on images was used for Unsupervised anomaly detection in brain images. These Unsupervised anomaly detection methods can be divided into two categories of

1. Reconstruction based methods
2. Restoration based methods

Reconstruction based methods

Reconstruction based methods use a pixel-wise difference between the reconstructions from Autoencoder based structures and the input image to determine the anomaly in input-image space. [38] apply the Autoencoder based approach to do anomaly segmentation in Brain CT in a successful manner. [3] and [1] apply Autoencoder based approaches to MRI brain anomaly segmentation. In [46] use concept of context inpainting to reconstruct healthy image slices from missing parts of image. All these above methods made use of linear bottleneck which hindered in the reconstruction of the healthy image slices. In [27] show that learning from a distribution warranties better quality reconstruction. In [45] show that using VAE helps in improving the anomaly segmentation for MRI scans by using both kl-divergence and reconstruction objectives for training the network.

Restoration based methods

Restoration based methods in [39] and [10] try to learn the image distribution of the normal data by moving to closest point in the normal manifold and then use pixel wise difference to find out anomaly detection.

2.3. Self Supervised Learning

Self supervised learning is gaining attention due to its superior performance in tasks of computer vision such as SimCLR [8]. The latest such self supervision has been used for Out of distribution detection is by [44]. Other self supervised learning techniques used in learning visual representation that is of interest to us is [35] which is Context inpainting to learn features of images. Self supervision has been used in medical imaging and specifically in Brain anomaly detection in [46] where a small part of the brain is masked and reconstructed using a Variational autoencoder. In [7] they use misplace small patches and try to learn the context of images by reconstructing the original images from the images where patches have been jumbled. They then proceed to do further downstream tasks of segmentation, classification etc. Further explanation of Self supervised learning has been done in 3.6

3. Background

3.1. Computed Tomography Imaging

Computed tomography is a very popular in-vivo technique for medical imaging of patients. The images from CT scans can be reformatted in multiple planes and can even generate 3-D images. An advantage of CT scans over X-rays is that they have better information regarding internal organs, bones, soft tissue and blood vessels.

Computed Tomography is an imaging procedure in which beam of X-rays are focused on a patient and rotated around the body producing signals that are processed to create slices of images of the patient. The X-ray detectors in the CT scanners are special and are located opposite to x-ray source. These signals are then passed on and by using numerical techniques of back-projection and radon transforms the signals are converted back to images that represent the organ of interest.Many organs such as Lung, Heart, Kidneys, Brain and Neck and Abdominal region can be imaged using CT imaging. In this thesis the organ of interest is the Brain tissue.[6]

The most common use of CT scans in brains is to detect pathologies of Bleeding, Brain Injury, Cavernoma, Atrophy, Aneurysm, Tumor etc. in a patient's brain. The CT scans can also be used in detection of Skull fracture and the associated bleeding in the Sub Arachnoidal regions.

3.2. Machine Learning

3.2.1. Introduction

Machine learning is a sub-field of Artificial Intelligence. The main advantage of machine learning over the existing techniques of Artificial intelligence inspired by logic is the ability to learn from experiences and the data collected from the experience. The main goal of machine learning is to optimize a model on this data and then generalise it to similar data distributions.

Definition 3.1 *Every machine learning problem consists of three main components of Performance P, task T and experience E. A computer program is said to learn if it improves performance P while doing task T based on experience E [33].*

In the case of this thesis the task T is finding the out of distribution sample, The Experience E is the data of Brain CT scans and the performance P is given by Metrics that measure the classification of the model, the segmentation of the model etc. In machine learning experiments generally the best practice is to split the data into 3 parts of Training data, Validation data and testing data according to [25]

- **Training Set:** The data which is used to optimize the weights of the given model to make accurate predictions.
- **Validation Set:** The data that is used to generalize by choosing hyperparameters for the model and measuring performance of the model.
- **Test set:** Once the model is frozen with hyperparameters the test set is then used to measure the generalization performance across the data. This set is not to be touched during the optimization or hyperparameter selection process.

3.2.2. Types of Learning

There are mainly two main types of learning approaches as mentioned in [34]

- **Supervised learning**: The main goal of prediction is to learn a mapping between the input data to the target variable or label. According to [41] Mathematically it is given as $f : X- > y$ where X is the domain of the data, y denotes the set of labels and f is the function that maps the data with the labels
- **Unsupervised learning**: In Unsupervised learning the goal is to find some structure from the data without the help of labels. Mathematical representation varies with the task.

3.3. Deep learning

Deep learning is a sub-field of machine learning which uses artificial neural networks [20]. Machine learning algorithms such as K nearest neighbor, Logistic regression, linear regression were considered as Linear models which couldn't handle much complexity. Kernel methods were a good replacement but were very slow and tedious in estimating complex functions [18].

Mathematically, this concept can be expressed as a nesting of functions:

$$f(x, w_0, w_1, ..., w_K) = \sigma_k({w_K}^T \sigma_{K-1}(w_{K-1}^T ... \sigma_0(w_0^T x))) \tag{3.1}$$

where σ_k are known as activation functions. They control the linearity or non-linearity of the function approximation. The above layered structured is called *Multi-Layered Perceptron* (MLP) or **Fully Connected Network**. The layers are known as Fully connected layers.

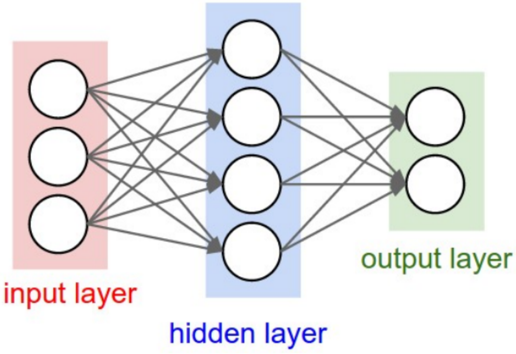


Figure 3.1.: Fully Connected Network from Lecture notes of Course CS231n [25]

Backpropagation

The backpropagation algorithm [30] is the bed rock of all deep learning methods. The back propagation algorithm computes the gradient of an objective function with respect to weights of a multi-layer stack of modules is a practical application of chain rule of derivatives. The key inference from the paper is that gradient of the objective with respect to the input of a modules can be computed by working backwards from the gradient with respect to Output of the module.

The backpropagation procedure can be applied repeatedly to propagate gradients through all functions starting from output to the input which creates a possibility to train a neural network end to end.

3.3.1. Activation functions

An activation function is a function used to introduce non-linearity in the neural networks. An activation function is generally used in between 2 layers. Some of the activation functions that are very common in usage are

- **Sigmoid**: The sigmoid function maps any real value to [0,1]. This is very useful for getting probabilities of a class. The main disadvantage of a sigmoid function is that when the derivatives are big or very small the gradients vanish and stops the optimization process [17]

- **Tanh:**Hyperbolic tangent function is a zero centered logistic function which maps the real input between -1 and 1. They also face with vanishing gradients problem since if the input is large or small the gradients saturate. [17]

- **Rectified Linear Unit (ReLU)**: Rectified Linear units is a function represented by ReLU = max(0,s). This function provides fast convergence and doesn't saturate like

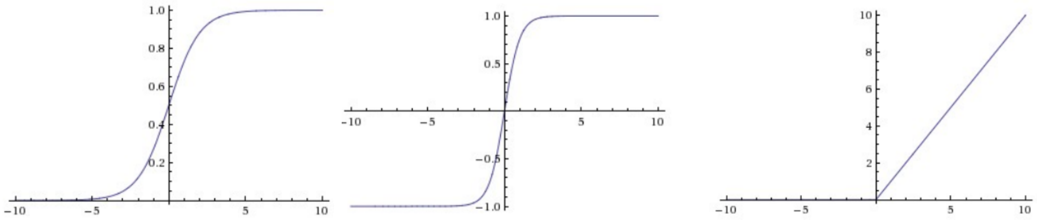


Figure 3.2.: (a)Sigmoid activation function (b) Tanh activation function (c) ReLU activation function. All images taken from [25]

sigmoid and tanh functions. The disadvantage is that when the output of a ReLU is negative then the gradients die out due to the flat part of the function. A small practical tip is to initialise the neurons with small positive biases (0.01). This is known as Leaky ReLU. [17]

3.3.2. Loss functions

Loss function is used for optimization of the gradients. Loss function is generally minimized by using Optimizers such as Stochastic Gradient Descent, Adaptive Momentum etc. Loss function generally measures the difference between the ground truth label and the data prediction at the point and with optimization of the loss function the model learns to predict the data better [17]. Some of the loss functions that are important in this thesis is

- Cross Entropy loss for multi class classification
- Mean Square error loss for the reconstruction of images
- KL-Divergence loss which measures the distance between two probability distributions.

3.3.3. Optimization

Optimization is an important component in the machine learning pipeline. The learning process happens when there is an optimization of loss or reward function. The general task of optimization can be described as

$$\theta^* = \arg\min_{\theta \in \mathcal{X}} f(\theta) \tag{3.2}$$

where θ is the vector of parameters which need to be optimized constrained by the domain $\mathcal{X}$. In Numerical analysis there are very well established techniques for optimization of Linear systems such as Least Squares method [16], Jacobi [12] and Gauss Seidel method which fall under class of iterative solvers and Conjugate gradient method [11].

But deep learning problems and the loss functions that are being optimized are highly non-linear in nature and the above mentioned algorithms don't work anymore. The above mentioned algorithms always look for a global minima which is not always attainable in the case of the Non linear problems. Hence we need optimizations that atleast achieve a local minima.

Gradient descent

For a differentiable function f is **gradient descent (GD)**. It is based on iterative scheme which can be given as

$$\theta^{k+1} = \theta^k - \alpha \nabla_\theta \mathcal{L}(\theta^k) \tag{3.3}$$

where θ^k represents the parameters of the algorithm at iteration k, α is a scalar representing the magnitude of the step and $\mathcal{L}(\theta^k)$ represents the loss function. α is also called learning rate. The size of learning rate has to be chosen carefully in order to attain the local minima. To prevent this issue *Learning rate schedules* can be used in increasing or decreasing the learning rate with different settings depending on loss value, iteration etc.

Stochastic Gradient Descent (SGD)

For large data one pass through the entire data will take a lot of time and computation of gradients can be slow. To combat this we can use stochastic optimization where there is an assumption that exact expectation value of the gradient can be approximated by the expected value of the smaller set of samples. Such sampling into mini batches so that we dont see the data twice. A complete iteration over the entire data is called epoch. SGD converges to a global minimum if the objective function is convex. To improve speed of the SGD optimization there are first and second moments that are applied that can cause faster convergence.

Adaptive momentum estimation

This method combines the first and second momentum from the Stochastic Gradient Descent.

$$m^{k+1} = \beta_1 m^k + (1 - \beta_1) \nabla_\theta(\theta^k) \tag{3.4}$$

$$s^{k+1} = \beta_2 s^k + (1 - \beta_2)[\nabla_\theta(\theta^k) \odot \nabla_\theta \mathcal{L}(\theta^k)] \tag{3.5}$$

$$\theta^{k+1} = \theta^k - \alpha \frac{m^{k+1}}{\sqrt{s^{k+1}} + \epsilon} \tag{3.6}$$

to correct for bias of the estimators, we use $\hat{m}^{k+1} = m^k/(1 - \beta_1)$ and $\hat{s}^{k+1} = s^k/(1 - \beta_2)$ replace m^{k+1} and s^{k+1} with $\hat{m}^{k+1}$ and $\hat{s}^{k+1}$ as shown in 3.6. Most of the algorithms used in this thesis use Adam estimation as it is expected to converge the fastest among other options such as AdaGrad, SGD with momentum etc.

3.3.4. Convolutional Neural Networks

The disadvantage with fully connected networks is that they take a lot of computations and occupy a lot space while computing. For images Convolutional neural networks are very common in usage. In general convolutional neural networks(CNNs) are used to process data that comes in the form of multiple arrays or tensors. The four key ideas that are reason for their success is the usage of shared weights, local connections, pooling and the depth that we can adapt while designing the architecture.

Shared weights restrict the degrees of freedom in an algorithm. This can be illustrated with the case of images. A fully connected network for a 100x100 image needs about 10^4 weights. As the layers increase the number of weights increase which make it impractical. Sahring this weights for different parts of image reduce the order of weight to 10^2.

Convolutional layers

In CNNs one filter would be equivalent of a neuron so a Convolution layer would be formed with several filters. Each convolution later is followed by an activation function typically ReLU as in 3.3.1. A 2d convolutional layer can be specified with filter width, filter height and number of filers. The depth of filter has to match and it implicitly given. Some of the other components important for understanding a CNN are

- **Stride**: The assumption is that now the filters have been moving pixel by pixel. But one can choose to apply the filter every n-th spatial location. This step of sliding is called a stride. In general , having an input of NxN and a filter of FxF with a stride of S, the output is of size $(\frac{N-F}{S}+1)(\frac{N-F}{S}+1)$ if $(\frac{N-F}{S}+1)$ is non-integer it is forbidden.
- **Padding** When applying convolutions the size of images shrink by a factor with respect to the input. With increasing depth of network the one can end up with very shrinked outputs. One workaround is to pad the images. Padding means adding a layer of values over the original image size to prevent this shrinking. Most common form of padding is zero-padding. There are other options available such as reflection padding etc. If we add a padding layer P then the output is expected to be $(\frac{N+2P-F}{S}+1)(\frac{N+2P-F}{S}+1)$

Pooling layers

In CNN architectures, it is common to include after some convolution layers an additional layer called the pool layer. This pooling operation is a fixed function, as a max(...) operation or an average operation. These layers however, do introduce their own hyperparameters F and S for their filter size and stride. Having for example a Max Pooling Layer over a matrix of 4 × 4, with F = 2 and S = 2, yields a pooled output of 2 × 2 in which the elements correspond to a block-wise maximum of the input.

The role of the convolutional layer is to detect local conjunctions of features from the previous layer, and the role of the pool layer is to merge semantically similar features into one. Pooling layers reduce the dimension of the representation and create an invariance to small shifts and distortions.

3.4. Autoencoder

An autoencoder [37] is a type of Neural network that learns the distribution of data by compressing and reconstructing it back to the input space. In this form of representation learning the High dimensional data points are reduced to Low dimensional data points by using an encoder part of neural network that compresses the information. It is an extension of Principal Component Analysis (PCA) where only linear transformations are used for the reduction of dimensionality of the data points. However PCA fails in the case of very high dimensional data as Linear transformations aren't sufficient to represent these data points.

The autoencoder consists of three main components Encoder, Decoder and Bottleneck. The encoder can be mathematically represented as $h = f(x)$ where x is the input data and f is the function that is being approximated by the Neural network h gives the latent space which is a compressed representation of the input data x. The decoder can be represented as $r = g(h)$ or $r = g(f(x))$ where the r is the reconstruction of data input x.

When the dimension of h is less than x the autoencoder is then called **undercomplete**. Learning using such an autoencoder helps us identify the most important features in the data. The Loss function that is used as objective for minimization is given by $L(x, g(f(x)))$. This function L is generally as reconstruction loss given by $L1$ or $L2$ norm in the euclidean space.

3.4.1. Autoencoder for Unsupervised anomaly detection

Autoencoder structures are one of the first architectures used for Unsupervised brain anomaly detection by [1]. The Autoencoder is made of convolutional layers and can be seen in figure 3.3

The autoencoder is trained using the Reconstruction loss between the input and the reconstructed output of the network. The reconstructed output is then thresholded to give the lesion mask and segmentation metrics are calculated.

3.5. Variational Autoencoder

The variational autoencoder was first introduced by Kingma et.al in [27] and is a model that constraints the latent space distribution to known prior distribution. The learning problem is formulated in a probabilistic method. Let x be the input data and z represent

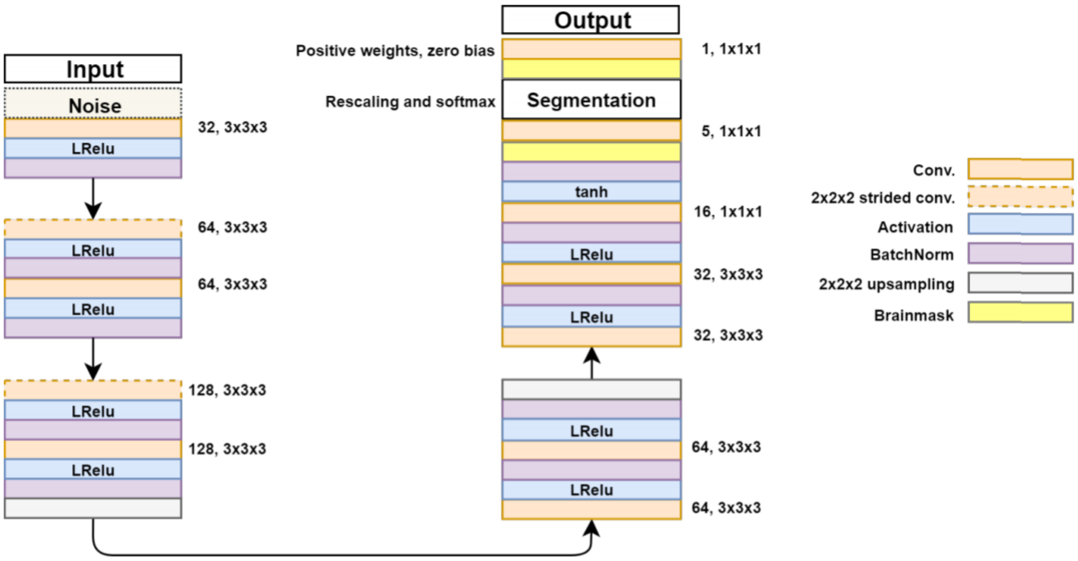


Figure 3.3.: Autoencoder architecture for Unsupervised anomaly detection [1]

the latent space representation which models the joint probability distribution as $p(x, z) = p(x|z)p(z)$. The prior can be defined as samples being drawn from a defined sitribution of latent samples. The objective is to minimize the likelihood $p(x|z)$ on conditioned on the prior. Generally the prior is chosen as Normal distribution or Gaussian distribution with zero mean and standard deviation eqaul to one which can be represented as $\mathcal{N}(0, 1)$. The output of the network which is to reconstruct the input from the minimization of likelihood condition on latent distribution is known as the posterior distribution.

This posterior distribution is intractable and an approximate posterior distribution is required. From Bayesian analysis we can use Variational Inference seeks this approximate posterior through a family of distributions.Variational inference [18] allows for this by optimisation by maximising the Evidence Lower Bound (ELBO). Similarity in distribution can be measured by minimising Kullback Leibler Divergence, Jensen Shannon(JS) divergence which are metrics that measure distances between distributions. Linking this probabilistic model to neural networks the VAE parametrises the approximate posterior distribution using the encoder and decoder. The Joint distribution is now parametrised by the neural network.

3.6. Self Supervised learning

In medical imaging dataset collection is a huge difficulty. Supervised learning limits the application of deep learning to medical data. Since the medical annotations need to be done by technical experts the cost of such methods go up drastically. So self supervised learning solves this problem by creating pseudo labels from the data itself.

Self supervised learning is a subset of unsupervised learning methods. In self supervised learning the neural networks learn features by using automatically generated labels. These self supervised learning are learnt from pretext tasks which are used for learning visual features from the data.

From [24] we can see that These pretext tasks have 2 common properties

1. Features need to be extracted by the convolutional neural networks for further processing
2. Pseudo labels for the pretext task can be generated from the attributes of images itself.

Generally shallow layers capture general low level features like edges,corners and textures while deeper layers capture task related features.

The general workflow of Self supervised learning is given in 3.4

3.6.1. Formulation

In formulation Self-supervised learning is similar to supervised learning setup. There dataset is represented by (X_i, y_i) where X_i represents the datapoint and y_i represents the labels which are human annotated in the supervised learning setup whereas in the self supervised learning setup these are determined by pretext tasks. The label is represented by P_i where it represents the pseudo label instead of ground truth y_i.

Given a set of N training data represented by $\{X_i, P_i\}_{j=1}^{n}$ the trainig loss function can be defined as

$$loss(D) = \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} loss(X_i, P_i) \tag{3.7}$$

The labels P_i if automatically generated then corresponds to Self supervised learning.The general schema of Self supervised learning is given in the figure 3.5

3.6.2. Common Pretext Tasks

A Pretext task is a task that is used to generated labels from the data or to extract features from the data. The pretext tasks can be put into four broad categories. They are

Generation based methods

This type of methods learns features by learning pretext tasks that involve Image generation. Some of the very common tasks are context inpainting as proposed by Pathak et.al [35] and using GANs for self supervision such as chen et.al [9]

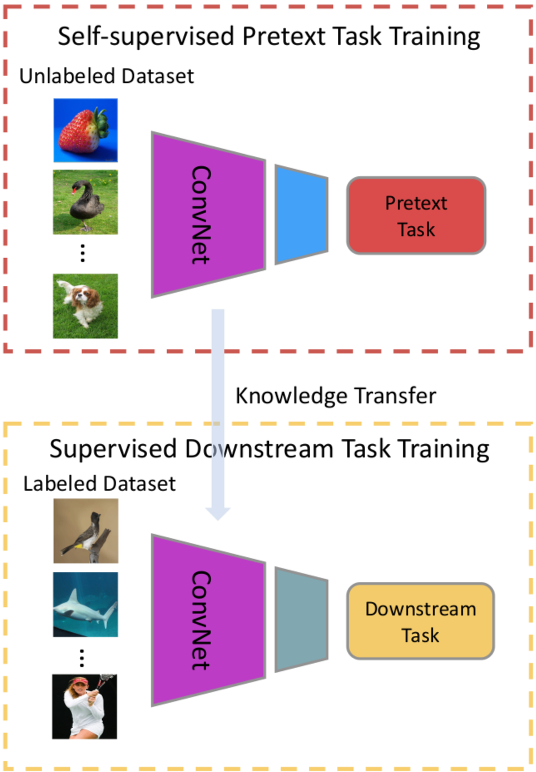


Figure 3.4.: General pipeline of the self supervised learning networks. The pretext tasks are used to learn features and then the downstream tasks are used for accomplishing task such as object detection, segmentation etc. [24]

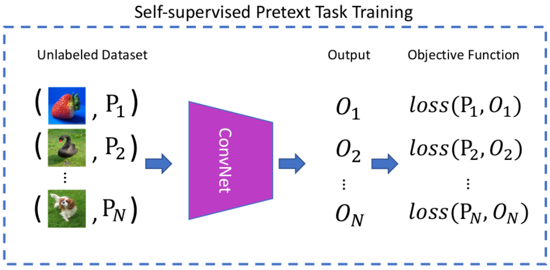


Figure 3.5.: Figure shows the Self supervised learning schema of generating labels and optimisation using loss function [24]

Context-Based methods

This method designs tasks based on context features of images such as context similarity,spatial structure, temporal structure etc.

Free Semantic label methods

In this method some transformations are applied on the data and trained on the prediction of these transformations as a pretext task [15]

Cross Modal based methods

This type of task design involves training convolutional neural networks to verify whether two different channels of input data are corresponding to each other. These are generally used in 3D images and in video data.

In this thesis we mainly use context based methods such as Context restoration [7], Context Inpainting[46], Geometric transformation [23].

Part II.

Methodology

4. Dataset and pre-processing

4.1. Dataset

The dataset used in the thesis is an In-house dataset from deepc gmbh and TUM Neuroradiology consisting of 179 healthy patient CT scans and 44 Anomalous CT scans. The distribution of the Anomalous samples is across different pathologies and are as follows

Pathology	Number of volumes
Atrophy	20
Intra cranial bleeding	11
Ischemia	9
Cavernoma	1
Aneurysm	1
Bleed	1
Tumor	1

The class imbalance has been shown in figure 4.1

The dataset is split as 149 volumes for training with 20632 usable slices (eg:non zero) The training data is then further split into 20% of healthy training volumes as Validation set i.e. 4126 slices will be used for validation from the training set while 16506 slices of images will be used for training. There are 30 healthy volumes for testing with 4314 usable slices.

Normal volumes	Number of volumes	Usable slices
Training set	149	20632
Test set	30	4314

Some examples of dataset has been given in figure 4.2

4.2. Pre-Processing pipeline

In the Preprocessing stage we have a pipeline that takes a raw unprocessed image in nifti format and then the pipeline converts it to an image usable for deep learning by performing operations of resampling, brain extraction, and rigid registration. The pipeline is illustrated in 4.3. Each step is explained briefly in the following paragraphs.

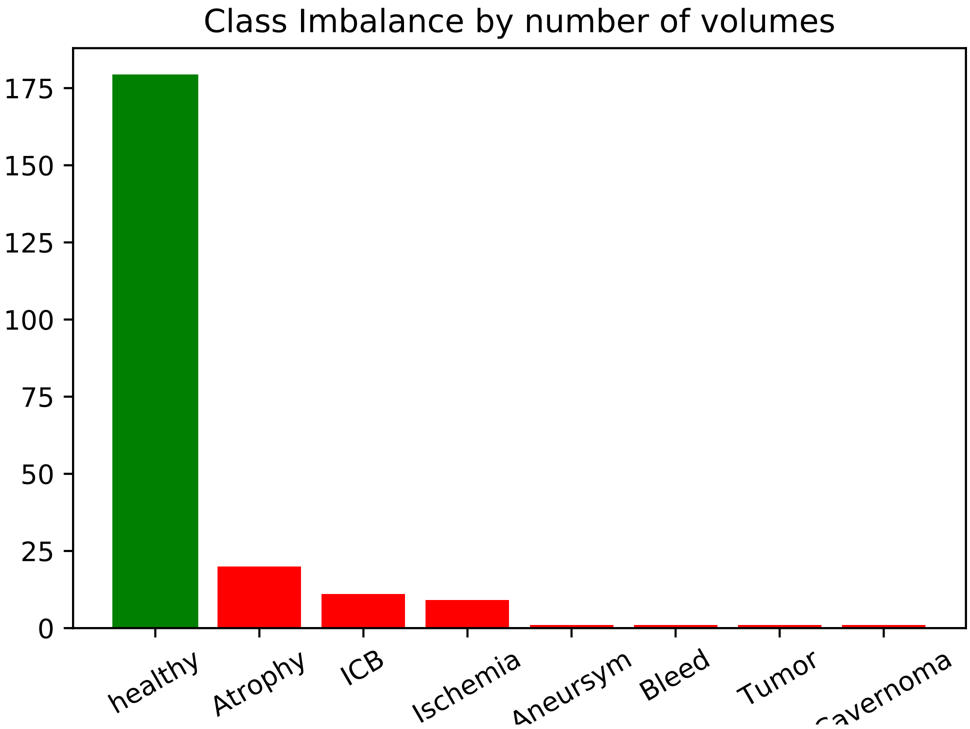


Figure 4.1.: Distribution of Normal and Pathology in the dataset, The green bar represents the Normal images and Red bars represents the anomalous samples

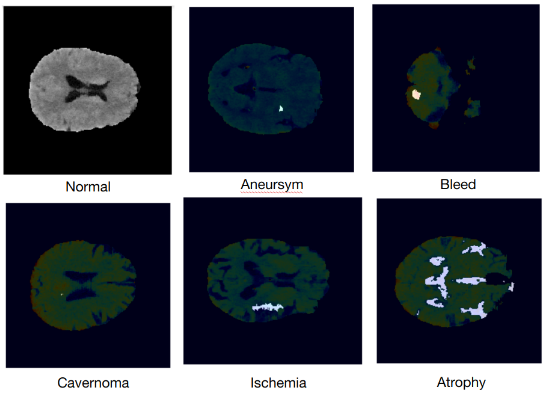


Figure 4.2.: Samples of Normal and Anomalous images with labels of the pathology

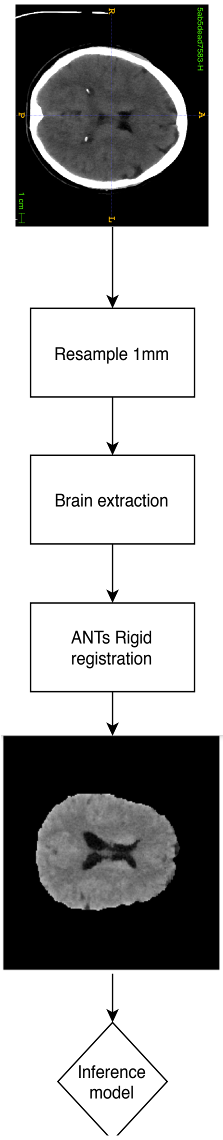


Figure 4.3.: Preprocessing pipeline for CT scans

In first step resampling is performed using ANTs a registration tool [2]. In the resampling process a linear interpolation is used to bring the resolution of the image to 1x1x1 mm. The command used for resampling is ResampleImageBySpacing from the ants library.

In the second step we use the library fsl which is generally used for MRI preprocessing to extract the brain image. We use the bet and fslmaths commands from the library. We extract brain only masks from which removes the optic nerve, traces of skull voxels. Skull stripping is then performed on these extracted masks.

In the third and final step the rigid registration is done using ANTs. The affine rigid registration uses nearest neighbor interpolation. The result of all these steps is a volume that has a shape of 240x240x155 where there are 155 slices. Then the slices are extracted from the volume of the nifti files to give 155 nifti images per volume each having size 240x240x1. The affine rigid registration was chosen over non-linear registration so as to keep original properties of the brain since non-linear registration deforms all brains to a similar template. An in-house ATLAS was used for the registration purposes.

5. Methodology

In this section the proposed Idea is explained in greater detail with relevant and distinct inputs being specifically mentioned.

5.1. Data Input

From previous literature and experiences it was decided that input would not be 3d volume as a whole but the 3d volume split into distinct slices as obtained after registration. Certain Image processing was done during loading the data to ease the load on the network training process.

Slice Selection

Slice selection is very common process in the field of anomaly detection. Generally only central slices are selected and given for training as they have maximum probability of finding lesions. But we dont do that in our dataset since there are cases in which the anomalies are in the initial and the rear end of the volumes. But the elimination of blank slices is done since there is no useful information to learn from these blank slices. Slices with brain matter less than 10 pixels have been removed. With this the total number of training slices amount to 20630 trianing slices.

5.2. Anomaly detection by Context restoration and Reconstruction

This anomaly detection involves a two stage training process. In the first stage an autoencoder is trained with the objective of restoring the context and learning the visual features from the context restoration task. In the second stage a fine tuning of the network is done to improve the reconstruction of the images. The following section explains in detail the components involved in building the model.

5.2.1. General Pre-processing

After the pre-processing done with raw images in 4.2 further image pre-processing is done. The image which is in Nifti format is loaded to the network as numpy array [43] using the Nibabel package [5]. After this we generally normalize the image between 0 and 1. The image is then scaled to 128x128 to reduce the computational load on the network.

Rescaling the image helps in not losing the parts of images but just interpolating them to smaller size.

5.2.2. Loading the data

The dataloader is a very important part of the pipeline which self generates labels or helps in learning features according to the methods as described in 3.6.In this section we provided a detail of how the dataloader is built for this model.

Pretraining using context restoration

For the context restoration two points are extracted from non-zero parts of the image in axial orientation and then a patch of size 1/8 image size is created with the extracted point at center of the patch. These patches are then swapped. The dataloader then returns the swapped image and original image.

VAE reconstruction

For the VAE reconstruction model the image is fed as axial orientation.

5.2.3. Network Architecture

Initial Designs

Various designs of architecture was experimented with before arriving at the final architecture. Since the main task is to get a segmentation a U-Net architecture with skip connections was attempted but since the main goal is to reconstruct the image the skip connections by feeding the gradients fooled the network to function as an Identity function. Hence an Autoencoder was the next choice of networks. However with the autoencoder also the reconstruction were quite poor and masks output of the autoencoder based architecture did not yield any positive results. So we moved to a Variational Autoencoder based architecture.

VAE Architecture

The problem in anomaly detection is that the classes are very different in size, shape and intensity of the anomalous pixels. Some pathological cases such as stroke or Ischemia is varying in intensity with compared to an anomaly like Intra cranial bleeding(ICB). Most of previous work in anomaly detection generally work with Bleeds which are hyper dense pixels. In this work we try to capture all variations from the normal brain as a Anomaly.To have a very high resolution reconstruction a modular custom Variational autoencoder architecture is used. The size of the latent space and the maximum filter size is set to 512.

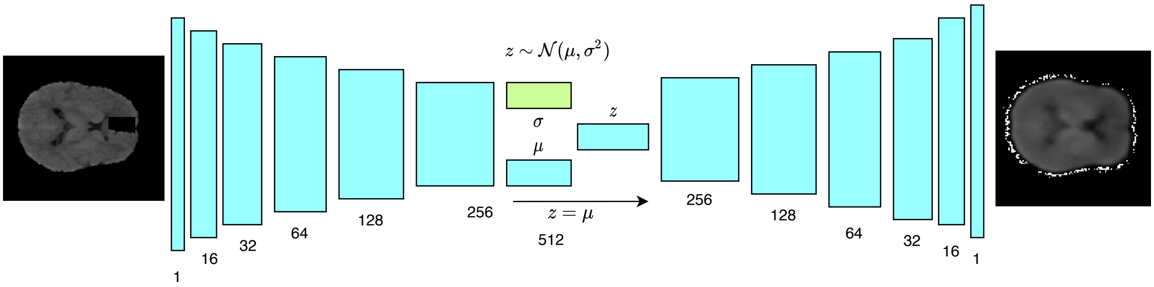


Figure 5.1.: Network architecture for Context restoration used as Pre-Training for the main architecture

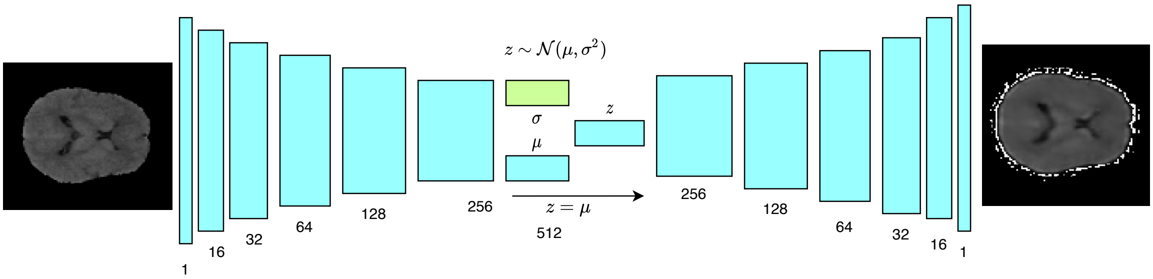


Figure 5.2.: Network architecture for Reconstruction fine tuning

The encoder is divided into 6 blocks. The convolutional filters have the same parameters throughout of kernel size 4, stride 2 and padding 1. In the first block the number of channels are increased from 1 to 16 and then in the further blocks until the latent space dimension the channel size has been increasing in order of 2 to give number of channels as 16,32,64,128,256 and 512.

We then apply the reparametrization trick from [27] to the convolutional latent space of 512 to split into two chunks of 256 channels each.A prior of Gaussian distribution with zero mean and unit covariance is applied. The decoder architecture is then applied which is same as encoder architecture with transpose convolutional parameters kernel size = 4, stride=2 and padding=1. The reconstruction is then obtained for the pretraining.

Both the encoder and decoder Batch normalisation is applied and the activation function is LeakyRelu where the slope is 0.1.

Loss functions

For the pretraining network the training objective of the loss function is the minimization of Mean Square error when comparing the reconstruction with the input image. If the input image is $\mathbf{x} \in \mathbb{R}^N$ and the reconstruction is given by $\hat{\mathbf{x}}$ the mean square error is given as

$$\mathcal{L}_{rec} = \|(x_i - \hat{x}_i)\|_2 \tag{5.1}$$

5.2.4. Training

For the training of the context restoration network we follow the steps given in Algorithm 1

Algorithm 1: Training Algorithm for Pre-training with Context Restoration

```
Initialization: parameters of encoder $Q_\phi$ ,decoder $G_\alpha$ for epoch in totalEpochs do
    for batch in totalBatches do
        Sample $x_i$ from Inputimages $\mathbf{x}$;
        Sample $\tilde{x}_i$ from Swapped Images $\tilde{\mathbf{x}}$;
        Obtain reconstructions $\hat{x}_i$ from model $G_\alpha(Q_\phi(\mathbf{z}|x_i\tilde{x}_i))$ for i= 1,..n
        Optimize: $\mathcal{L} = \|\hat{x}_i - x_i\|_2$ using Adam optimizer [26]
    end
end
```

The stopping criterion for the training is to check the plotted losses and image of the reconstruction for the overfitting or learning or the context. Once the loss starts to divulge it is preferred to stop the training else the network starts memorizing the swapped patches which is undesirable.

Following this the second part of the training is done where we fine tune the network to improve the reconstruction of images. This training algorithm is represented by Algorithm 2

Algorithm 2: Training Algorithm for VAE with pretraining

```
Initialization: parameters of encoder $Q_\phi$ ,decoder $G_\alpha$ for epoch in totalEpochs do
    for batch in totalBatches do
        Sample $x_i$ from Inputimages $\mathbf{x}$;
        Obtain reconstructions $\hat{x}_i$ from model $G_\alpha(Q_\phi(\mathbf{z}|x_i))$ for i= 1,..n
        Optimize: $\mathcal{L} = \|\hat{x}_i - x_i\|_2$ using Adam optimizer [26]
    end
end
```

5.3. Anomaly detection by Multi task learning

This anomaly detection involves a two stage training process. In the first stage an autoencoder is trained with the objective of restoring the context and learning the visual features from the context restoration task. In the second stage a classification head is added to the latent space of the encoder part and then a novel loss proposed is minimized. The motivation for this network is that it can classify anomalies based on self-supervision and can

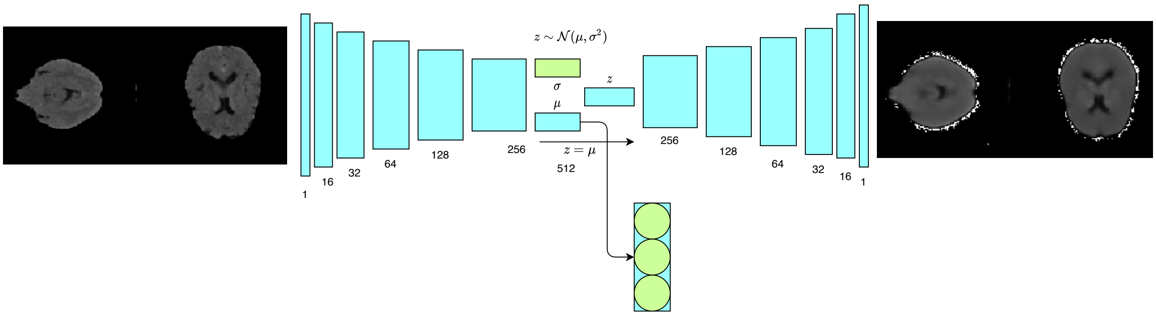


Figure 5.3.: Network architecture for Multi task learning

also offer reconstruction based segmentation masks for localization of anomaly.

The following section explains in detail the components involved in building the model. The general pre-processing of the data remains the same. But there is a variation in the self-supervision classification

5.3.1. Loading the data to the network

in Self supervised classification the dataloader is the most important component since it is in the dataloader that we generate automatic labels from existing data. In this case we randomly rotate the image by multiples of 90 and translate the image to approximately 1/8th image size on either sides. So the corresponding label for a data is a dimension 3 tensor which can be represented as follows $BxTxR$ where B represents batch size. T class for translation which has 6 classes, one class for each movement along the horizontal and vertical direction, R class for rotation which is a value in range 0-4.

5.3.2. Network Architecture

The architecture is similar to the 5.2.3 for the autoencoder part which does the anomaly localization. The new addition to the architecture is a fully connected layer from before the re-parametrization trick is applied. This fully connected layer has the total number of classes that are being predicted by the self supervised network.

5.3.3. Loss function

The loss function for the classification network is given by cross-entropy loss which can be mathematically represented as

$$\mathcal{L}_{classification} = -\sum_{i}^{C} y_i \log x_i \tag{5.2}$$

where y_i represents the label for the corresponding data point x_i. The reconstruction loss remains similar to equation 5.1.

For the multi task learning we combine both these loss functions with a scaling term to bring both the losses to similar scale and then reduce for the loss. The newer loss function is given by

$$\mathcal{L}_{multi_task} = \mathcal{L}_{classification} + \epsilon\mathcal{L}_{rec} \tag{5.3}$$

where ϵ is the scaling factor to bring both the losses to the same scale. The Adam optimizer [26] is used to optimize the loss in this case and a Learning rate scheduler for reducing the learning rate when the loss value attains a plateau is used.

Algorithm 3: Training Algorithm for Multi Task learning

```
Initialization: parameters of encoder $Q_\phi$ ,decoder $G_\alpha$ and classification head $C_\beta$,
for epoch in totalEpochs do
    for batch in totalBatches do
        Sample $x_i$ from Inputimages $\mathbf{x}$;
        Obtain reconstructions $\hat{x_i}$ from model $G_\alpha(Q_\phi(\mathbf{z}|x_i))$ for i= 1,..n
        Obtain cross-entropy loss from $\mathcal{L}_classification = c_\beta(G_\alpha(x_i))$
        Obtain the reconstruction residual $\mathcal{L}_{rec} = \|\hat{x}_i - x_i\|_2$
        Optimize for the joint loss $\mathcal{L}_{multi_task} = \mathcal{L}_{classification} + \epsilon\mathcal{L}_{rec}$ using Adam
          optimizer [26]
    end
end
```

The hyperparameters for training the models has been provided in

5.4. Inference

During inference a patient scan is fed into the network as 2-dimensional axial slices at a resolution of 128x128. Each slice is encoded back to the latent representation and then decoded back to form the reconstruction. Since model is trained only on healthy scans the anomalies fail to be reconstructed and the difference between the reconstructed output and the initial output provides the segmentation masks. A detailed approach is given in 6. For the multi task learning case 5.3 we get softmax scores from the classification head in addition to the reconstructions which are then combined to give meaninful results which are explained in detail in section 6.

Part III.

Results and Conclusion

6. Experiments

The main results that we aim to produce is segmentation masks from the reconstruction based models and a Classification score which is used to classify whether the sample is Out of distribution (Anomalous) or not. The main area of evaluation is how the discriminative power of the model is when classifying the anomalous and healthy images this is achieved with the help of metrics such as Area under Receiver operator characteristic curve and Area under Precision Recall curve. The approach proposed in this thesis is validated in several fronts. First We compare the results from the proposed architecture with the State of the art methods. Second we propose a new anomaly score for the multi task approach 5.3 Third we evaluate the ability of the reconstruction of methods proposed in 5.2 and 5.3 by checking the dice similarity coefficient. The reported results are averaged over 5 runs

6.1. Evaluation Metrics

AUROC - Area Under Receiver Operator characteristic curve

AUROC is a threshold independent performance evaluation. The ROC curve is a graph between False positive rate and True Positive rate . According to [21] AUROC can be interpreted as the probability that a positive sample has a greater detector score than a negative sample.

AUPR - Area Under precision recall curve

AUPR sometimes is more informative when compared to AUROC though both are threshold independent metrics [13]. The precision recall curve plots a graph between precision and recall. The baseline predictor has a AUPR value equal to precision in worst case scenario and in the worst case scenario the AUPR value is 1 (complete area under curve). The base rate of positive class influences the calculation and hence we must mention what is positive. In our case since we are aiming at finding Out of distribution samples we consider that as a positive case.

Dice Similarity coefficient

The dice similarity coefficient is a segmentation metric that is used to measure the similarity between two sets of data. It can be mathematically given as

$$DSC = \frac{2|X| \cap |Y|}{|X| + |Y|} \tag{6.1}$$

6.2. Anomaly score calculation

6.2.1. VAE with Pretraining

During inference the network is given 2 dimensional axial slices as input. The final anomaly score is the residual of the network output with the input image. This anomaly score is calculated for healthy images as negative class and Unhealthy images as positive class. We then calculate the AUROC and AUPR values and plot the ROC and PR curves.

6.2.2. Multi Task model with Pretraining

From the architecture of Multi task model 5.3 we can see that there would be two outputs from the model one from the classification head which gives the softmax score corresponding to the self supervision of rotation, translation etc and second being the corresponding reconstruction score by the Variational autoencoder. The combination of both the scores are done to get a uniform score. The score is calculated by using the normalized weighted averaging using parameter λ which serves as hyperparameter. The combination can be represented as follows.

$$score = (1 - \lambda)s_c + \lambda s_r \tag{6.2}$$

$$s_c = \frac{e^{f(y_i)}}{\sum_j e^{f(x_j)}} \tag{6.3}$$

$$s_r = \alpha \|x - \hat{x}\| \tag{6.4}$$

where s_c and s_r represents the anomaly scores from the classification head and reconstruction respectively. α is the scaling factor which brings the reconstruction score to same scale as the softmax score. In equation 6.3 the numerator represents probability assigned to label y_i and the denominator represents the sum of all probabilities. The relationship between the reconstruction and softmax score is inverse in nature. The λ value is chosen to be 0.5.

6.3. Results

6.3.1. Comparison with State of the Art

In this experiment we have implemented the methods [45] and [46] which are the superior anomaly detection methods during the writing of this thesis for MR datasets. We however train these networks on Our in house dataset and get the corresponding results for fair comparison. The dataset is split in two ways to get similarities between the calculated results and the clinical condition. The first case we call the method **slicewise scores** where there is a complete disjoint between the anomalous slices of image and healthy slices of image. The number of Healthy slices in this case of testing is 8120 and Anomalous cases is 3505.

In order to replicate the clinical condition we have a dataset where complete volumes of the image are fed in but broken down into slices before the pipeline and we dont have a previous knowledge of which slice has the anomaly hence the entire volume is marked as anomalous in this case. We have 4805 or healthy slices where there are no anomalies and 6820 mix of healthy and anomalous slices. We call this case as **clinical slicewise case**.

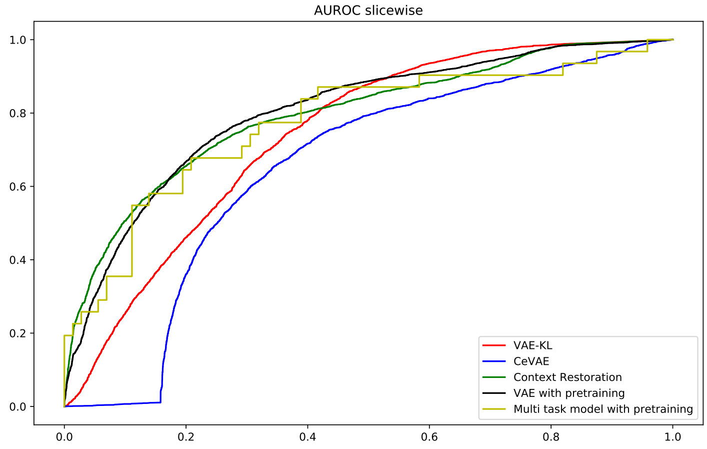
In the table below the AUROC scores AUPR scores have been mentioned. The methods are VAE-KL [45] , CeVAE [46], Anomaly detection by context restoration alone (Pretraining) 5.2.3 , Anomaly detection by VAE with pretraining 5.2, Anomaly detection with Multi Task learning with pretraining 5.3

	Slice wise score	Clinical Slice wise
VAE-KL [45]	0.652	0.668
CeVAE [46]	0.741	0.766
Context Restoration 5.2.3	0.795	0.636
VAE with Pretraining 5.2	0.803	0.673
Multi task model with Pretraining 5.3	0.783	0.822

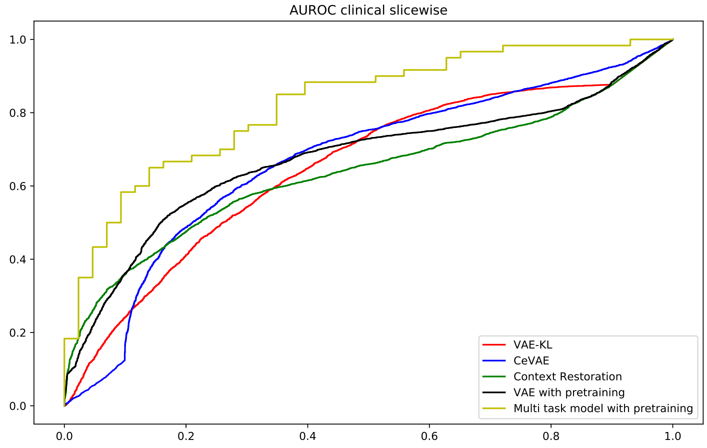
Table 6.1.: Table of AUROC scores comparison with state of the art anomaly detection methods

From the above results we can see that in terms of Slice wise scores the VAE with pretraining model performs the best. Another important thing to note is that the performance by just performing Context restoration has beaten the State of the art methods. This is a confirmation of our hypothesis that self supervision will have better impact over completely unsupervised learning.

In terms of AUPR we see that the multi task learning model with pretraining performs the best which confirms our hypothesis that combining the reconstruction scores and classification scores would lead to better anomaly detection of a sample. In ?? we can see again that AUPR values of context restoration has bettered the State of the art methods which is



(a) Slicewise ROC curves



(b) Clinical slicewise ROC curves

Figure 6.1.: Receiver Operator characteristic curves for all methods [45], [46], 5.2.3, 5.2,5.3

a positive response towards the hypothesis.

In figure 6.2 and 6.1 we plot the AUPR and AUROC values across all the models.

	Slice wise score	Clinical Slice wise
VAE-KL [45]	0.369	0.704
CeVAE [46]	0.471	0.640
Context Restoration 5.2.3	0.663	0.758
VAE with Pretraining 5.2	0.638	0.772
Multi task model with Pretraining 5.3	0.646	0.868

Table 6.2.: Table of AUPR scores comparison with state of the art anomaly detection methods

Further experiments were performed comparing the effect of the scaling factor in multi task model but most of them didnt yield good qualititative results and hence were not evaluated quantitatively.

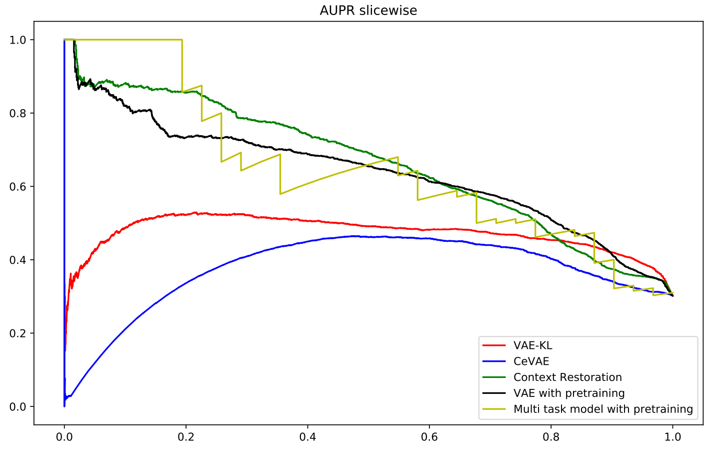
6.3.2. Segmentation

For localization of the anomalies we use just the Variational Autoencoder part in all models. In this experiment the images with anomalies are passed into the network and the network which is trained on only Healthy images fails to reconstruct the unhealthy part of the image leading to a blurry reconstruction after the decoder part. Taking a difference between the Output of the Variational Autoncoder and the Input gives the residual masks. But this residual cannot be used as a segmentation mask. For getting the segmentation mask we need to threshold the image and binarize the residual to get the mask. After getting the mask for each image we then proceed to calculate the Dice Similarity coefficient with respect to corresponding Ground truth slices.

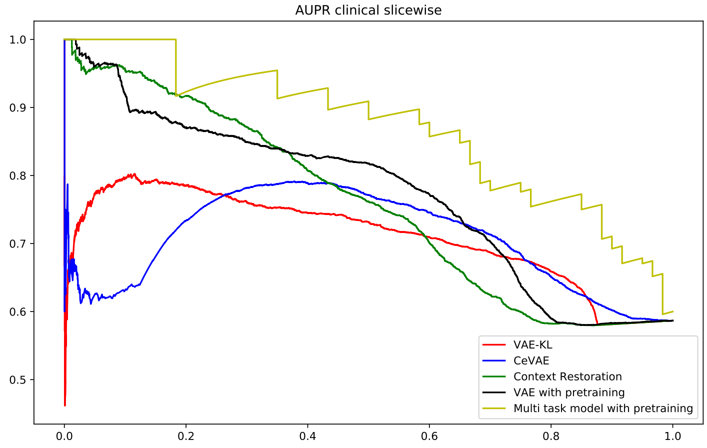
The dice scores for the dataset across all models have been computed and shown in table 6.3

Model	DSC ($\mu \pm \sigma$)
VAE-KL [45]	0.110 ± 0.0212
CeVAE [46]	0.112 ± 0.0210
Context Restoration 5.2.3	0.085 ± 0.0242
VAE with Pretraining 5.2	0.112 ± 0.0213
Multi task model with Pretraining 5.3	0.086 ± 0.0246

Table 6.3.: Table comparing the Dice Similarity coefficient for all the models



(a) Slicewise PR curves



(b) Clinical slicewise PR curves

Figure 6.2.: Precision Recall curves for all methods [45], [46], 5.2.3, 5.2,5.3

From the table 6.3 we can see that dice scores are very low. This can be attributed to lot of false positives in the edges of the brain which is due to generation from the Gaussian prior. The other reason to explain such a low dice score is the small size of anomalies. In most cases of pathology though there is many anomalies in the brain the focus has been just one particular pathology and the ground truth information lacks information about multiple anomalies in the sample and hence a huge change in the dice score.

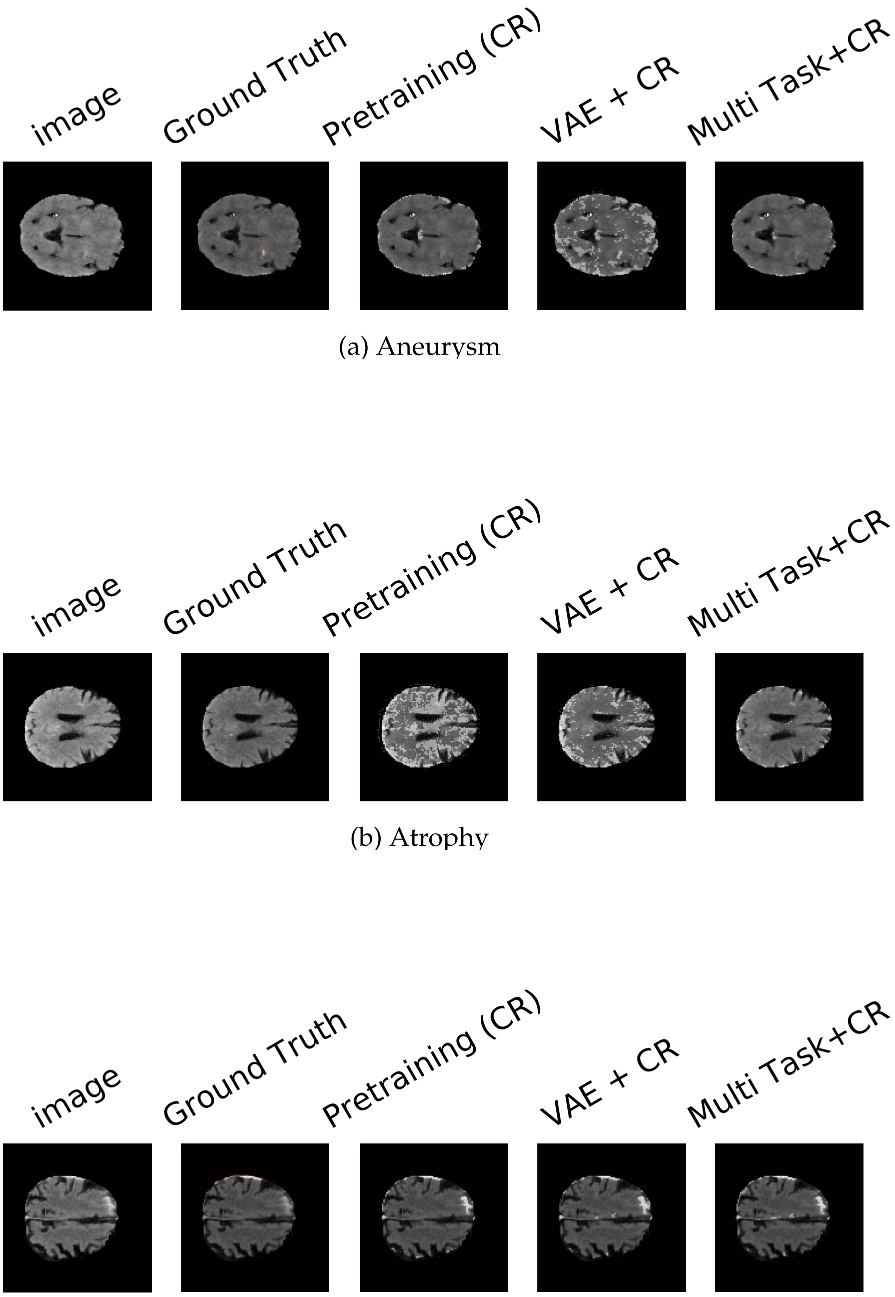
6.3.3. Pathology Wise Analysis

Since our data comprises of multiple pathological samples the generic anomaly detection evaluation based on segmentation and Area Under the Curve doesn't give use which pathology is being captured well and which pathology is not. For this experiment we consider only slice wise cases (disjoint set between healthy and pathological samples). The following table 6.4 gives us AUROC scores for different pathology. From the table we can see that Bleed, Intra Cranial Bleeding perform the best. The sample for cavernoma is pretty small and this is the reason why the Multi task learning is not able to differentiate between the normal and anomaly.

Model	Context Restoration	VAE with Pretraining	Multi task with Pretraining
Aneurysm	0.841	0.895	0.840
Atrophy	0.760	0.774	0.760
Bleed	0.951	0.944	0.956
Cavernoma	0.783	0.710	0.583
ICB	0.915	0.910	0.903
Ischemia	0.775	0.794	0.788
Tumor	0.848	0.843	0.828

Table 6.4.: Pathology wise AUROC scores for the 3 proposed models. ICB in the above table refers to Intra Cranial Bleeding

A few samples with a comparison to ground truth has been shown in the following figures.

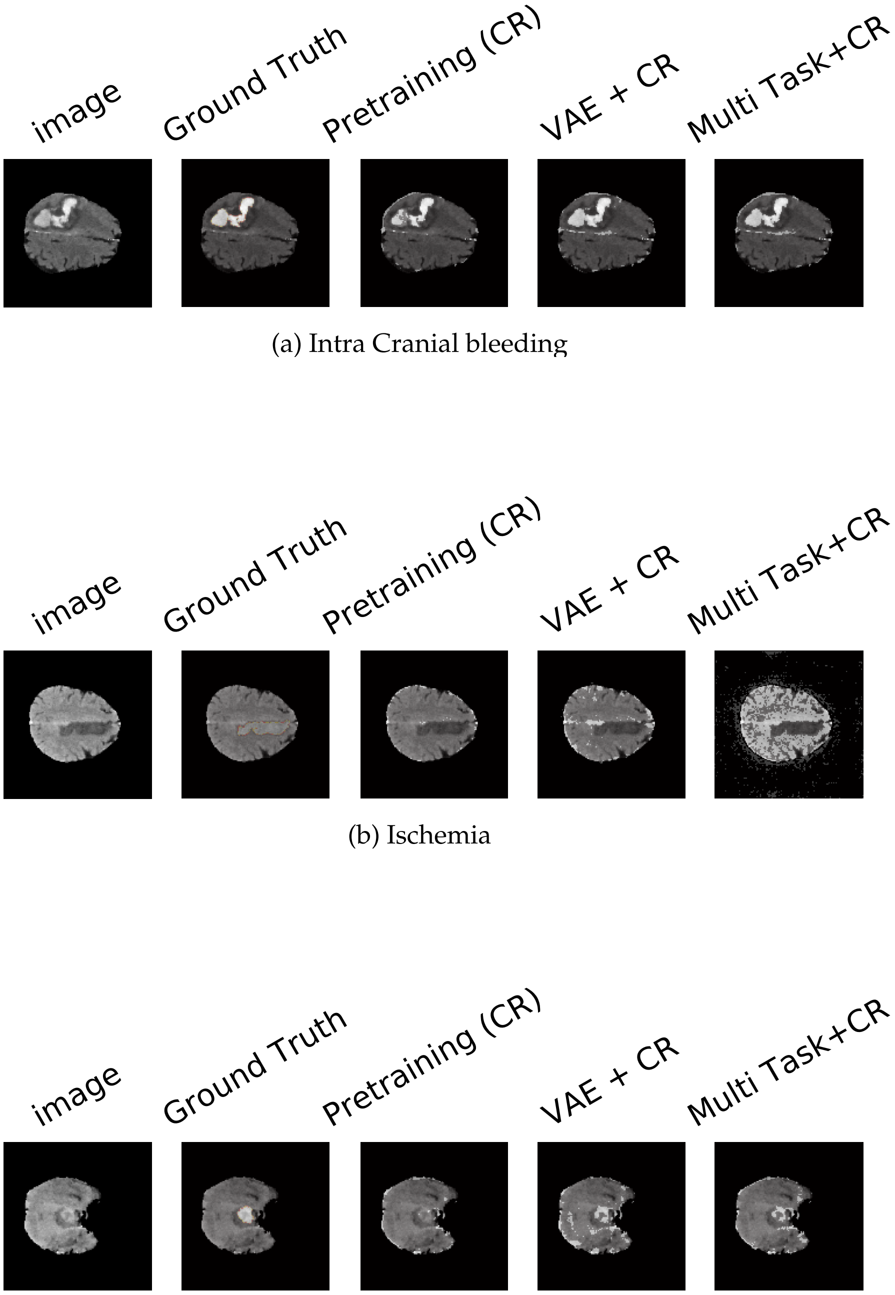


(a) Aneurysm

(b) Atrophy

(c) Bleed

Figure 6.3.: Segmentation samples for all models Pretraining(CR) is Context Restoration, VAE +CR is VAE with pretraining and Multi task+ CR is Multi task model with context restoration



(a) Intra Cranial bleeding

(b) Ischemia

(c) Tumor

Figure 6.4.: Segmentation samples for all models Pretraining(CR) is Context Restoration, VAE +CR is VAE with pretraining and Multi task+ CR is Multi task model with context restoration

7. Discussion and Outlook

7.1. Discussion

In this thesis, a self supervised method for detection of Anomalies in CT brain scans has been presented which uses novel ideas. Two models have been proposed with the most notable model being the multi task model with pretraining which uses context restoration in pretraining step to learn the features of healthy brain images and then adds a Classification head on top of the model and then trains the network in a multi task fashion. We then proceed to combine the scores from the classification and reconstruction to propose a novel anomaly score calculation.

Previous work in field of anomaly detection focused only on reconstruction based segmentation approaches in this approach we propose a novel contribution of combining it with a classifier that helps the sample being classified as anomalous or not based on self supervised out of distribution detection.

Experiments on In-house CT brain scans dataset has shown that the approaches achieve state of the art anomaly detection. It would be interesting to see how the approach performs on a Public dataset such as CQ-500 and Physionet where it would be validated further on its performance of anomaly detection.

The main advantage of the model is that if the anomalies are small enough not to be captured by the Reconstruction based methods the combination of self supervised classification and reconstruction residual scores helps in identifying the sample as anomalous. One more advantage over other anomaly detection methods is that this method is trained on all parts of the brain unlike the slice selection process that is done for training the other anomaly detectors. This method now pertaining to CT scans can also be applied for MRI scans with suitable preprocessing of the images and then changing the input of network to a Three channeled image corresponding to each of the modaililities.

7.2. Future Work

Though the model does well there are some limitation on the network as of now. The model is performing poorly on Ischemia and Cavernoma which are hypodense anomalies and the reconstruction based anomalies and the classification method fail to detect the anomalies individually but the multi-task model is very unsure of the samples from a case to case basis.

There are multiple ideas that are possible for future work which would make this work quite strong in anomaly detection

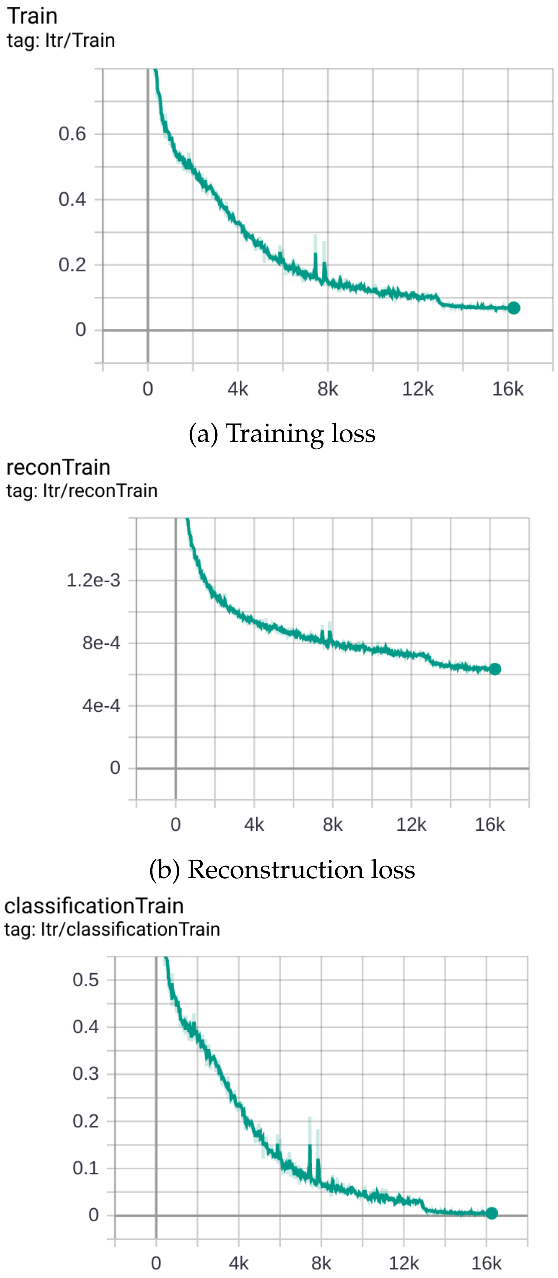
- Use of Adversarial loss function [32] to improve reconstruction. The effect of this is that the false positives currently created in the reconstruction boundaries of brain images will be removed and better segmentations can be obtained
- Use of concepts from Normalising flows [36] and invertible neural networks [4] to learn the complex latent space. Currently the prior assumes a Unit gaussian distribution with zero mean on the latent space. This can be improved with better probabilistic learning of the latent space. Another added advantage is that this would be improve the classification of the model.
- Use of Representation learning techniques to disentangle the features at the latent space to improve classification and enable better sampling for reconstruction.
- Improve the uncertainty of network by moving on to using techniques like Monte carlo dropout [14]. This shifts the model to complete bayesian perspective which is better in predicting uncertainities of the model.

Appendix

A. Appendix

A.1. Choice of α in Multi task learning for loss function in 5.3

The choice of α comes down to scale of the loss values of the reconstruction and the classification loss. Both the losses has to be of comparable scale and then the combination of these losses. Figure A.1 shows the scales of two losses and how the choice of α was done to bring them to similar scale. Since the scale difference is by order of magnitude 10^2 the α was also chosen to be the same.



(a) Training loss

(b) Reconstruction loss

(c) Classification loss

Figure A.1.: Loss values while training Multi task learning model. The x-axis shows the number of iterations of training and y-axis shows the loss values. These figures are obtained during the training period from logging of the loss values

Bibliography

[1] Hans E Atlason, Askell Love, Sigurdur Sigurdsson, Vilmundur Gudnason, and Lotta M Ellingsen. Unsupervised brain lesion segmentation from mri using a convolutional autoencoder. In *Medical Imaging 2019: Image Processing*, volume 10949, page 109491H. International Society for Optics and Photonics, 2019.

[2] Brian B Avants, Nick Tustison, and Gang Song. Advanced normalization tools (ants). *Insight j*, 2(365):1–35, 2009.

[3] 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*. Springer, 2018.

[4] Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger B Grosse, and Jörn-Henrik Jacobsen. On the invertibility of invertible neural networks. 2019.

[5] Matthew Brett, Christopher J. Markiewicz, Michael Hanke, Marc-Alexandre Côté, Ben Cipollini, Paul McCarthy, Christopher P. Cheng, Yaroslav O. Halchenko, Michiel Cottaar, Satrajit Ghosh, Eric Larson, Demian Wassermann, Stephan Gerhard, Gregory R. Lee, Erik Kastman, Ariel Rokem, Cindee Madison, Félix C. Morency, Brendan Moloney, Mathias Goncalves, Christopher Burns, Jarrod Millman, Alexandre Gramfort, Jaakko Leppäkangas, Ross Markello, Jasper J.F. van den Bosch, Robert D. Vincent, Krish Subramaniam, Pradeep Reddy Raamana, B. Nolan Nichols, Eric M. Baker, Soichi Hayashi, Basile Pinsard, Christian Haselgrove, Mark Hymers, Serge Koudoro, Nikolaas N. Oosterhof, Bago Amirbekian, Ian Nimmo-Smith, Ly Nguyen, Samir Reddigari, Samuel St-Jean, Eleftherios Garyfallidis, Gael Varoquaux, Jakub Kaczmarzyk, Jon Haitz Legarreta, Kevin S. Hahn, Oliver P. Hinds, Bennet Fauber, Egor Panfilov, Henry Braun, Jean-Baptiste Poline, Jon Stutters, Kesshi Jordan, Matthew Cieslak, Miguel Estevan Moreno, Valentin Haenel, Yannick Schwartz, Bertrand Thirion, Dimitri Papadopoulos Orfanos, Fernando Pérez-García, Igor Solovey, Ivan Gonzalez, Jath Palasubramaniam, Justin Lecher, Katrin Leinweber, Konstantinos Raktivan, Peter Fischer, Philippe Gervais, Syam Gadde, Thomas Ballinger, Thomas Roos, Venkateswara Reddy Reddam, and freec84. nipy/nibabel: 2.5.2, April 2020.

[6] Joshua Broder and Robert Preston. Chapter 1 - imaging the head and brain. In Joshua Broder, editor, *Diagnostic Imaging for the Emergency Physician*, pages 1 – 45. W.B. Saunders, Saint Louis, 2011.

[7] Liang Chen, Paul Bentley, Kensaku Mori, Kazunari Misawa, Michitaka Fujiwara, and Daniel Rueckert. Self-supervised learning for medical image analysis using image context restoration. *Medical image analysis*, 58:101539, 2019.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[9] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12154–12163, 2019.

[10] Xiaoran Chen, Suhang You, Kerem Can Tezcan, and Ender Konukoglu. Unsupervised lesion detection via image restoration with a normative prior. *Medical Image Analysis*, page 101713, 2020.

[11] Paul Concus, Gene H Golub, and Dianne P O'Leary. A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations. In *Sparse matrix computations*, pages 309–332. Elsevier, 1976.

[12] Alan R Curtis, Michael JD Powell, and John K Reid. On the estimation of sparse jacobian matrices. *J. Inst. Math. Appl*, 13(1):117–120, 1974.

[13] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[14] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Insights and applications. In *Deep Learning Workshop, ICML*, volume 1, page 2, 2015.

[15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[16] Gene Golub. Numerical methods for solving linear least squares problems. *Numerische Mathematik*, 7(3):206–216, 1965.

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[18] Stephan Guennemann. Lecture notes in machine learning, November 2018.

[19] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

[20] Mohamad H Hassoun et al. *Fundamentals of artificial neural networks*. MIT press, 1995.

[21] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[22] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[23] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 15637–15648. Curran Associates, Inc., 2019.

[24] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[25] Andrej Karpathy. University Lecture, 2017.

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. international conference on learning representations (2015), 2015.

[27] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[29] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc., 2017.

[30] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.

[31] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[32] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[33] Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

[34] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[35] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[36] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[37] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[38] Daisuke Sato, Shouhei Hanaoka, Yukihiro Nomura, Tomomi Takenaga, Soichiro Miki, Takeharu Yoshikawa, Naoto Hayashi, and Osamu Abe. A primitive study on unsupervised anomaly detection with an autoencoder in emergency head ct volumes. In *Medical Imaging 2018: Computer-Aided Diagnosis*, volume 10575, page 105751P. International Society for Optics and Photonics, 2018.

[39] 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.

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

[41] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[42] Alberto Taboada-Crispi, Hichem Sahli, Denis Hernandez-Pacheco, and Alexander Falcon-Ruiz. Anomaly detection in medical image analysis. In *Handbook of research on advanced techniques in diagnostic imaging and biomedical applications*, pages 426–446. IGI Global, 2009.

[43] S. van der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30, 2011.

[44] Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, et al. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

[45] 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.

[46] David Zimmerer, Jens Petersen, Fabian Isensee, and Klaus Maier-Hein. Context-encoding variational autoencoder for unsupervised anomaly detection. In *International Conference on Medical Imaging with Deep Learning – Extended Abstract Track*, London, United Kingdom, 08–10 Jul 2019.