Anomaly Localisation through Reconstruction-Loss Prediction

Student Name: C. Mavropoulos

Supervisor Name: C.G. Willcocks

Submitted as part of the degree of MEng Computer Science to the

Board of Examiners in the Department of Computer Sciences, Durham University

06/05/21

***Abstract* — Background** Recently, adversarial inpainting has been used for localisation and segmentation of abnormal regions in MRI imagery. The process of localisation is achieved through the generation of a heatmap through a sliding window inpainting which locates regions with a high reconstruction loss. These regions are later segmented to find boundaries between normal tissue and potential tumours. **Aims** As the process of reconstructing an inpainted image is expensive and the localisation module has an image reconstructed multiple times, there is a need for a faster approach for localisation. Our aim is to improve upon this architecture and produce a novel method to identify the region with the highest reconstruction loss and therefore the one which most probably contains an anomaly in a faster time. **Methods** To achieve this, we created a new Reconstruction Prediction Network that is tasked to predict the reconstruction loss that an image will have when passed through the original network that removes the inpainting. It does so by just looking at the inpainted image and outputs the predicted loss to be used in the heatmap generation. **Results** We have shown that the network is detect areas with high reconstruction loss and do so in 4 times the speed while using 28 less memory. The average difference between the predicted and original loss is approximately 3×10^{-5}. **Conclusion** While the proposed model is not as accurate as the original reconstructions, it is still able to detect areas which the original network deems anomalous in a much faster time.

***Keywords* —** Deep Learning, Inpainting, Anomaly Detection, Localisation

I INTRODUCTION

The medical sector has been an important focus of the deep learning research and application in the last years. Deep learning has been applied in many areas and relating to Magnetic Resonance Imaging (MRI) it has been employed in classification, improving efficiency, analysis and anomaly detection among others (Lundervold & Lundervold 2019). Anomaly detection is the task of identifying out-of-distribution samples. This task is challenging as there is a poor definition of what classifies an anomaly especially in the medical field as what can be constituted an anomaly in one person can be normal for another. Moreover, imaging techniques are not perfect and they often capture noise or introduce artifacts which make the task all the more difficult in a domain that calls for high accuracy. Thus, deep learning is often used to assist in this task and many different methods have been published tackling this problem (Fernando et al. 2020).

A Anomaly Detection

A.1 Supervised Learning

The most common approach to medical anomaly detection is supervised learning where the data is categorised or otherwise labelled as normal or abnormal. Labelling can also take the form of generated masks (Figure 1) that overlap an abnormal region in an image. Due to medical anomaly detection having a high degree of sensitivity and additional challenges as stated above, using datasets that have been generated with the assistance of experts in the field makes the task easier.

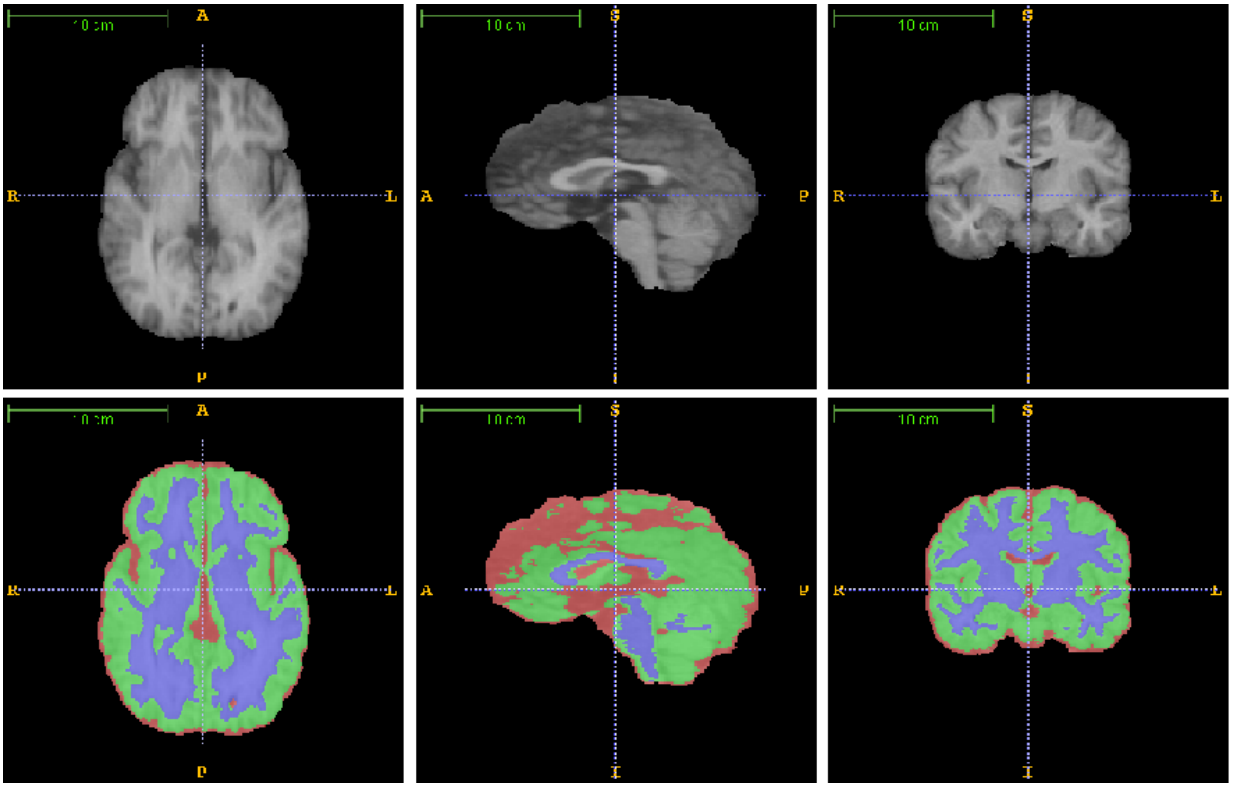

Figure 1: Segmented Brain MRI from (Jesacher et al. 2014)

However, there is a surplus of unlabelled datasets that can be used for anomaly detection where unsupervised learning must be employed (Lundervold & Lundervold 2019).

A.2 Unsupervised Learning

One method that has been used extensively is AutoEncoders that have been employed in the medical field since their introduction. These appear as feature extraction preprocessing techniques for other architectures as their ability to reconstruct high dimensional data is limited. Some of their improvements, such as, Variational AutoEncoders have been used for both 2D and 3D anomaly detection tasks.

Another prevalent method in unsupervised anomaly detection is the use of GANs which is primarily made of two components; the generator G and the discriminator D. The generator attempts to produce realistic samples while the discriminator attempts to classify between real samples taken from the dataset and fake samples generated by G. The result of this interaction is an adversarial min-max game between the two (Saase et al. 2020).
Some of the methods used for unsupervised anomaly detection rely on classification based on information extracted from the latent space representation of the data. These applications often include autoencoders used to map the input to the latent space for representation learning. These methods have applications in time-series data, X-ray and MRI images. However, recently, generative networks have been used for the purposes of anomaly detection. Specifically, Generative Adversarial Networks (GANs) and U-Nets have been used to generate segmentation maps or generate normal samples. However, inpainting is also used to detect anomalies based on the generated samples from such networks.

B Localisation

During the process of anomaly detection, some algorithms apply a localisation preprocessing step to segmentation. Segmentation of anomalous regions, such as a tumour, in medical imaging can be an important step in clinical routine as doctors refer to this segmentation to understand more about the patient. By using localisation, we highlight a region in a given image or sample that is most probable to contain or be an anomaly. Once such a region is identified, different methods can be used to either segment or more precisely identify the abnormal area in a given sample.
Localisation can be a crucial step in the process of anomaly detection as it allows us to focus on a specific region of a given sample instead of having to operate on its entirety. This enables us to use more precise and computationally expensive methods which will maintain high accuracy and precision. Specifically, a popular segmentation algorithm used in (Nguyen et al. 2020) is the Felzenszwalb's efficient graph-based which claims to have a running time nearly linearly related to the number of pixels in an image ($\mathcal{O}(n \log n)$, where n is the number of pixels) (Felzenszwalb & Huttenlocher 2004). Thus, a 32×32 region will be segmented much faster when compared to the full 256×256 image.
Recently, a paper published on Unsupervised Region-based Anomaly Detection uses such a localisation system before segmentation to focus on a specific anomalous region. Our work in this project revolved around improving the localisation part of that system as it required reconstructing hundreds of images to generate a heatmap of potentially anomalous regions (Nguyen et al. 2020). It achieved this through the use of adversarial image inpainting which was repeatedly applied to query MRI slices. Our focus is on producing the heatmap of such regions in a more efficient way and as such we keep the inpainting network the same and ignore the segmentation process. In the end, we'll compare our heatmap generated with our proposed model with a heatmap generated through the use of the original localisation system.

C Dataset

For training and testing we used the Neurofeedback Skull-stripped (NFBS) repository as its the one used in the original paper. It is a dataset consisting of 125 T1-weighted (T1w) anatomical

MRI scans which are manually skull-stripped. The subjects are between the ages of 21 and 45 with various clinical and sub-clinical psychiatric symptoms (Puccio et al. 2016). Anatomical MRI is used to study the shape, volume, integrity and other physical changes to the structure of the brain. The slices being T1w means that the Repetition Time (TR) between radio frequency pulses and the Time to Echo (TE), which is the time between sending and receiving the signal, is less than those of a T2-weighted MRI image (Radue et al. 2016). In total, for each subject there are 192 slices that depict different depths of a particular brain. Having the slices be skull-stripped means that only brain-matter remains visible after the application of a mask, while its surrounding area is removed. This process was done both through computer programs and experts manually creating the masks for certain slices. In this project we use the skull-stripped versions of these images as this allows us to focus more closely on the brain and learning its features.

II RELATED WORK

A Inpainting

Image inpainting is the task of reconstructing regions of an image. It is often used for problems such as object removal, image restoration and manipulation among others. In most cases, the input image is inpainted most often with a fixed-size white box which is then subsequently removed using a generative network to a varying degree of success. There are other ways to inpaint an image such as a Gaussian distribution, a series of smaller boxes or random pixel deletion.

For anomaly detection, the network learns how to reconstruct the region behind the white box and would learn so from normal data. Thus, when in testing we cover the region we are interested in, the network will give a normal reconstruction. If this reconstruction is too different based on a threshold from the original image then this would count as an abnormality. Most commonly, U-nets are used as the networks of choice for this task as they are able to retain more global information through copying feature maps.

However, some problems arise with this methodology. To start with, the region chosen to be inpainted is chosen at random though this might not be the best case. In the MRI example 2 , the black region retains no information and inpainting and reconstructing a white box in that region will not help the network model the normal distribution of the data. It may be obvious in our example that we are mostly interested in the regions where brain matter is located instead of the surrounding black region however, in other inpainting tasks this choice may not be so obvious.

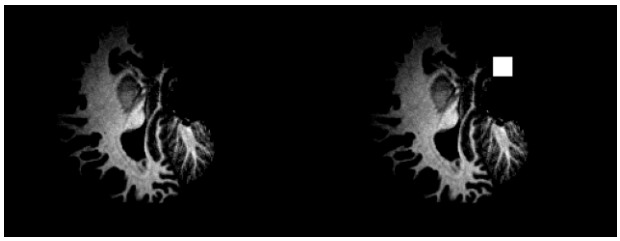

Figure 2: Inpainted Location

In the above example, the random mask was generated on the black region which was easy

for the network to remove by just turning the white pixels black. This is a two-fold problem as on one hand the network affected the batch loss greatly as the reconstruction is near perfect but on the other hand, the network learns to remove the white pixels in turn for black pixels. This is evident on some later reconstructions:

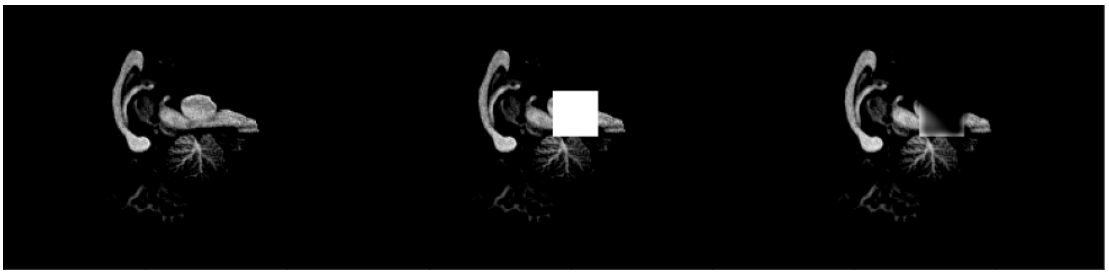

Figure 3: Reconstruction Removes Features

These two challenges make training take longer to account for more white box positions that fall in the target area but even then the white box may fall in less than ideal areas. These problems would need to be accounted for in our own solution.

B Image to Image Translation

In computer vision tasks, many problems rely on "translating" an image from one domain to another. Examples of such tasks are synthesising photos from label maps, colourising images among others. This problem has been extensively studied since the release of the $pix2pix$ network architecture and the subsequent paper which generalised image translation to be applied in different domains and tasks (Isola et al. 2018). The architecture uses a special GAN named U-Net, which utilises skip-connections to maintain detail in a deep network through propagating information in mirror encoder-decoder layers. Skip connections skip some layers and concatenate the input of a given layer with the output of a previous one. This enabled the network to produce realistic looking images for multiple pairs of input-target domains. However, a drawback of this architecture is that the task needs to be supervised which means there need to be image pairs, one for the input and one for the target domain. As stated previously, such datasets are expensive and time-consuming to create, especially in the medical sector where only highly trained individuals are able to accurately generate them. Thus, we reconstruct the inpainted image by training the network to remove the white box. Then, we are able to compare with the original image and calculate how different they are. We are then able to train our network based on that difference to get better at reconstructing these images.

C GANs

C.1 U-Net

U-Net was first introduced in 2015 as a convolutional network for biomedical image segmentation. The purpose of the segmentation was to recognise distinct neuronal structures in cells and output a mask that . The output was a segmentation mask which had the dimensions of the original image. These results were most importantly achieved with the use of a very small dataset by

(Ronneberger et al. 2015).
Many machine learning techniques on unsupervised anomaly detection work by learning the probability distribution of normal data samples during training. Thus, when an anomaly is passed, it would lie outside the probability distribution and the network would be able to detect it. A common problem with such approach is that these methods cannot differentiate between an outlier or noise and an abnormality. Especially in the field of medicine, as stated before outliers are common as individual differences between healthy samples can be many (Han et al. 2020).

D Adversarial Image Inpainting

Image inpainting is the process of adding a pattern on top of an image. This pattern is most commonly a white box however, some methods use probabilistic pixel deletion or a random pattern. Inpainting was used for visual feature learning in an unsupervised setting. The network was able to reproduce the image underneath the inpainting (Pathak et al. 2016). Thus, our input domain is the inpainted image and our target domain is the original one, before the addition of the white box pattern. This means that we now have a pair of images between the two domains and we can work in a somewhat supervised setting by utilising the reconstruction loss between the generated and original image. Specifically, the reconstruction loss can either be the mean average error (MAE or L1) or the mean squared error (MSE or L2):

$$Loss_{L1} = \mathbb{E}[|x - G(\bar{x})|] \tag{1}$$

Where x is the original image and $\bar{x}$ is the inpainted image. G is the network that takes $\bar{x}$ and attempts to generate x.

$$Loss_{L2} = \mathbb{E}[(x - G(\bar{x}))^2] \tag{2}$$

The gradient of L2 is high for large loss values and decreases as loss approaches 0, it has the potential to find solutions faster than L1, however, L1 has been shown to produce less blurry results (Isola et al. 2018).
A more recent method for unsupervised anomaly detection has the use of inpainting and has been successfully applied in domains which are consistent and predictable (Nguyen et al. 2020). In MRI applications, this paper has a three-step process to detect anomalies in samples 4. Firstly, adversarially train a Generator G to reconstruct missing healthy brain regions which have been randomly inpainted. The architecture also uses a discriminator D to detect if the final result looks real or fake. After the model has converged and the reconstructions are of satisfying quality, we move onto the localisation and segmentation steps. Given a query slice of the MRI, we inpaint it over a masked sliding window. Starting from the top left and moving 4 pixels each time we inpaint that region of the image and pass it to G. Then, G is able to reconstruct that image. For each reconstruction we calculate the reconstruction loss by comparing just the original and reconstructed region instead of the entire image. This is done as it can provide more accurate results for the latter heatmap. The heatmap can be constructed using the reconstruction losses which indicate areas of low and high reconstruction loss gained from G. As G has only been trained on normal data, its reconstructions will follow that distribution. Thus, when a tumor is inpainted for example, G will reconstruct it assuming there is normal brain tissue there. Thus, the difference between the normal brain tissue and the tumour will be higher than the other normal areas and thus the reconstruction loss will be higher there. Based on this heatmap, the areas with a high loss are more probable to contain an anomaly. Furthermore, segmentation algorithm is

applied to these regions which is able to exactly outline and depict the tumour.

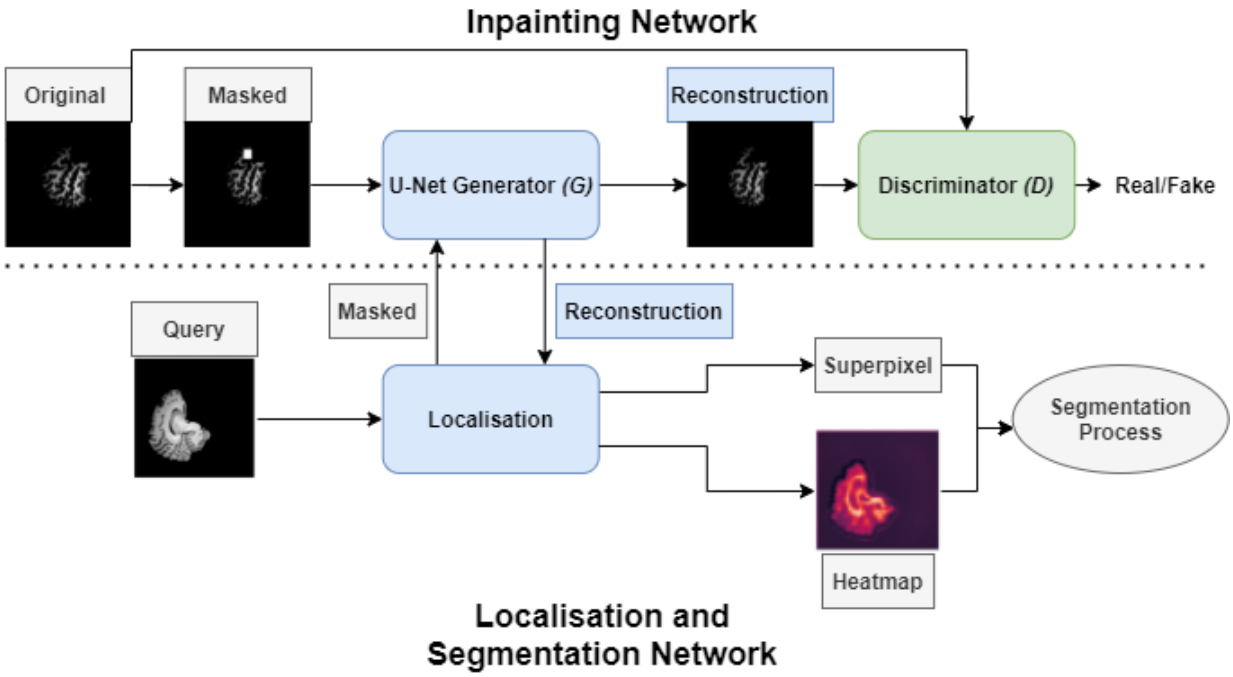

Figure 4: Original Architecture

With this method, the network needs to generate a reconstruction for all possible inpaintings which can is computationally expensive, especially for samples with a large resolution. U-Net is a large network with multiple parameters so that it can generate realistic reconstructions so when multiple images pass through the network they take time.

III SOLUTION

The process of iteratively reconstructing the image for each inpainting window position is expensive as it requires for the network to process or encode the image to some latent space before reconstructing. Moreover, after we calculate the loss of that region we have no more use of the image and thus we only reconstruct it for that loss. Our method aims to achieve a similar heatmap using a different method that could be more efficient both in terms of time and space.

A Data Pre-processing

Working with all 192 of the slices is computationally expensive as it essentially has the format of a video. Thus, the slices were split into individual images where the images holding no information were removed. These images were completely black and found at the very start and very end of the MRI images, were no brain matter was detected. They were also converted to Hierarchical Data Format (HDF or H5) for faster input and output. Finally, during development the images were resized to 128×128 while for the final implementation they were used in their

original 256×256 dimensions.
Normalisation was also applied to each image, as difference in depth and therefore illumination in parts of the brain caused significantly diminished reconstruction performance.

B Proposed Model

Our approach to localise the area where segmentation should be applied, follows similar steps with the process from (Nguyen et al. 2020). However, to improve on this architecture, instead of reconstructing the image for each position of the sliding window, we train a new network R that predicts the reconstruction loss generated by our network. This is much more efficient in practice as for an image to propagate through U-Net takes time as it needs to be encoded and then decoded. This process also requires a large amount of storage for the parameters used by the network.
The inpainting network $Fig: 4$ U-Net that reconstructs the image (U) and the Discriminator (D) that determines if an image is real or reconstructed (fake) are trained without any changes either to the architecture or to the parameters. After they have converged, the Reconstruction Network R is trained based on the U's reconstruction loss for each image.
The structure of R can be seen in Figure 5. Its input is a batch of one channel images of size 256×256. It has a series of convolutions to learn the features while the MaxPool2d operations reduce the size of the image enough for the output to be a single number relating to the prediction loss. Finally, it uses the sigmoid activation function before the output as the prediction lie between 0 and 1.
Some changes that were made during training were to bound the inpainted box in the middle of the image. This was done as a common problem faced during this project was that the inpainted location was often in the dark area were the network would learn to replace white pixels with black and erase parts of the images instead of reconstruct them. Furthermore, the network was trained on more instances where it needed to reconstruct more complex areas with brain tissue which sped up training. Bounding the inpainting in the middle third of each image made the network converge much faster than with completely randomly assigning the mask.

C Reconstruction Network

The reconstruction network tasked to approximate the reconstruction loss of the U-Net (U) is trained after U and the Discriminator (D) have converged and finished training. The reconstruction network R has similar structure to the discriminator. During training, as input to the network is an inpainted image. By reducing its dimensions and increasing the channels the network attempts to predict what the reconstruction loss for that image will be and produces a single number as an output which depicts that. The same image is passed to the generator G which follows the original architecture of reconstructing the image and then comparing it to the original to get the reconstruction loss. After we get the reconstruction loss from both network we compare them using a simple absolute difference for a given batch and take the mean:

$$[H]L_{rec} = |ReconstructionLoss - PredictedLoss| \tag{3}$$

Thereafter, we update the weights based on the results of these comparisons until the model has converged.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 128, 128]             640
         LeakyReLU-2         [-1, 64, 128, 128]               0
            Conv2d-3          [-1, 128, 64, 64]          73,856
         MaxPool2d-4          [-1, 128, 32, 32]               0
         LeakyReLU-5          [-1, 128, 32, 32]               0
       BatchNorm2d-6          [-1, 128, 32, 32]             256
            Conv2d-7          [-1, 256, 16, 16]         295,168
         MaxPool2d-8            [-1, 256, 8, 8]               0
         LeakyReLU-9            [-1, 256, 8, 8]               0
           Conv2d-10            [-1, 512, 4, 4]       1,180,160
        MaxPool2d-11            [-1, 512, 2, 2]               0
        LeakyReLU-12            [-1, 512, 2, 2]               0
           Conv2d-13              [-1, 1, 1, 1]           8,193
          Sigmoid-14              [-1, 1, 1, 1]               0
================================================================
Total params: 1,558,273
Trainable params: 1,558,273
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.25
Forward/backward pass size (MB): 23.84
Params size (MB): 5.94
Estimated Total Size (MB): 30.04
----------------------------------------------------------------
```

Figure 5: Structure of network R

During testing then, we just make use of the prediction network R over a series of inpainted images to generate the heatmap as can be seen in Figure 7.

The training algorithm can be found below. After initialising all the network parameters and defining the network, we move on to training. We have a set number of epochs for which the algorithm runs for and when reached the training stops. Firstly, we get a batch of MRI-slices from our training dataset. After that, we use the Inpaint function to generate an inpainted image p at a random location. As stated before, this random location is bound to the middle third of the image to improve training and feature learning. Then, we reconstruct p and we now have image g from our generator. We compare using the $L1$ loss as seen in line 7 of the algorithm the reconstructed image g and the original image x. We then predict the reconstruction loss with our network R by passing the inpainted image batch p. When

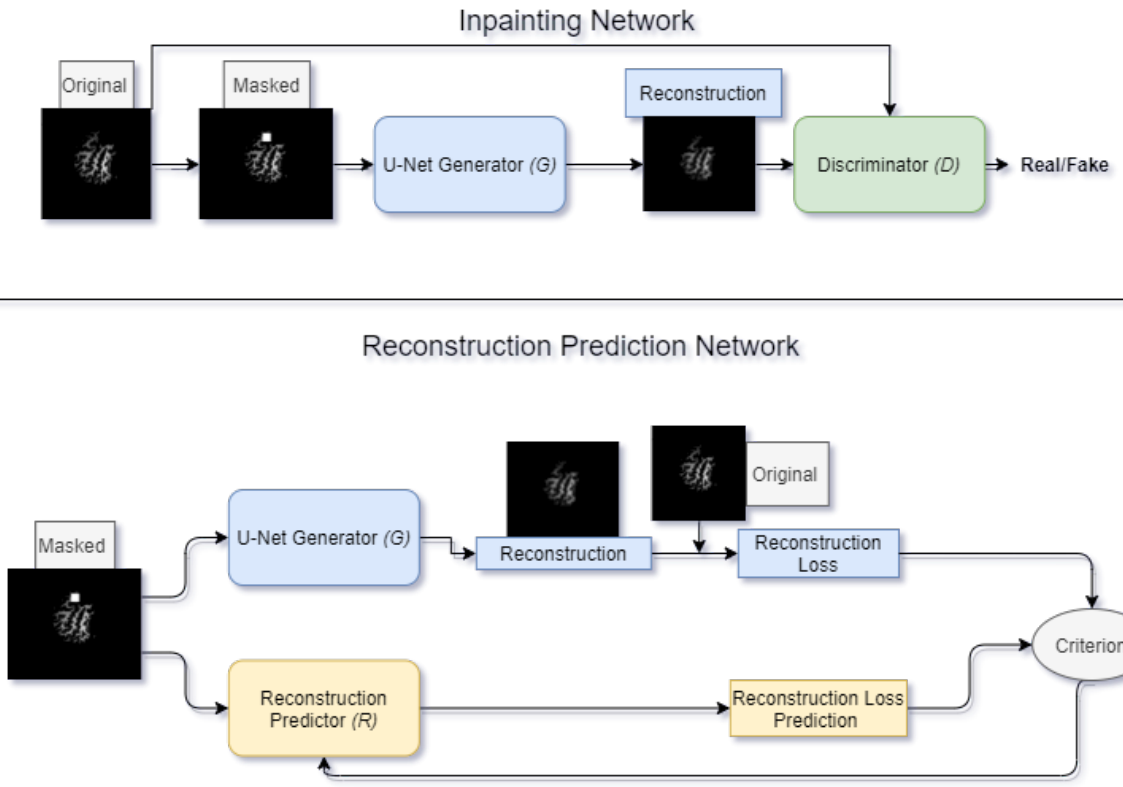

Figure 6: Proposed Model Training

we have the predicted loss, we compare it with the original loss (line 7) by taking the absolute difference between the two (line 9). We update our parameters based on that loss and continue until the number of max epochs has been reached or until the network converges.

Algorithm 1: Reconstruction-Loss Prediction Training

1 initialise window size γ
2 initialise learning rate a_R
3 **while** *current_epoch*<*max_epochs* **do**
4 x = random mini-batch of MRI slices
5 p = $Inpaint(x)$
6 g = $G(p)$
7 L_{rec} = $||g - x||$
8 $pred$ = $R(p)$
9 L_{pred} = $||pred - L_{rec}||$
10 $current\_epoch+ = 1$
11 **end**

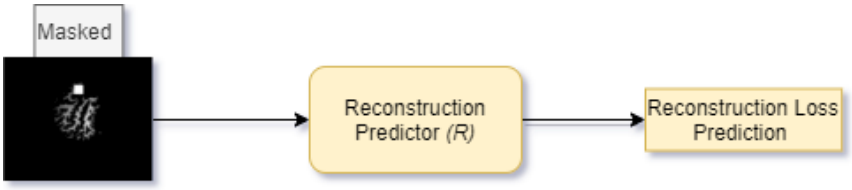

Figure 7: Proposed Model in Operation

IV RESULTS

All tests were run on the Google Colab platform which uses a Tesla V100 $16GB$ GPU and 13GB of RAM. Tests were run on CUDA version of 11.2. Input and output of images was done through google drive and the Wandb API which tracked loss and other statistics. The original U-Net architecture uses bilinear interpolation when upsampling however, this created some blurry results and generated some artifacts in the reconstructed images. Bicubic interpolation provided sharper images as it interpolates from a larger (4×4) neighbourhood area.
The window size for these tests was a 16×16 area as the original paper found that to be a good middle ground. The step size, which defined how far each inpaint window was from the previous was 4 while generating the heatmaps. This means that a 256×256 image will have a heatmap of 64×64 for a total of 4096 losses/pixels. However, accounting for the edges as we cannot inpaint multiple areas twice, the final dimensions of the heatmap were 60×60 for a total of 3600 reconstructions.
As with the original paper, the reconstruction loss was calculated just on the specific area that was reconstructed instead of the whole image. However, for our proposed model this cannot be achieved as its input is the entire image with the inpainting. This difference will lightly skew the results as discussed in the Evaluation.

Figure 8: Training Reconstruction Loss Difference

First, we take a look at the training loss. This loss is between the original reconstructed loss

and the predicted one. As we can see from the graph below, the model is able to quickly learn as it achieved good results in about 13 minutes of training. The results show the loss approaching zero after about 10 epochs while it is marginally improved over the next 30. The final number of epoch which the network trained for was 40 as very little improvement was seen at that stage. However, these results are on training data that it has seen before and thus these results can only be used to evaluate the training process.

To find out how the model performs with data it has not seen before, we generate batches of data from the unseen testing dataset and pass them both to the original model and the proposed one. The average difference calculated between the two was $3 * 10^{-5}$. This has a deviation of about 1% from the original loss.
While the above results look promising, qualitative results were the most insightful for this project. Below is a batch of images with the original image, the heatmap generated through the previous model and the heatmap generated with out proposed model.

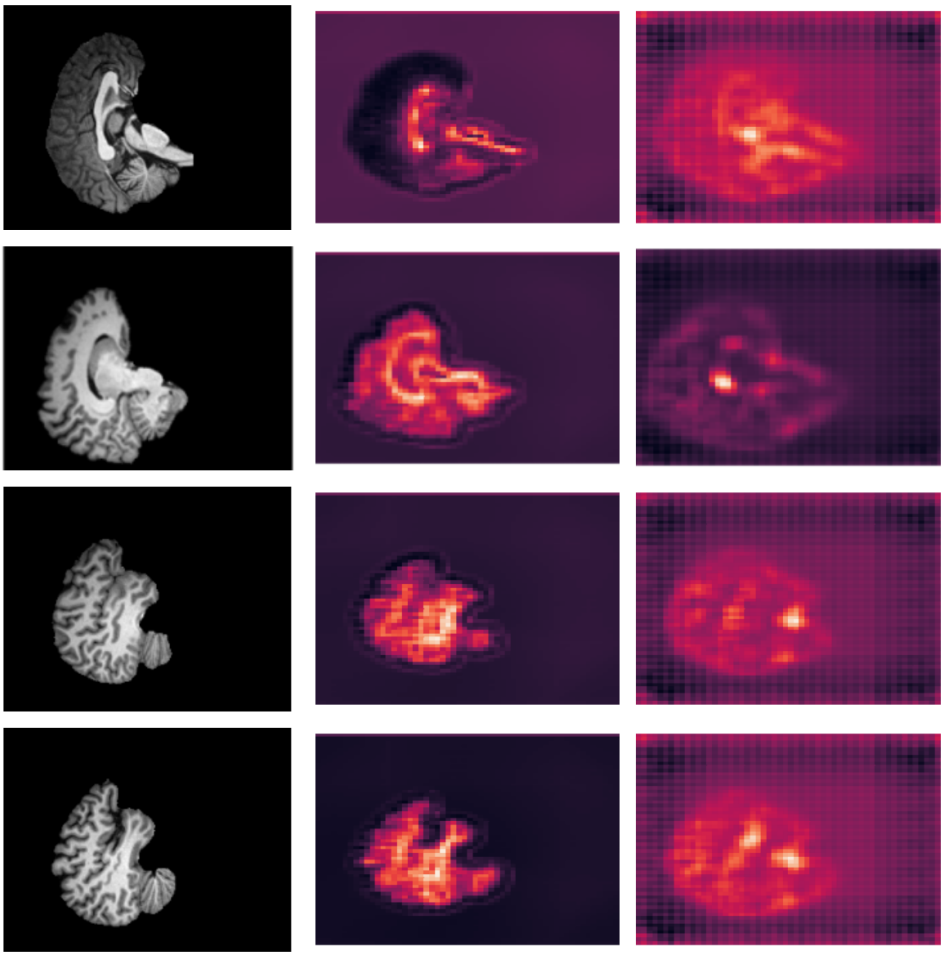

Figure 9: Original, Reconstructed Heatmap and Predicted Heatmap over a batch of 5 testing images

Finally, we need to look into the performance of the proposed model. For a single 256×256 image, the original localisation method took approximately 60 seconds to generate the heatmap. In our testing, our proposed model only took 14 seconds for the single image. Thus, our proposed model is 4 times faster at generating these heatmaps. Moreover, after training the network R with G reconstructions, during localisation we do not need to retain memory of G. This allows us to free up a lot of space as G had a total size $852.49MB$ while our model only uses $30.04MB$ as seen from its structure in Figure 5. This means that our localisation system is 28 times smaller than the original network.

V EVALUATION

As we introduce a new network in the architecture to replace the localisation system, we are adding computational overhead during training. As seen from 8 however, the model is light enough and the task is simple enough for it to train very quickly while approximating the original loss. Thus, we argue that the one-time overhead is less significant than the overhead added by the original architecture in the localisation step where the time taken to generate the heatmap is significantly more.

Moreover, these speeds are the average of just one slice however, sometimes multiple adjacent MRI slices might need to go through the network an be segmented. As mentioned before, a full MRI has 192 slices so if we wanted to generate a heatmap for all of them the time taken by the original network would be around 3.2 hours while our model would generate the heatmap at approximately 44.8 minutes. In a real world situation where there are multiple patients and procedures waiting on such results, this speedup could be very important.

Looking at the qualitative results, the predicted heatmaps have been able to learn the general areas of high and low reconstruction loss however, there is a loss in accuracy as it cannot exactly estimate the original results. Despite that, it can be seen from the images in 9 that the high reconstruction losses exist in the same areas as the original one which means that depending on the type of segmentation that we would like to do, these heatmaps might be enough. This is because the need only direct the segmentation algorithm to the correct area. This is also important as we can see a trade-off between some accuracy and speed which may be needed in some sectors. More interestingly, this is a novel approach of using a deep learning model to approximate the outputs of another which may have more applications in different sectors.

Furthermore, during testing, increasing the window size provides poorer results. This is because as the prediction network only looks at the inpainted image, if the inpainted section is too large it loses the ability to predict the reconstruction loss based on the surrounding area and thus 16×16 seemed to be the best window size for this task.

Moreover, below is the output of the network when instead of comparing the reconstructed areas for the heatmap, we compare the entire images for the original model.

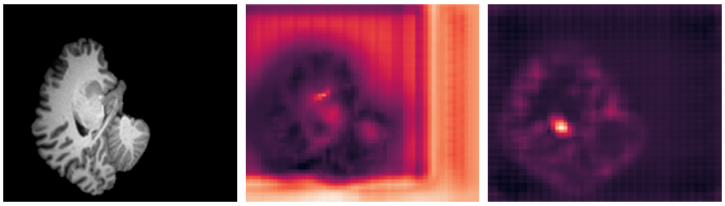

Figure 10: Original, Reconstructed Heatmap and Predicted Heatmap comparing the entire image

The reconstructed heatmap is now worse as we would have predicted, but interestingly, there are some high loss patterns surrounding the brain tissue. This is because, as stated before, we trained the networks by bounding the inpainted position to the center of the image. Thus, the network has not seen an inpainted location on the sides and thus has a high loss in these areas as it is struggling to reconstruct it. While this is expected, the predicted heatmap does not have this problem as it has learnt to compare the surrounding areas and thus it does not matter where the inpainted pixels are located in the dark background.
A pitfall of this method is that it is bound to the performance of the original reconstruction and it can only approximate the results generated initially by the original model. It will never be able to surpass these results although, as we do not need to keep the U-Net generator or use it during the localisation and segmentation process, we could make the generator G even deeper and bigger to increase the performance of R. This method would only add the initial overhead while increasing accuracy later on.

VI CONCLUSIONS

In conclusion, the adversarial inpainting method is able to recognise anomalous regions using the reconstruction loss of a generator G. After these regions have been identified, segmentation may be applied to differentiate between normal brain tissue and a potential tumour. This process of generating the heatmap needs to reconstruct the image a number of times to get the reconstruction loss for each area and then move on to the next image. We show that this process could take hours in a real world setting and for applications which may require faster methods we propose directly predicting the reconstruction loss using a lighter network.
This prediction has a diminished accuracy to the original method as it is attempting to approximate its results with less information. However, it is able to identify the high loss areas in a given image on which the segmentation would be applied. It can do so 4 times faster that the original method while using 28 times less memory and resources. This also means that after training it could be potentially run from less powerful machines which could be an incentive for deep learning methods to be adopted in the wider medical field.
These results show that having a deep learning model approximating the output of another could be a valid solution for a problem where we want to sacrifice some accuracy for speed. More experimentation is required to test if these results can be improved and to test the reliability of this method in the entire architecture, with the segmentation process.

References

Felzenszwalb, P. F. & Huttenlocher, D. (2004), 'Efficient graph-based image segmentation', *International Journal of Computer Vision* **59**, 167–181.

Fernando, T., Gammulle, H., Denman, S., Sridharan, S. & Fookes, C. (2020), 'Deep learning for medical anomaly detection – a survey'.

Han, C., Rundo, L., Murao, K., Noguchi, T., Shimahara, Y., Milacski, Z. Á., Koshino, S., Sala, E., Nakayama, H. & Satoh, S. (2020), 'MADGAN: unsupervised medical anomaly detection GAN using multiple adjacent brain MRI slice reconstruction', *CoRR* **abs/2007.13559**.
URL: *https://arxiv.org/abs/2007.13559*

Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. (2018), 'Image-to-image translation with conditional adversarial networks'.

Jesacher, B., Reimann, A., Meier, R., Wiest, R., Reyes, M. & Bauer, S. (2014), Brain tissue segmentation in mri by supervised classification and regularization.

Lundervold, A. S. & Lundervold, A. (2019), An overview of deep learning in medical imaging focusing on mri. Special Issue: Deep Learning in Medical Physics.
URL: *https://www.sciencedirect.com/science/article/pii/S0939388918301181*

Nguyen, B., Feldman, A., Bethapudi, S., Jennings, A. & Willcocks, C. G. (2020), 'Unsupervised region-based anomaly detection in brain mri with adversarial image inpainting'.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T. & Efros, A. A. (2016), 'Context encoders: Feature learning by inpainting'.

Puccio, B., Pooley, J. P., Pellman, J. S., Taverna, E. C. & Craddock, R. C. (2016), 'The preprocessed connectomes project repository of manually corrected skull-stripped T1-weighted anatomical MRI data', *GigaScience* **5**(1). s13742-016-0150-5.
URL: *https://doi.org/10.1186/s13742-016-0150-5*

Radue, E.-W., Weigel, M., Wiest, R. & Urbach, H. (2016), 'Introduction to magnetic resonance imaging for neurologists', *Continuum: Lifelong Learning in Neurology* **22**(5), 1379–1398.

Ronneberger, O., Fischer, P. & Brox, T. (2015), 'U-net: Convolutional networks for biomedical image segmentation'.

Saase, V., Wenz, H., Ganslandt, T., Groden, C. & Maros, M. E. (2020), 'Simple statistical methods for unsupervised brain anomaly detection on MRI are competitive to deep learning methods', *CoRR* **abs/2011.12735**.
URL: *https://arxiv.org/abs/2011.12735*