One Shot GAN for Medical Image Generation
(DH302 project report)

Team:
Akshat Kumar
Jash Kabra
Sandeepan Naskar
Niyati Mehta
Poojan Sojitra

Contents

1 Problem Statement

Training a Generative Adversarial Network (GAN) Model on single Brain Magnetic Resonance (MR) Images to generate similar images with the aim of Data Augmentation, henceforth enhancing datasets for Brain Tumour classification models and then further train a Classification Model for brain tumour using CNNs on our augmented dataset.

2 Introduction to GANs

The Generator: A generator network takes a random normal distribution (z), and outputs a generated sample that's close to the original distribution.

The Discriminator: A discriminator tries to evaluate the output generated by the generator with the original sample, and outputs a value between 0 and 1.

A random normal distribution is fed into the generator. The generator then outputs a random distribution, since it doesn't have a reference point. Meanwhile, an actual sample, or ground truth, is fed into the discriminator. The discriminator learns the distribution of the actual sample. When the generated sample from the generator is fed into the discriminator, it evaluates the distribution.

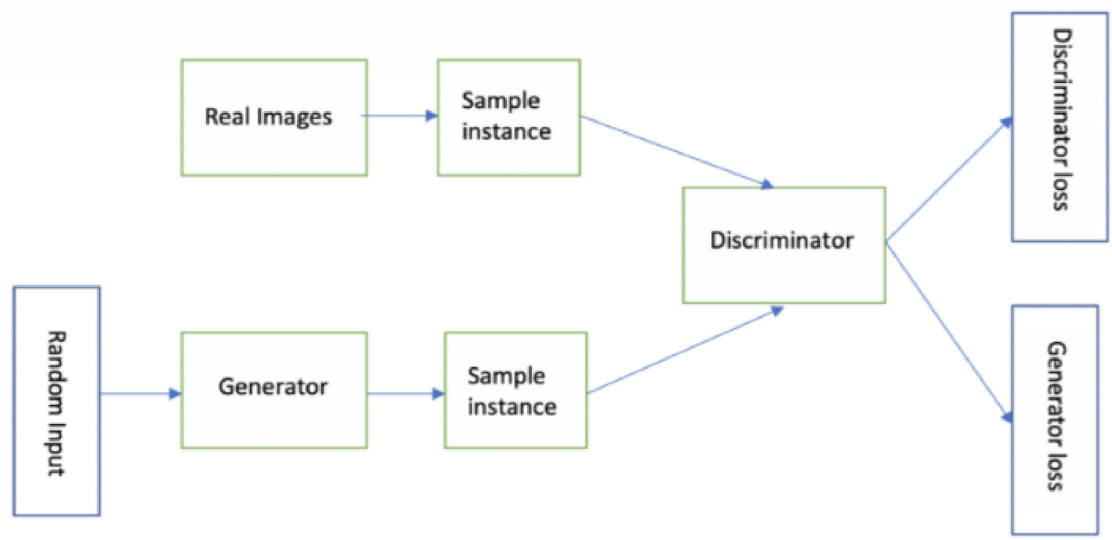

Figure 1: Architecture of GAN

If the distribution of the generated sample is close to the original sample, then the discriminator outputs a value close to '1' = real. If both the distribution doesn't match or they aren't even close to each other, then the discriminator outputs a value close to '0' = fake.

3 Papers implemented

3.1 The BrainGAN Paper

The BrainGAN is a framework for generating and classifying brain MRI images using GAN architectures and deep learning models. It aims to increase the limited brain MRI dataset using GANs to improve the accuracy of tumour classification models The two approaches used to implement it are as follows:

1. Using Vanilla GAN and DCGAN to generate additional brain MRI images to expand the dataset.

2. Automatically validating the generated images using CNN, MobileNetV2, and ResNet152V2 models. The models are trained on the generated images and tested on real brain MRI images to assess the quality of the generated images.

3.2 The OneShot GAN Paper

One Shot GAN: A GAN which can generate new images significantly different from the original image preserving the context. The modified Discriminator includes content for aggregating spatial information, and layout for aggregating information from different channels(global scene layout).

3.2.1 Paper novelties

1. A generative adversarial network that can learn to generate samples from a single image or video.

2. Uses a two-branch discriminator to judge the content and layout of images separately, which helps prevent overfitting and provides better guidance to the generator.

3. Diversity regularization is used to encourage the generator to produce diverse outputs.

4. Achieves higher quality and diversity compared to previous single-image GAN models, and can successfully learn from both single images and videos.

3.2.2 Architecture

We have used some blocks of De-convolution layers in the Generator. In the Discriminator, we used different blocks of Convolutional layers for content and layout branches. We have also used the Adam Optimizer, with a batch size of 4 and a learning rate of 0.0002.

3.2.3 Loss function

Diversity Regularization: the generator should produce images that are different from each other and substantially different from the original sample.

$$\mathcal{L}_{DR}(G) = E_{z_1,z_2}[\frac{1}{L}\sum_{l=1}^{L}||G^l(z_1) - G^l(z_2)||] \tag{1}$$

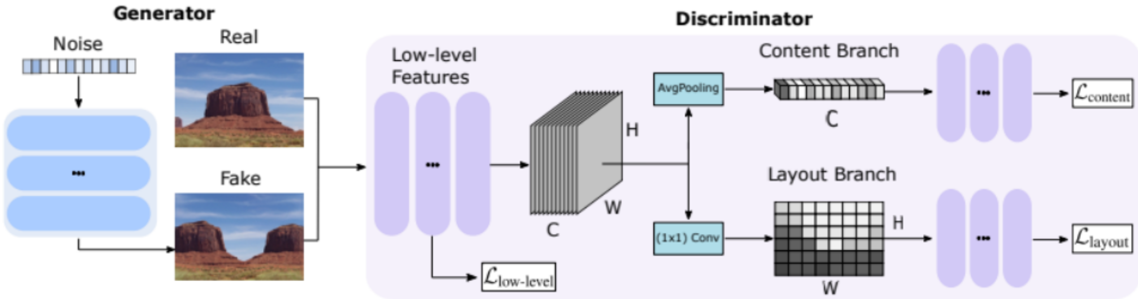

Figure 2: One-shot GAN Architecture

Adversarial Loss: The 3 discriminators decision is based on low-level details of images, like textures, and high-scale properties, such as content and layout. The double weight of low level encourages the generation of low level details with coherent layout content.

$$\mathcal{L}_{adv}(G, D) = \mathcal{L}_{D_{content}} + \mathcal{L}_{D_{layout}} + 2\mathcal{L}_{D_{low-level}} \tag{2}$$

where each $\mathcal{L}$ is binary cross entropy given by $logD(x) + log(1 - D(G(x)))$. Hence, our modified minimax loss function is given by

$$loss = \min_{G} \max_{D} [\mathcal{L}_{adv}(G, D) - \lambda \mathcal{L}_{DR}(G)] \tag{3}$$

The hyperparameter λ here acts like a regularization constant and was tuned and set to the value $\lambda = 0.15$.

4 Architecture modification

The original paper of BrainGAN uses DCGAN and Vanilla GAN. We adopted One Shot GAN to modify the quality of augmentation over here. Further, we train different instances of the GAN for generating tumour images and non-tumour images. And finally, we used pre-trained ResNet for classification.

5 Dataset

[1] The Dataset consists of 400 magnetic resonance images (MRIs) each of which falling into either the Normal or Tumor category having the following distribution:

- Normal (170 MRI)
- Tumor (230 MRI)

[1] **Dataset:** www.kaggle.com/datasets/mhantor/mri-based-brain-tumor-images

The training dataset and our validation dataset is hence of the following composition:

- Training set (Normal 119 and Tumor 161): 70% of the dataset
- Validation set (Normal 51 and Tumor 69): Remaining 30% of the dataset

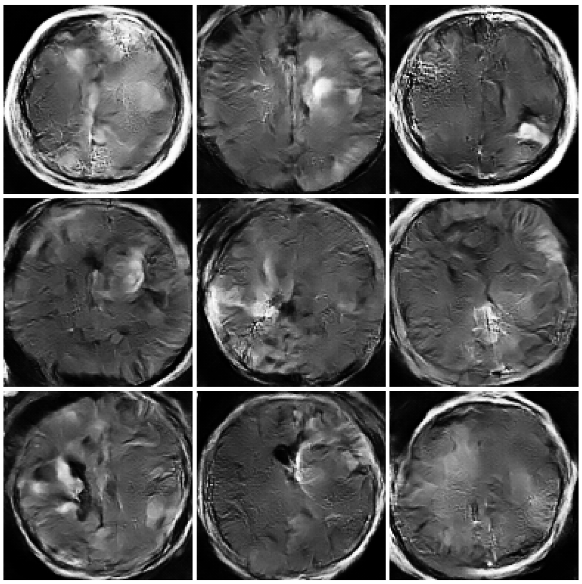

Figure 3: Normal Brain MRIs (Dataset)

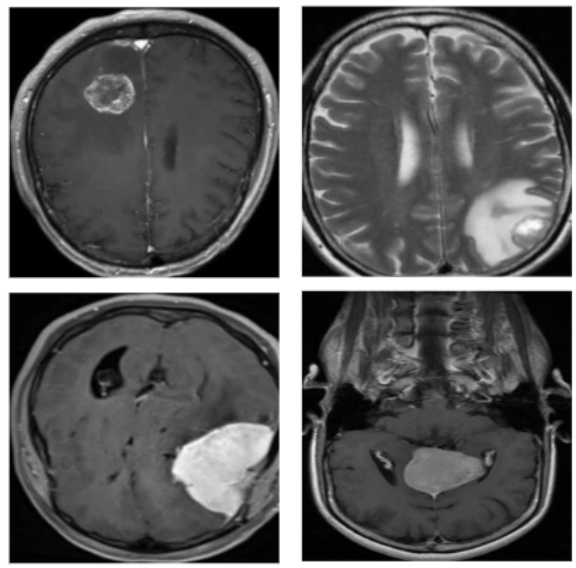

Figure 4: Tumour Brain MRIs (Dataset)

5.1 Data Pre-processing and Pre-training

Pre-processing: We first resized the input images to (128,128,1).

Pre-training: We trained the model on the training set for **10 epochs** with 3**750 steps per epoch** and a **batch size of 4**. Each GAN is trained on 70% of stratified split of the dataset. Similarly for classification and we test on remaining 30% dataset.

5.2 Data Augmentation

We initially had 170 Normal images and 230 Tumour images in our dataset Out of this, we have 119 Normal and 161 Tumour images in our training set for GAN. Using the GAN model, we generate a total of 700 Normal and 700 Tumour images. We then train a classification model on these 1400 images and then test on the 51 Normal and 69 Tumour images left in our validation set.

6 Results and Observations

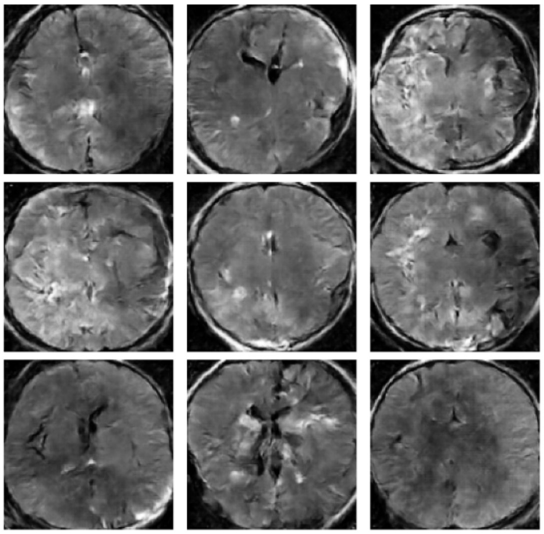

Figure 5: Generated Normal Images

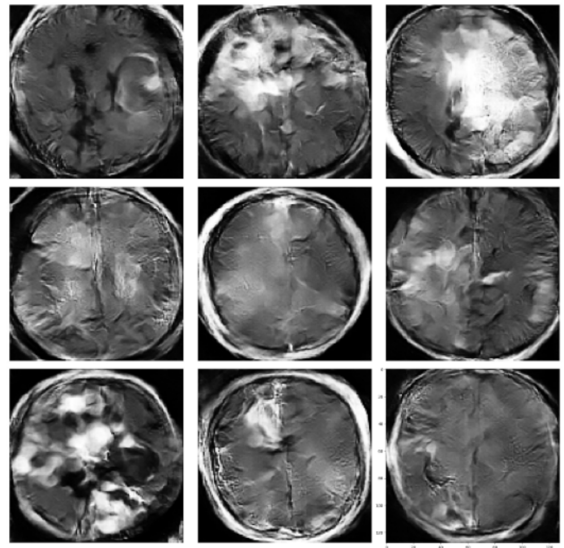

Figure 6: Generated Tumour Images

The distribution below plots the distribution of pixels in real images and generated images.

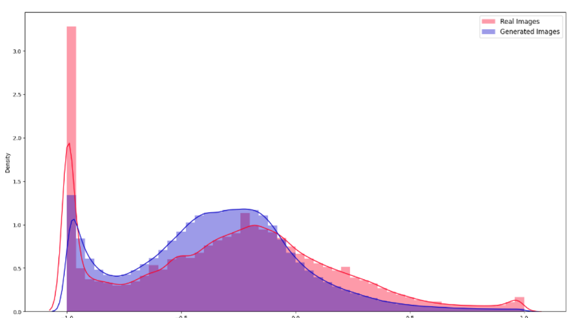

Figure 7: Distribution (for normal images)

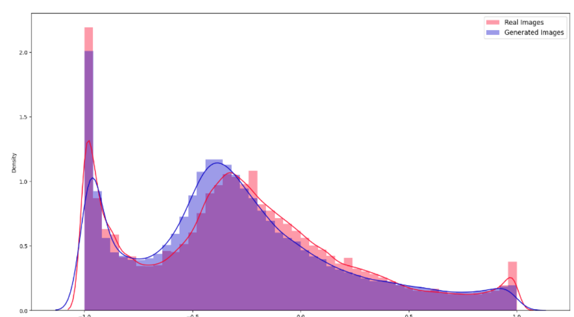

Figure 8: Distribution (for tumour images)

6.1 Controlling the Image variation

As one can see, there is enough difference between the original distribution and our generated distribution but not too much difference. This is an essential goal of data augmentation, to create new images which are sufficiently different from the original images to create more diversity, but not significantly different that they don't resemble the original distribution anymore.

To achieve this goal, we defined a min_threshold and a max_threshold. If L1 distance between original distribution and generated image is less than min_threshold or greater than max_threshold, we rejected the generated image. This significantly improved the results of our classification task.

7 Classification Task

7.1 Classification Head

The original paper tried 3 different architectures for the classification task: **Custom CNN, MobileNetV2, ResNet152V2. ResNet152V2** and gave the best results out of all the 3 models. We directly use a pre-trained ResNet152V2 model for our classification model.

Layer(type)	**Output Shape**	**Parameters**
resnet152v2 (Model)	(None, 4, 4, 2048)	54,331,648
reshape_2 (Reshape)	(None, 4, 4, 2048)	0
flatten_2 (Flatten)	(None, 100352)	0
dense_3 (Dense)	(None, 256)	25,690,368
dropout_2 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None,1)	257

Table 1: The pre-trained ResNet152V2 model architecture

7.2 Results

[2]

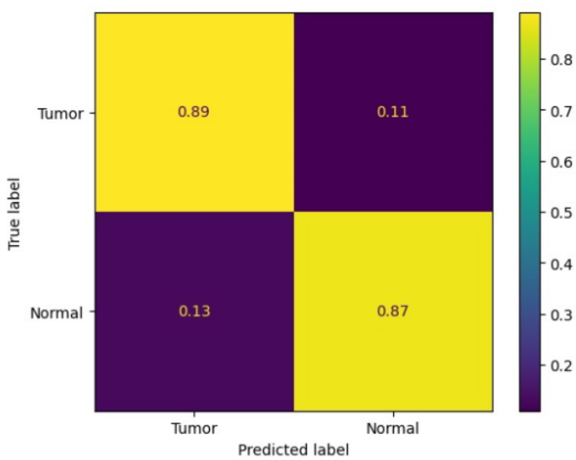

Figure 9: Without Data Augmentation

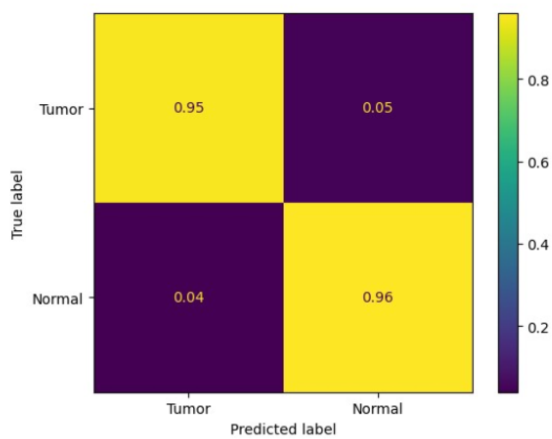

Figure 10: With Data Augmentation

[2] **Code:** https://colab.research.google.com/drive/1DbbvDsOF1UUoptsK1mXt8v18PT-DCpwx?authuser=1scrollTo=87cdc927uniqifier=1

8 Conclusion and Future Work

1. There is a significant improvement in accuracy and other metrics after augmenting our dataset
2. There is still a lot of room for improvement in defining a metric for controlling our generated distribution. There is not a lot of research papers which use this approach, so there is a lot of scope for independent research to be done on this topic.
3. GAN is a really useful model for data augmentation in tasks where training data size is small
4. One shot GAN can have other applications like Xrays, CT and other domains where data collection remains challenging.
5. Experiments can be performed with other MRI datasets also.[SGK21][Has21][HIH+22]

References

[Has21] Antor Mahamudul Hashan. Brain mri images, 2021.

[HIH+22] Dina Hussein, Dina Ibrahim, Halima Hamid, Atheer Fa, and Mohammad Ali. Braingan brain mri image generation and classification framework using gan architectures and cnn models. *Sensors*, 22:4297, 06 2022.

[SGK21] Vadim Sushko, Jürgen Gall, and Anna Khoreva. One-shot gan: Learning to generate samples from single images and videos. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2596–2600, 2021.