**Problem Statement:**

Misdiagnosis in the medical field is a very serious issue but it’s also uncomfortably common to occur. Imaging procedures in the medical field requires an expert radiologist’s opinion since interpreting them is not a simple binary process (Normal or Abnormal). Even so, one radiologist may see something that another does not. This can lead to conflicting reports and make it difficult to effectively recommend treatment options to the patient.

**Why Contrast MRI Scans:**

One of the complicated tasks in medical imaging is to diagnose MRI (Magnetic Resonance Imaging). Sometimes to interpret the scan, the radiologist needs different variations of the imaging which can drastically enhance the accuracy of diagnosis by providing practitioners with a more comprehensive understanding.

A picture containing mirror

Description automatically generated

In the above figure, we can find that a tumor in the MRI scan is more clearly visible after generating a contrast MRI image rather than in the original image.

But to have access to different imaging is difficult and expensive. With the help of deep learning, we can use style transfer to generate artificial MRI images of different contrast levels from existing MRI scans. This will help to provide a better diagnosis with the help of an additional image.

In this project, we use CycleGAN to translate the style of one MRI image to another, which will help in a better understanding of the scanned image. Using GANs we create T2 weighted images from T1 weighted MRI image and vice-versa.

**Objective:**

To build a Generative Adversarial Model which can generate artificial MRI images of different contrast levels from existing MRI scans.

**Method used for building the model:**

Style Transfer technique is used withGAN (Generative Adversarial Network) which runs based on the Convolutional neural networks.

**What is Style Transfer?**

Neural style transfer is an optimization technique used to take two images - a content image and a style reference image (such as an artwork by a famous painter) and blend them together so the output image looks like the content image, but “painted” in the style of the style reference image.

This is implemented by optimizing the output image to match the content statistics of the content image and the style statistics of the style reference image. These statistics are extracted from the images using a convolutional network.

A dog standing on grass

Description automatically generated with medium confidence 

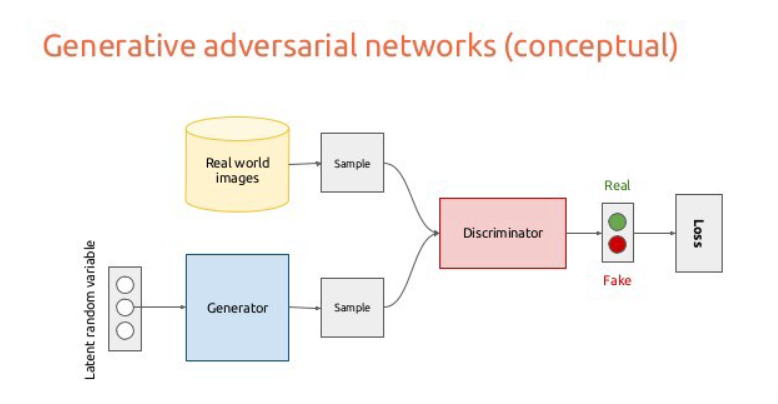
**Original image (Content image)** **Style reference image**

A picture containing grass, dog, outdoor, colorful

Description automatically generated

**Output image**

**How GAN works:**



A generative adversarial network (GAN) has two parts:

* The **generator** learns to generate plausible data. The generated instances become negative training examples for the discriminator.
* The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.

When training begins, the generator produces obviously fake data, and the discriminator quickly learns to tell that it is fake:

As training progresses, the generator gets closer to producing output that can fool the discriminator:

Finally, if generator training goes well, the discriminator gets worse at telling the difference between real and fake. It starts to classify fake data as real, and its accuracy decreases.

Both the generator and the discriminator are neural networks. The generator output is connected directly to the discriminator input. Through backpropagation the discriminator's classification provides a signal that the generator uses to update its weights.

**Dataset:** The dataset has been taken from github which can be found in the below link.

The data set contains 5000 images for training data and 2000 images for testing.

The whole process of how this data and the results is used can be found in the python code file submitted.

**Method**:

The proposed method efficiently and effectively learns the mappings among four contrasts of MRI [T1-weighted, T2-weighted, Proton Density (PD)-weighted, Magnetic Resonance Angiography (MRA)] to generate a fake image of target contrast given a real image and original contrast. For example, given an input image of T1-weighted contrast our model can generate fake T2-weighted, PDweighted and MRA images using only one generator. U-NET generator performs two synthesis: (i) generating a fake image given depth-wise concatenated real image and target contrast; (ii) reconstructing real image given fake image concatenated depth-wise with original contrast. Fake image is used to measure two losses: (i) Adversarial loss and (ii) contrast classification loss using PatchGAN discriminator. Reconstructed and real image are used to measure reconstruction loss to observe how close reconstructed image is to the real image in terms of structural (SSIM), perceptual (LPIPS) and global (L1) similarity.

Diagram

Description automatically generated

**Loss Functions**

**Adversarial Loss:** Instead of using adversarial loss which is reported to suffer from various training problems including mode collapse, vanishing gradients and sensitivity to hyper-parameters, we use regularized Wassersteing GAN with gradient penalty (WGAN-GP). This not only provides stable learning for deep generator and discriminator networks but also increase the quality of generated images. Generator G takes an input image x and a target label c to generate a fake image of the target contrast. While the discriminator D is responsible for finding out if the given image is real (from training set) or fake (generated by G).

**Contrast Classification Loss:** It forces the generator to produce image of correct contrast and allows discriminator to perform contrast classification for real and fake images .

**Generation Loss:** If the model generates a fake image T-1 belonging to T1 contrast using a real T2-weighted image then by using reverse mapping it should reconstruct the real T2-weighted image.

**Network Architecture:** We use UNet based generator for our model. The generator contains 7 down-sampling layers with strided convolutions of stride 2 followed by the 7 up-sampling layers with fractional strides. Each convolutional layer is followed by instance normalization and ReLU activation except for the final layer which uses tanh after convolution layer. We are using PatchGANsbased discriminator which can classify local patches for real or fake, providing efficiency over full image classifier.