UNIVERSITY OF TURKU

Data Augmentation with Conditional Generative Adversarial Networks (cGANs) for Deep Learning-based Classification of Brain Tumor Magnetic Resonance Images

Master's Degree Programme in Biomedical Imaging
Master's thesis
University of Turku
Faculty of Medicine
Institute of Biomedicine

Author(s):
Mahnoor Mahnoor

18.05.2025
Turku

Master's thesis

Subject: Master's Degree Programme in Biomedical Imaging
Author(s): Mahnoor Mahnoor
Title: Data augmentation with conditional generative adversarial networks (cGANs) for deep learning-based classification of brain tumor magnetic resonance images
Supervisor(s): Dr. Rainio, Oona; Dr. Tadi, Mojtaba Jafari; Professor Klén, Riku.
Number of pages: 59 pages
Date: 18.05.2025

Abstract

Background and aims: Generative adversarial networks (GAN) have been popularly used in generating augmented data in medical imaging. However, a classical GAN model is prone to model collapse, class imbalance, and instability. The purpose of this study was to validate a deep learning (DL) algorithm that generated brain tumor and non-tumor images from magnetic resonance images (MRI) and to compare its performance with that of true brain tumor and true brain non-tumor images from MRI.
Materials and methods: This single-center, retrospective study included MRI brain tumor and healthy control images from a public repository. Datasets were divided into training (63%), validation (6%), and test (31%) sets, with stratification by presence and absence of brain tumor. A conditional-generative adversarial network was trained to produce brain tumor and non-tumor images. The generated images were trained on a modified U-Net CNN multiple times with different numbers of generated images, and their classification accuracy was evaluated from a separate set of unseen dataset of 2000 images. The Mann-Whitney U-test was used to estimate the statistical significance between generated and true images. The generated images are systematically evaluated by multiple evaluation metrics, such as the Inception Score (IS), Frechet Inception Distance (FID), Structural Similarity Index Measure (SSIM), Mean Squared Error (MSE), Dice Similarity Coefficient (DSC), and Peak Signal-to-Noise Ratio (PSNR).
Results: A total of 2000 MRI images were generated, having an equal number of brain tumor and non-tumor images. The CNN trained with 1000 true and 1500 generated images worked the best, achieving 92% accuracy, 90% sensitivity, and 85% specificity for the diagnosis of brain tumors. The generated images exhibited an IS, SSIM, MSE, DSE, FID, and average PSNR of 2.09, 0.16, 7609, 0.89, 0.64, and 28.78 dB, respectively.
Conclusion: The classification performance of convolutional neural networks (CNNs) increased when its training set was augmented with generated MRI brain tumor and non-tumor images, suggesting that synthetic images can serve as effective alternatives to real images in deep learning-based classification models.
Significance: The results highlight the potential of generative MRI images as viable alternatives to real MRI scans for CNN-based brain tumor classification. It addresses data scarcity,

enhances model robustness, preserves patient privacy, and reduces costs, making AI-driven diagnostics more scalable and efficient.

Keywords: Conditional Generative Adversarial Networks, Convolutional Neural Networks, Brain tumor, Magnetic Resonance Images

Contents

List of Figures

List of Tables

Abbreviations

2D	Two-Dimensional.
3D	Three-Dimensional.
AUC	Area Under the Curve.
AI	Artificial Intelligence.
ANN	Artificial Neural Network.
BUSI	Breast Ultrasound Image Dataset.
cGAN	Conditional Generative Adversarial network.
CNNs	Convolutional Neural Networks.
CT	Computed tomography.
D	Discriminator.
db	Decibel.
DL	Deep learning.
DSC	Dice Similarity Coefficient.
FCN	Fully convolutional networks.
FID	Frechet Inception Distance.
FLAIR	Fluid-Attenuated Inversion Recovery.
FN	False Negative.
FP	False Positive.
FPR	False-Positive Rate.
G	Generator.
GAN	Generative Adversarial Network.
GCN	Graph Convolutional Networks.
IS	Inception Score.
JS	Jensen-Shannon.
KL	Kullback-Leibler.
MBA	Median Balance Accuracies.
ML	Machine Learning.
MRI	Magnetic Resonance Imaging.
MSE	Mean Squared Error.
NCIA	National Cancer Imaging Archive.
NTITM	Non-Tumor Images with Tumor Masks.
NTT-NTM	Non-Tumor-Trained and Non-Tumor-Mask.
NTT-TINTM	Non-Tumor-Trained and Tumor-Image with Non-Tumor mask.
PET	Positron Emission Tomography.
PSNR	Peak Signal-to-Noise Ratio.
RDN	Residual Dense Network.
ROC	Receiver operating characteristics.
SSIM	Structural Similarity Index Measure.

TM	Tumor Mask.
TN	True Negative.
TP	True Positive.
TPR	True positive rate.
TT	Tumor Trained.
TT-NTITM	Tumor-Trained and Non-Tumor Image with Tumor Mask.
TT-TM	Tumor-Trained and Tumor-Mask.

1 Introduction

The central nervous system, assessed by the brain and spinal cord, coordinates all sensory information and the actions that go along with it [68]. The brain is a complex organ that controls our breathing, heartbeat, temperature, hunger, emotions, and movements [5]. However, tumors in the brain tend to disturb the brain's normal functioning. A brain tumor is the aberrant proliferation of brain cells [7]. The National Foundation for Cancer Research [57], estimates that about 25,400 people will be diagnosed with a malignant brain tumor, with about 18,760 deaths expected in the United States. Benign brain tumors are non-cancerous tumors, while malignant brain tumors are cancer-causing [9]. Malignant tumors are of main concern as they can metastasize to other body organs and spread cancer. Nonetheless, early tumor detection allows for timely treatment, increasing the chance of survival.

Brain tumors are detected and analyzed using imaging technologies such as positron emission tomography (PET), computed tomography (CT), and magnetic resonance imaging (MRI) [3]. However, MRI is recognized as the most prevalent and efficient diagnostic tool for identifying brain tumors within the clinical community. It is a non-ionizing and non-invasive modality that provides valuable information regarding the size, type, shape, and location of tumors [63]. MRI offers essential information for observing brain anatomy that is helpful for diagnosing some brain irregularities like brain tumors [62].

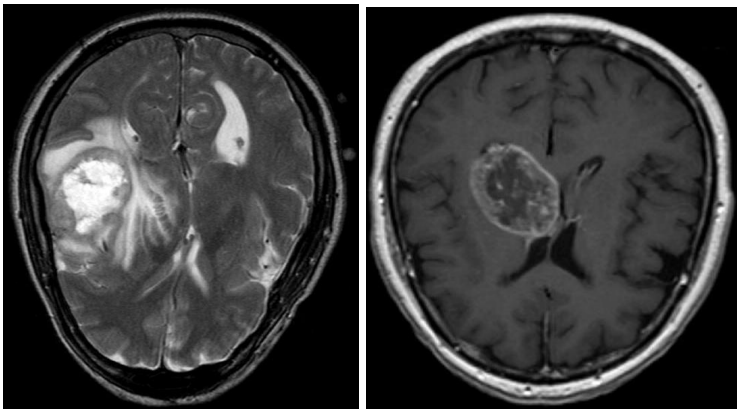

Figure 1.1: Transaxial view of magnetic resonance brain tumor images [53].

Preprocessing MRI images provides valuable information and can be utilized in various clinical applications and research studies. For example, brain segmentation images have been widely used in locating brain lesions [12], tumors [29], neurodegenerative diseases such as Alzheimer's [64], and dementia [22], etc. However, with the advancement in artificial intelligence (AI) and machine learning (ML), MRI image augmentation provides realistic-looking artificial brain images that enlarge datasets and help improve machine learning algorithms [51]. Many deep learning models have been prepared to retrieve valuable and distinctive features from the brain MRI images that help diagnose brain tumors [70].

Medical imaging classification holds significant importance in categorizing and diagnosing human diseases. The classification of medical images has become an easy process with the

current advancements in AI. Machine learning is a type of AI that has become a powerful tool with its optimized ability in disease diagnosis, tissue segmentation, and image classification [76], [50]. One ML model type, a convolutional Neural Network (CNN), is thought to be the most efficient approach with high diagnostic accuracy given the large dataset [33]. Autoencoder, being an unsupervised ML model, utilizes a neural network for representation learning [14]. Remarkably, numerous machine learning models have been developed to accurately diagnose tumors, including brain tumors, with high diagnostic accuracy [40].

Deep learning (DL) is a subtype of machine learning-based neural networks that are used to learn data representations and can be applied to supervised, semi-supervised, and unsupervised learning [47] and has been making many breakthroughs in the field of medical diagnosis. Dozens of DL algorithms aid in image processing, like classification and segmentation [13]. These deep-learning algorithms demand a significant quantity of data for training, highlighting the problem of data scarcity [45] and class imbalance [10]. Many solutions have been proposed to overcome data scarcity, like data augmentation [75], and k-fold cross-validation [25].

To train any DL model in medical imaging, the original dataset is split into two sets: a training set and a test set. The training dataset is used to train a given algorithm that runs through multiple iterations, and the parameters of the model are optimized to improve the model's performance. The test dataset is used to analyze the model's final performance. Usually, DL models require a large dataset to train. However, ML models are unable to function effectively and collapse due to limited data. To avoid this situation, the data augmentation technique is applied.

Data augmentation techniques are applied to amplify the size and diversity of training data to improve the performance of DL algorithms. Traditional data augmentation techniques involve reflection, rotation, translation, cropping, or adding a Gaussian blur to increase the dataset, which is quick and simple to implement [66]. Nonetheless, these techniques enhance the data quantity but never reach the level of a realistic situation. This increases the need for deep learning-based generative models to create realistic-looking augmented data [46] and to reduce dataset biases and class imbalance [26].

Goodfellow et al. [27] presented a generative adversarial network (GAN) to generate synthetically augmented data, performing superior to traditional augmentation techniques. A GAN architecture consists of two neural networks: a generator and a discriminator. Data is fed into the generator to generate artificial medical images, while the discriminator separates artificially generated images from the original images. The generator functions to improve its generated data so that it can forge the discriminator and produce realistic-looking artificial images. Since then, many models have been proposed to synthesize artificial images. A recent advancement in GAN is conditional GAN (cGAN), where neural networks are conditioned to auxiliary input such as class labels and feed to the generator and the discriminator to guide data generation, producing augmented labeled data [65].

Scientifically speaking, diagnosing tumors from medical images could be more fallacious and rely heavily on radiologists expertise and fatigue level. Computational intelligence acts

as an assisting tool for radiologists and physicians in identifying and diagnosing brain tumors. This study aims to examine the innovative application and modification of cGAN for data augmentation in medical imaging, particularly for MRI brain tumor images. The performance of the presented model will be assessed through different quality metrics, and the model's effect on the performance of binary classification in medical images will be evaluated. This approach will tackle the challenges of class imbalance in the brain tumor MRI dataset and address the limitations posed by small data sizes.

1.1 Research questions

In this thesis, it is proposed to utilize cGAN originally presented by [52], a variation of traditional GAN, to generate synthetic images with conditioned input data. In cGAN, a condition is given to the generator and discriminator model along with random noise to produce condition-specific images. The condition could be a label or any structured input. In this way, we get a diverse set of outputs through targeted image generation. Our experiment will focus on generating 2-dimensional (2D) MRI brain tumor images with segmentation maps.

This thesis will research the generation of artificially augmented medical image datasets. The thesis will aim to answer the following research question:

How do augmented data obtained from conditional generative adversarial networks affect the performance of the binary classifier for medical image datasets?

To answer the research question above, we answer several sub-questions:

- **Question 1:** Does cGAN performing multi-domain image-to-image translation produce realistic brain tumor magnetic resonance images?
- **Question 2:** What parameters need to be optimized to get high-quality output from cGAN? What evaluation metrics should be used to judge generated image quality, and how are these metrics affected by the proposed cGAN model?
- **Question 3:** Can the classification accuracy of state-of-art classification CNN be increased via augmentation with cGAN-generated images, and should Transfer Learning be considered given the small size of the dataset?
- **Question 4:** What will be the outcomes of the classifier when no augmented data is used, and how accurate will it be when cGAN-generated images are used?

2 Aims and Hypothesis

This thesis aims to address the challenge of class imbalance in brain tumor MRI datasets by developing a cGAN architecture capable of generating high-quality, label-consistent synthetic images. By generating additional MRI scans for underrepresented classes—specifically, tumor-positive and tumor-negative samples—the proposed method aims to enhance the diversity and balance of the training data. The generated images will be used to augment datasets for deep learning classifiers, aiming to improve classification accuracy and robustness in identifying brain tumors. Unlike existing studies that focus on comparative analyses, this work emphasizes demonstrating the standalone effectiveness and viability of the proposed cGAN-based augmentation approach for deep learning-based brain tumor diagnosis.

For the research question, we have the following hypotheses:

- **Hypothesis 1:** We expect the cGAN model will perform better in multi-modal label-to-image translation based on the evaluation metrics of generated images. For cGAN, some state-of-the-art models usually do not have enough open-source code. So, we expect to build an open-source cGAN code.
- **Hypothesis 2:** We expect that Inception V3 will work best on binary classification. We expect to have synthetic images obtained from cGAN having similarity close to real datasets in terms of contrast and image quality. We also expect the synthetic images to perform better with the classification model than when no augmented data is used.

2.1 Objectives

Based on research questions, the objective of the thesis is to train a cGAN to generate better-quality MRI brain tumor images. The objectives of the thesis are:

- **Objective 1**: Implement and train the cGAN algorithm that could generate synthetic brain MRI tumor images with the given labels.

Figure 2.1 illustrates the process for Objective 1.

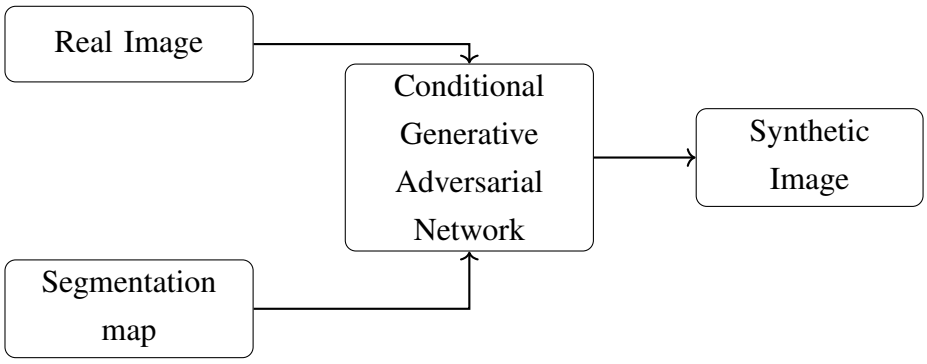

Figure 2.1: Original MRI dataset along with tumor labels given to cGAN to generate augmented MRI dataset.

- **Objective 2**: Implement and train a classifier and determine if using synthetic MRI data can improve the accuracy.

Figure 2.2 illustrates the process for objective 2.

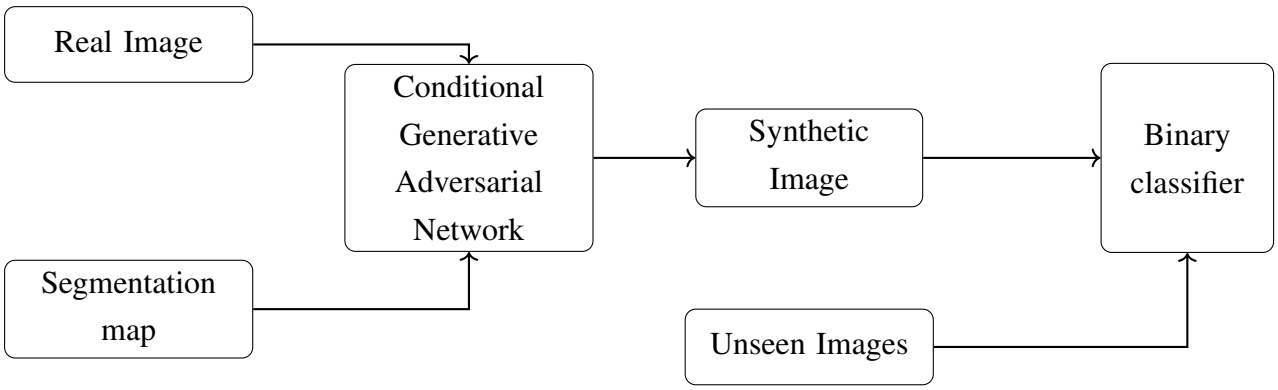

Figure 2.2: The generated images from conditional-GAN (cGAN) feed to the binary classifier along with an unseen set of the original dataset.

3 Background and Literature Review

3.1 Artificial Neural Networks

Machine learning is a subclass of AI in which data is given to the machine, and the machine analyzes the data, learns from it, and draws conclusions or makes decisions from the data [18]. The machine model is either supervised, semi-supervised, or unsupervised [81]. An artificial neural network (ANN) is a computational model composed of layers of interconnected artificial neurons. These neurons are organized into an input layer (which receives the data), one or more hidden layers (which perform intermediate computations and feature transformations), and an output layer (which produces the final result). A classical ANN model consists of many hidden layers [80]. Recently, many new types of ANN algorithms have been developed, such as Deep Learning Neural Networks [21] and Radial Basis Function Networks [73].

ANN has been used widely to classify medical images. Nalbalwar et al. [56] conducted a study using ANN as a classifier to identify different types of tumors and CNN for early detection of skin cancer [80]. ANN is used to classify heart states as diastole and systole from echocardiographic images in an attempt for early diagnosis of heart diseases [77].

3.2 Deep Learning and Convolutional Neural Networks

Deep learning (DL) is a type of ML algorithm that uses ANNs with many neural layers between the input and output layers. DL has more deep hidden layers than a typical ML model, which enables DL to learn complex patterns and depictions of intricate datasets [92]. Because of this, DL is well suited for complicated data, such as speech recognition [42], natural language processing, and computer vision [84].

Convolutional neural network (CNN) is one of the DL networks [87] with its ability to identify the significant features automatically without human oversight [20]. Forsyth et al. [23] introduced CNNs. Like ANNs, CNNs have many hidden layers consisting of convolutional layers, pooling layers, and many connected layers. Convolutional layers are the backbone of CNNs.

CNN is used for image classification in medical imaging. Tiwari et al. [82] in their paper proposes the CNN model for classifying brain tumor MRI images. Xie et al. [85] presented a CNN model in their research that could perform three-dimensional (3D) medical image segmentation.

3.3 Generative Adversarial Networks

Generative adversarial networks, or GAN were first introduced by Ian Goodfellow et al. in 2014 [52]. It consist of two neural networks responsible for learning realistic distribution from the training dataset in a competitive setting. One framework is generator network or generator (G) that is responsible for generating fake inputs from noise. The other network is discriminator network or discriminator (D) where the fake inputs from generator to fed to and is responsible

for discriminating real data (X) from the fake data ($G(X)$). Fundamentally, the generator aim to produced inputs that are indistinguishable from real, while discriminator tries to correctly classify the real and fake generated inputs.

This adversarial optimization approach has received significant attraction from academia and industrial practitioners. Its potential to mitigate domain shift and produce high-quality synthetic images has enabled GAN models to achieved state-of-art performance in various computer vision application such as optimizing image resolution [86], text-to-image synthesis [48], and image-to-image translation [91].

The original GAN model proposed in 2014 [52] is a generative model that learns to generate new samples directly from a target distribution, without the need of explicit modeling of the underlying probability distribution. Typically, the generator G takes the noise vector z from prior distribution such as Gaussian or uniform distribution $p(z)$ and outputs a sample $x_g = G(z;\theta_g)$, where θ_g denotes the parameters of G. Given the real data distribution $p_r(x)$, the output of generator or (G) x_g is expected to have visual similarity close to real sample x. The discriminator (D) is either given real $x_r \sim p_r(x)$ or fake generated inputs $x_g \sim p_g(x)$, and the output $y_1 = D(x;\theta_d)$ of D is a scalar value indicating the probability of input being fake or real. The objective of GAN is to train G to generate fake distributions $p_g(x)$ close enough to real distribution $p_r(x)$ so that to make it hard for D to discriminate between real and fake data.

Mathematically, the optimization of generator G and discriminator D is given as:

Generator (G) :

$$\mathscr{L}_G^{\text{GAN}} = \min_G \mathbb{E}_{x_g \sim p_g(x)} [\log(1 - D(x_g))]$$

Discriminator (D):

$$\mathscr{L}_D^{\text{GAN}} = \max_D \mathbb{E}_{x_r \sim p_r(x)} [\log D(x_r)] + \mathbb{E}_{x_g \sim p_g(x)} [\log(1 - D(x_g))]$$

In this equation, $\mathbb{E}_{x_g \sim p_g(x)}$ is the expected value of the generated sample x_g drawn from the probaility distribution $p_g(x)$ while $\mathbb{E}_{x_r \sim p_r(x)}$ is the expected value of the real data x_r drawn from the real data data distribution $p_r(x)$.

In this setting, the D is a binary function trained to optimize the log-likelihood of correct classification. If the D is optimized fully before the generator updates, then minimizing the $\mathscr{L}_G^{\text{GAN}}$ becomes equivalent of reducing divergence between $p_r(x)$ and $p_g(x)$. The expected outcome from a well-trained GAN results in generator producing $p_g(x)$ approximate to real data distribution $p_r(x)$ of the real input images.

GANs are generative models for producing natural images. There are significant challenges with GAN training, such as model collapse, instability, and non-convergence, due to improper network architecture and selection of optimization algorithms [72]. To address these challenges, many solutions for better design and algorithm optimization have been developed. We proposed cGAN, a type of GAN that uses labels as input for generating targeted images, in our thesis as a solution to avoid model collapse.

3.4 Conditional Generative Adversarial Networks

The cGAN is an extension of simple GAN that uses labels in the form of conditional input for generating the data. In cGAN, the images are generated with specific conditions, instructing the model to render images belonging to a particularly category [48]. The addition of class label serves two main purposes: to generate the targeted images, and to enhance to model performance. The additional information in the form of class labels leads to more stable model, faster training, and domain specific high-quality images. One limitation with basic GAN is its lack of control on its output - the mapping of latent space from input to output images is complex and unpredictable, and it generate random images from the domain. cGANs restrict the model performance by putting condition on the generator and discriminator through the provided class label. In results, when trained standalone on a generator, it generates images with given class label only. There are many ways to integrate class labels in the GAN architecture. One of the method proposed by Denton et al. [19], involves adding embedding layers on the top of CNN architecture followed by a fully connected layer with a linear activation. This layer scales the embedding to the dimension of the input image after which concatenated in the model as additional layer. Importantly, GANs can not only be conditioned to class labels, but can be conditioned to other labels such as image, enabling image-to-image translation application of GANs.

3.5 Evaluation cGANs

The performance and efficiency of a model are assessed by evaluation metrics. These metrics evaluate the performance of the model, which is crucial to assess the quality of the model and to make improvements in it. A cGAN model generates synthetic data that is comparable to real-world data, and evaluation metrics are a way to measure its quality and assess how much generated data is different from the original data.

3.5.1 Inception Score

The Inception Score (IS) is a mathematical algorithm used for measuring the quality and variety of generated images, introduced by [71]. A well-known CNN image classification network for Inception Score introduced by Szegedy et al. [79] was used to evaluate generated images.

The class probability of each generated image is measured by the inception score. IS is calculated by comparing the distribution of class probabilities for each individually created image with the distribution of class probabilities for all generated images. Kullback-Leibler divergence [41] is a mathematical formula that measures the comparison. The final IS value is the exponential value of the expected comparison value.

A high IS score suggests a good variety of features in generated images, while a low IS score suggests less diversity or lower quality of generated images.

3.5.2 Frechet Inception Distance

Heusel et al. [32] introduced the Frechet Inception Distance (FID), a metric that quantifies the similarity between generated images and real images by comparing their underlying feature distributions. This measure was proposed as an improvement over the Inception Score (IS) and computes the distance between the distributions of feature vectors extracted from a pretrained Inception network for real versus generated images [90].

A low FID means generated images are similar to real images and are of high quality, whereas a high FID means poor image quality.

3.5.3 Structural Similarity Index Measure

The structural similarity index measure (SSIM) is a mathematical way of calculating the predicted perceived quality of image [61]. It is used to measure similarity between images. In terms of GANs, the measurement or prediction of the perceived augmented image quality is done in comparison with initial real image a reference. SSIM is a perception-based model that calculates the changes in structural information of images in comparison. It is based on the idea that the pixels display string inter-dependencies, specially when they are spatially closed to each other - a relationship that convey critical structural information about object in comparison [54].

3.5.4 Mean Squared Error

The mean squared error (MSE) or mean square deviation is a mathematical way of measuring the average squared of the difference between estimated error and real error [34]. It is the risk function that correspond to the expected value of the squared error loss. The fact that MSE is always strictly positive (rather than zero) because of its intrinsic nature of randomness in the data. In machine learning, MSE is refer to as the empirical risk - i.e., the average squared loss calculated on a given dataset [43].

3.5.5 Peak Signal-to-Noise Ratio

The peak signal-to-noise ratio (PSNR) is a mathematical metric of calculating the ratio between the maximum signal to maximum noise that effects the fidelity of the data. Because many data have diverse dynamic range. PSNR is expressed as logarithmic quantity in decibels (db). PSNR is defined via the MSE values, is widely used to asses the reconstruction quality for images and videos. Higher PSNR generally indicates better image reconstruction quality [78].

3.5.6 Deep Scatter Estimation

The deep scatter estimation or DSE is the term used in machine learning to estimate the scatter patterns in the data, typically in context of signal processing or image quality. When paired with deep learning methods, such as neural networks, DSE uses model to learn from data to estimate the scatter of the generated data [30].

3.6 Literature Review

Among the studies on medical images, Naderi et al. [55]; Amirrajab et al. [4]; Dar et al. [17]; and Oh et al. [44] used cGAN to reconstruct MRI images. HaoQi et al. used the conventional cGAN to augment the retinal fundus images and vessel segmented images of retina [28]; Yi used cGAN with sharpness detection networks to potentially denoise the low-dose computed tomographic images; Ben-Cohen et al. used fully convolutional networks (FCN) and cGAN to artificially generate PET images from CT data [8]; The relevant studies under this topic are summarized in Table 3.1.

Table 3.1: Summary of classification model for classifying synthetically generated medical images

Research Topic	**GAN Model**	**Dataset**	**No. of Images & Image dimensions**	**Performance**	**Ref.**
Retinal fundus image augmentation with cGAN	Conventional Conditional GAN	Retinal fundus images	600, 512 × 512 pixels	F1-score: 82.75%	[28]
Pix2Pix medical image segmentation by cGAN	cGAN	HC18 ultrasound & Montgomery Chest X-ray	999 & 114, 256 × 256 pixels	Dice score: 97.92%	[55]
Application of conditional generative adversarial networks for generating muti-contrast MRI images	cGAN	MIDAS [11], IXI [37], & BRATS dataset (Brain MRI images) [6]	4000-5000	SSIM: 89.5%	[17]
Generating low-dose CT images by sharpness detection of blur images using conditional generative adversarial networks	cGAN (Sharpness Awareness GAN using sharpness detection network)	CT images from National Cancer Imaging Archive	2832, 256 × 256 pixels	SSIM: 87.01%	[88]

Research Topic	GAN Model	Dataset	No. of Images	Performance	Ref.
Generating artificial PET images from CT data using DC-GAN	cGAN image-to-image translation	Data taken from Sheba Medical Center	25 CT and PET pairs of liver region	True positive rate (TPR): 92.3%	[8]
Application of pix2pixGAN for generating synthetic images of brain slices	pix2pix cGAN	Brain slices images [39]	2300, 256 × 256 pixels	SSIM:0354	[2]
cGAN application for generating breast cancer images	Conditional Generative Adversarial Network (cGAN)	Ki67-stained whole slide images from breast cancer patients	694 images, 256 × 256 pixels	Immuno Ratio Value: 0.53	[74]
Data augmentation through deep learning models	DreamOn, a conditional generative adversarial network (GAN) based on REM-dream-inspired interpolations	Breast Ultrasound Image Dataset	780, image resolution not defined	Median Balance Accuracies: 0.55	[49]
Low-dose CT contrast enhancement and volume quantification using Deep Learning Model	Residual Dense Network and conditional Generative Adversarial Network (cGAN)	Computed Tomographic images	Not explicity mentioned	SNR: 13.3 +- 1.9	[89]
Radiomic prediction based on MRI images of breast cancer using cGAN	Conditional Generative Adversarial Network (cGAN)	MRI breast cancer images	187, 128 × 128	FID score: 1.31	[35]

Research Topic	**GAN Model**	**Dataset**	**No. of Images**	**Performance**	**Ref.**
Cardiac MRI image generation through conditional generative adversarial networks	Conditional Generative Adversarial Networks (cGANs)	Cardiac MRI images	number of images not specified, 256×256	Dice score: 0.95	[4]
Graph-based conditional generative adversarial networks for functional connectivity regeneration from MRI images	Graph Convolutional Networks-based Conditional GAN with Class-Aware Discriminator	Brain MRI	477, image dimensions not specified	F1-score: 0.69	[59]
Brain tumor segmentation with conditional synthesis	Conditional Generative Adversarial Network (cGAN)	Brain MRI images	443, image dimensions not specified	Mean SSIM for T1W + FDA: 0.82	[44]

4 Materials and Methodology

This chapter describes the methodology for generating synthetic MRI images using cGAN. The methodology includes the software requirements, the dataset used, data division, data preprocessing, data segmentation, training and testing strategies, cGAN architecture, data postprocessing, binary classifier, statistical testing, evaluation metrics, and hyperparameter optimization.

4.1 Software requirements

The cGAN model was built and tested in Python [83] (version: 3.9.9) with additional packages including Keras [15] (version: 2.15.0) and TensorFlow [1] (version: 2.15.0). We conducted training using 2 NVIDIA Volta V100 GPUs, each with 32 GPU memory, on the Puhti supercomputer of CSC - IT Center for Sciences, Finland [16] (version: 3.1.10). The model was executed in TensorFlow 2.15 and data parallelism was applied across all GPUs to accelerate the training process.

4.2 Dataset Used

This thesis utilizes the publicly available dataset of the Brain Tumor MRI Dataset [53]. The dataset consisted of 2-dimensional MRI images of the human brain region, with a subset of MRI brain images with and without tumors. It includes images acquired from multiple imaging sequences, including T1-weighted, T2-weighted, fluid-attenuated inversion recovery (FLAIR), and others. The resolution range across the dataset spanned from 150×168 pixels (minimum) to 1920×1446 pixels (maximum). Patient information and treatment details were not included in the dataset. However, only the images were utilized in this thesis.

4.3 Data Division

The Figure 4.1 explains the dataset division. A total of 8,400 images were used, which were partitioned into four distinct subsets. The train dataset consisted of 4,000 images (47.6%), the test dataset consisted of 2,000 images (23.8%), the validation dataset consisted of 400 images (4.8%), and the totally unseen dataset consisted of 2,000 images (23.8%). The totally unseen dataset was used to test an encoder part of U-Net CNN for classification purposes, where images were used to evaluate the accuracy of generated images when tested for binary classification. Each dataset had an equal number of brain images with and without a tumor. All images were in .jpg format.

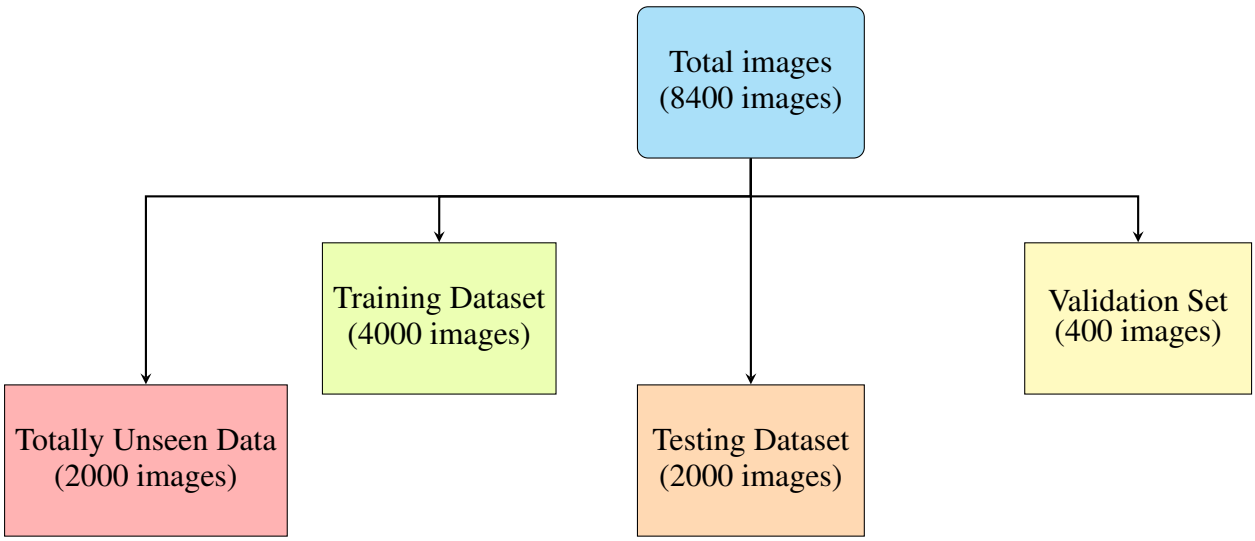

Figure 4.1: Data splitting methodology showing how the total dataset was divided into training, testing, validation, and unseen sets for model development and evaluation.

4.4 Data Preprocessing

The images in the dataset had a background that was not of interest. All the images were cropped to remove rows and columns that were entirely black (pixel value of 0). We used Python libraries cv2 (version: 4.9.0) [60] for image processing to load the data. Once the images were loaded, the Python script calculated the boundaries to crop by finding the first and last rows and columns that contain non-zero pixel values. The cropped images were resized to 256×256 and saved to the new directory for later use. Images were saved in .jpg format and were in greyscale. This step is shown in Figure 4.2.

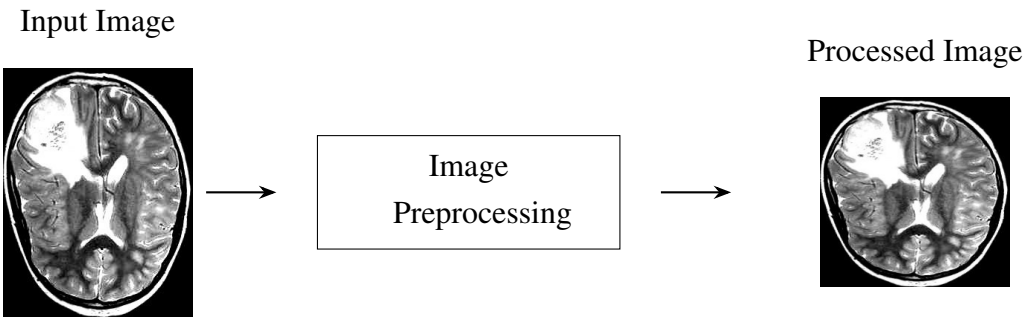

Figure 4.2: Image preprocessing pipeline showing the transformation from input to processed image.

4.5 Data Segmentation

Before training a cGAN, we created segmentation maps for our dataset. cGAN later used these segmentation maps to construct new images based on the labels. Adaptive thresholding was used to generate segmentation maps.

Adaptive thresholding determines the threshold for a pixel based on a small region around it. So, we get different thresholds for different regions of the same image, which gives better results

for images with varying illumination. Adaptive mean thresholding uses the mean value of the neighborhood area (a small square region around each pixel in the image) minus the constant **C** with the given kernel. This means that instead of applying a single global threshold to all pixels, the algorithm will look at the pixels in a local neighborhood around the current pixel, calculate a threshold based on the average (mean) of the pixel values in that neighborhood. The constant **C** effectively allows for fine-tuning the thresholding, acting as a bias term that can adjust how lenient or strict the binarization is. By subtracting **C**, the threshold value is slightly lowered, which can help distinguish features from the background under different lighting conditions in the image.

We used the cv2, NumPy (version: 1.26.4) [58], and pyplot (version: 3.8.4) [36] libraries. The images were imported from the folder, and a median blur with a kernel size of 5 was applied. This reduced noise in the image, which is particularly helpful before performing thresholding. Next, adaptive mean thresholding converts the image to a binary form. It uses a neighborhood size of 11 and a constant of 2, which applies the threshold value to each pixel based on the local region of the pixel. With a neighborhood size of 11, the algorithm will look at the window from (x-5, y-5) to (x+5, y+5) to compute the threshold for just that pixel. Figure 4.3 explains the process of applying adaptive thresholding on MRI images to create segmentation maps. These segmentation maps were used in training and testing datasets for the model testing. Figure 4.4 presents the brain MRI images and their corresponding segmentation maps. Every image and its segmentation map were combined into a single .jpg file.

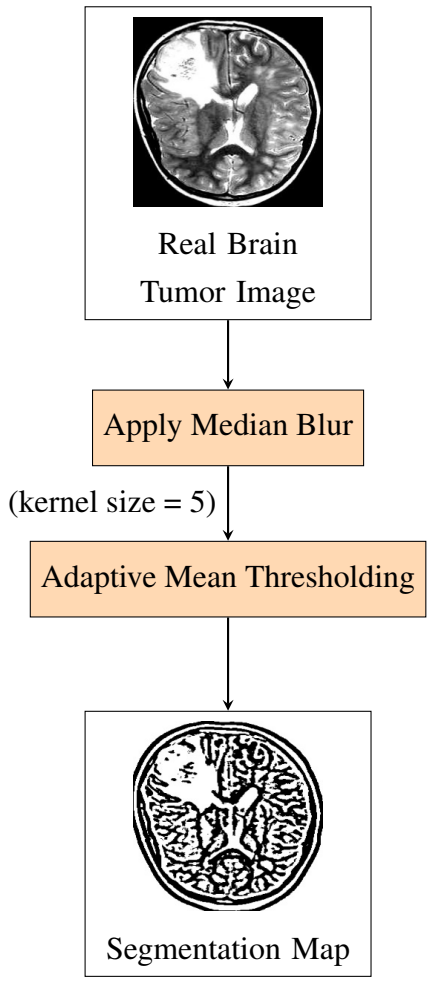

Figure 4.3: Applying median threshold to MRI image for generating segmentation masks.

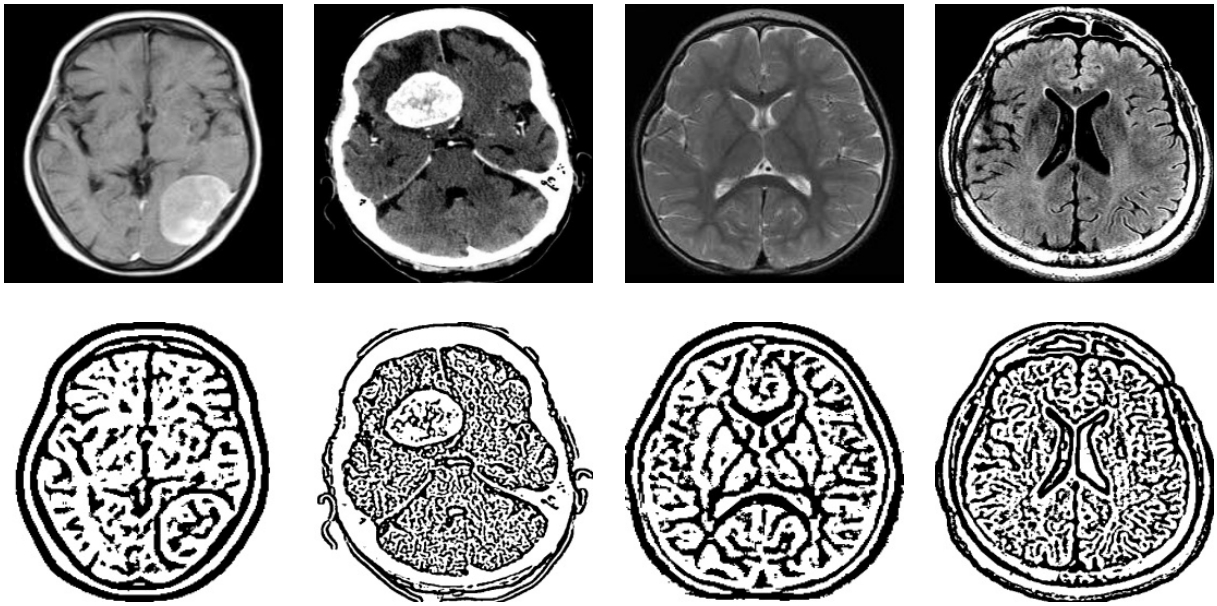

Figure 4.4: MRI Brain images before and after deploying median blur to create segmentation maps.

4.6 Training and Testing Strategies

To evaluate the performance and generalizability of the cGAN model, four distinct training and testing configurations were designed. These experiments evaluate the model's ability to reconstruct masks when trained and tested under the same conditions, as well as its capacity to translate between tumor and non-tumor masks across different domains.

Each configuration follows a structured naming convention: (Training Domain) - (Testing Domain), where "TT" denotes tumor-trained, "NTT" denotes non-tumor-trained, "TM" refers to tumor masks, and "NTM" refers to non-tumor masks. "NTITM" stands for non-tumor image with tumor mask, while "TINTM" stands for tumor image with non-tumor mask. The four configurations are as follows:

4.6.1 Tumor-Trained, Non-Tumor Image with Tumor Mask Tested (TT-NTITM)

In this configuration, the model was trained with brain tumor images paired with their corresponding masks (TT), and then tested on a separate set of brain non-tumor images paired with tumor masks (NTITM). The objective of this configuration was to investigate the model's response to non-tumor images when trained only on tumor images, highlighting its adaptability to cross-domain scenarios. Figure 4.5 shows the training and testing data used in this configuration for running the cGAN.

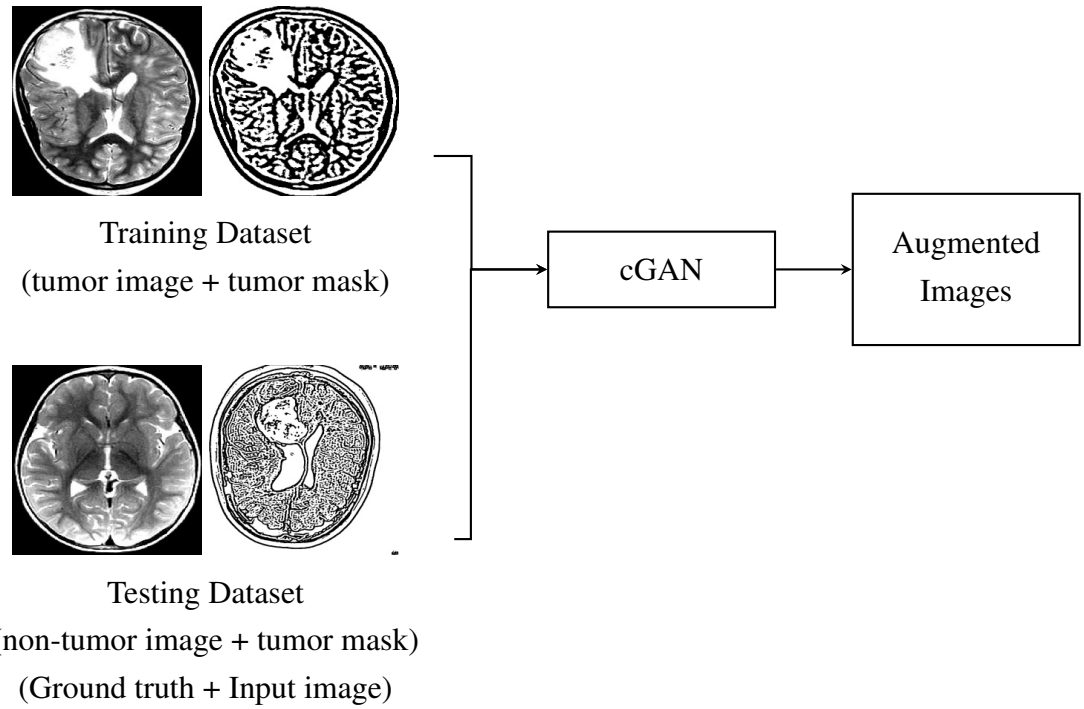

Figure 4.5: Working scheme of tumor-trained and non-tumor image with tumor mask tested (TT-NTITM) as inputs, processing through cGAN, and the output of the augmented image.

4.6.2 Tumor-Trained, Tumor Mask Tested (TT-TM)

In this configuration, the model was trained with brain tumor images paired with their corresponding masks (TT), while the testing was performed with a separate set of brain tumor images paired with their corresponding masks (TM). The objective of this configuration was to assess the model's performance in learning tumor image synthesis when exposed to the same data configuration. Figure 4.6 shows the training and testing data used in TT-TM configuration for running the cGAN model.

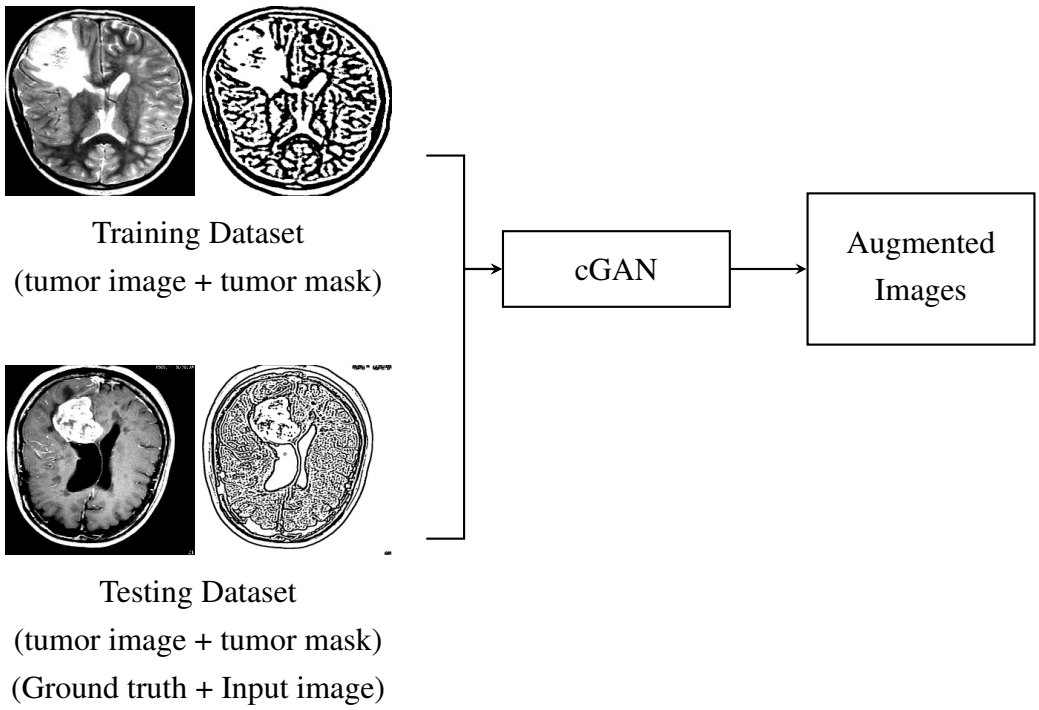

Figure 4.6: Working scheme of tumor-trained and tumor mask-tested (TT-TM)as inputs, processing through cGAN, and the output of the augmented image.

4.6.3 Non-Tumor-Trained, Non-Tumor Mask Tested (NTT-NTM)

In this experimental analysis, the model was trained with brain non-tumor images paired with their corresponding non-tumor masks and tested with a separate set of brain non-tumor images paired with their corresponding non-tumor masks. The goal was to assess the model's performance to learn non-tumor image synthesis when exposed to the same dataset configuration. Figure 4.7 shows the training and testing images used in this configuration for generating augmented images through cGAN model.

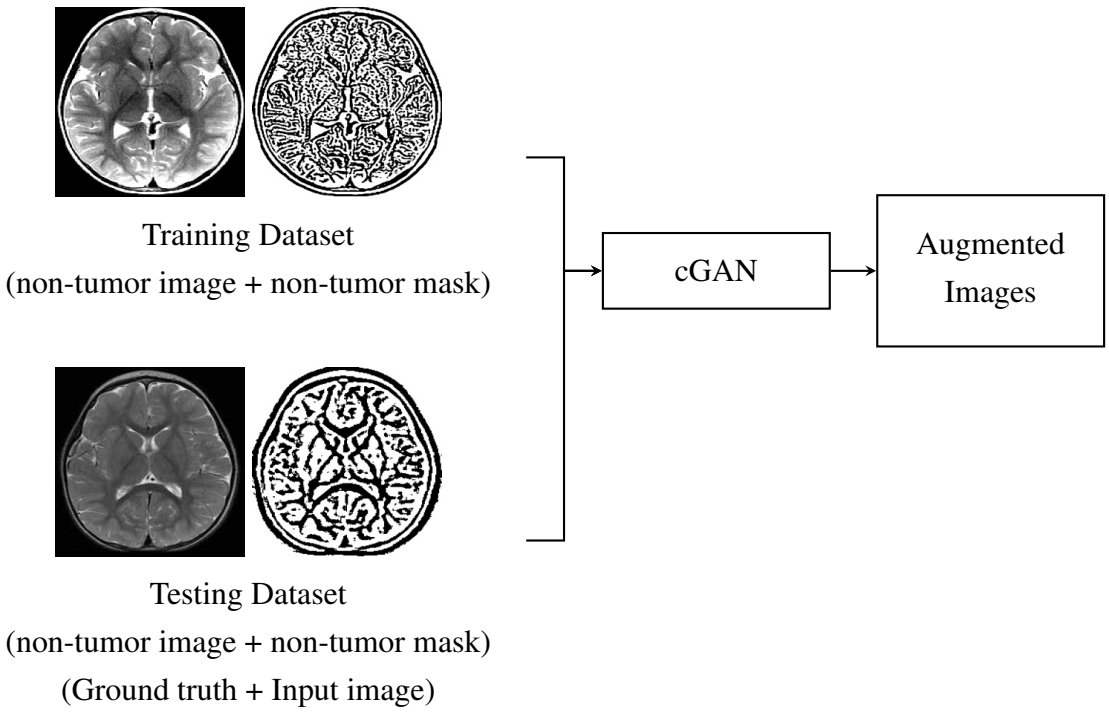

Figure 4.7: Working scheme of non-tumor-trained and non-tumor mask tested (NTT-NTM) as inputs, processing through cGAN, and the output of the augmented image.

4.6.4 Non-Tumor-Trained, Tumor Image with Non-Tumor Mask Tested (NTT-TINTM)

In this experimental analysis, the model was trained with brain non-tumor images paired with their corresponding non-tumor masks while testing was performed with a separate set of brain tumor images paired with non-tumor masks. The goal of this experiment was to investigate the model's response to tumor masks when trained only on non-tumor images, highlighting its adaptability to cross-domain scenarios. Figure 4.8 shows the training and testing images used in this configuration for generating augmented images through cGAN model.

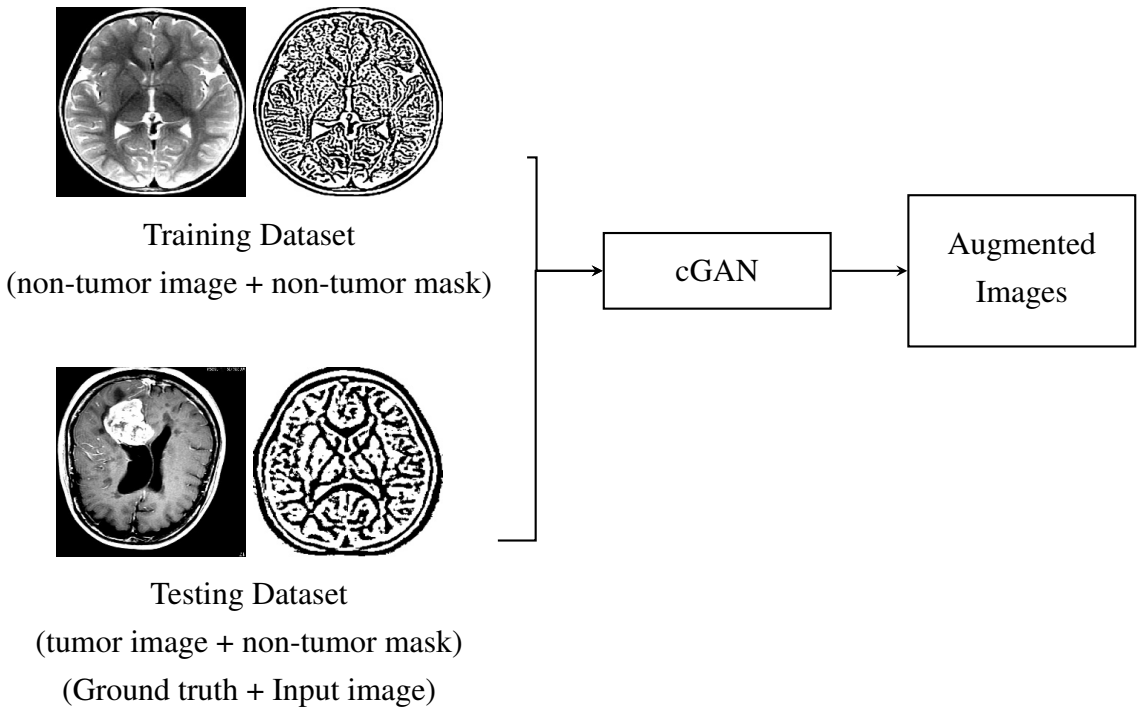

Figure 4.8: Working scheme of non-tumor-trained and tumor image with non-tumor mask tested (NTT-TINTM) as inputs, processing through cGAN, and the output of the augmented image.

These configurations provide structural evaluations of the cGAN model's ability to generalize to previously unseen data while maintaining its accuracy under domain shifts. The findings provide critical information about the model's resilience and flexibility in generating medical data. The validation and totally unseen dataset had an equal number of brain tumor and non-tumor images paired with their corresponding tumor and non-tumor masks.

4.7 cGAN Architectures

This subsection describes the architectural setup of the code for the image-to-image transition with a conditional generative adversarial network (cGAN) used in this thesis. It is developed using Python and Tensorflow and Keras. The architecture was modified through experimentation. This subsection provides a detailed examination of the network architecture, highlighting the key features of each part of the GAN model employed in the study.

4.7.1 Libraries

The cGAN implementation was carried out using Python, with key libraries including TensorFlow [1], NumPy [58], Matplotlib [36] (version: 3.8.4) and IPython.display (version: 8.22.2) [38] and pathlib (version: 3.10.14) [24].

4.7.2 Generator Architecture

The generator part of our cGAN consisted of a U-Net architecture for image-to-image translation, a common approach used in models like Pix2Pix [55]. The U-Net model is fundamentally

composed of two parts: an encoder and a decoder, connected by skip connections.

The encoder decreased the spatial dimension of the image while increasing the depth of feature maps. Many convolutional layers were used to extract features and reduce dimensions, followed by batch normalization and Leaky ReLU Activation. The batch normalization functioned by normalizing the output of the previous layer, subtracting the batch mean, and then dividing by the batch standard deviation, thereby improving training stability and convergence. At the same time, Leaky ReLU Activation allows a small, positive gradient when the unit is not active, unlike ReLU, which is strictly zero when its input has the value zero.

The decoder performed the reverse of a convolutional operation, increasing the spatial dimensions of the input feature maps. These convolutional layers were followed by batch normalization, dropout, and ReLU Activation. The dropout function randomly sets input units to 0 at each training step, which helps prevent overfitting.

Generator loss is measured through GAN loss and L1 loss, which measure generator loss and average absolute difference between the generated images and the real images. The GAN loss is measured using binary cross-entropy that aims to treat generated images as real images. Figure 4.9 displays the architecture of the generator used in cGAN.

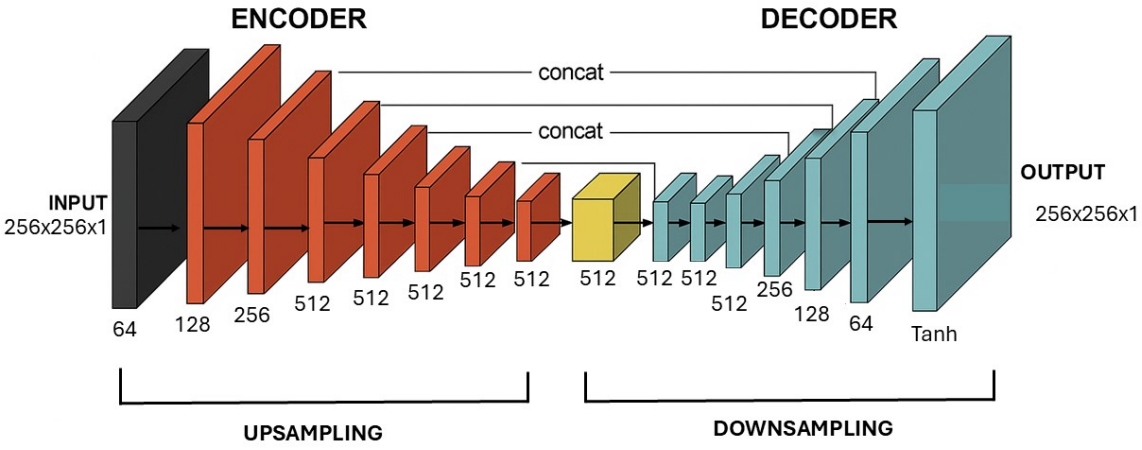

Figure 4.9: 3D visualization of the U-Net generator architecture used in pix2pix. The model takes a 256×256×1 grayscale input and passes it through 8 encoder blocks (red) with progressively increasing filter sizes, down to a bottleneck layer (yellow) of 512 filters. It then passes through 7 decoder blocks (teal), using transposed convolutions to upsample and concatenating skip connections from the corresponding encoder layers. The final output passes through a tanh activation to produce a 256×256×1 image. (Numbers along each block represent the number of feature channels (filters) at that layer)

4.7.3 Discriminator Architecture

The discriminator part of our cGAN used in this thesis, known as the PatchGAN discriminator, is designed to classify whether sections (or patches) of an image are real or fake. This approach focuses on the texture and local content of the image rather than the whole image itself.

Instead of classifying the entire image as real or fake, PatchGAN classifies patches of the image. Each output from the discriminator represents a 70x70 patch of the input image. This method helps the discriminator focus on finer details in the image, making it effective in tasks like image-to-image translation where local texture and structure are essential.

The discriminator receives two images as input: the real target image and the input image, which is combined with the generated image from the generator. These two inputs are concatenated along the channel dimension using tf.keras.layers.concatenate. This allows the discriminator to simultaneously consider the condition (input image) and the output (real or generated).

The discriminator is built using several layers of convolutions, each followed by batch normalization (except for the first layer if specified) and Leaky ReLU activation. The output shape of the last layer is designed to be (batch size, 30, 30, 1), where each 30x30 unit outputs a single value representing the classification of corresponding 70x70 patches of the input image.

Utilizing tf.keras.utils.plot_model the structure of the discriminator can be visualized. This shows the flow and transformation of input through the model and clarifies how inputs are processed through each layer.

The loss function of the discriminator was measured as real loss, generated loss, and total loss. Real loss was computed as a sigmoid cross-entropy loss between the discriminator's output for real images concatenated with their corresponding input images and a matrix of ones (indicating real), while generated loss was computed as a sigmoid cross-entropy loss between the discriminator's output for generated images concatenated with the same input images and a matrix of zeros (indicating fake). The total loss was the sum of the real loss and the generated loss, which the training procedure aims to minimize to improve the discriminator's ability to distinguish real images from fake ones. Figure 4.10 display the architecture of discriminator used in cGAN.

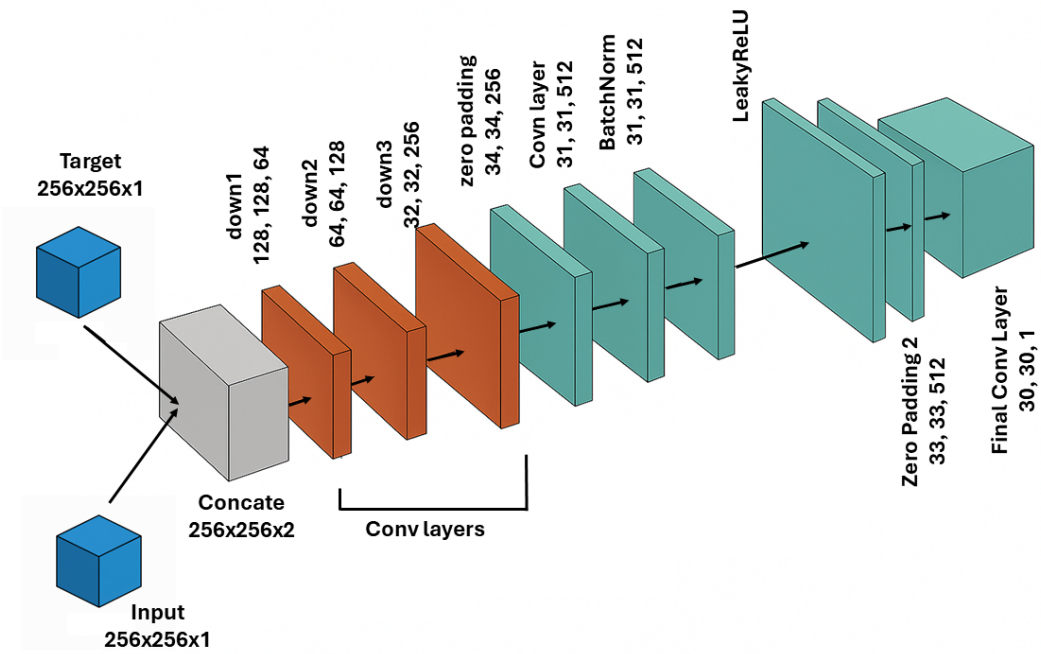

Figure 4.10: Modified architecture of the Discriminator in a Pix2Pix cGAN. (Numbers inside each block represent the number of feature channels (filters) at that layer)

4.8 Hyperparameters

Hyperparameters control the training dynamics and model behavior. These are crucial for achieving stable training and high-quality results. The various hyperparameters used in our model of this thesis are mentioned in Table 4.1.

Hyperparameters	**GAN Model**
Buffer Size	2000
Batch size	1
Epochs	50
Steps per epoch	2000
Learning Rate	2e-4
Adam Optimizer	0.5
Dropout Rate	0.5
Image Dimension of training data	256×256
Lambda	100
Channels	1
Kernel Initializer Standard Deviation	0.02

Table 4.1: Hyperparameters used in the conditional-Generative Adversarial Network code for generating augmented data.

4.8.1 Generated Images

The generated images function visualizes the results of the model during training. It takes three arguments: the model, which predicts the output image; test input for testing; and the target or ground truth image, again, for which the model's output is evaluated. The model is saved at checkpoints after every 20,000 steps to preserve training progress.

4.9 Data Postprocessing

The images generated from cGAN were saved. The segmentation maps of these images were created by applying a threshold as outlined in section 3.4. The process is shown in Figure 4.3.

4.10 Binary Classifier

U-Net, originally introduced by Ronneberger et al. [69] for medical image segmentation, is a lightweight CNN architecture consisting of two paths: the first reduces the image dimensions, and the latter increases them back to the original size. In this way, U-Net can first see the whole image at once to understand its context and then focus on the details required for accurate segmentation. However, in research by Hellström et al., [31], it was noted that the sole constricting path of a typical U-Net can be used to create an efficient CNN for classifying medical images based on the presence of cancer. Here, we used the encoder part of U-Net from [31]. The CNN consists of four sequences of two convolutional layers and one max pooling layer, followed by four dense layers. We proposed using stochastic gradient descent as the optimizer, with a learning rate of 0.001, and binary entropy as the loss function. During the training, 50% of the training data was used for validation. The number of epochs was set to 100. Different combinations of real and generated images were fed to the U-Net CNN for classification, and evaluation metrics, including accuracy, sensitivity, specificity, and AUC, were used to assess how well the U-Net CNN identifies brain tumor and non-tumor MRI images. Figure 4.11 displays the CNN architecture used for binary classification. Figure 4.12 shows the working mechanism of the U-Net CNN for the classification of real and generated images.

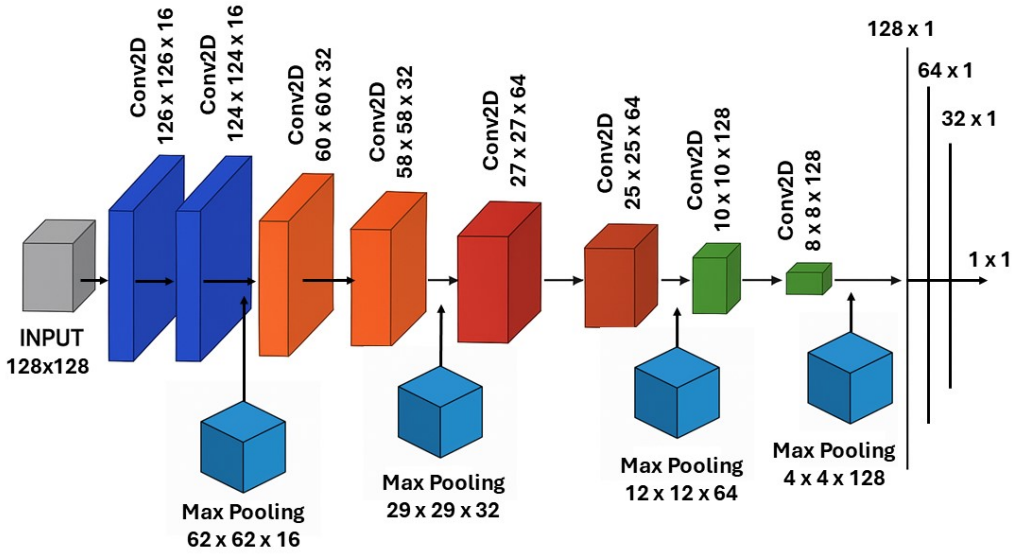

Figure 4.11: The architecture of Convolutional Neural Network (CNN) used for classifying images. One maximum pooling layer is added after every two convolutional layers. The last three layers before the 1×1 output layer are one-dimensional dense layers.

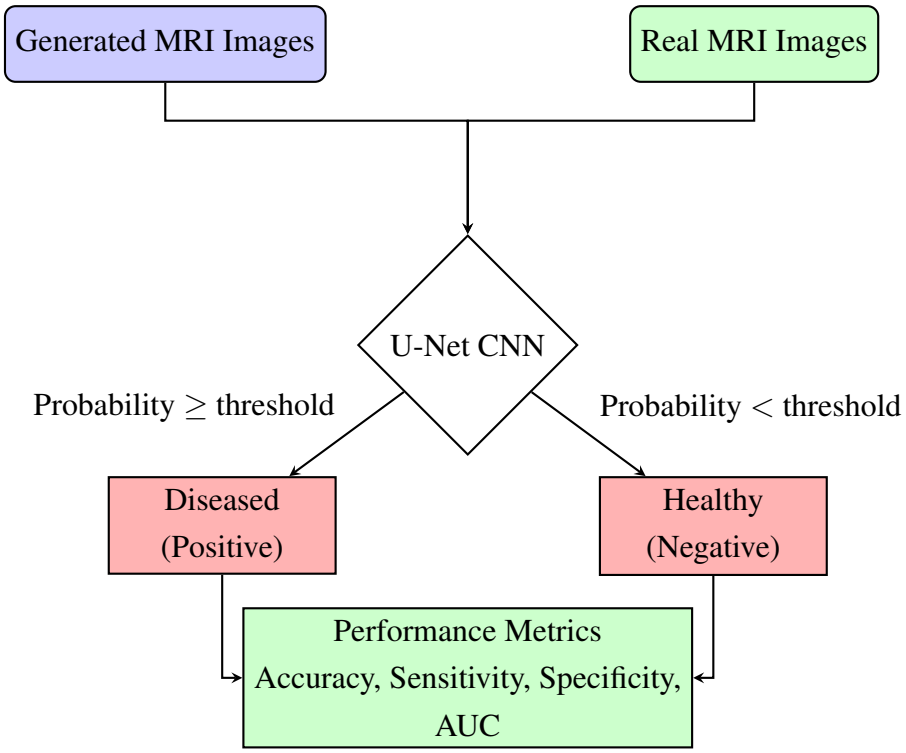

Figure 4.12: U-Net CNN classification of brain tumor and non-tumor images from generated data

4.11 Evaluation Metrics

The performance and efficiency of a model was assessed by evaluation metrics. These metrics evaluate the performance of the model, which is crucial to assess the quality of the model and to make improvements in it. A cGAN model generates synthetic data that is comparable to real-world data and evaluation metrics are a way to measure to its quality and assess how much generated data is different from the original data.

4.11.1 Inception Score

We calculated the inception score (IS) using Keras with Kullback-Leibler (KL) divergence (a statistical measure of how one probability distribution differs from a second, reference probability distribution). The generated images were loaded. All images were of the same size and configuration. We predicted the activations for KL divergence. These divergences were later used to calculate IS. We also computed the IS score for segmentation maps of generated images. The process is outlined in Figure 4.13.

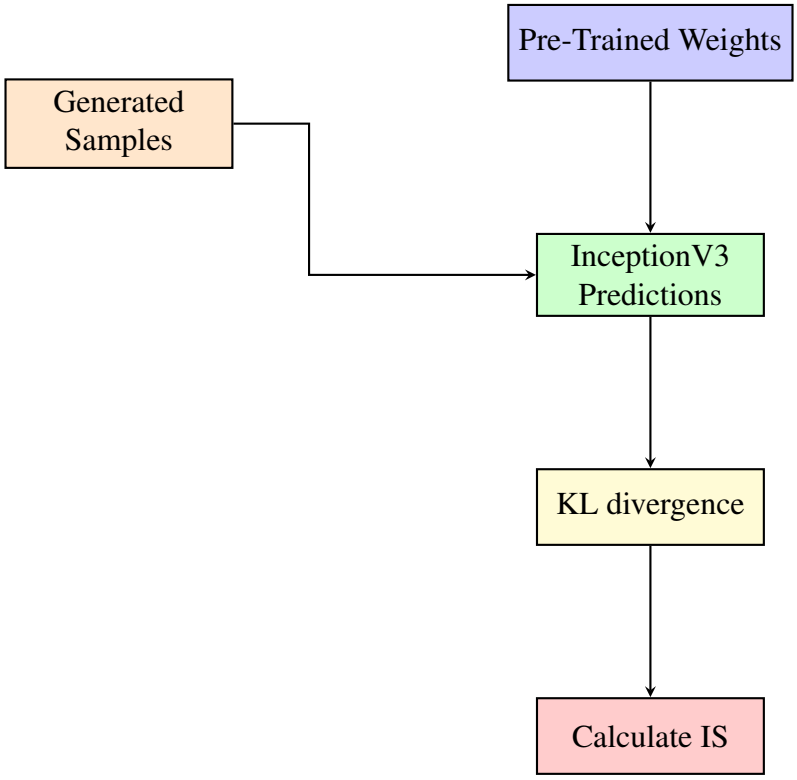

Figure 4.13: Calculating IS using InceptionV3 and pre-trained weights.

We calculated the IS by calculating the conditional class distribution and marginal class distributions of the generated images. We then calculated how much the conditional class distribution deviates from the marginal class distribution. The formula used is mentioned in equation (4.1).

$$\mathrm{IS} = \exp\left(\mathbb{E}_x\left[\mathrm{KL}\left(p(y|x) \parallel p(y)\right)\right]\right) \tag{4.1}$$

where $p(y|x)$ is conditional class distribution, $p(y)$ is marginal class distribution. KL is the KL divergence of the conditional and marginal class distributions. $\mathbb{E}_x$ is the expected value with respect to x.

4.11.2 Frechet Inception Distance

We calculated the FID score using Keras with inceptionV3 as a feature extractor. We obtained pre-trained weights from ImageNet for the model. The real images and generated images were loaded. All the images were the same size and configuration. We predicted on images to extract feature activations. These feature activations were later used to calculate the FID score. We also computed the FID score for segmentation maps of real and generated images. The process is outlined in Figure 4.14.

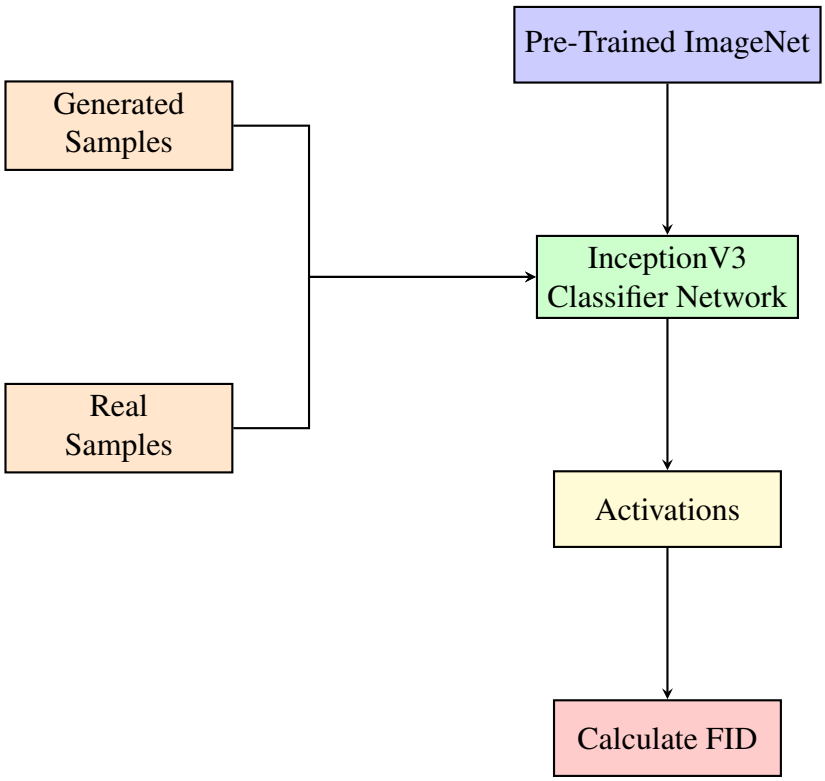

Figure 4.14: Calculating FID using ResNet-50 and ImageNet as pre-trained weights.

The FID was calculated using feature vectors of feature activations and covariance matrices of real and generated activations. We calculated the mean squared difference between the mean feature vectors of real and generated images. We then computed covariance matrices of these real and generated activations. The formula used is mentioned in equation (4.2).

$$\text{FID} = \|\mu_{real} - \mu_{gen}\|^2 + \text{Tr}\left(\Sigma_{real} + \Sigma_{gen} - 2(\Sigma_{real}\Sigma_{gen})^{1/2}\right) \tag{4.2}$$

where μ_{real} and μ_{gen} are mean feature vectors for real and generated images. Σ_{real} and Σ_{gen} are covariance matrices of real and generated feature vectors. Tr is the sum of diagonal elements, measuring the differences between distributions.

4.11.3 Structure Similarity Index Metric

We calculated the Structural Similarity Index Measure (SSIM) score using the OpenCV [60] and NumPy (version: 1.26.4) [58] libraries. The real and generated images were loaded. All images were of the same size and configuration. We predicted pixel intensities and covariance between real and generated images, which was used to calculate the SSIM. We also computed the SSIM score between segmentation maps of real and generated images.

SSIM was calculated using the mean of pixel intensity of real and generated images. We also calculated the covariance between real and generated images. We also computed deviations in pixel intensity of real and generated images. The formula used is mentioned in equation (4.3).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4.3}$$

where μ_x and μ_y are mean intensity values of real and generated images. σ_x and σ_y are variations in pixel intensities of real and generated images. σ_{xy} is the correlation of pixel patterns between real and generated images. C_1 and C_2 are constants.

4.11.4 Mean Squared Error

We calculated the mean squared error (MSE) score using the OpenCV and NumPy libraries. The real and generated images were loaded. All images were of the same size and configuration. We calculated the square of the differences between pixels of the real and generated images. We then computed the average of the squared differences of all pixels between real and generated images. We also computed the MSE score between segmentation maps of real and generated images. The formula used is mentioned in equation (4.4).

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(I_{\text{ref}}(i,j) - I_{\text{gen}}(i,j) \right)^2 \tag{4.4}$$

where $I_{ref}(i,j)$ and $I_{gen}(i,j)$ refers to pixels of real and generated images. M and N refer to the height and width of the images. In our case, M and N were 256×256 pixels image.

4.11.5 Peak Signal-to-Noise Ratio

We calculated the peak signal-to-noise ratio (PSNR) score using the OpenCV and NumPy libraries. The real and generated images were loaded. All images were of the same size and configuration. We determined the ratio of the maximum possible pixel value in the images to the MSE value of the same images. We also computed the PSNR score between segmentation maps of real and generated images. The formula used is mentioned in equation (4.5).

$$\text{PSNR} = 20 \cdot \log_{10} \left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right) \tag{4.5}$$

where MAX_I is the maximum possible pixel value in the image (255 for 8-bit images) and MSE is the mean squared error calculated between $I_{ref}(i,j)$ and $I_{gen}(i,j)$.

4.12 Performance Evaluation of the CNN Classification

The CNN performance was measured with and without augmented data, and a comparison of these results was made. We used Youden's threshold based on a prediction of the training data of a specified set of images to convert numerical predictions into binary labels [67]. The classification results of the U-Net CNN classifier were assessed in terms of the following metrics after each iteration.

4.12.1 Accuracy

Accuracy is the ratio of the number of correctly predicted instances (positive and negative) to the total number of predictions made. We calculated accuracy by calculating the ratio of the sum of true positive and true negative predictions, consisting of both tumor and non-tumor predictions, to total numbers of predictions. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) instances were calculated based on comparing generated and real pixels, and accuracy was calculated.

Mathematically, it is expressed as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.6}$$

where TP refers to pixels correctly classified as tumor positive, TN refers to pixels correctly classified as tumor negative, FP refers to pixels incorrectly classified as tumor positive, and FN refers to pixels incorrectly classified as tumor negative.

4.12.2 Sensitivity

Sensitivity is the ratio of true positive (tumor positive) predictions to the total number of predictions made. We calculated sensitivity by calculating the ratio of pixels correctly classified as tumor positive.

Mathematically, it is expressed as:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.7}$$

4.12.3 Specificity

Specificity is the ratio of true negative (tumor negative) predictions to the total number of predictions made. We calculated sensitivity by calculating the ratio of pixels correctly classified as tumor negative.

Mathematically, it is expressed as:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4.8}$$

4.12.4 Area Under the Curve (AUC)

Area under the curve (AUC) is the receiver operating characteristics (ROC) curve, which plots the true positive rate (sensitivity) (TPR) against the false positive rate (1- specificity) (FPR) for various threshold values. We calculated AUC by plotting sensitivity and specificity values as calculated previously.

Mathematically, it is given as:

$$\text{AUC} = \int_0^1 \text{ROC Curve}(x)\, dx \tag{4.9}$$

where ROC is a plot of TPR against FPR, and x is the threshold value for classification.

4.13 Statistical testing

We then computed the mean and the standard deviation of the classification accuracy of the test dataset over 20 iterations. We used the Mann-Whitney U test to compare the accuracy values obtained with training data sets that included and excluded cGAN-generated images. A significance level of 0.05 was used as the threshold to identify statistically significant differences.

5 Results

This chapter describes the results of training the cGAN model. These results include values of the evaluation metrics used to test the quality of generated images and the performance evaluation of CNN classification with generated images. It also includes statistical testing used to assess CNN classification.

5.1 Generated Images

After encountering memory issues with high-resolution images, we made a cGAN model to run for 256×256 image resolution, at 50 epochs. The cGAN model was run four times to generate images with four different configurations, as mentioned in subsection 3.6. The hyperparameters used in running the cGAN were the same for all configurations except for different datasets used for training and testing the model. The images generated from 4 different configurations of the cGAN model and their corresponding masks are shown in Figure 5.1, 5.2, 5.3, and 5.4. The masks of these synthetic images were generated as mentioned in subsection 3.8. All generated images and their masks were of 256×256 image resolution.

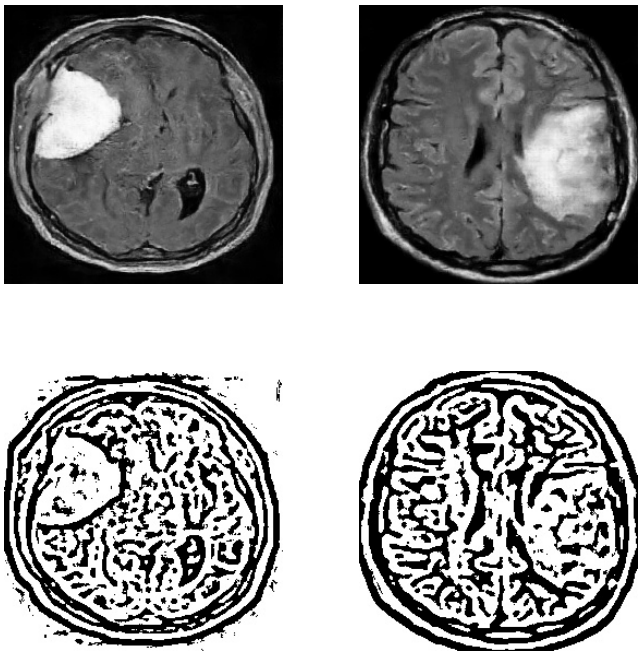

Figure 5.1: Images and their corresponding segmentation maps generated from TT-TM configuration used for cGAN training and testing.

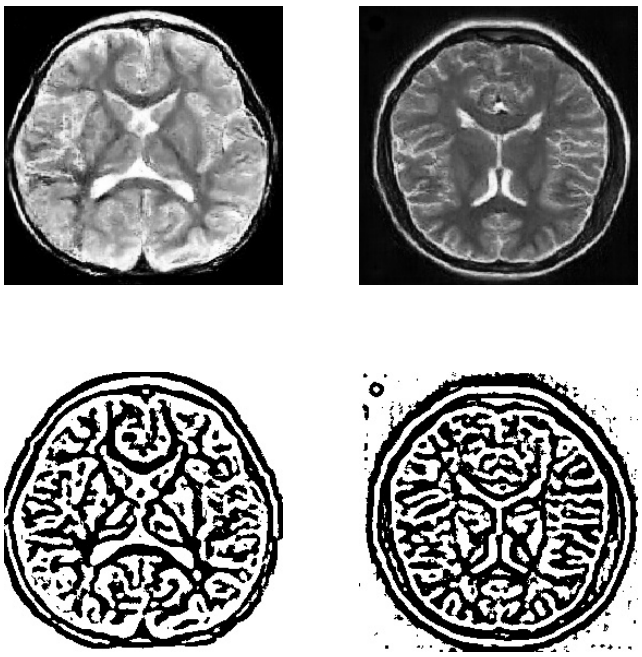

Figure 5.2: Images and their corresponding segmentation maps generated from NTT-NTM configuration used for cGAN training and testing.

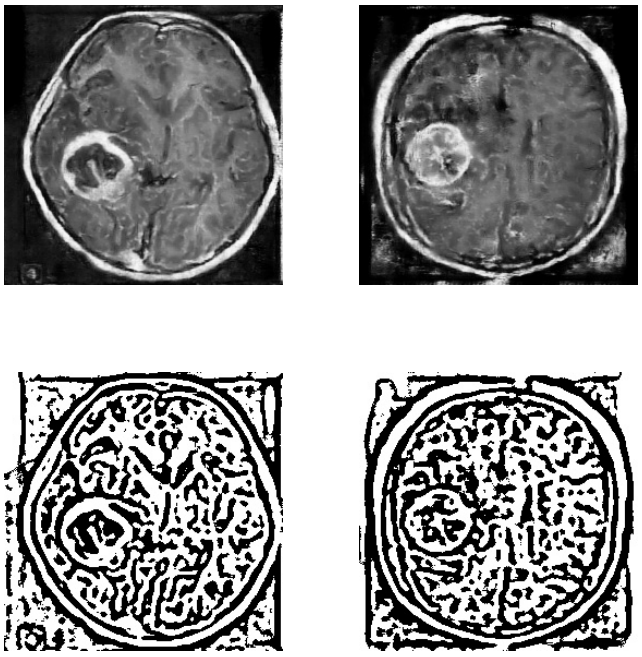

Figure 5.3: Images and their corresponding segmentation maps generated from the TT-NTITM configuration used for cGAN training and testing.

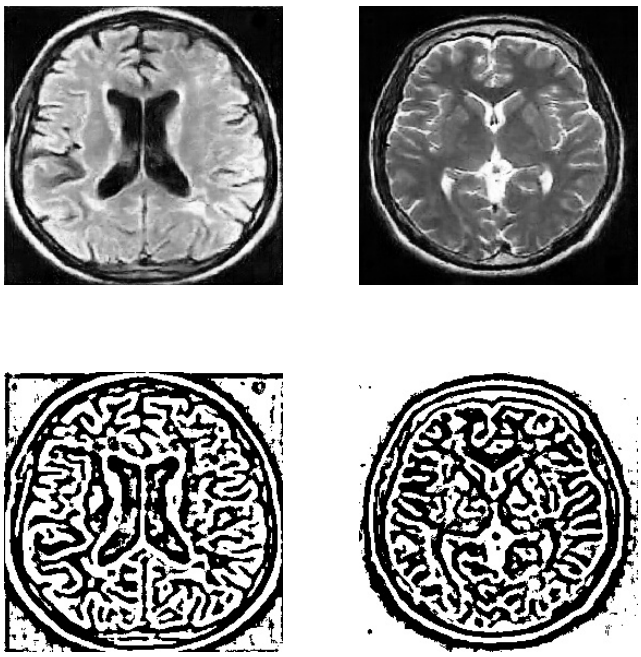

Figure 5.4: Images and their corresponding segmentation maps generated from NTT-TINTM configuration used for cGAN training and testing.

5.2 Evaluation Metrics

5.2.1 TT-TM and NTT-NTM

The inception score was calculated for images generated through TT-TM and NTT-NTM configurations and their corresponding masks, and the results are given in Table 5.1. As expected, the real images give the highest IS (2.0$\pm$0.08 and 2.1$\pm$0.06), demonstrating real medical data's natural variability and quality. Among the generated data, the TT-TM configuration gave an IS of 2.09$\pm$0.11, and TT-TM mask gave an IS of 2.06$\pm$0.1, while the NTT-NTM gave an IS score of 2.01$\pm$0.11 and NTT-NTM mask gave an IS of 1.82$\pm$0.07. Both these IS are close to real medical data IS (2.1$\pm$0.06).

Image Configuration	Inception Score
Real tumor positive image	2.0$\pm$0.08
Real tumor negative image	2.1$\pm$0.06
TT-TM	2.09$\pm$0.11
TT-TM mask	2.06$\pm$0.10
NTT-NTM	2.01$\pm$0.11
NTT-NTM mask	1.82$\pm$0.07

Table 5.1: Mean $\pm$ standard deviation values of Inception score (IS) for images generated from TT-TM and NTT-NTM configuration and their masks.

The SSIM and MSE values were calculated for images generated from TT-TM and NTT-NTM configurations and their corresponding masks, and the results are given in Table 5.2. The

real medical data have SSIM and MSE values of 1.0 and 0.0, respectively, showing perfect structure similarity and no error. Among the generated data, the TT-TM configuration has an SSIM of 0.16, and the TT-TM mask had an SSIM of 0.1, while the NTT-NTM configuration has a similar SSIM of 0.17, and the NTT-NTM mask had an SSIM of 0.12. The PSNR values for masks are shown in Table 5.3.

Image Configuration	**SSIM**	**MSE**
Real images	1.0	0.0
TT-TM	0.16	7609
TT-TM mask	0.1	24117
NTT-NTM	0.17	5708
NTT-NTM mask	0.12	24145

Table 5.2: Structure Similarity Index (SSIM) and Mean Squared Error (MSE) for images generated from TT-TM and NTT-NTM configuration and their masks.

The PSNR was calculated for images generated from TT-TM and NTT-NTM configurations and their corresponding masks, and the results are shown in Table 5.3. We calculated the average, highest, and lowest PSNR values of our generated data. The real images yielded infinite PSNR values when compared to themselves, reflecting perfect quality and no noisy images. Among synthetic images, the TT-TM configuration achieved the average PSNR of 28.78, with the highest PSNR at 31.73 and the lowest at 27.25. The NTT-NTM configuration achieves an average PSNR of 28.74, with the highest PSNR of 32.45 and the lowest of 27.26.

Image Configuration	**Avg. PSNR**	**Highest PSNR**	**Lowest PSNR**
Original pos img	Infinity	Infinity	Infinity
Original neg img	Infinity	Infinity	Infinity
TT-TM	28.78	31.73	27.25
TT-TM mask	31.26	33.79	28.58
NTT-NTM	28.74	32.45	27.26
NTT-NTM mask	31.78	34.08	29.06

Table 5.3: Peak Signal-to-Noise Ratio (PSNR) for images generated from TT-TM and NTT-NTM configuration and their mask.

The FID and DSE were calculated for images generated from TT-TM and NTT-NTM configurations, and the results are shown in Table 5.4. Images generated from TT-TM configurations had FID of 64, while images generated from NTT-NTM configurations had FID of 31, indicating better similarity to real data. Despite differences in FID, the DSE score was similar for both configurations (0.89 ± 0.07 and 0.89 ± 0.05). The FID and DSE values for generated masks are given in Table 5.4.

Image Configuration	FID	DSE
TT-TM	64	0.89±0.07
TT-TM mask	47.92	0.84±0.03
NTT-NTM	31	0.89±0.05
NTT-NTM mask	40.11	0.83±0.03

Table 5.4: Frechet Inception Distance (FID) and Mean ± standard deviation values of DSE for images generated from TT-TM and NTT-NTM configuration and their mask.

5.2.2 TT-NTITM and NTT-TINTM

The IS was calculated for generated images through the TT-NTITM and NTT-TINTM configurations and their corresponding masks, and the results are given in Table 5.5. The results showed that the images generated through the TT-NTITM configuration provide the IS of 1.6±0.04. However, NTT-TINTM configuration shows an IS score (1.9±0.06) close to the real data (reference to Table 5.1: 2.1±0.06). Table 5.5 shows the IS for generated masks.

Image Configuration	Inception Score
TT-NTITM	1.6±0.04
TT-NTITM mask	1.82±0.07
NTT-TINTM	1.9±0.06
NTT-TINTM mask	1.86±0.1

Table 5.5: Mean ± standard deviation values of Inception score for real images and images generated from cGAN through TT-NTITM and NTT-TINTM configuration and their mask.

The SSIM and MSE were calculated for real images and images generated from TT-NTITM and NTT-TINTM configurations and their corresponding masks, and the results are given in Table 5.6. The results showed that the NTT-TINTM configuration showed a higher SSIM (0.18) compared to the TT-NTITM configuration.

Image Configuration	SSIM	MSE
TT-NTITM	0.16	6569.3
TT-NTITM mask	0.08	25068
NTT-TINTM	0.18	5545.94
NTT-TINTM mask	0.11	24262

Table 5.6: Structure Similarity Index (SSIM) and Mean Squared Error (MSE) for real images and images generated from cGAN through TT-NTITM and NTT-TINTM configuration.

The PSNR was calculated for real images and images generated from TT-NTITM and NTT-TINTM configurations and their corresponding masks, and the results are shown in Table 5.7. We calculated the average, highest, and lowest PSNR values of our generated data. In the TT-NTITM configuration, the generated images yielded an average PSNR of 28.72, with the

highest PSNR of 32.41 and the lowest PSNR of 26.94. The NTT-TINTM configuration yielded images with an average PSNR of 28.57, a highest PSNR of 29.81 and a lowest PSNR of 27.27.

Image Configuration	Avg. PSNR	Highest PSNR	Lowest PSNR
TT-NTITM	28.72	32.41	26.94
TT-NTITM mask	31.24	33.60	28.59
NTT-TINTM	28.57	29.81	27.27
NTT-TINTM mask	31.78	34.17	29.03

Table 5.7: Peak Signal-to-Noise Ratio (PSNR) for real images and images generated from cGAN through TT-NTITM and NTT-TINTM configuration and their mask.

The FID and DSE scores were calculated for images generated from TT-NTITM and NTT-TINTM configurations and their corresponding masks, and the results are shown in Table 5.8. The results showed that the TT-NTIM gave FID and DSE of 69.78 and 0.92$\pm$0.04 respectively, while the NTT-TINTM configuration gave FID and DSE of 40.18 and 0.90 $\pm$0.07 respectively.

Image Configuration	FID	DSE
TT-NTITM	69.78	0.92$\pm$0.04
TT-NTITM mask	79.577	0.82$\pm$0.02
NTT-TINTM	40.18	0.90 $\pm$0.07
NTT-TINTM mask	44.17	0.83$\pm$0.03

Table 5.8: Frechet Inception Distance (FID) and Mean $\pm$ standard deviation values of DSE for images generated from TT-NTITM and NTT-TINTM configuration and their mask.

5.3 Performance Evaluation of Binary Classification

5.3.1 TT-TM and NTT-NTM

Different numbers of original and generated MRI images were tested for the classification accuracy of the modified U-Net CNN. The results are shown in Table 5.9. Here, images generated from TT-TM and NTT-NTM configurations were tested.

		Original images		
		10	500	1000
cGAN images	0	0.57±0.09	0.88±0.03	0.88±0.04
	250	0.67±0.07	0.77±0.14	0.73±0.01
	500	0.62±0.03	0.85±0.04	0.88±0.05
	750	0.72±0.05	0.79±0.05	0.88±0.05
	1000	0.74±0.05	0.81±0.05	0.89±0.04
	1250	0.70±0.03	0.82±0.04	0.90±0.03
	1500	0.74±0.04	**0.88±0.04**	**0.92±0.03**
	1750	0.74±0.03	0.83±0.04	0.89±0.03
	2000	**0.76±0.04**	0.85±0.04	0.89±0.03

Table 5.9: Mean ± standard deviation values for the accuracy over 20 iteration rounds when the CNN is trained by using a dataset consisting of the specified numbers of original images and synthetic images created by the cGAN. (images generated from TT-TM and NTT-NTM configuration). Bold digits in the table indicates the highest accuracy reached for the given number of real images.

The mean accuracy values from Table 5.9 were tested using the Mann-Whitney U test to determine the statistically significant differences. The comparison was made between mean accuracy values for a specified number of images as shown in Table 5.10. The actual values are in the supplementary Table .1.

		Original images		
		10	500	1000
cGAN images	250	***	***	***
	500	***	**	
	750	***	***	
	1000	***	***	
	1250	***	***	*
	1500	***		**
	1750	***	***	
	2000	***	***	

Table 5.10: The p-values of the Mann-Whitney U test comparing the classification accuracies of the CNNs trained with training data containing both the specified numbers of original images and synthetic cGAN images, compared to the CNN trained with the same number of original images but no cGAN images. Significance levels: * $p \leq 0.05$, ** $p \leq 0.01$, *** $p \leq 0.001$, no symbol: $p > 0.05$.

We calculated AUC for our data generated from TT-TM and NTT-NTM configurations, and the results are plotted in Figure 5.5. The graph was plotted between AUC values (y-axis) to number of times AUC values (x-axis) were calculated. Each line corresponds to a number of original images, and each dot corresponds to a number of generated images that were used. Both the number of original and generated images are in accordance with Table 5.9.

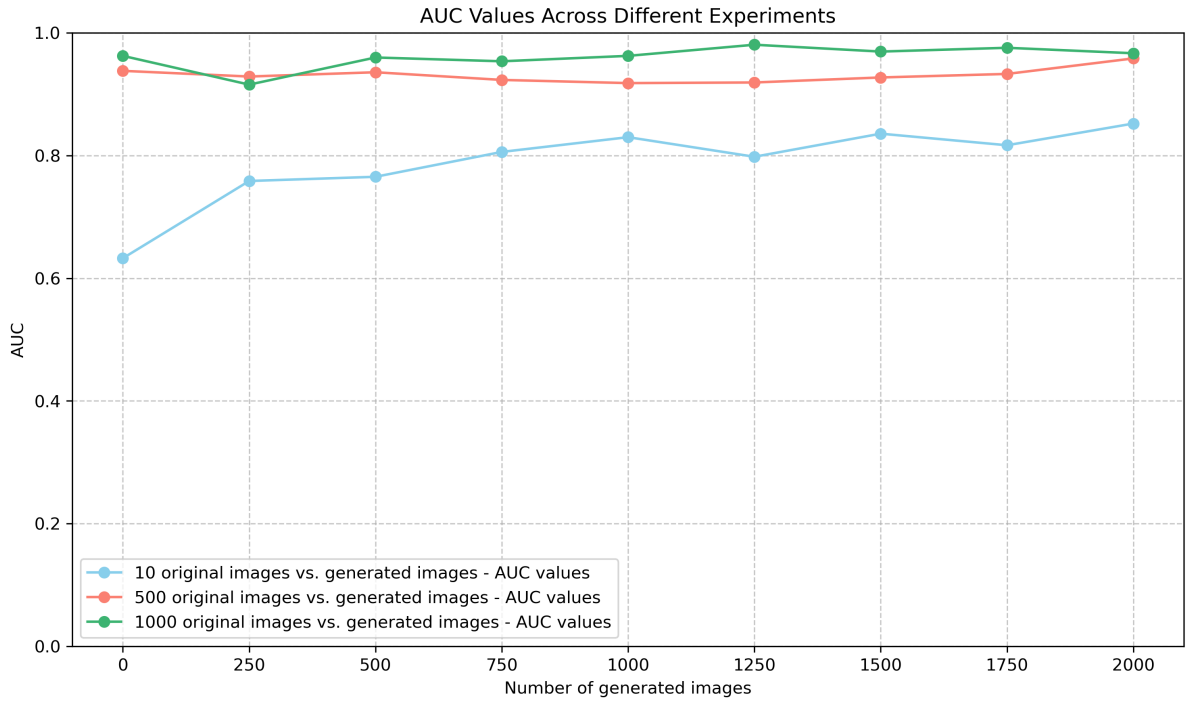

Figure 5.5: AUC plot comparing the performance of three different experiments. (10 original vs. generated images, 500 original vs. generated images, 1000 original vs. generated images). The third experiment (1000 original vs. generated images) outperforms the others. (Images generated from TT-TM and NTT-NTM configurations were used.)

5.3.2 TT-NTITM and NTT-TINTM

Different numbers of original and generated MRI images were tested for the classification accuracy of the modified U-Net CNN. Table 5.11 demonstrates the model performance with and without augmented data. Here, images generated from TT-NTITM and NTT-TINTM configurations were tested.

		Original images		
		10	500	1000
cGAN images	0	0.59±0.1	**0.88±0.04**	0.88±0.04
	250	0.53±0.07	0.81±0.09	0.79±0.07
	500	0.58±0.08	0.82±0.03	0.87±0.04
	750	0.64±0.04	0.83±0.03	0.88±0.04
	1000	0.65±0.04	0.83±0.03	**0.89±0.04**
	1250	0.62±0.06	0.82±0.03	0.88±0.03
	1500	**0.66±0.06**	0.83±0.02	0.88±0.03
	1750	0.66±0.05	0.83±0.05	0.89±0.02
	2000	**0.66±0.06**	0.82±0.03	0.87±0.03

Table 5.11: Mean ± standard deviation values for the accuracy over 20 iteration rounds when the CNN is trained by using a dataset consisting of the specified numbers of original images and synthetic images created by the cGAN. (images generated from TT-NTITM and NTT-TINTIM configuration). Bold digits in the table indicates the highest accuracy reached for the given number of real images.

The mean accuracy values from Table 5.11 were tested using the Mann-Whitney U test to determine statistically significant differences. The comparison was made between mean accuracy values for a specified number of images as shown in Table 5.12. The actual values are in the supplementary Table .2.

		Original images		
		10	500	1000
cGAN images	250		***	***
	500		***	
	750		***	
	1000	*	***	
	1250		***	
	1500	*	***	
	1750		***	
	2000	**	***	

Table 5.12: The p-values of the Mann-Whitney U test comparing the classification accuracies of the CNNs trained with training data containing both the specified numbers of original images and synthetic cGAN images, compared to the CNN trained with the same number of original images but no cGAN images. Significance levels: * $p \leq 0.05$, ** $p \leq 0.01$, *** $p \leq 0.001$, no symbol: $p > 0.05$.

We calculated the AUC for our data with the TT-NTITM and NTT-TINTM configurations. The results are plotted in Figure 5.6. The graph was plotted between AUC values (y-axis) and the number of times AUC values (x-axis) were calculated. Each line corresponds to a number

of original images, and each dot corresponds to a number of generated images that were used. Both the number of original and generated images are in accordance with Table 5.11.

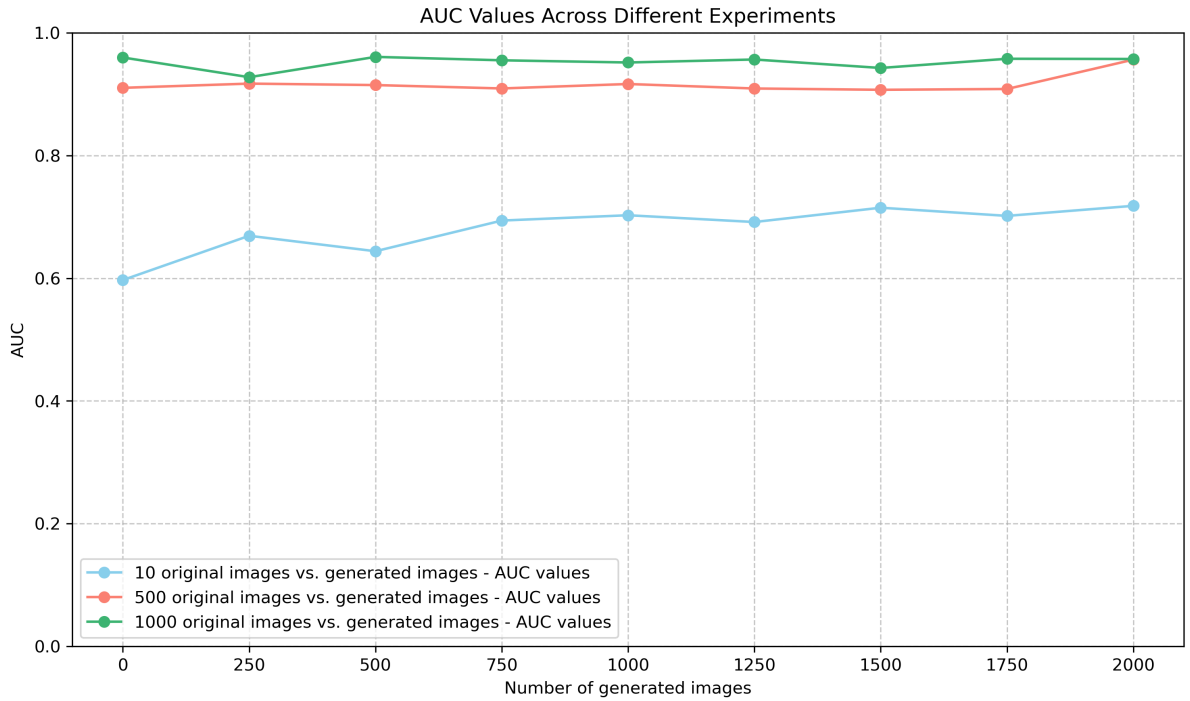

Figure 5.6: AUC plot comparing the performance of three different experiments. (10 original vs. generated images, 500 original vs. generated images, 1000 original vs. generated images.) The third experiment (1000 original vs. generated images) outperforms the others. (Images generated from TT-NTITM and NTT-TINTM configurations were used.)

We calculated sensitivity and specificity for our generated data and the results are mentioned in Table 5.13. This table shows that the maximum sensitivity and maximum specificity of our generated data is 98% and 94% respectively, indicating better image quality.

Image Configuration	Sensitivity	Specificity
TT-TM & NTT-NTM	0.98±0.01	0.94±0.03
TT-NTITM & NTT-TINTM	0.95±0.03	0.94±0.02

Table 5.13: The sensitivity and specificity of images generated through given four configurations.

6 Discussion

MRI imaging offers in-depth insights into various aspects of brain health, including brain structures, function, disorders (brain tumor, dementia, stroke, etc.), and brain blood flow dynamics. The brain MRI holds significant importance in clinical diagnosis and therapeutic interventions. Notably, deep-learning-based classification models stand out for their ability to classify images and facilitate the clinical assessment of patient health. However, the limited publicly available dataset pose a significant challenge in testing the classification models, impeding the application in the real world. As such, artificial generation of augmented data with high image quality is critical.

This thesis aims to investigate the impact of augmented data on the performance of a binary classifier for disease classification. The primary objective was to build and train a cGAN algorithm to synthesize brain MRI tumor images and explore the effect of a number of augmented data on the binary classifier. This work contributes to the growing body of generative AI in medical imaging, and proposes deep learning-based models to generate synthetic data that improves brain tumor classification tasks.

The image generation through four different configurations highlight the influence of training data on the quality and effectiveness of generated images. These experiments tested the model's performance when exposed to same dataset configuration and for cross-domain image synthesis when a mismatched training and testing dataset was used. When the cGAN was trained with tumor images and tested separately with tumor and non-tumor images, it introduced tumors in the generated images. This outcome highlights the model's ability to generalize learned tumor features and add them to unseen images. The generated images showed the tumor features embedded into images, suggesting the incorporation of tumor-related characteristics during training. This behavior confirms the model's cross-domain scenarios, transferring tumor features into the non-tumor image. Conversely, when the model was trained with non-tumor images and tested separately with tumor and non-tumor images, it did not introduce any tumor features in the generated images. Instead, the model learn non-tumor features and transfers the learned features to generate non-tumor images even when tested with tumor images. The observed response indicates the model's performance across cross domains and highlights the model's potential to learned the training features and generate augmented data.

Qualitative metrics such as IS, SSIM, MSE, PSNR, FID, and DSE mutually provides detail assessment of visual realism and structure similarity to real data. Regarding IS, the real images - both tumor-positive and tumor-negative - achieved high values (2.0 and 2.1, respectively), reflecting the inherent image quality and realism. The TT-TM and NTT-NTM generated images yielded comparable IS (2.09 and 2.01, respectively), demonstrating the ability of TT-TM and NTT-NTM to produce realistic and diverse images, proving their value for data augmentation and simulation tasks in medical imaging workflow. Conversely, the TT-NTITM and NTT-TINTM generated images yielded comparatively low IS (1.6 and 1.9, respectively), suggesting diminished diversity and realism in generated data. The masks of generated images give the same IS as their corresponding images, indicating the model's performance in retaining

generative diversity.

The SSIM and MSE values further affirm these trends. The real images, as expected, show a perfect SSIM of 1.0 and MSE of 0.0. As for augmented data, the NTT-NTM and NTT-TINTM generated images gave slightly better SSIM (0.17 and 0.18, respectively), depicting moderate structural fidelity. MSE values for all generated images were constantly higher than real data, highlighting the pixel-level dissimilarity. Mask-based evaluations consistently gave low SSIM and high MSE, reflecting that the condition alone on the mask may lead to the loss of finer structural details.

The PSNR results complement the MSE trends. The real images returned infinity PSNR values as expected with zero reconstruction error. Among generated data, the mask-based analysis (for example, TT-TM and NTT-NTM masks) gave high PSNR (average PSNR: 31.78 and highest PSNR: 29.06) than their non-mask counterparts, suggesting that while the overall image brightness and noise level may correspond with real images, the spatial structures may still diverge.

The FID further gives arguments for testing the image quality. The TT-NTM configuration gave the lowest FID (31), suggesting that the generated data distribution is close to real data in the feature space. This is in contrast to where TT-NTM gave lower IS (2.01), suggesting that while it matches the feature distribution, it might lack class-wise diversity. Conversely, TT-NTITM gave the highest FID (69.78), implying weak alignment with real data feature distribution, which aligns with its lowest IS (1.6) and SSIM (0.16).

Lastly, the DSE, which reflects the alignment of real and generated data in terms of task performance, further facilitates these findings. Most configuration gave DSE in range 0.83 - 0.92 with TT-NTITM suprassing the highest DSE (0.92), suggesting that even with low image quality, this configuration may still produce beneficial representations for the downstream classifier.

The modified U-Net CNN performance is shown in Table 5.9 and 5.11 to depict the model's accuracy with the augmented data, achieving an accuracy of 92% for the CNN classification. The Figure 5.5 and 5.6 display AUC curves across the three experiments, indicating strong performance of the classifier with more augmented data. Finally, Table 5.13 showcases the maximum sensitivity and specificity the model could achieve with the augmented data.

The peak performance achieved can be attributed to the hyperparameters used for training the model, including the learning rate, Adam optimizer, dropout rate, kernel initializer standard deviation, lambda, buffer size, batch size, and the number of epochs, along with steps per epoch. In case of TT-TM and NTT-NTM configurations, with 10 original images, the model accuracy was as low as 57% which refers to the low diversity of the dataset used. By adding generated data, the model's accuracy increases. For example, after adding 1000 generated images, the model's performance improved up to 74%. With 1000 real data, the model reached up to 92% of accuracy when tested with 1500 original images. This shows the performance of the CNN classifier when diversity is introduced to the model as the number of generated images increases.

In case of TT-NTITM and NTT-TINTM configurations, as expected, the training with 10

real images and no generated data exhibits a low accuracy of 59% which shows low performance of the model with limited real data. With 1500 images added to real data, the model performance reached 66%, demonstrating the positive impact of adding cGAN-generated data. As the number of generated images increased, the model's performance increased, achieving 89% accuracy when 1000 generated images were added to 1000 real images. After 1000 images, the plateau of accuracy was steady with accuracy at about 87-89 %, suggesting the model's ability to correctly classify the images. These results indicated that the synthetic images generated from cGAN, regardless of cross-domain data generation, can effectively augment the data and improve the classification performance of CNN. This supports using augmented data for model training in data-scarce medical scenarios. These findings suggest that the model's classification is potentially sensitive to augmented data. These results underscore the careful selection of image-mask configuration in generative models training to ensure that the generated data effectively contributes to the downstream task while maintaining the image quality and diversity.

The results from Table 5.10 show the behavior of significant differences between real and generated images. With a small number of original images (10 images), there were significant differences when the generated images were added. The significant difference was also noted with 500 original images, except for 1500. In the case of 1000 original images, the results were mixed. The statistically significant differences were observed only with a few generated images (250, 1250, 1750). This highlights the effect of cGAN augmentation in addressing data scarcity, where synthetic images provided meaningful improvements in model performance.

The results from Table 5.12 were mixed for an extremely low number of real data points (10 original images) in the data regimen. The presence of significant differences in multiple configurations (1000, 1500, 2000) shows that the generated data often led to significant improvements when a large amount of generated data was used. Interestingly, with 500 original images, adding cGAN-generated images showed statistically significant improvements in most cases. In contrast, no significant difference was noted with 1000 original images across all generated image counts, except for testing with 250 generated images. Although no correction for multiple comparisons was applied, the p-values were sufficiently small to suggest that the observed significance is unlikely to be due to chance alone.

The graphs of AUC show the improvement in AUC with an increase in the number of generated images, proving the meaningful performance of augmented data. This graph in Figure 5.5 shows that our model achieved a maximum AUC of 98%, indicating that the model performs better in classification with augmented data. The other graph in Figure 5.6 shows a positive correlation between the number of images and the resulting AUC performance. The gap between 10 original images and 500-1000 original images is noticeable, highlighting the limitation in the model's performance with a small dataset. The model's performance improved in terms of AUC with a high number of generated images.

In addition, the classification performance of CNN was evaluated using sensitivity (true positive rate) and specificity (true negative rate), which is important to configure in medical

imaging to understand how well the model classifies the brain tumors and avoid false alarms. All four configurations of image generation yielded high sensitivity and specificity. The TT-TM and NTT-NTM configuration gave sensitivity of 98% and specificity of 94%, while the TT-NTITM and NTT-TITNM configuration gave sensitivity of 95% and specificity of 94%. These high values indicate the model's performance in correctly classifying brain tumors and minimizing incorrect tumor predictions. This suggests that the cGAN-generated tumor and non-tumor data, when well aligned with the image-mask, boost the model performance and detection capabilities, reinforcing the viability of synthetic data augmentation for medical image classification tasks.

Initially, the model crashed at high epochs because of the high computational demand for generating 256×256 images. But when run on 50 epochs, we successfully generated some realistic-looking images. All image configurations demonstrated the capability to generate good-quality images, with the TT-TM image-mask pair outperforming with a favorable IS score.

7 Limitations

One notable limitation of this study is that the training data was minimal in terms of anatomical diversity, tumor type, and patient demographics, leading the model to learn from a given dataset. This restricted the generalizability of the findings and led the model to perform well only on images similar to the training set used.

The generated images were not evaluated by any medical professional (radiologist or oncologist). As a result, it was unclear if generated images were diagnostically useful. Visual quality inspection solely based on evaluation metrics may not reflect the clinical relevance.

The primary limitation of cGAN is its computational complexity. The combination of segmentation maps and ground truth for model training requires memory and computational resources. This restricts the application of our model for disease classification in institutions with limited access to advanced hardware and impedes the real-world processing capability of cGAN.

8 Conclusions and Future Work

The importance of early cancer classification cannot be denied, primarily considering the enhanced patient outcomes and improved survival rates. This thesis was initiated with the central question: How can augmented data obtained from conditional generative adversarial networks affect the performance of the binary classifier for medical image datasets? The answer to this question has been explored extensively in this thesis with a focus on building a deep-learning model for generating augmented data, amplifying the limited real-world datasets, and then implementing this augmented data to find the accuracy of the binary classifier.

This thesis deployed and evaluated the cGAN model with four different configurations: TT-TM, NTT-NTM, TT-NTITM, and NTT-TINTM. The primary training dataset was 4000 MRI brain images with masks, while testing was done with 2000 MRI brain images with masks, having an equal proportion of tumor-positive and tumor-negative images. The purpose of the four different configurations was to train the model with one combination of image and mask and test the model on a different combination of image and mask. The evaluation was tested with various quality metrics, including IS, SSIM, MSE, PSNR, FID, and DSE. Later, images were tested for the classification performance of the modified U-Net CNN.

The real unseen and generated data were used as training data for the modified U-Net CNN for image classification. The model was tested with different combinations of real and generated data via 20-fold cross-validation. The results showed that including generated data in testing the model improved the model maximum accuracy by 92%, sensitivity by 98%, and specificity by 94%, thereby verifying the augmented data's effectiveness. Furthermore, the Mann-Whitney U test was conducted, showing that the synthetic and real data had statistically significant differences in accuracy values.

In conclusion, this thesis has shown that using a cGAN model can successfully address the challenge of the limited publicly available medical data that can be used as training data for machine learning classification models in cancer classification. By generating high-quality and high-resolution augmented data, this model proves to be a valuable tool to augment real-world datasets, enhancing the accuracy and effectiveness of classification models. This work thus offers a robust response to the research question posed, demonstrating the ability of the cGAN model to successfully augment the data and its effect on classification models. Since the cGAN crashed quite often in an unsystematic way, its successful training requires further studies.

8.1 Future Work

Looking ahead, several novel methodologies and data-driven approaches enhance the new level of accuracy and robustness in machine learning models for cancer detection. One such approach should focus on the type of brain cancer. Our analysis was restricted to the presence or absence of a tumor, which particularly restricts the wide application. Although our experiment produces high accuracy, the future work should focus on the classification of brain tumors and identifying their types.

Moreover, although our experiment included cross-domain image analysis, the future work should focus on other domain work, for-example, generating contract enhanced images from non-contrast enhanced images.

It is important to acknowledge that the model was tested with MRI images only and restricted to the brain, which limits its generalizability to other imaging modalities and organs. To address this concern, the feature endeavors should focus on training the cGAN model using a dataset from CT and PET imaging, and experiment with tumors from other body regions. In addition, rare diseases, such as Amyotrophic Lateral Sclerosis, should be explored and tested with cGAN for data augmentation and classification.

Additionally, the images were evaluated based on statistical figures and values; no healthcare professional was involved in testing the images. Future directions should involve medical personnel in evaluating the realism of synthetic data to further confirm the clinical application of the generated images.

Acknowledgement

I would like to express my deepest gratitude to the individuals and institutions whose support has been instrumental in the completion of this thesis.

First and foremost, I am sincerely thankful to my supervisors, Dr. Rainio, Dr. Tadi, and Professor Klén, for their invaluable guidance, feedback, and encouragement throughout this journey. Their expertise and mentorship have greatly contributed to the direction and quality of this research.

I also gratefully acknowledge the support provided by The Valto Takala Fund Scholarship, which made this work possible by allowing me to fully dedicate myself to my studies and research.

Special thanks go to my colleagues and friends at the Turku PET Centre. Their collaboration, insightful discussions, and continuous support created a stimulating environment that enriched my academic experience.

Finally, I would like to thank my family for their unwavering support, understanding, and patience during this journey. Their belief in me has been a constant source of strength.

Some diagrams in this thesis were created with the assistance of ChatGPT-4o, used as a tool for conceptual design and formatting, in partial fulfillment of the graphical requirements.

Turku, 12.05.2025

Mahnoor Mahnoor

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

[2] Aljohani, A., & Alharbe, N. (2022). Generating synthetic images for healthcare with novel deep pix2pix gan. Electronics, 11(21), 3470.

[3] Amin, J., Sharif, M., Haldorai, A., Yasmin, M., & Nayak, R. S. (2022). Brain tumor detection and classification using machine learning: a comprehensive survey. Complex & intelligent systems, 8(4), 3161-3183.

[4] Amirrajab, S., Al Khalil, Y., Lorenz, C., Weese, J., Pluim, J., & Breeuwer, M. (2022). Label-informed cardiac magnetic resonance image synthesis through conditional generative adversarial networks. Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society, 101, 102123. https://doi.org/10.1016/j.compmedimag.2022.102123

[5] Anantharajan, S., Gunasekaran, S., Subramanian, T., & Venkatesh, R. (2024). MRI brain tumor detection using deep learning and machine learning approaches. Measurement: Sensors, 31, 101026.

[6] awsaf49. (n.d.). *Brain Tumor Segmentation (BraTS2020)* [Data set]. Kaggle. Retrieved May 13, 2025, from `https://www.kaggle.com/datasets/awsaf49/brats2020-training-data`

[7] Bandyopadhyay, S. K. (2011). Detection of brain tumor-a proposed method. Journal of global research in computer science, 2(1), 56-64.

[8] Ben-Cohen, A., Klang, E., Raskin, S. P., Amitai, M. M., & Greenspan, H. (2017). Virtual PET images from CT data using deep convolutional networks: initial results. In Simulation and Synthesis in Medical Imaging: Second International Workshop, SASHIMI 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 10, 2017, Proceedings 2 (pp. 49-57). Springer International Publishing.

[9] Bhuiya, N. (2023). A review on the occurrence of brain tumor in adults and pediatrics and the associated risk factors (Doctoral dissertation, Brac University).

[10] Bria, A., Marrocco, C., & Tortorella, F. (2020). Addressing class imbalance in deep learning for small lesion detection on medical images. Computers in biology and medicine, 120, 103735.

[11] Bullitt, E., Zeng, D., Gerig, G., Aylward, S., Joshi, S., Smith, J. K., ... & Ewend, M. G. (2005). Vessel tortuosity and brain tumor malignancy: a blinded study1. Academic radiology, 12(10), 1232-1240.

[12] Cabezas, M., Oliver, A., Lladó, X., Freixenet, J., & Cuadra, M. B. (2011). A review of atlas-based segmentation for magnetic resonance brain images. Computer methods and programs in biomedicine, 104(3), e158-e177.

[13] Cai, L., Gao, J., & Zhao, D. (2020). A review of the application of deep learning in medical image classification and segmentation. Annals of translational medicine, 8(11).

[14] Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., ... & Wang, J. (2024). Context autoencoder for self-supervised representation learning. International Journal of Computer Vision, 132(1), 208-223.

[15] Chollet, F., et al. (2015). Keras. GitHub.

[16] CSC – IT Center for Science. (n.d.). *CSC – IT Center for Science*. Retrieved September 13, 2024, from `https://csc.fi`

[17] Dar, S. U., Yurt, M., Karacan, L., Erdem, A., Erdem, E., & Cukur, T. (2019). Image synthesis in multi-contrast MRI with conditional generative adversarial networks. IEEE transactions on medical imaging, 38(10), 2375-2388.

[18] De Bock, K. W., Coussement, K., De Caigny, A., Słowiński, R., Baesens, B., Boute, R. N., ... & Weber, R. (2024). Explainable AI for operational research: A defining framework, methods, applications, and a research agenda. European Journal of Operational Research, 317(2), 249-272.

[19] Denton, E. L., Chintala, S., & Fergus, R. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. Advances in neural information processing systems, 28.

[20] Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. Computer Science Review, 40, 100379.

[21] Ding, S., Xu, L., Su, C., & Jin, F. (2012). An optimizing method of RBF neural network based on genetic algorithm. Neural Computing and Applications, 21, 333-336.

[22] Fellhauer, I., Zöllner, F. G., Schröder, J., Degen, C., Kong, L., Essig, M., ... & Schad, L. R. (2015). Comparison of automated brain segmentation using a brain phantom and patients with early Alzheimer's dementia or mild cognitive impairment. Psychiatry Research: Neuroimaging, 233(3), 299-305.

[23] Forsyth, D. A., Mundy, J. L., di Gesú, V., Cipolla, R., LeCun, Y., Haffner, P., ... & Bengio, Y. (1999). Object recognition with gradient-based learning. Shape, contour and grouping in computer vision, 319-345.

[24] Fulton, A., & Rzepka, N. (n.d.). Universal Pathlib (Version 0.2.6) [Python package]. PyPI. Retrieved April 4, 2025. https://pypi.org/project/universal-pathlib/

[25] Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. Statistics and Computing, 21, 137-146.

[26] Garcea, F., Serra, A., Lamberti, F., & Morra, L. (2023). Data augmentation for medical imaging: A systematic literature review. Computers in Biology and Medicine, 152, 106391.

[27] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... , Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

[28] HaoQi, G., & Ogawara, K. (2020, April). CGAN-based synthetic medical image augmentation between retinal fundus images and vessel segmented images. In 2020 5th International Conference on Control and Robotics Engineering (ICCRE) (pp. 218-223). IEEE.

[29] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., ... & Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. Medical image analysis, 35, 18-31.

[30] Heinkele, A., Erath, J., Hennemann, L., Maier, J., Fournié, E., Sunnegaardh, J., ... & Kachelrieß, M. (2025, April). Deep scatter estimation for static CT using multiple projections. In Medical Imaging 2025: Physics of Medical Imaging (Vol. 13405, pp. 848-853). SPIE.

[31] Hellström, H., Liedes, J., Rainio, O., Malaspina, S., Kemppainen, J., & Klén, R. (2023). Classification of head and neck cancer from PET images using convolutional neural networks. Scientific Reports, 13(1), 10528.

[32] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30.

[33] Hijazi, S., Kumar, R., & Rowen, C. (2015). Using convolutional neural networks for image recognition. Cadence Design Systems Inc.: San Jose, CA, USA, 9(1).

[34] Hodson, T. O. (2022). Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. Geoscientific Model Development Discussions, 2022, 1-10.

[35] Huang, Z. H., Chen, L., Sun, Y., Liu, Q., & Hu, P. (2024). Conditional generative adversarial network driven radiomic prediction of mutation status based on magnetic resonance imaging of breast cancer. Journal of translational medicine, 22(1), 226. https://doi.org/10.1186/s12967-024-05018-9

[36] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

[37] Imperial College London. (n.d.). *IXI Dataset – Information eXtraction from Images*. Retrieved September 13, 2024, from `https://brain-development.org/ixi-dataset/`

[38] IPython Development Team. (n.d.). IPython.display — IPython 8.22.0 documentation. Retrieved April 4, 2025. https://ipython.readthedocs.io/en/stable/api/generated/IPython.display.html

[39] Ixi Dataset. Available online: https://brain-development.org/ixi-dataset/ (accessed on 21 October 2022).

[40] Işın, A., Direkoğlu, C., & Şah, M. (2016). Review of MRI-based brain tumor image segmentation using deep learning methods. Procedia Computer Science, 102, 317-324.

[41] Joyce, J. M. (2011). Kullback-leibler divergence. In International encyclopedia of statistical science (pp. 720-722). Springer, Berlin, Heidelberg.

[42] Kamath, U., Liu, J., & Whitaker, J. (2019). Deep learning for NLP and speech recognition (Vol. 84). Cham, Switzerland: Springer.

[43] Karunasingha, D. S. K. (2022). Root mean square error or mean absolute error? Use their ratio as well. Information Sciences, 585, 609-629.

[44] Khalil, Y. A., Ayaz, A., Lorenz, C., Weese, J., Pluim, J., & Breeuwer, M. (2024). Multi-modal brain tumor segmentation via conditional synthesis with Fourier domain adaptation. Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society, 112, 102332. https://doi.org/10.1016/j.compmedimag.2024.102332

[45] Kim, H. E., Cosa-Linan, A., Santhanam, N., Jannesari, M., Maros, M. E., & Ganslandt, T. (2022). Transfer learning for medical image classification: a literature review. BMC medical imaging, 22(1), 69.

[46] Lan, X. L. (2023). Traditional Augmentation Versus Deep Generative Diffusion Augmentation for Addressing Class Imbalance in Chest X-ray Classification (Master's thesis).

[47] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[48] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4681-4690).

[49] Lerch, L., Huber, L. S., Kamath, A., Pöllinger, A., Pahud de Mortanges, A., Obmann, V. C., Dammann, F., Senn, W., & Reyes, M. (2024). DreamOn: a data augmentation strategy to narrow the robustness gap between expert radiologists and deep learning classifiers. Frontiers in radiology, 4, 1420545. https://doi.org/10.3389/fradi.2024.1420545

[50] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.

[51] Mahnoor, M., Rainio, O., & Klén, R. (2024). Exploring Generative Adversarial Network-Based Augmentation of Magnetic Resonance Brain Tumor Images. Applied Sciences, 14(24), 11822.

[52] Mirza, M. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

[53] Msoud Nickparvar. (2021). Brain Tumor MRI Dataset [Data set]. Kaggle., accessed on January 15, 2025. https://doi.org/10.34740/KAGGLE/DSV/2645886

[54] Mudeng, V., Kim, M., & Choe, S. W. (2022). Prospects of structural similarity index for medical image analysis. Applied Sciences, 12(8), 3754.

[55] Naderi, M., Karimi, N., Emami, A., Shirani, S., & Samavi, S. (2023). Dynamic-Pix2Pix: Medical image segmentation by injecting noise to cGAN for modeling input and target domain joint distributions with limited training data. Biomedical Signal Processing and Control, 85, 104877.

[56] Nalbalwar, R., Majhi, U., Patil, R., & Gonge, S. (2014). Detection of brain tumor by using ANN. image, 2(3), 7.

[57] National Foundation for Cancer Research. (n.d.). *Brain cancer*. Retrieved September 13, 2024, from https://www.nfcr.org/cancer-types/cancer-types-brain-cancer/

[58] NumPy Developers. (n.d.). NumPy documentation. Retrieved April 1, 2025, from https://numpy.org/devdocs/index.html

[59] Oh, J. H., Lee, D. J., Ji, C. H., Shin, D. H., Han, J. W., Son, Y. H., & Kam, T. E. (2024). Graph-Based Conditional Generative Adversarial Networks for Major Depressive Disorder Diagnosis With Synthetic Functional Brain Network Generation. IEEE journal of biomedical and health informatics, 28(3), 1504–1515. https://doi.org/10.1109/JBHI.2023.3340325

[60] OpenCV Team. (n.d.). opencv-python (Version 4.11.0.86) [Python package]. PyPI. Reterieved April 5, 2025. https://pypi.org/project/opencv-python/

[61] Osorio, F., Vallejos, R., Barraza, W., Ojeda, S. M., & Landi, M. A. (2022). Statistical estimation of the structural similarity index for image quality assessment. Signal, Image and Video Processing, 1-8.

[62] Park, J. G., & Lee, C. (2009). Skull stripping based on region growing for magnetic resonance brain images. NeuroImage, 47(4), 1394-1407.

[63] Pereira, S., Pinto, A., Alves, V., & Silva, C. A. (2016). Brain tumor segmentation using convolutional neural networks in MRI images. IEEE transactions on medical imaging, 35(5), 1240-1251.

[64] Petrella, J. R., Coleman, R. E., & Doraiswamy, P. M. (2003). Neuroimaging and early diagnosis of Alzheimer disease: a look to the future. Radiology, 226(2), 315-336.

[65] Patel, M., Wang, X., & Mao, S. (2020, July). Data augmentation with conditional GAN for automatic modulation classification. In Proceedings of the 2nd ACM Workshop on wireless security and machine learning (pp. 31-36).

[66] Rainio, O., & Klén, R. (2024). Comparison of simple augmentation transformations for a convolutional neural network classifying medical images. Signal, Image and Video Processing, 18(4), 3353-3360.

[67] Rainio, O., Tamminen, J., Venäläinen, M.S. et al. Comparison of thresholds for a convolutional neural network classifying medical images. Int J Data Sci Anal (2024). https://doi.org/10.1007/s41060-024-00584-z

[68] Raza, M., Sharif, M., Yasmin, M., Masood, S., & Mohsin, S. (2012). Brain image representation and rendering: A survey. Research Journal of Applied Sciences, Engineering and Technology, 4(18), 3274-3282.

[69] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation (pp. 234–241). In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

[70] Saeedi, S., Rezayi, S., Keshavarz, H., & R. Niakan Kalhori, S. (2023). MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques. BMC Medical Informatics and Decision Making, 23(1), 16.

[71] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. Advances in neural information processing systems, 29.

[72] Saxena, D., & Cao, J. (2021). Generative adversarial networks (GANs) challenges, solutions, and future directions. ACM Computing Surveys (CSUR), 54(3), 1-42.

[73] Segal, R., Kothari, M. L., & Madnani, S. (2000). Radial basis function (RBF) network adaptive power system stabilizer. IEEE Transactions on Power Systems, 15(2), 722-727.

[74] Senaras, C., Niazi, M. K. K., Sahiner, B., Pennell, M. P., Tozbikian, G., Lozanski, G., & Gurcan, M. N. (2018). Optimized generation of high-resolution phantom images using cGAN: Application to quantification of Ki67 breast cancer images. PloS one, 13(5), e0196846.

[75] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of big data, 6(1), 1-48.

[76] Shen, D., Wu, G., & Suk, H. I. (2017). Deep learning in medical image analysis. Annual review of biomedical engineering, 19(1), 221-248.

[77] Sundaramurthy, S., Wahi, A., Devi, L. P., & Yamuna, S. (2014, March). Cardiac cycle phase detection in echocardiography images using ANN. In 2014 International Conference on Intelligent Computing Applications (pp. 275-279). IEEE.

[78] Suriyan, K., Ramaingam, N., Rajagopal, S., Sakkarai, J., Asokan, B., & Alagarsamy, M. (2022). Performance analysis of peak signal-to-noise ratio and multipath source routing using different denoising method. Bulletin of Electrical Engineering and Informatics, 11(1), 286-292.

[79] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[80] Shah, A., Shah, M., Pandya, A., Sushra, R., Sushra, R., Mehta, M., ... & Patel, K. (2023). A comprehensive study on skin cancer detection using artificial neural network (ANN) and convolutional neural network (CNN). Clinical eHealth.

[81] Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavaf, N., & Fox, E. A. (2020). Natural language processing advancements by deep learning: A survey. arXiv preprint arXiv:2003.01200.

[82] Tiwari, P., Pant, B., Elarabawy, M. M., Abd-Elnaby, M., Mohd, N., Dhiman, G., & Sharma, S. (2022). Cnn based multiclass brain tumor detection using medical imaging. Computational Intelligence and Neuroscience, 2022(1), 1830010.

[83] Van Rossum, G., & Drake, F.L. (2009). Python 3 Reference Manual. CreateSpace.

[84] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. Computational intelligence and neuroscience, 2018(1), 7068349.

[85] Xie, Y., Zhang, J., Shen, C., & Xia, Y. (2021). Cotr: Efficiently bridging cnn and transformer for 3d medical image segmentation. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24 (pp. 171-180). Springer International Publishing.

[86] Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., & He, X. (2018). Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1316-1324).

[87] Yao, G., Lei, T., & Zhong, J. (2019). A review of convolutional-neural-network-based action recognition. Pattern Recognition Letters, 118, 14-22.

[88] Yi, X., & Babyn, P. (2018). Sharpness-aware low-dose CT denoising using conditional generative adversarial network. Journal of digital imaging, 31, 655-669.

[89] Yu, Y., Wu, D., Lan, Z., Dai, X., Yang, W., Yuan, J., Xu, Z., Wang, J., Tao, Z., Ling, R., Zhang, S., & Zhang, J. (2024). Deep learning model for low-dose CT late iodine enhancement imaging and extracellular volume quantification. European radiology, 10.1007/s00330-024-11288-0. Advance online publication. https://doi.org/10.1007/s00330-024-11288-0

[90] Yu, Y., Zhang, W., & Deng, Y. (2021). Frechet inception distance (fid) for evaluating gans. China University of Mining Technology Beijing Graduate School, 3.

[91] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).

[92] Zhou, Z. H. (2021). Machine learning. Springer nature.

Appendices

- Code available on Github: github.com/mahnork/Conditional-Generative-Adversarial-Network

		Original images		
		10	500	1000
cGAN images	250	0.00^{***}	0.00^{***}	0.00^{***}
	500	0.00^{***}	0.01^{**}	0.78
	750	0.00^{***}	0.00^{***}	0.75
	1000	0.00^{***}	0.00^{***}	0.34
	1250	0.00^{***}	0.00^{***}	0.02^{*}
	1500	0.00^{***}	0.8	0.002^{**}
	1750	0.00^{***}	0.00^{***}	0.22
	2000	0.00^{***}	0.00^{***}	0.24

Table .1: The p-values of Mann-Whitney U test comparing the classification accuracies of the CNNs trained with training data containing both the specified numbers of original images and synthetic cGAN images compared to the CNN trained with the same number of original images but no GAN images (same image and mask). Significance levels: * $p \leq 0.05$, ** $p \leq 0.01$, *** $p \leq 0.001$. (same image and mask)

		Original images		
		10	500	1000
cGAN images	250	0.33	0.0001^{***}	0.0001^{***}
	500	0.87	0.00^{***}	0.3
	750	0.38	0.0^{***}	0.46
	1000	0.046^{*}	0.0002^{***}	0.9676
	1250	0.13	0.0^{***}	0.39
	1500	0.046^{*}	0.0001^{***}	0.49
	1750	0.06	0.0001^{***}	0.94
	2000	0.01^{**}	0.00^{***}	0.24

Table .2: The p-values of the Mann-Whitney U test comparing the classification accuracies of the CNNs trained with training data containing both the specified numbers of original images and synthetic cGAN images, compared to the CNN trained with the same number of original images but no cGAN images. Significance levels: * $p \leq 0.05$, ** $p \leq 0.01$, *** $p \leq 0.001$. (opposite image and mask)

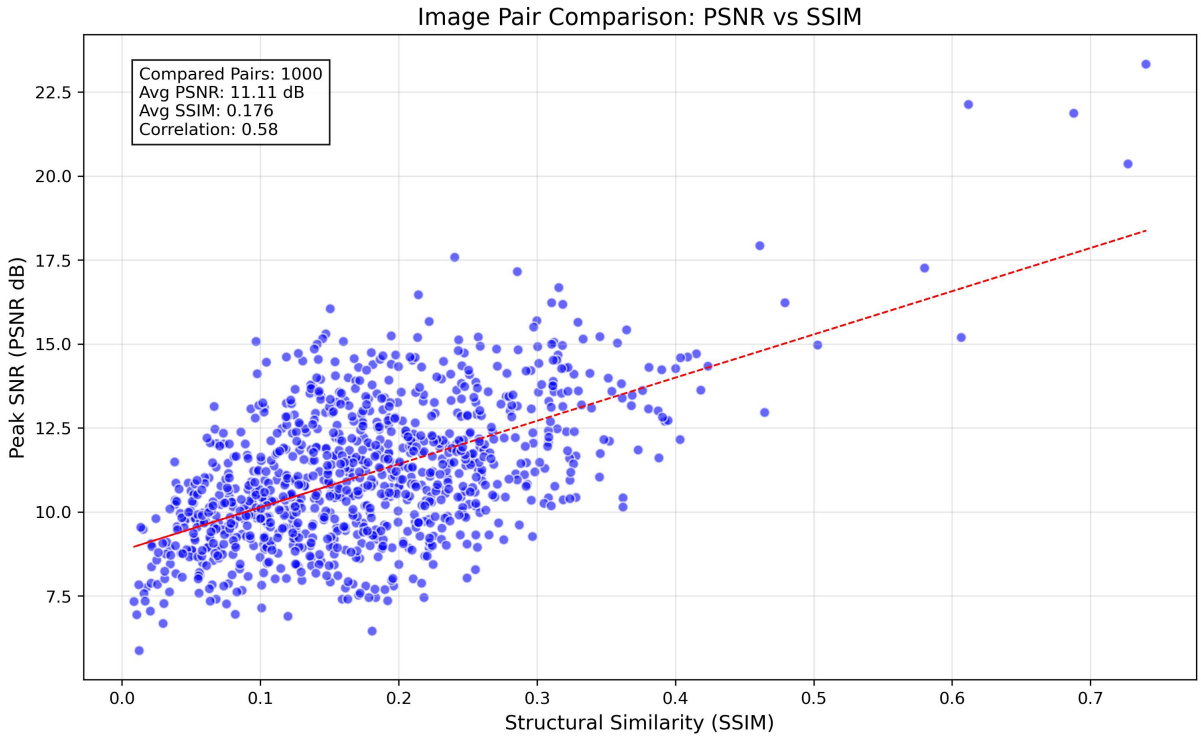

Figure .1: Scatter plot for image pair comparison of PSNR vs SSIM for image generated with pos img vs testing data having neg img

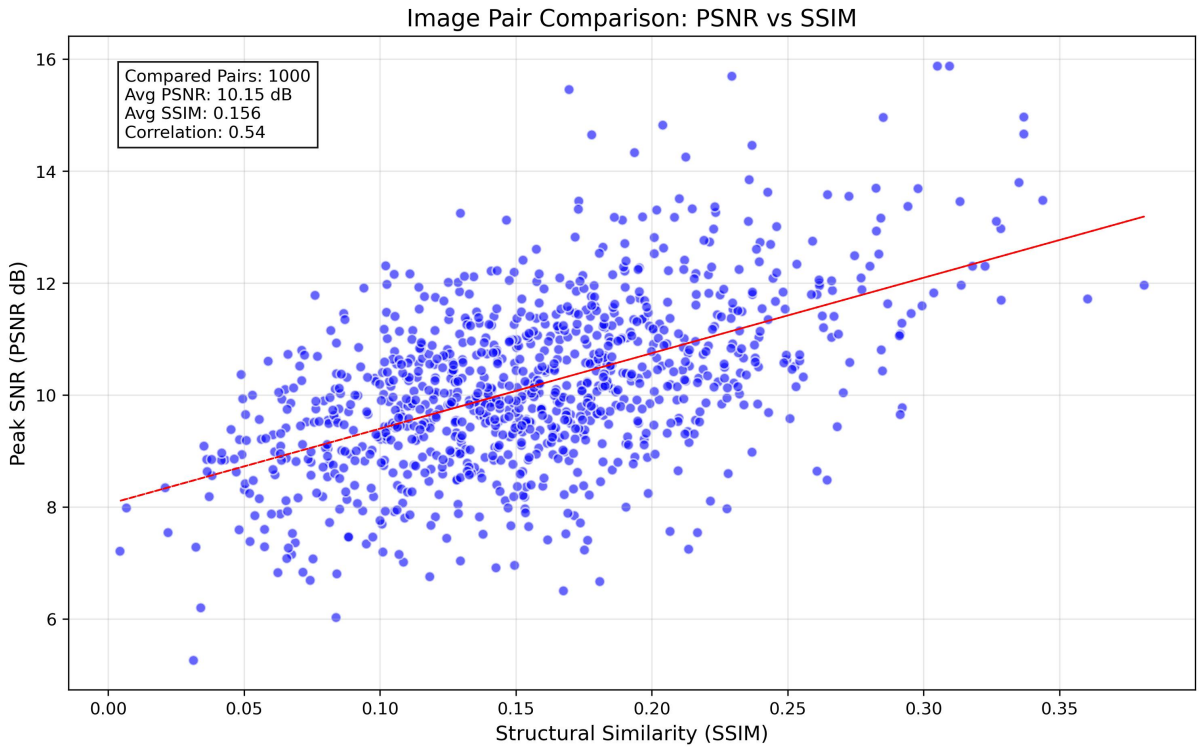

Figure .2: Scatter plot for image pair comparison of PSNR vs SSIM for image generated with neg img vs testing data having pos img

GAN images	Original images: 10	500	1000
250	0.00 ↓	0.00 ↓	0.00 ↓
500	0.00 ↓	0.01 ↓	0.78 ↑
750	0.00 ↓	0.00 ↓	0.75 ↑
1000	0.00 ↓	0.00 ↓	0.34 ↑
1250	0.00 ↓	0.00 ↓	0.02 ↓
1500	0.00 ↓	0.8 ↑	0.002 ↓
1750	0.00 ↓	0.00 ↓	0.22 ↑
2000	0.00 ↓	0.00 ↓	0.24 ↑

Table .3: The p-values of Mann-Whitney U test comparing the classification accuracies of the CNNs trained with training data containing both the specified numbers of original images and synthetic GAN images compared to the CNN trained with same number of original images but no GAN images.(same image and mask)

GAN images	Original images: 10	500	1000
250	**0.33** ↑	0.0001 ↓	0.0001 ↓
500	0.87 ↑	**0.00** ↓	**0.3** ↑
750	0.38 ↑	0.0 ↓	0.46 ↑
1000	**0.046** ↓	**0.0002** ↓	0.9676 ↑
1250	0.13 ↑	0.0 ↓	0.39 ↑
1500	**0.046** ↓	**0.0001** ↓	**0.49** ↑
1750	0.06 ↑	0.0001 ↓	0.94 ↑
2000	**0.01** ↓	**0.00** ↓	**0.24** ↑

Table .4: The p-values of Mann-Whitney U test comparing the classification accuracies of the CNNs trained with training data containing both the specified numbers of original images and synthetic GAN images compared to the CNN trained with same number of original images but no GAN images.(opposite image and mask)