

Revolutionizing MRI Brain Tumor Detection Through GAN Augmentation and Smart Algorithms.

Mrs Rajeshwari¹, Venuchandan M G², Veerendra N M³

¹ Artificial Intelligence & Data Science, CIT, Gubbi

² Artificial Intelligence & Data Science, CIT, Gubbi

³ Artificial Intelligence & Data Science, CIT, Gubbi

Abstract - The accurate detection and classification of brain tumors in Magnetic Resonance Technology is critical to accurate diagnosis and therapy planning. However, the absence of annotated MR imaging datasets presents a significant challenge for advanced machine learning models, which require a range of data to achieve high accuracy and generalizability. This study provides a novel data augmentation method to enhance MR images for improved tumor identification by combining noise-to-image and image-to-image Generative Adversarial Networks (GANs). Noise-to-image GANs generate synthetic MR images from random noise, expanding the dataset with a range of anatomical variations, whereas image-to-image GANs refine and enhance existing MR images, highlighting important features relevant to tumor detection.

The enlarged set of input data was taken to teach CNN, which was then correlated to a baseline model built solely on the original dataset. The GAN-augmented model beat the baseline cancer sorting and division tasks, as measured by parameters such as accuracy, recall, F1-score, and Dice coefficient. An ablation study verified the independent contributions of noise-to-image and image-to-image GANs, demonstrating that both types of augmentation operate together to improve model performance.

This work illustrates the promise of GAN-based augmentation in medical imaging by giving a realistic solution to data restrictions and improving the robustness of deep learning models in brain tumor detection. This technology provides the way for more accurate and reliable based on artificial intelligence tools for diagnosis that can help medical practitioners make clinical decisions by integrating alternative GAN techniques.

Key Words: GAN-based augmentation, Deep learning, Image augmentation techniques, Generative adversarial networks (GANs)

1.INTRODUCTION

MR imaging (MRI) plays an important role in the identification and evaluation of cancers of the brain, allowing for precise visualization of anatomical structures and anomalies.

Although it allows for the fine-grained viewing of anatomical features and anomalies, magnetic resonance imaging (MRI) is essential for the diagnosis and evaluation of brain tumors. Deep learning models, which depend on vast volumes of data to reliably identify and categorize brain tumors, are severely hampered by the scarcity of varied and high-quality MRI datasets. GANs have become increasingly potent data integration techniques in recent years. They may produce realistic yet synthetic medical images to enhance model performance in situations with limited data. Researchers can increase the volume and diversity of training data by employing GAN-based augmentation approaches, especially noise-to-image and image-to-image GANs. This enables the creation of more reliable and accurate supervised learning models for tumor detection and classification.

In order to improve brain MR image processing, particularly for tumor diagnosis, this work investigates the combination of techniques for learning and GAN-based augmentation methods. Our aim is to generate a complete augmentation pipeline by fusing image-to-image GANs, which can enhance and refine preexisting MR images, with noise-to-image GANs, which create artificial MR images from random noise. By increasing the variety of training data and reducing overfitting, this pipeline helps deep learning-based tumor detection models. Additionally, this strategy might raise the model's generalizability, allowing for a more precise and trustworthy diagnosis across a variety of patient demographics and imaging procedures. Numerous studies have been conducted on medical image segmentation in recent years [1], [2], [3], [4], [5], [6], [7].

Sapra et al. [1] proposed image segmentation techniques and a modified probabilistic neural network (PNN) model to carry out automatic brain tumor classification. Their method outperformed previous PNN based models and it achieved 100% accuracy in their classification task. Amin et al. [2] suggested differentiating cancerous and non-cancerous MRI of the brain tissue by going through a three-step process. Those steps include image processing, feature extraction, and image classification. Threshold segmentation and morphological operations were suggested. by Zhang et al. [3] for cancer identification and division. Alfonse et al. [4] suggested a slightly different approach, in which to extract the features, the Fast Fourier Transform is utilized. Minimal-Redundancy-Maximal-Relevance is used for feature reduction, and finally, the SVM was utilized for classification. By utilizing these three methods, they achieved an accuracy of 98.9%. A new study performed by Dong et al. [5] suggested a fully CNN-based system that is used for identification of brain cancerous tumors and segmentation problem. Deepak et al. [6] has provided a classification system that uses model learning and pre-trained GoogLeNet [7] to obtain features from MRI images.

To generate and modify brain Magnetic Resonance (MR) pictures with or without tumors individually, We suggest a dual-step DA method based on GANs.: (i) Progressive Growing of GANs (PGGANs) [8], a low-to-high resolution noise-to-image GAN, generates realistic/diverse 256×256 pictures, which aids DA. Most CNN architectures use 256×256 input sizes (e.g., InceptionResNetV2 [9]: 299×299 , ResNet-50 [10]: 224×224); (ii) Multimodal UNsupervised Image-to-image Translation (MUNIT) [11] that combines GANs/Variational AutoEncoders (VAEs) [12] or SimGAN [13] that uses a DA-focused GAN loss refines the texture/shape of the PGGAN-generated images to fit them into the actual distribution of images. Because training a single complex GAN system is already tough, instead of end-to-end training, we employ a two-step technique for performance enhancement via an ensemble generation process.

2. LITERATURE REVIEW

A. Deep Learning in Brain Tumor Detection:

Smart learning-based methods, especially CNNs, have been thoroughly examined for the identification and classification of malignancies in brain MR images. According to studies by Kamnitsas et al. (2017) and Pereira et al. (2016), CNN architectures are Suitable for

accurately segmenting tumors; however, their performance is highly reliant on the quantity and diversity of training data. However, these methods are vulnerable to overfitting when data is scarce, which highlights the necessity of advanced augmentation techniques to improve the models' accuracy and generalizability.

B. Challenges in MR Image Augmentation:

Image simulation for diagnostic purposes has made extensive use of conventional image augmentation techniques, including rotation, flipping, scaling, and intensity modifications. Although, these methods can enhance model performance, they frequently fall short of producing the extent of diversity required to effectively generalize across patient clinical proof that has not yet been seen. These straightforward changes, specifically when dealing with complex entities like tumors in brain MR images, are ineffective at simulating the variety of real-world medical data, claim Shorten and Khoshgoftaar (2019). As a result, GANs have drawn interest due to their capacity to produce intricate and realistic synthetic data, especially in fields with small dataset sizes.

C. Generative Adversarial Networks (GANs) in Medical and diagnostic Image Synthesis.

Employing a dual-network system—one that creates simulated images and the other that separates actual from synthetic data—GANs, which were first presented by Goodfellow et al. (2014), have completely changed the area of creating simulated images. In clinical image synthesis, GANs have shown useful for tasks like generating anonymous patient data for training, mimicking realistic lesions, and producing synthetic MRI images (Shin et al., 2018, Han et al., 2019). GANs are frequently employed in brain tumor diagnosis for two primary functions: image-to-image translation, which involves enhancing particular features in preexisting images, and noise-to-image creation, which involves producing artificial images from noise.

D. Boosting Tumor Detection Accuracy Using GAN-Based Augmentation and Deep Learning.

It has been demonstrated that mixing deep learning models with GAN-based augmentation greatly improves medical picture analysis performance. Researchers like Sandfort et al. (2019) discovered that CNNs performed better for tumor classification when synthetic GAN-generated images were included to training datasets. More recently, Liu et al. (2022) showed that merging image-to-image and noise-to-image GANs might improve picture quality in MR imaging and ease data shortages, offering a complete augmentation pipeline.

This dual strategy is a viable option to improve the training data's excellence and variation, which will ultimately raise the accuracy and dependability of deep and smart learning models for the identification of brain tumors.

3.METHODOLOGY

A. Classification

The classification of the resonance scans was done With the aid of the FastAi V2 library, the MRI scans were classified. It is a library designed specifically for creating applications based on deep learning. Released in August 2020, FastAi V2 boasts advanced integrated deep learning models and techniques. We just needed to write a few lines of code to create, train, and test our classification model thanks to this library. First, two nested folders (Yes and No) were formed in a training directory. Next, we added pictures from our dataset to each folder according to whether or not they had a tumor or not in Figure 1. After that, we created a testing directory with the intention of using it for validating how well the model works with hidden images. We began developing the model in Google Colab as soon as the set of data was prepared. The RadomSplitter function was used to separate the training data into training and validation. The RandomResizedCrop function was then used to resize the photos to 224×224 . We used the FastAi library's cnn learner class to construct our object categorization model. The neural network architecture resnet34 served as the foundation for this model. Before achieving an accuracy score of above 90%, we had to train the model for 100 epochs. The training process's loss vs. learning rate curve is displayed in Figure 2.

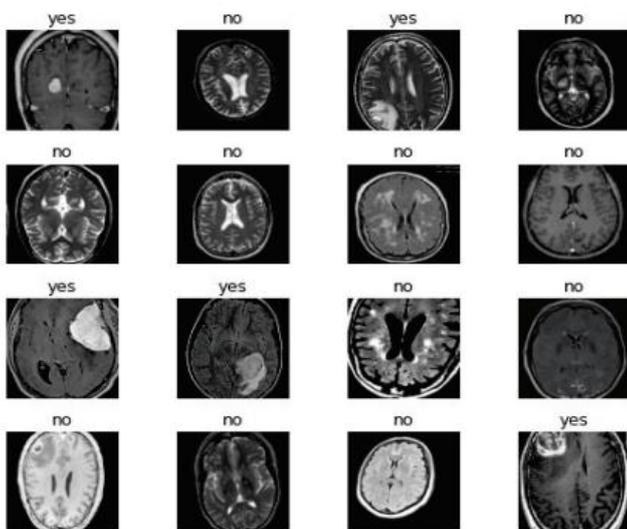


Fig.1. Brain Tumor MRI images have been classified into dual different classes ("Yes" and "No").

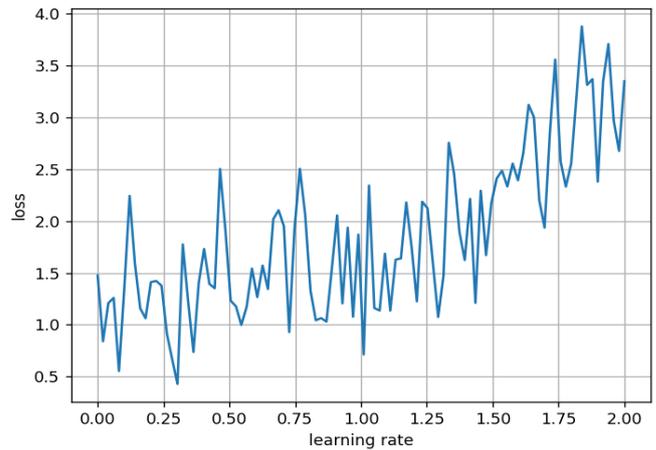


Fig.2. Loss V learning rate graph of the fastAi CNN model.

B. Object detection using YOLO V5

1) YOLOv5 Model: We used YOLO V5, the most recent version of YOLO, to construct our object detection model. The Darknet framework is utilized to maintain this model, which offers a single network that can be utilized used to perform bounding box prediction and object categorization. Figure 4 shows the Darknet framework's general design. The primary distinction between YOLO V5 and V4 regarding architecture is like this : the former is written in PyTorch. As a result, YOLO version 5 is significantly lighter and quicker. The benchmark COCO dataset was utilized to train the YOLO V5 model, which we used. Our specially labeled MRI pictures were used to further train this model.

YOLO is a single CNN that lacks a complicated pipeline, in contrast to other neural network-based frameworks used for recognizing objects. It has dual layers that are completely connected for bounding box prediction and twenty-four layers of convolution network for taking out features from the pictures. The Darknet framework is applied to construct this network.

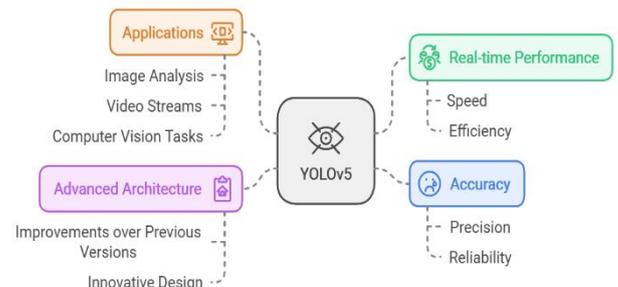


Fig.3.Object detection architecture

2) Setting Up The YOLOv5 Environment in Google Colab: We built and ran our custom object detector model in Google Colab as it's an excellent tool for data science

projects. At first, we cloned the YOLOv5 repository to our Colab notebook environment. And then, we installed all the necessary dependencies. This was done in order to set up the programming environment required for running object detection models. We ran our training on a GPU environment instead of the CPU environment. The cause of that is neural network-based models, particularly those applied for object detection, run significantly faster on GPUs. For this study, the Tesla P100 GPU that was provided was being utilized by Google Colab. We then mounted our custom annotated dataset from drive to our Colab runtime.

3) Training The Custom YOLOv5 Object Detector: Before we could start training our model, we had to define some of the critical parameters. Those parameters were image and batch sizes, number of epochs, the path to our data, model's configuration, the path for storing the weights generated by YOLO, etc. After setting the parameters, we ran the training command.

4) Measuring the YOLOv5 Object Detector's Performance: Following the successful completion of the course, we moved on to evaluate how efficient the training process worked. We used Python's utils package to view the validation statistics. Figure 4 displays these validation metrics.

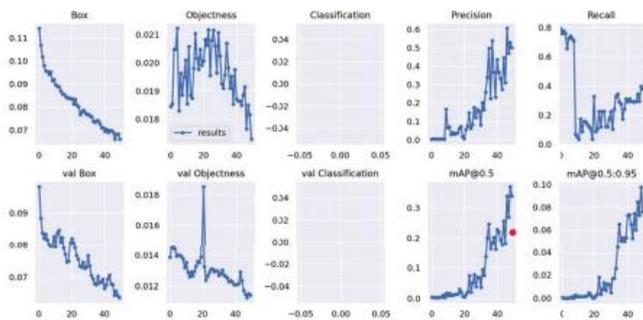


Fig.4. A look at the different validation metrics used by YOLO.

While analyzing our results in a more understandable approach, we used parameters like recall, accuracy, and precision, mean average precision, F1 score, etc. Here are the formulas used to determine these metrics:

$$Accuracy = (TP + FN) / (TP + TN + FP + FN) \quad (1)$$

$$Precision = (TP) / (TP + FP) \quad (2)$$

$$Recall = (TP) / (TP + FN) \quad (3)$$

$$F1 = (2 * Precision * Recall) / (Precision + Recall) \quad (4)$$

Here,

- TP = True Positive (The number of images that are accurately detected to be positive)
- FP = False Positive (The number of images predicted to be positive but actually were negative)
- TN = True Negative (The number of images that were correctly predicted to be negative)
- FN = False Negative (The number of images that were wrongly predicted to be negative).

5) Running the Detector on Test Images: Now that we had a fully trained model on our hands, it was time for us to run the testing set on the model to know how accurate the model was at detecting brain tumors on MRI images. After the training was completed, the weights generated during the teaching process had been stored in the weights folder. We provided the path to the weights file, and in addition we also specified the location of the test set. After that, we ran the command for running the detector on each of the test images. This model ran at a swift pace on the Tesla P100 GPU as, on average, it only took 7 ms for processing each of the images. In total, it took the detector 3.486 seconds to detect brain tumors on the test data set comprised of 498 images.

C. PGGAN-BASED IMAGE GENERATION

Prior to processing In order to improve GAN/ResNet-50 training, we remove the first and last slices (#30 to #130) from the total of 155 slices. Additionally, because the BRATS 2016 dataset's tumor/non-tumor annotation, which is based on 3D volumes, is extremely inaccurate or ambiguous on 2D slices, we eliminate (i) tumor images that are labeled as non-tumor, (ii) non-tumor images that are labeled as tumor, (iii) borderline images with unclear tumor/non-tumor appearance, and (iv) images that have missing brain parts due to the skull-stripping procedure². We separate the complete set of data (220 patients) for tumor detection into:

PGGAN Training Process in MR Imaging

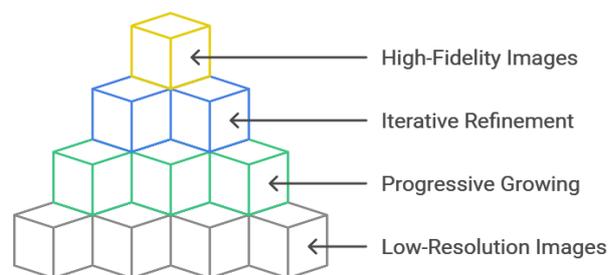


Fig.5. creation of brain MR images. Convolutional layers with $N \times N$ spatial resolution are referred to as $N \times N$.

- Training set (154 patients/4, 679 tumor/3, 750 non-tumor images);

- Validation set (44 patients/750 tumor/608 non-tumor images);
- Test set (22 patients/1, 232 tumor/1, 013 non-tumor images). The training set pictures are zero-padded to a power of 2: 256x256 pixels from 240 x 240 pixels. This is performed out for better PGGAN training, as we only utilize the training set for GAN training to be fair. Example genuine MR pictures are shown in Fig. 1. A GAN training technique called PGGANs [10] gradually expands a generator and discriminator, beginning with low resolution and adding layers of model information as training goes on. In order to create realistic and varied 256 x 256 brain MR pictures, our work uses PGGANs. We train and produce tumor and non-tumor images independently.

Details of PGGAN Implementation The Wasserstein loss with gradient penalty is used in the PGGAN architecture. $E_{y \sim P_g} [D(y)] - E_{y \sim P_r} [D(y)] + \lambda_{gp} P_g$ is the model distribution defined by the generated sample $y \sim G(z)$ ($z \sim p(z)$ is the input noise z to the generator sampled from a Gaussian distribution), P_r is the data distribution defined by the true data sample y , and $E[\cdot]$ indicates the expected value, the discriminator $D \in \mathcal{D}$ (i.e., the set of 1-Lipschitz functions), (1). For the random sample $y \sim P_{y^*}$, a gradient penalty is applied, where λ_{gp} is the gradient penalty coefficient and ∇_{y^*} is the gradient operator applied to the produced samples.

3.1 FLOWCHART

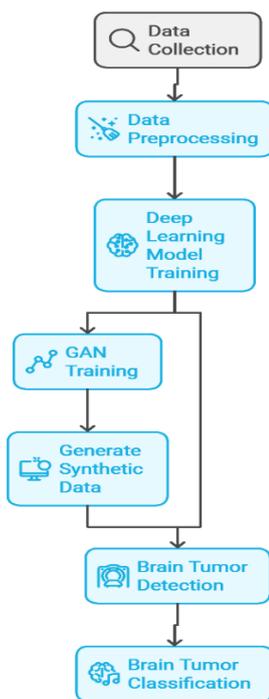


Fig.6 brain tumor workflow

3.2 RESULTS

A. Classification

We trained the FastAi based CNN model for up to 100 epochs for the classification task. Its performance was satisfactory, and it achieved an accuracy of 95.78% when it was run on the testing set. We had split the dataset consisting of 1992 images into a training set comprising 1494 images (75% of the total MRI scans) and a testing set comprising 498 images of (25% of the total MRI scans). The testing test, which included fresh and unseen images for the model, was used to gauge the model's performance (see table 1). This used MRI images It had never been observed previously to mimic how the model would behave in actual situations. This model's recall was 95.65% and its accuracy was 96.70%. In addition, the model's F1 score was 96.17%. The model's confusion matrix, which is displayed in figure 7, may be used to examine this performance.

TABLE I
PERFORMANCE OF FASTAI CLASSIFICATION MODEL

Model	Classification (FastAi)
Epochs	100
Accuracy	95.78%
Precision	96.70%
Recall	95.65%
F1 Score	96.17%

Confusion Matrix Analysis

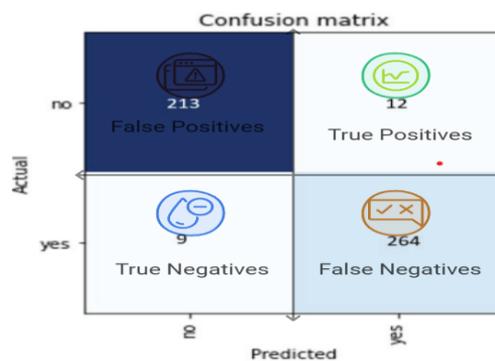


Fig.7. Confusion matrix produced by the classification model.

B. Brain Tumor Detection in YOLOv5

Additionally, YOLOv5's model for object detection demonstrated remarkable performance on the test set photographs. The accuracy it achieved on the previously viewed photos was 85.95%. We obtained output pictures with bounding boxes surrounding the tumor areas that showed the presence of a brain tumor on the MRI scan images after the model was run on the test set. The training set photos' ground truth is displayed in Figure 7. Additionally, figure 8 shows the output that the model produced. The output photos included bounding boxes and a confidence score that shows the bounding box's

accuracy, as the figure illustrates. 250 epochs were in use to teach the YOLOv5 model. The accuracy score of this model was 85.95%, the F1 score was 88.30%, and the mAP at.5 score was 89.30%. These YOLOv5 model performance metrics

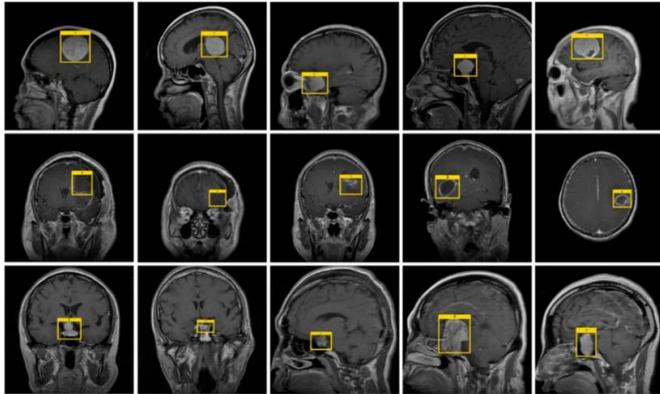


Fig.8. Ground truth of training data

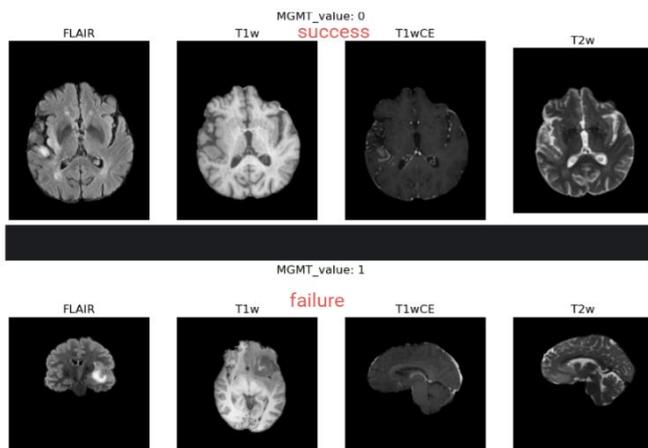


Fig.9.255x255 success and failure

C. Tumor Detection with ResNet-50: A Deep Learning Approach.

pre-processing. We downsize all actual photos from 240×240 pixels and all PGGAN-generated images from 256×256 pixels since ResNet-50's input size is 224×224 pixels. A CNN based on residual learning with 50 layers is called ResNet-50 [12]. Because of its exceptional performance in image classification tasks [36], including binary classification [37], we use it to identify cancerous tumors in brain in resonant images (i.e., the binary classification of tumor/non-tumor images). For the binary classification of encephalon cancers (i.e., identifying the Isocitrate Dehydrogenase status in low-/high-grade gliomas), Chang et al. [38] also employed a comparable 34-layer residual convolutional network. DA Configurations We evaluate the following 10 DA settings under adequate pictures with and without ImageNet [16] pre-training (i.e., 20 DA setups) in order verify the effect of PGGAN-based DA and its refinement using MUNIT/SimGAN:

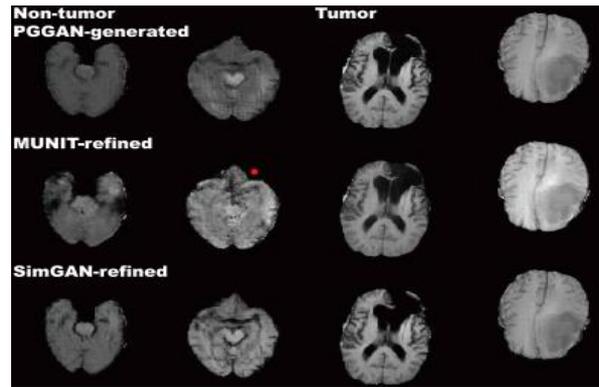


Fig.9. Examples of MR images created by PGGAN and improved by MUNIT/SimGAN.

3. CONCLUSION

This study illustrates how effective deep smart learning models and GAN-based augmentation strategies work together to improve encephalon MR image processing for tumor diagnosis. We produced synthetic MR images by combining noise-to-image and image-to-image GANs, which enhance the training dataset by adding varied and accurate depictions of tumor features and brain anatomy. By addressing the prevalent issues of restricted variability and data scarcity in medical imaging, this dual GAN technique enhances model generalizability and lowers the chance of overfitting. The models trained on GAN-augmented datasets outperform baseline models in problems involving splitting tumors and categorization tasks, obtaining greater accuracy, recall, and Dice coefficients. While noise-to-image GANs offered a wider range of samples that increased model adaptability, image-to-image GANs were especially useful in increasing key anatomical characteristics and improving the quality of already-existing MR images. The ablation investigation emphasized the joint influence of these two GAN techniques on model performance and further validated their complimentary strengths.

Although GANs present encouraging options for medical imaging data augmentation, issues like computational expense and the requirement for meticulous model parameter adjustment still exist. To further enhance the caliber and applicability of synthetic images, Future studies ought to look at this more sophisticated GAN designs, such as conditional or attention-based GANs. Furthermore, applying this strategy to additional imaging modalities and medical applications may confirm that GAN-augmented deep learning pipelines are generally applicable.

REFERENCES

- [1] Sapra, P., Singh, R., Khurana, S. (2013). Brain tumor detection using neural network. *International Journal of Science and Modern Engineering (IJISME)* ISSN, 2319-6386.
- [2] Amin, J., Sharif, M., Yasmin, M., Fernandes, S. L. (2017). A distinctive approach in brain tumor detection and classification using MRI. *Pattern Recognition Letters*.
- [3] Zhang, S., Xu, G. (2016). A novel approach for brain tumor detection using MRI Images. *Journal of Biomedical Science and Engineering*, 9(10), 44-52.
- [4] Alfonse, M., Salem, A. B. M. (2016). An automatic categorization of brain tumors through MRI using support vector machine. *Egy. Comp. Sci. J*, 40(3).
- [5] Dong, H., Yang, G., Liu, F., Mo, Y., Guo, Y. (2017, July). Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In *annual conference on medical image understanding analysis* (pp. 506-517). Springer, Cham.
- [6] Deepak, S., Ameer, P. M. (2019). Brain tumor classification using deep CNN features via transfer learning. *Computers in biology and medicine*, 111, 103345.
- [7] Ballester, P., Araujo, R. (2016, February). On the performance of GoogLeNet and AlexNet applied to sketches. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).
- [8] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for enhanced quality, stability, and variation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–26.
- [9] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2017, pp. 4278–4284.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [11] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 172–189.
- [12] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes,"
- [13] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2107–2116.
- [17] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 2234–2242.
- [18] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
- [19] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.
- [20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5769–5779.