DCGAN-Based Data Augmentation and GridSearch-Optimized CNN for Imbalanced Brain Tumor MRI Classification

[1]Pradeep Shree Adhikari and [2]Amit Sharma*

[1, 2]Department of Computer Science & Engineering, Vivekananda Global University, Jaipur, Rajasthan, India

[1]psadhikary@gmail.com
[2*]dr.amittech@gmail.com

Abstract. This paper focuses on more accurately and efficiently detecting the presence of brain tumors from their Magnetic Resonance Imaging (MRI) scans, leading to improved clinical decision-making and treatment outcomes. Medical image classification frequently uses deep learning models that must deal with complications like class imbalance and inappropriate hyperparameter settings, which can harm their performance and accuracy. To overcome these problems, this paper proposes a unified framework of DCGAN–based data augmentation, along with GridSearch-based hyperparameter optimisation for multiclass brain tumor classification. The DCGAN model generates synthetic MRI samples to balance the dataset and increase feature variance, whereas GridSearchCV optimises hyperparameters for the CNN classifier. The experiment is conducted on deep learning models, Custom CNN, VGG19, EfficientNet and BPNN, with different optimisers. Out of these models, the Custom CNN with Adam optimiser attained the best performance, registering an accuracy of 88.79%, recall of 87.96, and an F1-score of 88.13. The results show that using generative data augmentation effectively enhances classification reliability, while hyperparameter optimisation enables automated brain Tumor diagnosis with consistently better performance.

Keywords: CNN, Hyperparameter optimization, GridSearchCV, Medical image classification, Brain tumor.

B. Singh et al. (eds.), *Proceedings of the International Conference on Advances in Computing Technology and Artificial Intelligence (COMPUTATIA 2026)*, Atlantis Highlights in Intelligent Systems 18,
https://doi.org/10.2991/978-94-6239-713-2_12

1 Introduction

This study uses a public brain Tumor MRI dataset comprising four categories: glioma, meningioma, pituitary, and non-Tumor. Manual MRI analysis is time-consuming and prone to errors. Convolutional Neural Networks (CNNs), on the other hand, automatically extract hierarchical features for disease classification, making them a powerful alternative [9,12]. However, CNNs don't always perform well when there are too many classes and require fine-tuning. To address these problems, this study uses Deep Convolutional Generative Adversarial Networks (DCGAN) to augment the dataset, making it more balanced and helping the model generalise better [13]. Grid Search is also a systematic way to find the best values for key hyperparameters such as kernel size, learning rate, and batch size [14]. This ensures that the CNN is set up optimally and neither overfits nor underfits [15].

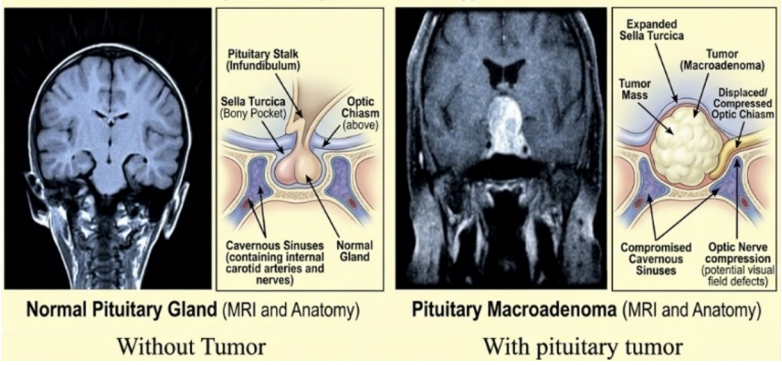

Fig. 1 MRI and comparative anatomy of the pituitary gland

Fig. 1 illustrates the differences in the anatomy and radiology of a healthy pituitary gland and a pituitary macroadenoma. The "Without Tumor" section shows the normal pituitary stalk, sella turcica, optic chiasm, and cavernous sinuses [16]. The "With Pituitary Tumor" section, on the other hand, shows abnormal changes, such as an enlarged sella turcica, a Tumor mass, and pressure on the optic chiasm and cavernous sinuses [17]. This comparison shows how the macroadenoma alters the structure and what that means for the patient, such as how it could make it hard for them to see

1.1 Image Classification Process

This paper improves multiclass image classification by using DCGAN-based data augmentation and systematic hyperparameter tuning to address data skew. To improve minority-class accuracy, a Deep Convolutional Generative Adversarial Network (DCGAN) generates synthetic images for each class [18]. The study then evaluates the classification accuracy of a CNN trained with and without these augmented samples.

As illustrated in Fig. 2, the pre-processed images are then processed within the BTDN framework to ensure robust analytical performance.

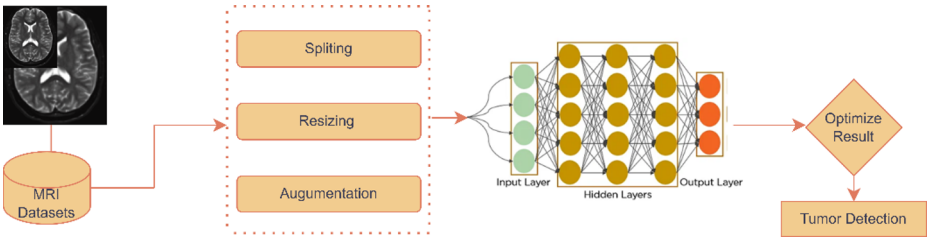

Fig. 2: BTDN Model used for brain tumor detection [20]

1.2 Hyperparameter tuning

Hyperparameter tuning optimizes model parameters that are set before training rather than learned from data. In this study, Grid Search is used to evaluate candidate values, including learning rate, dropout fraction, filter count, and kernel size, on a validation set to identify the configuration that yields the highest accuracy [19,20]. It includes Learning Rate, which controls the magnitude of weight adjustments during training; Dropout, a regularisation technique that deactivates a portion of the network to prevent overfitting; and Filters & Kernel Size, which define the number of features extracted and the dimensions of the convolutional filters [21]. By carefully adjusting these settings, the model is better suited to the unique features of the MRI dataset. This process works well with DCGAN-based augmentation, improving classification accuracy and ensuring the model generalises well to new data.

1.2.2 Hyperparameter Tuning in Machine Learning

GridSearchCV searches through a parameter grid to find the best settings for a model [22,23]. For a CNN, this involves testing combinations of hyperparameters such as learning rates (for example, 0.001, 0.01), filter counts (for example, 32, 64), and kernel sizes (for example, 3x3, 5x5). The algorithm trains the model on each combination and evaluates performance using a scoring function on a validation set[24,25]. The configuration yielding the highest score is then selected to train the final model, ensuring robust and generalised results across various estimators [26,27].

1.3 Contributions to Work

i. A CNN-based classification model is developed to perform multiclass brain tumor MRI classification [28].
ii. A GridSearchCV-based hyperparameter optimization approach is applied to determine the optimal configuration of CNN parameters [29].
iii. A comparative analysis of baseline, augmented, and optimised CNN models is conducted to evaluate the effectiveness of the proposed approach [30].

2. Literature Review

The study focuses on ML methods for solving complex analytical healthcare tasks. However, class imbalance remains a significant challenge that hinders model performance. The following section reviews approaches to class imbalance and maps all approaches to be studied.

2.1 Related Work

This study evaluated a nationwide cervical cancer screening program in Bangladesh using the DHIS2 electronic registry [24]. While primary health facilities provided most VIA tests, the colposcopy follow-up rate remained low at 40.4%. The DHIS2 tracker proved effective for monitoring malignancy management pathways and tracking patient progress across pre-cancerous stages [1]. A quasi-experimental study employed the Health Belief Model (HBM) to evaluate a nurse-led educational intervention. The results showed that participants' knowledge increased by 10% and their confidence in their abilities by 25%. The proportion screened rose from 16% to 57%, indicating that targeted education can change health behaviours [2]. A ten-year study in rural China examined a cancer prevention programme founded on Social Cognitive Theory. Knowledge and self-efficacy in the intervention group improved significantly at both the 3-month and 18-month follow-ups ($p<0.001$). These findings indicate that community health programmes are not yet useless in resource-constrained settings [3]. In a retrospective cohort study in Japan, immediate colposcopy was examined with regard to the international retesting guidelines. The study found that certain high-risk genotypes were more likely to have extended observation periods when referral was delayed. The findings indicate that enhancing clinical pathways and risk stratification would be useful to prevent cancer and ensure that local policies are consistent with international standards [4]. The study involved 446 married women in rural Uttarakhand and examined their health behaviours. The study showed that socio-demographic factors such as education and income played a significant role in people's understanding of diseases. Understanding such social and cultural barriers is the initial step in developing health promotion programmes that are specific to some groups [5]. In this study, machine learning (ML) models, including XGBoost and Random Forest, were used to predict water quality. With careful tuning of the hyperparameters, the XGBoost model achieved 97.06% accuracy. The results indicate that environmental monitoring is significantly enhanced by improving model architectures to make correct predictions and generalizations [6]. Our modified U-Net architecture produced a deep learning model that automatically segmented the appendix on CT scans. The model performed well on the Jaccard index and the dice coefficient, with pooling layers and filters used to extract information. Such a CAD system reduces the likelihood that various doctors will perceive clinical radiological images differently [7]. The study on predicting breast cancer recurrence looked into how hyperparameter optimization (HPO) affected the performance of the model. The study employed Grid Search to prevent overfitting, and the findings indicated that enhanced models significantly outperformed baseline models. This method provides strong risk stratification for complex biomedical datasets [8]. This study used LASSO regression and incremental learning to predict long-term risk

to address class imbalance and data that changes over time. This method makes mass screening programmes work better by keeping high sensitivity for high-grade lesions and lowering over-screening in low-risk groups [9,10]. This study used machine learning classification and ideas from the Health Belief Model to predict who would not show up for their appointment. The sequential modelling method identified people at high risk by analysing administrative and psychosocial data. This makes it easier to make public health programmes more effective by tailoring interventions to each person and using resources more wisely [11].

2.2. Research Gap

Deep learning-based medical image classification has made a lot of progress, but there are still some problems with the current studies that screen for brain tumors. A lot of previous studies have focused on using traditional machine learning or deep learning algorithms to make classification models. But these studies don't always talk about the problems that come up when the dataset isn't balanced or there aren't enough samples to train on. The main problems with the research that were found are as follows:

1. Classification studies often face class imbalance issues that adversely affect model performance.
2. There hasn't been much research on how to use DCGAN-based synthetic image generation to improve brain Tumor image datasets.
3. Numerous CNN-based classification models rely on manual or default hyperparameter configurations, leading to subpar performance.
4. There are a few studies that combine data augmentation with GANs and systematic hyperparameter optimisation methods.

2.3 Comparison table

Table 1: Comparison of few Studies on Image Classification

Study	Method Used	Dataset / Application	Key Findings	Limitations
[21]	Generative Adversarial Networks	Image generation tasks	Introduced GAN framework for generating realistic images	Did not focus on medical datasets
[22]	DCGAN	Unsupervised image representation	Demonstrated deep convolutional GAN for realistic image synthesis	Limited exploration in medical image classification
[23]	Deep Learning Survey	Medical image analysis	Highlighted potential of CNN models in medical diagnosis	Did not address dataset imbalance issues

[24]	CNN	Skin cancer image classification	Achieved dermatologist-level classification accuracy	Large dataset requirement
[25]	Data Augmentation Techniques	Image classification	Showed that augmentation improves model performance	Traditional augmentation limited to image transformations
[16]	ML-based classification	Cancer detection	Demonstrated improved classification using ML models	Hyperparameter tuning not fully explored
[17]	CNN with Hyperparameter Tuning	Tumor image classification	Improved accuracy through hyperparameter optimization	No GAN-based data augmentation
[18]	Deep Learning Model	Brain tumor detection	Demonstrated improved classification with deep CNN	Dataset imbalance not addressed
[19]	Hybrid Deep Learning Model	Brain tumor MRI classification	Achieved improved classification using hybrid models	Lack of systematic parameter optimization
[20]	Hybrid CNN Framework	Medical image classification	Showed improved classification with hybrid networks	No GAN-based augmentation strategy

3. Methodology

To assess the efficacy of the proposed framework, it is compared with established baseline models using standard performance metrics. In this case, the baseline models include both the most recent architectures for multi-class image classification and older machine learning algorithms. To ensure a fair comparison, both the proposed and baseline models are trained on an identical dataset and evaluated on a distinct set. If the proposed model demonstrates improved accuracy or other performance metrics, it is considered an advancement over current methodologies.

3.1 Dataset Description

The Intel MobileODT Brain Tumor Screening dataset is used for the experiment. It contains images split into three groups: Type 1, Type 2, and Type 3. The dataset is imbalanced, with some classes having far fewer samples than others. This imbalance could cause the model to perform worse and reduce its accuracy in sorting images. The study utilised a publicly available MRI dataset of brain Tumors, classified into four categories: glioma, meningioma, pituitary Tumor, and non-Tumor images. The dataset had 5712 training images and 1311 testing images, with training and testing images in

separate folders. All images were resized to 128 × 128 pixels so that all neural networks would receive the same input size.

3.2 Algorithm steps to implement GridSearchCV

GridSearchCV is a type of hyperparameter optimization technique used to find the best parameters using an exhaustive grid search that results in the highest cross-validation score. The following are the steps used in our work:

- Define the hyperparameters and their possible values
- Select the machine learning estimator for tuning
- Specify the search space using a parameter grid
- Split the dataset to evaluate model performance.
- Fit the model using GridSearchCV with multiple hyperparameter combinations and cross-validation
- Select the best hyperparameters based on the highest recall score
- Train the final model on the full dataset using the optimal hyperparameters.

3.3 DCGAN-Based Data Augmentation

To address class imbalance, a Deep Convolutional Generative Adversarial Network (DCGAN) is used to generate synthetic images for the underrepresented classes. The DCGAN has two main components: the Generator and the Discriminator. Adversarial training helps the generator learn to produce realistic synthetic images that resemble the original dataset. The discriminator, on the other hand, distinguishes between real and fake images. These images are then added to the training dataset to increase diversity and help the model learn.

3.4 CNN Model Architecture

A Convolutional Neural Network (CNN) is used to classify brain Tumor images into more than one group. The CNN model has many convolutional, pooling, and fully connected layers that it uses to extract features and sort them.

The structure has:
• Convolution layers to get features
• Max pooling layers to make the number of dimensions smaller
• Dropout layers to stop overfitting
• Fully connected layers for the final classification

3.5 Hyperparameter Optimization using GridSearchCV

According to this approach, numerous combinations of hyperparameters are tested, and the best one is selected. The hyperparameters to be optimised include the learning rate, the batch size, the number of filters, the kernel size, and the dropout rate. GridSearchCV

performs a comprehensive search over the specified parameter set to identify the optimal combination of hyperparameters, based on its effectiveness in cross-validation.

3.6 Data Preprocessing and Augmentation

Images were scaled to [0, 1] before training to ensure the model could converge. We divided the data into two sections: 90% for training and 10% for testing. Additional diversity and prevention of overfitting were achieved using the Keras Image Data Generator, which rotated, zoomed, shifted, and flipped the data horizontally. Such pre-processing techniques enhance the stability and generalization of the models on multiclass data. Where X represents the original collection of images, the augmented dataset X0 is modified to be in the form of:

$$X' = T(X)$$

where Trepresents operations such as rotation, scaling, and flipping.

3.7 Proposed Algorithm

i. Import the necessary libraries: NumPy, Pandas, Keras, Scikit-learn and TensorFlow's
ii. Load the dataset using Pandas.
iii. Split the dataset into training and testing sets using Scikit-learn'strain_test_split() function.
iv. Create a function that builds the CNN model with hyperparameters to be tuned.
v. Define the hyperparameters to be tuned and their values for GridSearchCV.
vi. Instantiate the GridSearchCV object, passing in the CNN model, hyperparameters to be tuned, and number of folds for cross-validation.
vii. Fit the GridSearchCV object to the training data.
viii. Evaluate the performance of the best model returned by GridSearchCV on the testing data.
ix. Compare the performance of the GridSearchCV-tuned model with the baseline models.
x. Draw conclusions and report the findings.

3.8 Proposed Flow layout

The flowchart in Fig. 3 shows a structured way to set up a convolutional neural network model. The first step is to load and preprocess the dataset into the right tensors. After that, it is split into training and testing sets. Then, a search is done over the possible values of different hyperparameters using a parameter grid for the training set to set up the model. The CNN model is trained, tweaked, and improved by adding new images. Before training, hyperparameter tuning is used to make the model even better. Once the model has been trained, it is used to make predictions based on new data. The last step is to look at and compare those results.

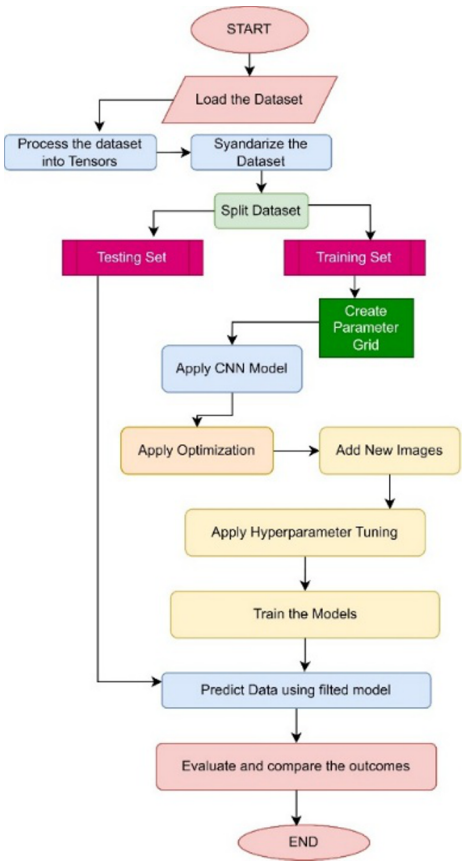

Fig 3: Proposed Flow Layout

3.9 Mathematical Representation of the Proposed Model

i. Convolution Operation in CNN

The convolution operation uses a filter (kernel) to detect spatial features in the input image by sliding across the image matrix. The resulting feature map shows how each filter captures different patterns, such as shapes, edges, or textures. This process helps the CNN model learn hierarchical image features that are useful for tasks like classification.

$$S(i,j) = (I * K)(i,j) = \sum_{m} \sum_{n} I(m,n)K(i-m, j-n)$$

Where I represents the input image, K represents the convolution kernel (filter) and $S(i,j)$ represents the resulting feature map after convolution

The convolution operation enables the CNN model to extract key spatial features from the input images.

ii. ReLU Activation Function

The Rectified Linear Unit (ReLU) activation function makes the neural network non-linear by passing only positive values and setting negative values to zero. This function speeds up the training process and helps with the problem of the gradient disappearing. ReLU makes deep convolutional neural networks better at learning as a result.

$$f(x) = \max(0, x)$$

iii. Softmax Classification Function

The Softmax function is used in the last layer of the CNN model to convert the output scores into a probability distribution over several classes. The output values indicate how likely it is that the input image belongs to a particular class. The predicted class label for the input image is the one with the highest probability.

$$P(y = i \mid x) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

Where: z_i= output score of class i, k= number of classes, $P(y = i \mid x)$= probability of class i

iv. DCGAN Objective Function

The objective function of the Generative Adversarial Network is like a minimax game between the generator and the discriminator. The generator creates fake images that look like real ones, and the discriminator tries to tell the difference between the two. Through this adversarial training process, the generator learns to produce realistic fake images that can be used to augment the dataset.

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Where: G= Generator network, D= Discriminator network, x= real data sample and z= random noise vector

Generator Loss : $L_G = -E[\log(D(G(z)))]$

The generator attempts to produce synthetic images that can fool the discriminator. The objective is to maximize the probability that generated images are classified as real by the discriminator.

Discriminator Loss : $L_D = -E[\log(D(x))] - E[\log(1 - D(G(z)))]$

The discriminator learns to distinguish between real MRI images and synthetic images produced by the generator by minimizing classification error.

4. Results and Discussion

The experiments were conducted to evaluate the effectiveness of the proposed deep learning framework for multiclass brain Tumor classification. The experiments were implemented in Python using TensorFlow 2.10 and Keras, with GPU acceleration. All MRI images were resized to 128 × 128 pixels and normalized to the [0, 1] range. The dataset, containing 5,712 training and 1,311 testing images, was further split 90:10 for validation. To address class imbalance and overfitting, DCGAN-based synthetic augmentation and standard techniques (rotation, shifting, flipping) were applied via the Image Data Generator. Models including a Custom CNN, BPNN, VGG19, and EfficientNetV2B3 were trained using the Adam optimizer and categorical cross-

entropy loss. We employed early stopping and batch sizes of 32 for up to 50 epochs. For transfer learning, we used ImageNet weights that had already been trained, along with modified classification heads. The hyperparameters were set to the best values. We evaluated performance by examining accuracy, precision, recall, F1 score, confusion matrices, and ROC curves.

4.1 Model Performance Comparison

Model Comparison

The model's performance improves across the measures. The base CNN prediction accuracy is 64.45 out of 100, and the F-score is 58.16, which, in other words, does not give it much predictive power. These numbers improved significantly when images generated by DCGAN were introduced, reaching 73.57 and 73.95, respectively. This demonstrates the significance of providing models with additional data to improve them. The best results came from using automated hyperparameter tuning, which achieved an accuracy of 77.86% and an F-score of 77.99. This change shows that data augmentation needs to be added to systematic optimisation to make classification models that are both accurate and stable.

Table 2. Model Performance output

Model	Optimizer	Accuracy	Recall	F1	Training_Time
Custom_CNN	Adam	88.787185	87.955701	88.126328	1581.369077
VGG19	Adam	88.634630	88.097858	88.142518	7898.150440
VGG19	Adagrad	87.871854	87.098039	86.996931	9514.549475
VGG19	SGD	77.955759	76.418845	73.350868	6994.383452
Custom_CNN	SGD	69.565217	68.724038	67.283076	1380.944834
Custom_CNN	Adagrad	65.369947	63.706972	63.309161	1453.447346
BPNN	SGD	54.462243	55.797749	48.371260	1203.568866
BPNN	Adagrad	49.199085	50.570806	41.610988	281.660662
BPNN	Adam	46.681922	48.385802	38.966520	300.608462
EfficientNet	SGD	42.944317	42.487654	35.842790	2089.574840
EfficientNet	Adagrad	42.791762	42.688272	36.727549	4344.115301
EfficientNet	Adam	36.918383	37.157407	28.451835	22078.592183

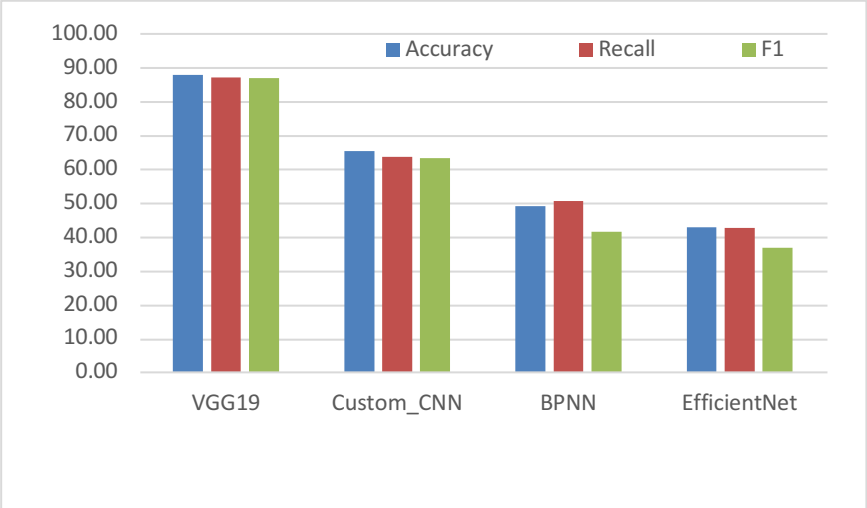

Fig 4: All models performance on Adagrad optimizer

The four neural network architectures, VGG19, Custom_CNN, BPNN, and EfficientNet, work very differently, as shown in Fig 4. The VGG19 model was the best for this classification task, with accuracy, recall, and F1-score all close to 90%. The Custom_CNN, on the other hand, performed only moderately (65–70%). The BPNN and EfficientNet performed much worse, with scores of about 45–50% and 35–40%, respectively. These results show a significant performance gap between traditional architectures and more advanced models. This indicates that VGG19 is better at handling the many different parts of the brain Tumor dataset.

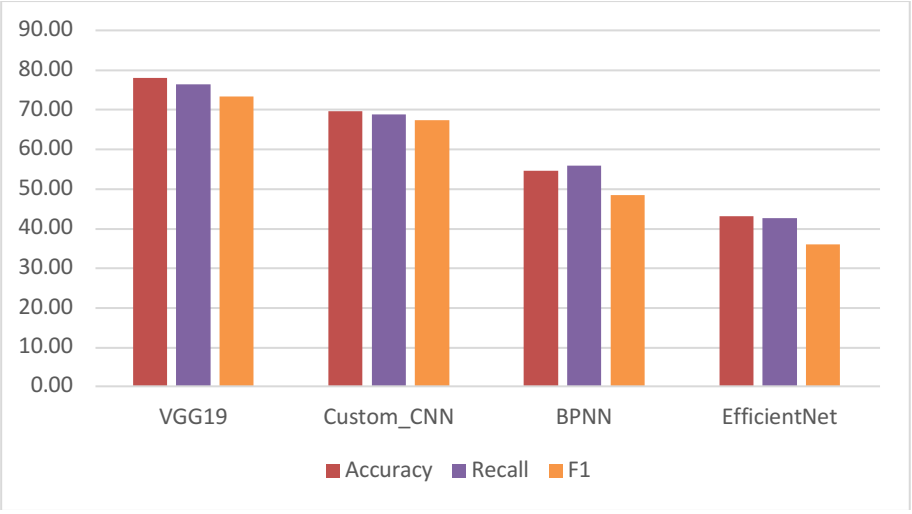

Fig :5 All models performance on SGD optimizer

Fig. 5 shows how well VGG19, Custom_CNN, BPNN, and EfficientNet performed on tests of accuracy, recall, and F1-score. VGG19 consistently scores around 80, indicating that it is more efficient than the others. The Custom_CNN ranks second, with results similar but somewhat lower. The BPNN, however, performs fairly well, with accuracy and recall of approximately 55%. However, its lower F1-score shows that it doesn't balance precision and recall well. EfficientNet performed the poorest, with all metrics falling below 45%, suggesting it is ill-suited for this specific dataset. Overall, the results establish VGG19 as the most reliable architecture, while the Custom_CNN serves as a viable alternative.

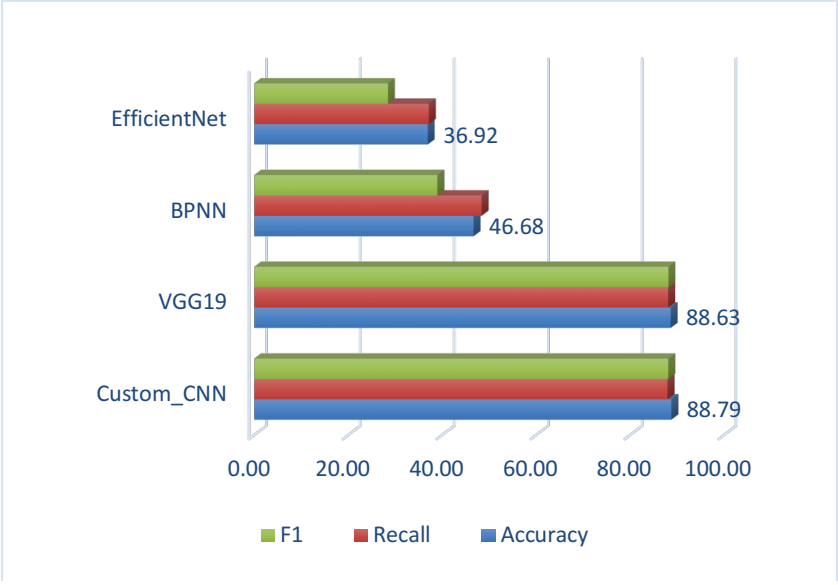

Fig 6: All models performance on Adam optimizer

The chart in Fig 6 shows a clear difference in performance when the Adam optimizer is used across all models. VGG19 and Custom_CNN both achieve very high scores, with accuracy, recall, and F1 scores all around 88%. This shows that they are very stable and balanced when Adam's adaptive learning is used. BPNN, on the other hand, performs only moderately, with accuracy and recall values around 47% and a slightly lower F1 score, indicating it doesn't generalise well. EfficientNet is way behind, with all of its scores below 38%. This shows that Adam doesn't make up for the fact that it doesn't fit this dataset as well. Overall, Adam helps show how VGG19 and Custom_CNN are better than BPNN and EfficientNet in this test.

4.2 Confusion Matrix Analysis

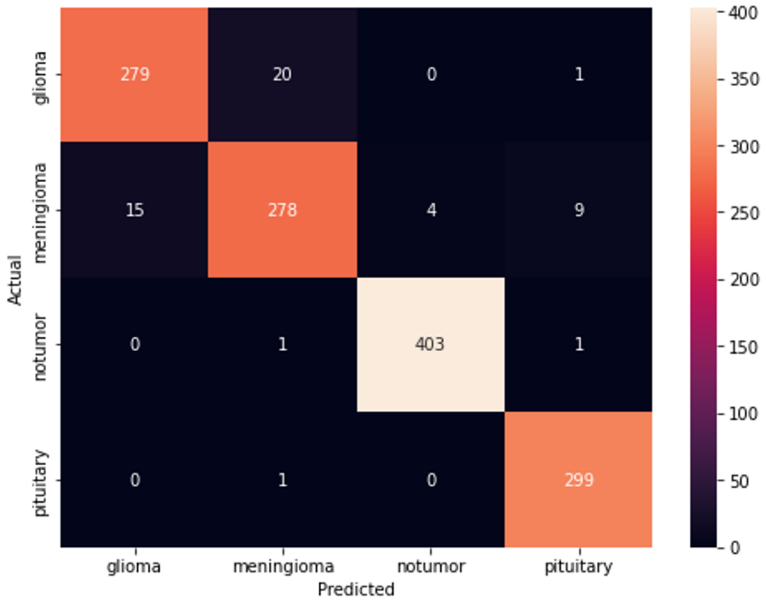

Fig: 7 Confusion Matrix for Brain Tumor Classification.

The confusion matrix evaluates how well the classification performs across four groups: glioma, meningioma, no Tumor, and pituitary. The model demonstrates high precision for "No Tumor" (403 correct) and "Pituitary" (299 correct). Glioma (279 correct) and Meningioma (278 correct) yield commendable results; however, there is significant mutual misclassification between the two. This pattern indicates that both Glioma and Meningioma share certain radiological appearances, highlighting the need to focus on a specific aspect to enhance the architecture in the future to assist in distinguishing between the two categories.

5. Conclusion

This paper has proposed a framework that integrates purposeful hyperparameter optimisation and data augmentation patterns for DCGAN to enhance the classification of brain Tumor MRIs. The approach addresses two major issues in medical imaging, namely class imbalance and the failure to select appropriate parameters. The experimental results indicate that the starting point is a strong CNN. However, it is significantly more accurate when DCGAN-generated fake images are added to the dataset. The model was further enhanced by performing grid search optimisation, where the maximum accuracy reached 77.86. It outperformed the baseline and the models that were only improved. These findings substantiate the fact that generative augmentation, coupled with systematic tweaking, can produce more valid diagnostic tools. To promote model generalisability to different medical imaging tasks, future studies will focus on advanced optimisation methods, such as evolutionary and Bayesian methods.

References

1. Nessa "Cervical cancer screening data from the case-based national electronic registry in Bangladesh," *BMC Global and Public Health*, vol. 3, no. 34, 2025, doi: 10.1186/s44263-025-00145-x.
2. M. Ogoncho, S. Wakasiaka, I. G. Mageto, and M. Chege, "Effect of a Nurse-Led Educational Intervention on the Knowledge, Perceptions and Uptake of Cervical Cancer Screening among HIV-Infected Women in Kenya," *Asian Pac. J. Cancer Prev.*, vol. 26, no. 5, pp. 1591-?, 2025, doi: 10.31557/APJCP.2025.26.5.1591.
3. M. Zhang., "Effects of a theory driven and culturally tailored educational program on promoting cervical cancer screening in rural populations," *Sci. Rep.*, vol. 15, no. 18540, 2025, doi: 10.1038/s41598-025-02600-z.
4. Y. Umezaki., "Evaluation of Cervical Cancer Screening in Japan: Challenges and Future Directions for Negative Intraepithelial Lesion or Malignancy/High-Risk Human Papillomavirus Positive Case Management," *Curr. Oncol.*, vol. 32, no. 6, p. 295, May 2025, doi: 10.3390/curroncol32060295.
5. N. Upadhyai, R. Upadhyai, A. Mittal, and K. Gupta, "Factors influencing the attitudes of rural women in Uttarakhand towards cervical cancer screening," *Indian J. Forensic Community Med.*, vol. 12, no. 3, pp. 173-179, 2025.
6. E. Wibowo, "Forecasting water quality through machine learning and hyperparameter optimization," *Indonesian J. Elect. Eng. Comput. Sci.*, vol. 33, no. 1, pp. 496-506, Jan. 2024, doi: 10.11591/ijeecs.v33.i1.pp496-506.
7. T. Baştuğ., "Fully Automated Detection of the Appendix Using U-Net Deep Learning Architecture in CT Scans," *J. Clin. Med.*, vol. 13, no. 19, p. 5893, Oct. 2024, doi: 10.3390/jcm13195893.
8. L. González-Castro., "Impact of Hyperparameter Optimization to Enhance Machine Learning Performance: A Case Study on Breast Cancer Recurrence Prediction," *Appl. Sci.*, vol. 14, no. 13, p. 5909, Jul. 2024, doi: 10.3390/app14135909.
9. B. Ferro, S. Bayer, S. Brailsford, and H. Smith, "Improving intervention design to promote cervical cancer screening among hard-to-reach women: assessing beliefs and predicting individual attendance probabilities in Bogotá, Colombia," *BMC Women's Health*, vol. 22, no. 212, 2022, doi: 10.1186/s12905-022-01800-3.
10. G. S. R. E. Langberg., "Towards a data-driven system for personalized cervical cancer risk stratification," *Sci. Rep.*, vol. 12, no. 12083, 2022, doi: 10.1038/s41598-022-16361-6.
11. A. Sharma and A. Sharma, "KNN-DBSCAN: Using k-nearest neighbor information for parameter-free density based clustering," presented at the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2017, 2017, pp. 787–792. doi: 10.1109/ICICICT1.2017.8342664.
12. K. Magade and A. Sharma, "Significant role of IoT in Cyber-Physical Systems, Context Awareness, and Ambient Intelligence," in *The Next Generation Innovation in IoT and Cloud Computing with Applications*, 2024, pp. 16–34. doi: 10.1201/9781003406723-2.
13. V. Kumar, S. K. Gupta, A. Hussain, and A. Sharma, "A Systematic Approach to Prevent Threats Using IDS in IoT Based Devices," *GMSARN International Journal*, vol. 19, no. 1, pp. 107–112, 2025.
14. A. Hussain, A. J. Obaid, G. Tyagi, and A. Sharma, *The Next Generation Innovation in IoT and Cloud Computing with Applications*. in The Next Generation Innovation in IoT and Cloud Computing with Applications. 2024, p. 173. doi: 10.1201/9781003406723.

15. P. Garg and A. Sharma, "A distributed algorithm for local decision of cluster heads in wireless sensor networks," presented at the IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017, 2018, pp. 2411–2415. doi: 10.1109/ICPCSI.2017.8392150.
16. A. Jopek "Improving platelet- RNA -based diagnostics: a comparative analysis of machine learning models for cancer detection and multiclass classification," Molecular Oncology, vol. 18, no. 11, pp. 2743–2754, Nov. 2024, doi: 10.1002/1878-0261.13689.
17. S. M. Hussain ,"Hyper Parameter Tuning In Convolutional Neural Networks for Precise Tumor Image Classification," Mar. 11, 2025, Computer Science and Mathematics. doi: 10.20944/preprints202503.0734.v1.
18. Nhlapho, W., Atemkeng, M., Brima, Y., & Ndogmo, J.-C. (2024). Bridging the Gap: Exploring Interpretability in Deep Learning Models for Brain Tumor Detection and Diagnosis from MRI Images. Information, 15(4), 182. https://doi.org/10.3390/info15040182
19. Ilani, M.A., Shi, D. & Banad, Y.M. T1-weighted MRI-based brain tumor classification using hybrid deep learning models. Sci Rep 15, 7010 (2025). https://doi.org/10.1038/s41598-025-92020-w
20. Mathivanan, S.K., Srinivasan, S., Koti, M.S. A secure hybrid deep learning framework for brain tumor detection and classification. J Big Data 12, 72 (2025). https://doi.org/10.1186/s40537-025-01117-6
21. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2672–2680.
22. Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations (ICLR)*.
23. Litjens, G., Kooi, T., Bejnordi, B., Setio, A., Ciompi, F., Ghafoorian, M., van der Laak, J., van Ginneken, B., & Sánchez, C. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88. https://doi.org/10.1016/j.media.2017.07.005
24. Esteva, A., Kuprel, B., Novoa, R., Ko, J., Swetter, S., Blau, H., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542, 115–118. https://doi.org/10.1038/nature21056
25. Shorten, C., & Khoshgoftaar, T. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 60. https://doi.org/10.1186/s40537-019-0197-0
26. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
27. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281–305.
28. Snoek, J., Larochelle, H., & Adams, R. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25.
29. Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25.
30. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.