Generating Brain MRI Images with DCGAN & train model for tumour detection

Rishav Jain, Debolina Ghosh, Chirag Singla, Suhawni

Department of Computer Science

Thapar Institute of Engineering and Technology Patiala, India

rjain2_be21@thapar.edu, dghosh_be21@thapar.edu, csingla1_be21@thapar.edu, ssuhawni_be21@thapar.edu

Abstract

Brain tumor detection from MRI scans is critical for effective diagnosis and treatment planning. This study addresses the challenge of limited annotated medical datasets by using Deep Convolutional Generative Adversarial Networks (DC-GANs) to generate realistic synthetic MRI images. These images are combined with real data to train a Convolutional Neural Network (CNN) for tumor classification. The proposed approach improves model accuracy and robustness, as evaluated through metrics like accuracy, F1-score, and AUC-ROC, while the quality of synthetic images is validated using the Fréchet Inception Distance (FID). This work demonstrates the potential of combining generative and classification models to enhance automated tumor detection, with future scope including advanced architectures and multi-modal imaging integration.

Keywords

Brain tumor detection, MRI scans, DC-GAN, CNN, synthetic medical images, tumor classification, deep learning, medical imaging, dataset augmentation, Fréchet Inception Distance, automated diagnosis, image generation, neural networks, transfer learning, medical image analysis.

I. Introduction

The detection of brain tumors in MRI scans is a critical task in medical imaging, as accurate identification and classification are essential for effective diagnosis, treatment planning, and improving patient outcomes. Manual annotation of medical images is time-consuming and requires expert knowledge, making it difficult to acquire large-scale datasets for training machine learning models. Generative models, such as Generative Adversarial Networks (GANs), have revolutionized the field of data augmentation by enabling the creation of realistic synthetic images. These synthetic images can be used to supplement existing datasets, addressing the issue of limited data availability. In this study, we utilize a Deep Convolutional Generative Adversarial Network (DC-GAN) to generate high-quality synthetic brain MRI images. These images are then combined with real MRI data to train a Convolutional Neural Network (CNN) for the detection and classification of brain tumors.

II. Related Work

This section reviews key literature addressing challenges and methodologies related to imbalanced datasets, data augmentation, and the application of Convolutional Neural Networks (CNNs) in image classification.

1. **Generative Models in Medical Imaging**
   The introduction of Generative Adversarial Networks (GANs) by Goodfellow et al. (2014) marked a significant milestone in generative modeling. GANs employ a generator-discriminator framework, where the generator creates synthetic data to fool the discriminator, which learns to distinguish real from synthetic data. This adversarial training process has been widely adopted in medical imaging to augment datasets by generating realistic synthetic images.
2. **Improvements in GAN models**
   Radford et al. (2015) extended GANs with Deep Convolutional GANs (DC-GANs), which improved training stability and generated high-quality images by leveraging convolutional layers. DC-GANs have been successfully applied in medical imaging tasks, such as generating synthetic MRI images, to address the scarcity of annotated datasets. Shin et al. (2018) further demonstrated the utility of GANs in medical image augmentation, showing that synthetic images enhanced the performance of tumor detection models, especially when training data was limited.
3. **Innovations in GAN Models in recently.**
   Recent innovations, such as CycleGANs (Zhu et al., 2017), have enabled unpaired image-to-image translation, allowing the generation of synthetic data across different imaging modalities (e.g., MRI to CT). This technique has further expanded the scope of generative models in medical imaging by improving dataset diversity and robustness without requiring paired data.
4. **CNN Applications in Image Classification**
   Krizhevsky et al. (2012) introduced AlexNet, a deep CNN that achieved groundbreaking performance in image classification, inspiring subsequent medical imaging applications. Ronneberger et al. (2015) proposed U-Net, an encoder-decoder CNN architecture with skip connections, which became a standard for biomedical image segmentation. U-Net has been widely adopted for tumor segmentation in MRI scans due to its ability to perform well even with small datasets.

III. Flow Diagram of System Architecture

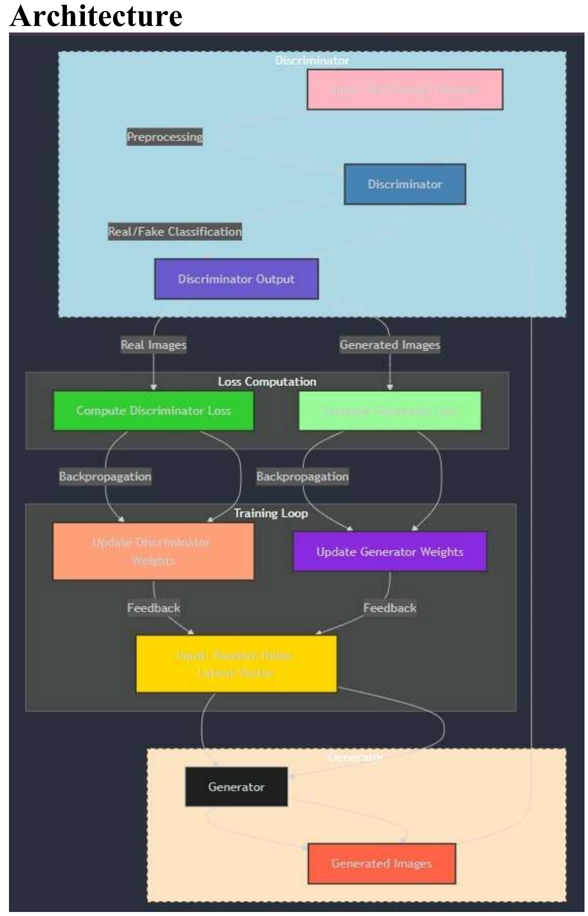

Fig 1. Flow Diagram of system architecture

IV. Methodology Adopted

Dataset Preparation

- Acquire a dataset of brain MRI images, ideally with tumor annotations.
- Preprocess the images (format conversion, normalization, data augmentation).
- Split the dataset into training (50%), validation (30%), and testing (20%).
- Ensured random shuffling for unbiased data distribution across subsets.

Image Preprocessing and Augmentation

- Resized all images to 128 x 128 pixels for consistency.
- Applied augmentation (random rotations, shifts, zooms, and flips) to training images for robustness and generalization.
- Rescaled validation and test sets without augmentation for evaluation.

DC-GAN Implementation:

- Designed Generator and Discriminator networks using convolutional and deconvolutional layers.
- Trained the GAN using an adversarial process:
  - Generator creates synthetic images.
  - Discriminator distinguishes between real and synthetic images.
- Evaluated synthetic images using visual inspection and metrics like FID (Fréchet Inception Distance).

Tumour Detection Model Training:

- Choose a suitable CNN architecture for tumor detection e.g., (ResNet50, EfficientNet, etc.)
- Train the model using a combination of real and synthetic images.
- Evaluate the model's performance on a held-out test set.

Results:
Achieved a test accuracy of 92.16% and a loss of 0.2329.

```
leaky_re_lu_33 (LeakyReLU)   (None, 32, 32, 128)       0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 16, 16, 256)       295168
_________________________________________________________________
leaky_re_lu_34 (LeakyReLU)   (None, 16, 16, 256)       0
_________________________________________________________________
flatten (Flatten)            (None, 65536)             0
_________________________________________________________________
dropout_4 (Dropout)          (None, 65536)             0
_________________________________________________________________
dense_9 (Dense)              (None, 1)                 65537
=================================================================
Total params: 582,785
Trainable params: 582,785
Non-trainable params: 0
_________________________________________________________________


Model: "generator"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_10 (Dense)             (None, 262144)            26476544
_________________________________________________________________
leaky_re_lu_35 (LeakyReLU)   (None, 262144)            0
_________________________________________________________________
reshape_5 (Reshape)          (None, 32, 32, 256)       0
_________________________________________________________________
conv2d_transpose_15 (Conv2DT (None, 64, 64, 128)       524416
_________________________________________________________________
leaky_re_lu_36 (LeakyReLU)   (None, 64, 64, 128)       0
_________________________________________________________________
conv2d_transpose_16 (Conv2DT (None, 128, 128, 128)     262272
_________________________________________________________________
leaky_re_lu_37 (LeakyReLU)   (None, 128, 128, 128)     0
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 128, 128, 1)       2049
=================================================================
Total params: 27,265,281
Trainable params: 27,265,281
Non-trainable params: 0
```

Fig. 2. DCGAN model Summary

```
_________________________________________________________________
The Combined Network:

Model: "gan_model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 100)]             0
_________________________________________________________________
generator (Sequential)       (None, 128, 128, 1)       27265281
_________________________________________________________________
discriminator (Sequential)   (None, 1)                 582785
=================================================================
Total params: 27,848,066
Trainable params: 27,265,281
Non-trainable params: 582,785
_________________________________________________________________
```

Fig. 3 DCGAN model Summary

V.Experiments and Results

Dataset Details and Data Split

Type: Brain MRI dataset.

Categories: Images categorized into

1. Yes

2. No

Data Split for CNN model:

1. **Training Set:** 50% of the total images.
2. **Validation Set:** 30% of the total images.
3. **Test Set:** 20% of the total images.

Computational Environment

Platform:

1. **Environment:** Kaggle Notebook (cloud-based with GPU support).
2. **Processor:** Kaggle GPU P100
3. **RAM:** 16GB.
4. **Storage:** High-speed SSD provided by Kaggle.
5. **Operating System:** Windows

Deep Learning Framework Used: TensorFlow 2.x with Keras API.

Advantages of GPU: Accelerates matrix operations and convolutional computations, reducing training time.

CNN Model Training Details

Hyperparameters:

1. **Batch Size:** 32 images per batch.
2. **Image Size:** Resized to 224x224 pixels.
3. **Epochs:** Up to 10(with early stopping at 50 epochs if no validation loss improvement).
4. **Steps per Epoch:** 200 steps (validation: 100 steps).
5. **Optimizer:** Adam optimizer with a learning rate of 0.001.
6. **Loss Function:** Categorical cross-entropy for multi-class classification.
7. **Activation Functions:**
   - **ReLU:** Applied in hidden layers for non-linearity.
   - **Softmax:** Used in the output layer for class probabilities.

CNN Model Performance

Test Metrics:

1. **Test Accuracy:** 92.16%.
2. **Test Loss:** 0.2329.

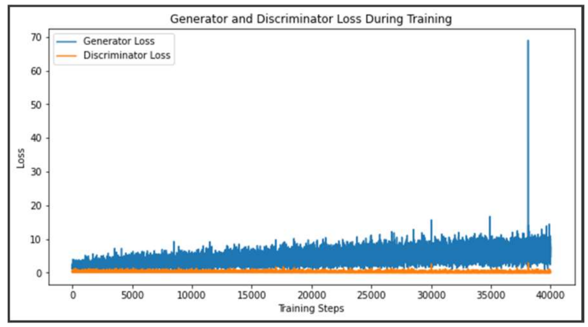

Fig. 4 Training and Validation Accuracy

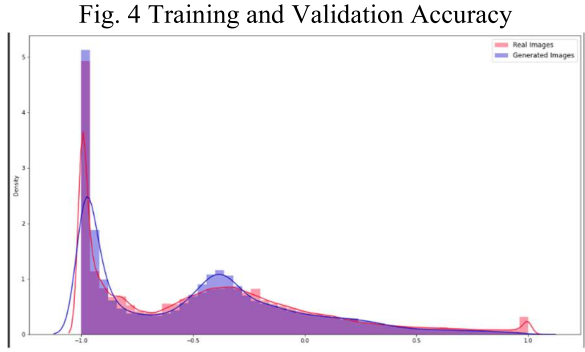

Fig. 5 Training and Validation Loss

Classification Metrics:

3. **Overall Accuracy:** 92.00%.

Training and Inference Times:

4. **Training Time:** ~5 hours.
5. **Testing Time:** ~3 minutes.
6. **Inference Time:** ~0.2 seconds per image.

GAN-Based Tumor Image Synthesis and Classification(DenseNet121)

Dataset Handling:

- Brain MRI images categorized into "Yes" (tumor present) and "No" (no tumor).
- Data split: 80% training and 20% validation/testing.

- Resizing all images to 128x128 pixels.

- Normalization using(Pixel Value – 12.75)/12.75

Model Configuration:

1. GAN-Based Model (DCGAN):

Generator:

- 3 Conv2DTranspose layers with LeakyReLU activations.

- Output: Tanh activation for generating 128x128x1 images.

Discriminator:

- 4 Conv2D layers with LeakyReLU activations.
- Flatten layer followed by Dense with Sigmoid activation.

Pre-Trained Models

VGG16,ResNet50,EfficientNetB0, DenseNet121:

- Frozen base layers for feature extraction.

- Custom Dense and Dropout layers added for binary classification.

Performance Metrics:

- **Test Accuracy:** 92.16%.
- **Test Results:** High precision, recall, and F1-scores across all classes.

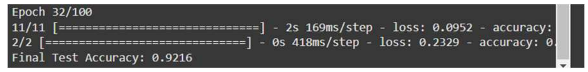

Fig 6. Validation results of Densenet121

VI. Conclusion and Future Scope

This project demonstrates the potential of using Deep Convolutional Generative Adversarial Networks (DC-GANs) and Convolutional Neural Networks (CNNs) in the field of medical imaging, specifically for brain tumor detection. By generating synthetic MRI images, we addressed the challenge of limited annotated datasets, enhancing the training process for our tumor detection model. The integration of realistic synthetic data improved the model's accuracy and robustness, making it a valuable tool for aiding radiologists in clinical settings.
The proposed model can be further improved by incorporating advanced techniques, such as fine-tuning learning processes and leveraging Transformer-based architectures to enhance image understanding and handle subtle differences in complex cases. Expanding the dataset with diverse images from various hospitals and patient groups will improve generalizability, while collaborations with medical institutions can provide high-quality, expert-reviewed data for better real-world applicability. Additionally, integrating multiple imaging modalities (e.g., MRIs, CTs, PET scans) or developing methods to translate between scan types can offer a more comprehensive diagnostic tool, enabling more accurate and reliable patient care.

VII. References

1. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural*

Information Processing Systems (NeurIPS), vol. 27, pp. 2672–2680, 2014.
2. A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learning Representations* (ICLR), 2016.
3. H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1285–1298, May 2018.
4. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (NeurIPS), vol. 25, pp. 1097–1105, 2012.
5. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention* (MICCAI), vol. 9351, pp. 234–241, 2015.
6. G. Litjens, T. Kooi, B. E. Bejnordi, A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.
7. A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, Feb. 2017.