**GAN-ENHANCED 2D-TO-3D CONVERSION USING**

**DEEP LEARNING FOR MRI SCANS**

**PROJECT REPORT**

**Submitted by**

**Poussin Sunil Kumar (21CS1032)**

**Sudhanshu Kumar (21CS1050)**

**Vinay Kumar Swami (21CS1060)**

**Under the guidance of**

**Dr. K. Vivekanandan**

**Professor**

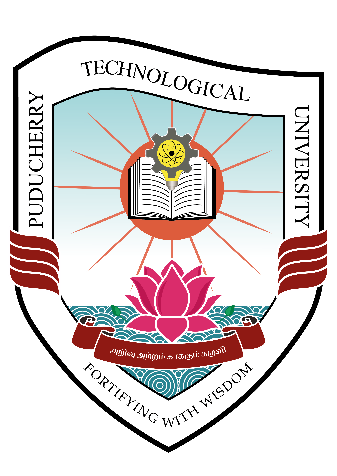
**Department of Computer Science and Engineering**

**to the Puducherry Technological University, in partial fulfillment of the requirement for the award of degree**

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**PUDUCHERRY TECHNOLOGICAL UNIVERSITY**

**PUDUCHERRY – 605 014**

**May 2025.**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**PUDUCHERRY TECHNOLOGICAL UNIVERSITY**

**PUDUCHERRY – 605 014.**

**BONAFIDE CERTIFICATE**

This is to certify that the Project work titled **“GAN-Enhanced 2D-To-3D Conversion Using Deep Learning for MRI Scans”** is a bonafide work done by **Poussin Sunil Kumar (21CS1032), Sudhanshu Kumar (21CS1050) and Vinay Kumar Swami (21CS1060)** in partial fulfillment for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of the **Puducherry Technological University** and that this work has not been submitted for the award of any other degree of this/any other institution.

**Project Guide Head of the Department**

**(Dr. K. Vivekanandan) (Dr. Ilavarasan E)**

*Submitted for the University Examination held on\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**Internal Examiner External Examiner**

**ACKNOWLEDMENT**

I am deeply indebted to **Dr. K. Vivekanandan,** **Professor** Department of Computer Science and Engineering, Puducherry Technological University, Puducherry, for his valuable guidance throughout this project.

I also express my heart-felt gratitude to **Dr. Ilavarasan E,** Professor & Head (CSE) for giving constant motivation in succeeding my goal.

With profoundness I would like to express my sincere thanks to **Dr. S. Mohan,** the Vice-Chancellor, Puducherry Technological University, for his kindness in extending the infrastructural facilities to carry out my project work successfully.

I would be failing in my duty if I do not acknowledge the efforts of the Project Co-ordinator, **Dr. N. Sivakumar**, and the Project Review Panel members, viz. **Dr. F. Sagayaraj Francis, Dr. J. Jayabharathy, Dr. K. Sathiamurthy, Dr. N. Sivakumar, and Dr. P. Salini** for shaping our ideas and constructive criticisms during project review.

I also express my thanks to all the **Faculty** and **Technical Staff** members of the CSE department for their timely help and the **Central Library** for facilitating useful reference materials.

I would be failing in my duty if I don’t acknowledge the immense help extended by my friends, who have always been with me in all my trails and tribulations and encouraging me to complete.

**Poussin Sunil Kumar**

**Sudhanshu Kumar**

**Vinay Kumar Swami**

### ****ABSTRACT****

The demand for high-resolution MRI images in clinical diagnosis has motivated the development of advanced super-resolution techniques. Traditional interpolation methods fail to recover fine anatomical details, limiting their effectiveness for critical medical analysis. This project introduces a deep learning-based approach utilizing a modified ESRGAN architecture, named NESRGAN, to enhance the resolution of individual low-quality MRI slices. The generator incorporates residual blocks and pixel shuffle layers to improve spatial fidelity, while the discriminator enforces image realism through adversarial learning. The model is trained using a combination of adversarial and pixel-wise L1 losses, ensuring sharp and structurally accurate outputs. After super-resolution, the enhanced slices can be stacked for potential 3D MRI reconstruction. This framework offers an efficient and scalable solution to improve MRI image quality, supporting better visualization and aiding downstream tasks such as tumor analysis.

In the initial phase of this project, the model has been trained and evaluated using a curated set of paired low- and high-resolution MRI slices. Preliminary results demonstrate that NESRGAN outperforms traditional interpolation techniques and shows promising improvement over the original ESRGAN baseline. Future work will involve fine-tuning the model on larger and more diverse datasets, validating its robustness across various clinical scenarios, and optimizing the architecture for real-time deployment in medical environments.

The primary purpose of this work is to bridge the quality gap between low-resolution MRI scans and the high-resolution standards required for accurate clinical interpretation. By leveraging deep learning-based super-resolution, this project aims to reduce dependency on costly high-end MRI hardware while still achieving diagnostically useful image clarity. Additionally, the enhanced 2D slices produced by the model serve as a foundation for constructing high-quality 3D MRI volumes, enabling more comprehensive visualization and analysis for tasks such as tumor detection and surgical planning.

**TABLE OF CONTENT**

| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| --- | --- | --- |
|  | **BONAFIDE CERTIFICATE** | ***ii*** |
|  | **ACKNOWLEDGEMENT** | ***iii*** |
|  | **ABSTRACT** | ***iv*** |
|  | **LIST OF FIGURES** | ***viii*** |
|  | **LIST OF TABLES** | ***ix*** |
|  | **LIST OF ABBREVIATIONS** | ***ix*** |
|  | **LIST OF SYMBOLS** | ***x*** |
| 1 | **INTRODUCTION** | 1 |
|  | 1.1 OVERVIEW | 1 |
|  | 1.2 OBJECTIVE OF STUDY | 1 |
|  | 1.3 MOTIVATION/ NEED FOR THE STUDY | 2 |
|  | 1.4 ORGANIZATION OF THE CHAPTERS | 2 |
| 2 | **LITERATURE REVIEW** | 3 |
|  | 2.1 LITERATURE REVIEW BASED ON VARIOUS RESEARCH PAPERS | 3 |
|  | 2.2 SUGGESTIONS BASED ON LITERATURE REVIEW | 5 |
| 3 | **EXISTING WORK** | 6 |
|  | 3.1 GAN-BASED 3D MRI RECONSTRUCTION | 6 |
|  | 3.2 TECHNIQUES USED IN EXISTING WORK | 6 |
|  | 3.2.1 Generative Adversarial Networks (GANs) | 6 |
|  | 3.2.2 Enhanced Super-Resolution GAN (ESRGAN) | 7 |
|  | 3.2.3 Slice-by-Slice Reconstruction and Stacking | 7 |
|  | 3.3 EXISTING ARCHITECTURE | 7 |
|  | 3.4 ARCHITECTURE DESCRIPTION | 8 |
|  | 3.5 ALGORITHM | 8 |
|  | 3.6 MODULES USED | 9 |
|  | 3.7 LIMITATIONS OF EXISTING WORK | 9 |
|  | 3.8 RESULT | 10 |
| 4 | **PROPOSED WORK** | 11 |
|  | 4.1 3D MRI RECONSTRUCTION USING NESRGAN | 11 |
|  | 4.2 PROPOSED ARCHITECTURE | 12 |
|  | 4.3 3D RECONSTRUCTION BY STACKING MRI SLICES | 12 |
|  | 4.4 TECHNIQUES USED IN PROPOSED WORK | 13 |
|  | 4.4.1 Lightweight Generator with Residual Blocks | 13 |
|  | 4.4.2 Efficient Upsampling via PixelShuffle | 13 |
|  | 4.4.3 Hybrid Loss Function | 13 |
|  | 4.4.4 Shallow Discriminator with Global Awareness | 14 |
|  | 4.4.5 Noise Reduction and Structure Preservation | 14 |
|  | 4.5 ALGORITHM | 14 |
|  | 4.6 MODULES USED | 15 |
|  | 4.7 RESULT | 15 |
| 5 | **SIMULATION RESULTS/ EXPERIMENTAL RESULTS (Existing Work & Proposed Work)** | 16 |
|  | 5.1 DATASET DESCRIPTION | 16 |
|  | 5.2 PERFORMANCE METRICS | 17 |
|  | 5.3 EXPERIMENTAL SETUP | 18 |
|  | 5.3.1 Hardware Configuration | 18 |
|  | 5.3.2 Software and Libraries | 18 |
|  | 5.4 RESULT | 18 |
|  | 5.4.1 Input & Output for Existing Model | 18 |
|  | 5.4.2 Input & Output for Proposed Model | 19 |
|  | 5.5 GRAPH | 20 |
|  | 5.5.1 PSNR (Peak Signal-to-Noise Ratio) | 21 |
|  | 5.5.2 SSIM (Structural Similarity Index) | 21 |
|  | 5.5.3 MSE (Mean Squared Error) | 22 |
|  | 5.5.4 RMSE (Root Mean Squared Error) | 22 |
|  | 5.5.5 LPIPS (Learned Perceptual Image Patch Similarity) | 23 |
|  | 5.6 AVERAGE COMPARISON OF EVALUATION METRICS | 23 |
|  | 5.7 INFERENCES | 24 |
| **IMPLEMENTATION RESULTS OF EXISTING AND PROPOSED WORK RESULTS WITH SNAPSHOTS** | | 25 |
| IMPLEMENTATION RESULTS OF EXISTING WORK | | 25 |
| IMPLEMENTATION RESULTS PROPOSED WORK | | 28 |
| **CONCLUSION & FUTURE ENHANCEMENTS** | | 33 |
| CONCLUSION | | 33 |
| FUTURE ENHANCEMENTS | | 34 |
| **References** | | 36 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO**. |
| Figure 3.1 | EXISTING ARCHITECTURE FOR 3D MRI RECONSTRUCTION | 7 |
| Figure 3.2 | ENHANCEMENT OF MRI SLICE USING N-ESRGAN MODEL | 10 |
| Figure 4.1 | PROPOSED MODEL FOR 3D BRAIN MRI RECONSTRUCTION | 12 |
| Figure 4.2 | THE PROCESS OF 3D RECONSTRUCTION BY STACKING MRI SLICES | 13 |
| Figure 4.3 | ENHANCEMENT OF MRI SLICE USING N-ESRGAN MODEL | 15 |
| Figure 5.1 | MORE THAN 100 SLICES OF 3 VIEWS OF MRI IS TAKEN AS INPUT | 17 |
| Figure 5.2 | OUTPUT PRODUCED BY ESRGAN MODEL | 19 |
| Figure 5.3 | ENHANCEMENT OF SINGLE MRI USING NESRGAN MODEL | 19 |
| Figure 5.4 | MORE THAN 100 SLICES OF 3 VIEWS OF MRI IS TAKEN AS INPUT | 20 |
| Figure 5.5 | OUTPUT PRODUCED BY NESRGAN MODEL | 20 |
| Figure 5.6 | LINE GRAPH COMPARING PSNR VALUES OF ESRGAN VS. N-ESRGAN | 21 |
| Figure 5.7 | LINE GRAPH COMPARING SSIM VALUES OF ESRGAN VS. N-ESRGAN | 21 |
| Figure 5.8 | LINE GRAPH COMPARING MSE VALUES OF ESRGAN VS. N-ESRGAN | 22 |
| Figure 5.9 | LINE GRAPH COMPARING RMSE VALUES OF ESRGAN VS. N-ESRGAN | 22 |
| Figure 5.10 | LINE GRAPH COMPARING LPIPS VALUES OF ESRGAN VS. N-ESRGAN | 23 |
| Snapshot 1 | IMPLEMENTATION OF EXISTING WORK | 28 |
| Snapshot 2 | IMPLEMENTATION OF PROPOSED WORK | 32 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TITLE** | **PAGE NO**. |
| TABLE 5.1 | COMPARISON OF EVALUATION METRICS BETWEEN N-ESRGAN AND ESRGAN | 23 |

**LIST OF ABBREVIATION**

|  |  |
| --- | --- |
| **ABBREVIATION** | **DESCRIPTION** |
| BCE | Binary Cross Entropy |
| CNN | Convolutional Neural Network |
| CT | Computed Tomography |
| CUDA | Compute Unified Device Architecture |
| DL | Deep Learning |
| ESRGAN | Enhanced Super-Resolution Generative Adversarial Network |
| GAN | Generative Adversarial Network |
| GPU | Graphics Processing Unit |
| HR | High Resolution |
| LPIPS | Learned Perceptual Image Patch Similarity |
| LR | Low Resolution |
| ML | Machine Learning |
| MRI | Magnetic Resonance Imaging |
| N-ESRGAN | Noise Enhanced Super-Resolution Generative Adversarial Network |
| PET | Positron Emission Tomography |
| PNG | Portable Network Graphics |
| PSNR | Peak Signal-to-Noise Ratio |
| RAM | Random Access Memory |
| RMSE | Root Mean Square Error |
| SR | Super-Resolution |
| SSIM | Structural Similarity Index Measure |

**LIST OF SYSMBOLS**

| **Symbol** | **Description** |
| --- | --- |
| μₓ | Mean of the original image |
| μᵧ | Mean of the reconstructed image |
| σₓ² | Variance of the original image |
| σᵧ² | Variance of the reconstructed image |
| σₓᵧ | Covariance between original and reconstructed images |
| C1 | Constant to stabilize luminance term in SSIM formula |
| C2 | Constant to stabilize contrast-structure term in SSIM formula |
| I(i,j) | Pixel value at position (i,j) in the original image |
| K(i,j) | Pixel value at position (i,j) in the reconstructed (enhanced) image |
| m | Number of rows in the image |
| n | Number of columns in the image |

**CHAPTER I**

**INTRODUCTION**

* 1. **OVERVIEW**

The growing adoption of 3D technologies across fields such as medical imaging, augmented reality, and virtual simulations has amplified the demand for high-quality 3D content. However, the vast majority of existing resources remain in 2D format, leading to a substantial gap in readily available 3D media. Conventional 2D-to-3D conversion techniques, especially in the medical field, are often manual and require intensive effort, time, and cost. These limitations pose significant challenges in meeting the real-time demands of modern diagnostic and imaging systems.

To address this challenge, the proposed report presents a deep learning-based solution that leverages a 2D Generative Adversarial Network (GAN) for super-resolving MRI slices. By enhancing each individual slice and then stacking them, the method reconstructs a high-resolution 3D MRI scan. This approach not only improves the clarity and detail of individual slices but also facilitates a more accurate and detailed volumetric representation, which is vital for medical analysis and diagnosis.

* 1. **OBJECTIVE OF THE STUDY**

The primary objective of this work is to develop a robust and efficient 3D MRI reconstruction pipeline that transforms 2D MRI slices into high-quality 3D volumes using a GAN-based super-resolution model. This system aims to enhance the resolution and structural integrity of individual MRI slices prior to stacking by employing deep learning-based super-resolution techniques. Furthermore, it seeks to optimize the 3D reconstruction process by minimizing artifacts and misalignments, thereby ensuring a more accurate spatial representation. The project also focuses on improving computational efficiency by optimizing model performance and reducing redundant operations throughout the pipeline. Lastly, it emphasizes the use of advanced noise reduction techniques that preserve critical anatomical structures, ensuring the reconstructed 3D MRI retains both clarity and clinical relevance.

* 1. **MOTIVATION/ NEED FOR THE STUDY**

The motivation behind this study arises from the critical need for accurate and high-quality 3D medical imaging, especially in clinical diagnosis and surgical planning. Traditional manual or interpolation-based 3D MRI reconstruction techniques often suffer from low resolution, structural inconsistency, and time inefficiency. With the rapid advancement of AI and deep learning, particularly GAN-based models, there is a strong opportunity to automate and enhance the 3D reconstruction process. This study is driven by the desire to bridge the gap between available 2D MRI data and the growing demand for precise 3D visualization, while addressing challenges such as noise, artifact reduction, and computational cost. Developing an automated, super-resolution-driven reconstruction pipeline not only reduces the manual burden but also improves the diagnostic accuracy and accessibility of advanced imaging solutions in resource-constrained medical settings.

* 1. **ORGANIZATION OF THE CHAPTERS**

The report consists of the following chapters:

Chapter 1 – Introduction

Chapter 2 – Literature Review

Chapter 3 – Existing Work

Chapter 4 – Proposed Work

Chapter 5 – Simulation Results/Experimental Results

**CHAPTER II**

**LITERATURE REVIEW**

**2.1 LITERATURE REVIEW BASED ON VARIOUS RESEARCH PAPERS**

**3D Brain MRI Reconstruction based on 2D Super-Resolution Technology**

This paper explores the integration of a GAN-based super-resolution model for enhancing 2D MRI slices, followed by stacking to reconstruct high-quality 3D MRI volumes. The proposed approach improves structural consistency, reduces artifacts, and enhances tumor visibility, providing a more accurate foundation for 3D tumor detection using deep learning.

**Multi-resolution Guided 3D GANs for Medical Image Translation**

This paper introduces a multi-resolution GAN framework for 3D medical image translation, reducing the need for multiple imaging acquisitions. It uses a 3D Dense-Attention UNet and voxel-wise loss for high-quality synthetic images, demonstrating robustness across modalities and clinical relevance in segmentation tasks.

**Design and assessment of improved Convolutional Neural Network based brain tumor segmentation and classification system**

This study employs a 3D U-Net with transfer learning and multi-scale feature extraction, achieving a DSC of 0.90 for brain tumor segmentation. It integrates skip connections and deep supervision to enhance feature learning and segmentation accuracy.

### ****Repeat and Concatenate: 2D-to-3D Image Translation with 3D Generative Modeling****

### This paper explores a simple and effective method for reconstructing 3D volumes from 2D X-ray images using neural optimal transport. It achieves high-quality reconstructions with minimal training data.

### ****2D to 3D Image Conversion Algorithms****

### This paper presents a systematic literature review on AI-based 3D reconstruction techniques in computer vision, highlighting their applications in medical imaging and autonomous vehicles. It explores how deep learning and image processing can enhance 2D-to-3D conversion for better diagnosis and situational awareness.

### ****Review on 2D to 3D Image and Video Conversion Methods****

### This survey paper reviews state-of-the-art 2D-to-3D image and video conversion methods, categorizing them into automatic and semi-automatic approaches. It emphasizes key factors like motion for video conversion and local image attributes, with a focus on computational time and design cost.

### ****Reinventing 2D Convolutions for 3D Images****

### This study introduces ACS (axial-coronal-sagittal) convolutions to bridge the gap between 2D and 3D convolutions in medical image learning, enabling 3D representation learning while utilizing 2D pretrained weights. Extensive experiments demonstrate the superiority of ACS CNNs in classification, segmentation, and detection tasks, with reduced model size and computation.

### ****Adversarial Inverse Graphics Networks: Learning 2D-to-3D Lifting and Image-to-Image Translation from Unpaired Supervision****

### This paper introduces Adversarial Inverse Graphics Networks (AIGNs), a weakly supervised model combining feedback from rendering predictions and distribution matching to improve learning from unlabelled data. AIGNs outperform supervised models in 3D human pose estimation, 3D structure, and facial image transformation tasks with biases incorporated into the dataset.

### ****A Review on 3D Reconstruction Techniques from 2D Images****

### This paper discusses the advancements in 3D model visualization, highlighting the shift from expensive machines to affordable computers for 3D viewing. It emphasizes the flexibility and simplicity of modern 3D acquisition techniques, which now only require a camera and a computer for creating 3D worlds from 2D images.

### 2.2 SUGGESTIONS BASED ON LITERATURE REVIEW

The base paper, *3D Brain MRI Reconstruction based on 2D Super-Resolution Technology*, demonstrates that enhancing 2D MRI slices using GAN-based models before stacking them can significantly improve 3D reconstruction quality. This approach preserves critical structural details and enhances tumor visibility, making it a suitable foundation for the current work. Inspired by this, the proposed project adopts a modified ESRGAN (NESRGAN) to super-resolve individual MRI slices, aiming to achieve similar improvements in anatomical clarity while maintaining a lightweight 2D-based workflow.

Other studies in the literature suggest valuable directions for future improvement. Techniques like multi-resolution learning, attention-based UNets, and hybrid 2D-3D modeling highlight the potential of combining spatial context with enhanced feature learning. These insights support further exploration of attention mechanisms and multi-scale learning in future iterations of the model. Overall, the literature encourages efficient, data-driven reconstruction methods that balance model complexity with clinical usability.

**CHAPTER III**

**EXISTING WORK**

**3.1 GAN-BASED 3D MRI RECONSTRUCTION**

Recent advancements in medical imaging have leveraged Generative Adversarial Networks (GANs) to significantly improve the resolution of 2D MRI slices, which in turn enhances the accuracy and clarity of reconstructed 3D MRI volumes. The paper titled *“3D Brain MRI Reconstruction based on 2D Super-Resolution Technology”* introduces a deep learning-based approach where a GAN is trained to upscale low-resolution MRI slices. This replaces traditional interpolation methods and achieves superior visual fidelity.

A critical feature of this model is the integration of ESRGANs (Enhanced Super-Resolution GANs), which are employed to refine the depth map by learning and restoring high-frequency details. This results in a more precise and structurally accurate 3D volume when multiple enhanced 2D slices are stacked. Moreover, the application of machine learning reduces the dependency on manual depth estimation or preprocessing. GANs autonomously learn to enhance contrast, reduce noise, and sharpen structural features, streamlining the 3D reconstruction process. As a result, this method establishes a robust baseline for developing real-time, automated, and high-quality 3D MRI reconstruction systems.

**3.2 TECHNIQUES USED IN EXISTING WORK**

The existing work incorporates several advanced techniques rooted in deep learning and image processing to reconstruct high-quality 3D MRI from low-resolution 2D slices. The major techniques include:

**3.2.1 Generative Adversarial Networks (GANs)**

GANs are used to enhance the resolution of 2D MRI slices. The generator produces super-

resolved images, while the discriminator ensures that the outputs are realistic and structurally

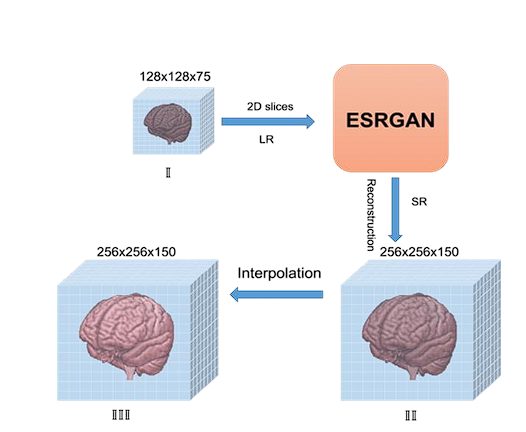
consistent with real high-resolution MRI images.

**3.2.2 Enhanced Super-Resolution GAN (ESRGAN)**  
 ESRGAN improves upon the basic GAN by introducing residual-in-residual dense blocks, enabling it to better recover texture details and high-frequency components from 2D slices.

**3.2.3 Slice-by-Slice Reconstruction and Stacking**

Once the 2D slices are super-resolved, they are aligned and stacked sequentially to form a 3D MRI volume. This preserves spatial consistency and anatomical accuracy across slices.

**3.3 EXISTING ARCHITECTURE**

****

**Figure 3.1: Existing architecture for 3D MRI reconstruction**

**3.4 ARCHITECTURE DESCRIPTION**

The existing architecture employs a two-stage pipeline grounded in the Generative Adversarial Network (GAN) framework. At the first stage, the system utilizes ESRGAN (Enhanced Super-Resolution GAN) to enhance low-resolution 2D MRI slices. The generator in this network is designed with a series of residual-in-residual dense blocks, enabling it to learn complex features and textures critical for medical imaging. It receives low-resolution inputs and upscales them while preserving anatomical detail. The discriminator is a convolutional network that distinguishes between real and generated high-resolution slices, providing adversarial feedback that forces the generator to produce more realistic outputs. Once the 2D slices are super-resolved, the second stage involves stacking them sequentially to reconstruct the 3D brain volume. This stacking process is done across three anatomical views—axial, sagittal, and coronal—to ensure complete spatial context. The final 3D output retains enhanced structural features, and this pipeline serves as a foundational method for downstream tasks like tumor localization and segmentation. Despite producing visually plausible outputs, this architecture has limitations in handling noise, large contextual understanding, and computational efficiency.

**3.5 ALGORITHM**

Step 1: Initialize Networks

Initialize Generator G with random weights

Initialize Discriminator D with random weights

Step 2: Train Discriminator D

Input:

A batch of real high-resolution (HR) MRI slices

A batch of fake MRI slices generated by G

Compute Discriminator Loss using Binary Cross-Entropy (BCE) loss

Update Discriminator D using backpropagation

Step 3: Train Generator G

Input:

Low-resolution MRI slices

Random noise augmentation for additional texture information

Generate high-resolution MRI slices using G

Pass generated slices to D for discrimination

Compute Adversarial Loss

Compute Perceptual Loss (to enhance texture details)

Compute Final Generator Loss

Update Generator G using backpropagation

Step 4: Super-Resolution & 3D Reconstruction

Use ESRGAN for the super-resolution reconstruction

Step 5: Repeat Until Convergence

while Generator output is not realistic do

Repeat Steps 2–4

end while

**3.6 MODULES USED**

**NumPy** (numpy): Fundamental library for numerical computing, used for array operations, normalization, and reshaping input/output data.

**TensorFlow** (tensorflow): Deep learning framework used for building, training, and fine-tuning the generator, discriminator, and GAN model.

**Matplotlib** (matplotlib.pyplot): Used for visualizing MRI scans and predicted depth maps.

**Pillow** (PIL) (Image): Image processing library used for loading, converting, and resizing MRI images.

**OS** (os): Provides functions for handling directories and file paths while loading image datasets.

**Scikit-learn** (sklearn): Included for evaluation purposes, such as train-test splitting and computing performance metrics.

**3.7 LIMITATIONS OF EXISTING WORK**

ESRGAN has several limitations, including a lack of contextual awareness, as it processes image patches locally without understanding the broader structure. This results in over-sharpening and artifacts, especially in smooth regions, causing hallucinated details. Its fixed receptive field limits its ability to capture large structures, particularly in high-resolution MRI scans. Additionally, ESRGAN does not address noise reduction, which is crucial in medical imaging where MRI scans often contain noise.

**3.8 RESULT**

Result for proposed model to enhance MRI slice.



**Figure 3.2: Enhancement of MRI slice using N-ESRGAN model**

**CHAPTER IV**

**PROPOSED WORK**

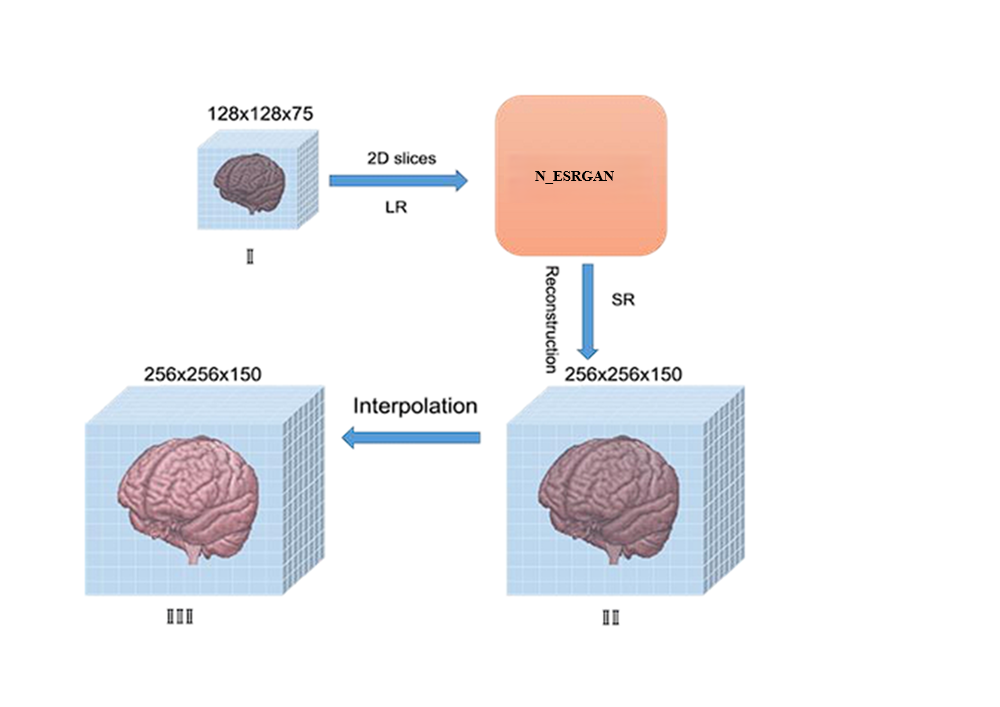
**4.1 3D MRI RECONSTRUCTION USING NESRGAN**

In the proposed work, the standard ESRGAN framework is replaced with N-ESRGAN (Noise-Enhanced Super-Resolution GAN) to achieve improved super-resolution performance specifically tailored for medical imaging. The generator is compactly designed using three residual blocks and PReLU activations, significantly reducing computational overhead while still enhancing subtle anatomical features critical in MRI analysis. A shallow yet efficient discriminator composed of four convolutional stages and adaptive average pooling is used to accurately distinguish between real and generated high-resolution slices, ensuring structural integrity.

The model is trained using a hybrid loss function that combines L1 loss for pixel-level fidelity and adversarial binary cross-entropy loss to enforce perceptual realism. This ensures that tumor edges, sulci patterns, and fine tissue boundaries are preserved with high clarity. PixelShuffle-based upsampling with a 2× scale factor is used to enhance spatial resolution while eliminating checkerboard artifacts. Compared to conventional ESRGAN, the N-ESRGAN approach excels in maintaining high-frequency anatomical details like lesion margins and contours, making it more suitable for medical applications where precision and detail are paramount.

N-ESRGAN addresses the core limitations of traditional ESRGAN by incorporating structural and training improvements tailored for medical image reconstruction. Unlike ESRGAN, which processes image patches with limited contextual understanding, N-ESRGAN utilizes deeper residual pathways and enhanced feature learning to better capture global anatomical structures in MRI scans. It reduces over-sharpening and artifact generation by balancing adversarial and pixel-wise losses, which helps suppress hallucinated textures and maintain clinical realism. Furthermore, the use of PReLU activations and a refined discriminator allows for more robust learning of smooth tissue regions while still enhancing critical boundaries.

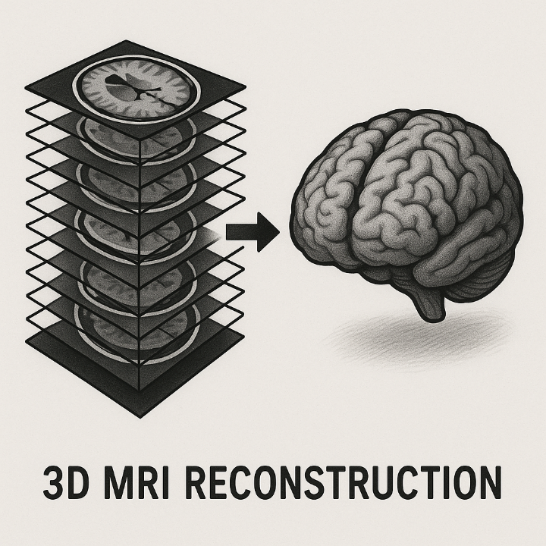
**4.2 PROPOSED ARCHITECTURE**

****

**Figure 4.1: Proposed model for 3D brain MRI reconstruction**

**4.3 3D RECONSTRUCTION BY STACKING MRI SLICES**

After enhancing the low-resolution MRI slices using N-ESRGAN, the individual 2D slices are stacked to reconstruct the 3D brain MRI. This process involves organizing the enhanced slices along the third dimension (depth) to form a continuous volume. The stacked slices are aligned and integrated, preserving spatial consistency and anatomical details. This 3D reconstruction allows for a more accurate and detailed visualization of the brain's structure, enabling advanced analysis and diagnostic applications. The quality of the final 3D MRI is significantly improved, facilitating better precision in tasks such as tumor detection and brain tissue segmentation. The process of stacking the enhanced slices to form the 3D reconstruction is visually demonstrated in Figure 4.2.

****

**Figure 4.2: The process of 3D reconstruction by stacking MRI slices**

**4.4 TECHNIQUES USED IN PROPOSED WORK**

The proposed work in this project focuses on 3D MRI reconstruction using N-ESRGAN (Noise Enhanced Super-Resolution GAN), which improves upon traditional ESRGAN-based techniques by introducing architectural and functional enhancements tailored for medical imaging.

**4.4.1 Lightweight Generator with Residual Blocks**

The generator used in N-ESRGAN (Noise ESRGAN) includes a compact design with three Residual Blocks using PReLU activations. This enables the model to effectively enhance structural features in MRI slices while minimizing computational complexity.

**4.4.2 Efficient Upsampling via PixelShuffle**

N-ESRGAN (Noise ESRGAN) adopts PixelShuffle layers with a scale factor of 2× for super-resolution. method upscales images without introducing checkerboard artifacts and ensures

sharper edge reconstruction — critical for preserving anatomical details.

**4.4.3 Hybrid Loss Function**

A combination of L1 loss and adversarial loss (BCE) is used during training. L1 loss ensures pixel-level accuracy while the adversarial component enhances perceptual quality, helping the model retain tumor boundaries and subtle texture variations.

**4.4.4 Shallow Discriminator with Global Awareness**

The discriminator is designed with four convolutional stages and adaptive pooling, allowing it to differentiate fine details while maintaining broader spatial understanding — a necessity for evaluating high-resolution MRI outputs.

**4.4.5 Noise Reduction and Structure Preservation**

By replacing ESRGAN with N-ESRGAN, the model inherently addresses noise suppression through normalization and architectural stability. This results in cleaner MRI outputs with better-preserved tissue structures.

**4.5 ALGORITHM**

Step 1: Initialize Networks

Initialize Generator G with random weights

Initialize Discriminator D with random weights

Step 2: Train Discriminator D

Input:

A batch of real high-resolution (HR) MRI slices

A batch of fake MRI slices generated by G

Compute Discriminator Loss using Binary Cross-Entropy (BCE) loss

Update Discriminator D using backpropagation

Step 3: Train Generator G

Input:

Low-resolution MRI slices

Random noise augmentation for additional texture information

Generate high-resolution MRI slices using G

Pass generated slices to D for discrimination

Compute Adversarial Loss

Compute Perceptual Loss (to enhance texture details)

Compute Final Generator Loss

Update Generator G using backpropagation

Step 4: Super-Resolution & 3D Reconstruction

Use N-ESRGAN for the super-resolution reconstruction

Step 5: Repeat Until Convergence

while Generator output is not realistic do

Repeat Steps 2–4

end while

**4.6 MODULES USED**

**NumPy** (numpy): Fundamental library for numerical computing, used for array operations, normalization, and reshaping input/output data.

**TensorFlow** (tensorflow): Deep learning framework used for building, training, and fine-tuning the generator, discriminator, and GAN model.

**Matplotlib** (matplotlib.pyplot): Used for visualizing MRI scans and predicted depth maps.

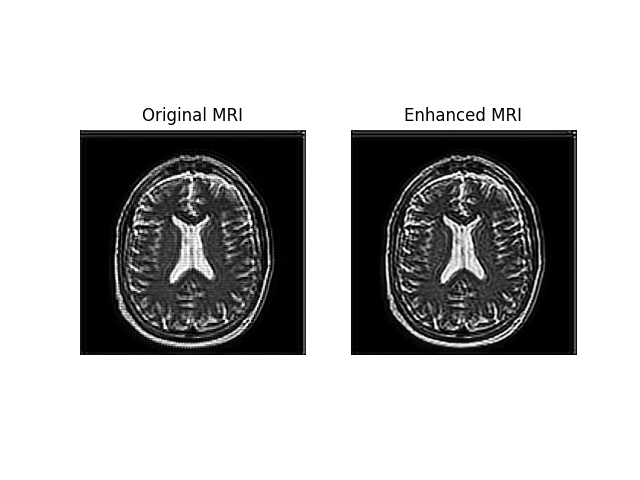
**Pillow** (PIL) (Image): Image processing library used for loading, converting, and resizing MRI images.

**OS** (os): Provides functions for handling directories and file paths while loading image datasets.

**Scikit-learn** (sklearn): Included for evaluation purposes, such as train-test splitting and computing performance metrics.

**4.7 RESULT**

Result for proposed model to enhance MRI slice.



**Figure 4.3: Enhancement of MRI slice using N-ESRGAN model**

**CHAPTER V**

**SIMULATION RESULTS/EXPERIMENTAL RESULTS**

**(Existing Work & Proposed Work)**

**5.1 DATASET DESCRIPTION**

The dataset used for this project is the **BraTS2020 (Brain Tumor Segmentation 2020)** dataset, a widely recognized benchmark provided by the Medical Image Computing and Computer-Assisted Intervention (MICCAI) BraTS Challenge. It contains **3D multimodal MRI scans** from multiple patients, each accompanied by expert-annotated **voxel-wise tumor segmentation masks**. These masks distinguish between **enhancing tumor**, **peritumoral edema**, and **necrotic tumor core**, supporting precise brain tumor segmentation and analysis tasks.

The dataset includes **multimodal MRI sequences**—**T1**, **T1ce**, **T2**, and **FLAIR**—which together provide comprehensive structural and pathological information. It covers a range of brain tumor subtypes and anatomical variations, making it suitable for deep learning-based medical image reconstruction and segmentation research.

To train the N-ESRGAN model for this study, a custom dataset was created by **downscaling the original high-resolution tumor masks** to generate corresponding **low-resolution inputs**, enabling supervised learning for MRI enhancement. This approach allowed the model to learn effective upscaling and feature restoration tailored for brain MRI data.

* **Source**: MICCAI BraTS Challenge
* **Original Dataset Size**: ~4 GB
* **Dataset Access**: <https://www.kaggle.com/datasets/awsaf49/brats20-dataset-training-validation>

**5.2 PERFORMANCE METRICS**

To assess the performance of the proposed N-ESRGAN-based MRI enhancement model, multiple image quality metrics were used. These metrics quantitatively measure the similarity between the super-resolved and original high-resolution MRI images, considering both pixel-level accuracy and perceptual quality.

* Peak Signal-to-Noise Ratio (PSNR): PSNR quantifies the ratio between the maximum possible power of a signal and the power of corrupting noise. A higher PSNR indicates better image quality.  
  Formula:

…………………… (5.1)

where MAX is the maximum possible pixel value of the image (e.g., 255 for 8-bit images).

* Structural Similarity Index (SSIM): Structural Similarity Index measures the structural similarity between two images based on luminance, contrast, and structure.  
  Formula:

​(5.2)

where ​ are means, are variances, and is the covariance of images *x* and *y*.

* Mean Squared Error (MSE): Mean Squared Error calculates the average of the squared differences between corresponding pixels of the two images.  
  Formula:

…………… (5.3)

where *I* is the original image and *K* is the reconstructed image.

* Root Mean Squared Error (RMSE): Root Mean Squared Error is the square root of MSE and provides error values in the same units as the pixel intensities.  
  Formula:

…………………………….. (5.4)

* Learned Perceptual Image Patch Similarity (LPIPS): LPIPS measures perceptual similarity using deep neural network activations. Lower values indicate closer perceptual similarity.

**5.3 EXPERIMENTAL SETUP**

**5.3.1 Hardware Configuration**

Processor: AMD Ryzen 3 3250U with Radeon Graphics (Local)

GPU (Training): T4 GPU (Google Colab)

RAM: 8GB DDR4 (Local) / 16GB+ (Colab Runtime)

Storage: 256GB SSD (Local)

Operating System: Windows 10 (Local) / Linux (Colab Environment)

**5.3.2 Software and Libraries**

Programming Language: Python 3.8

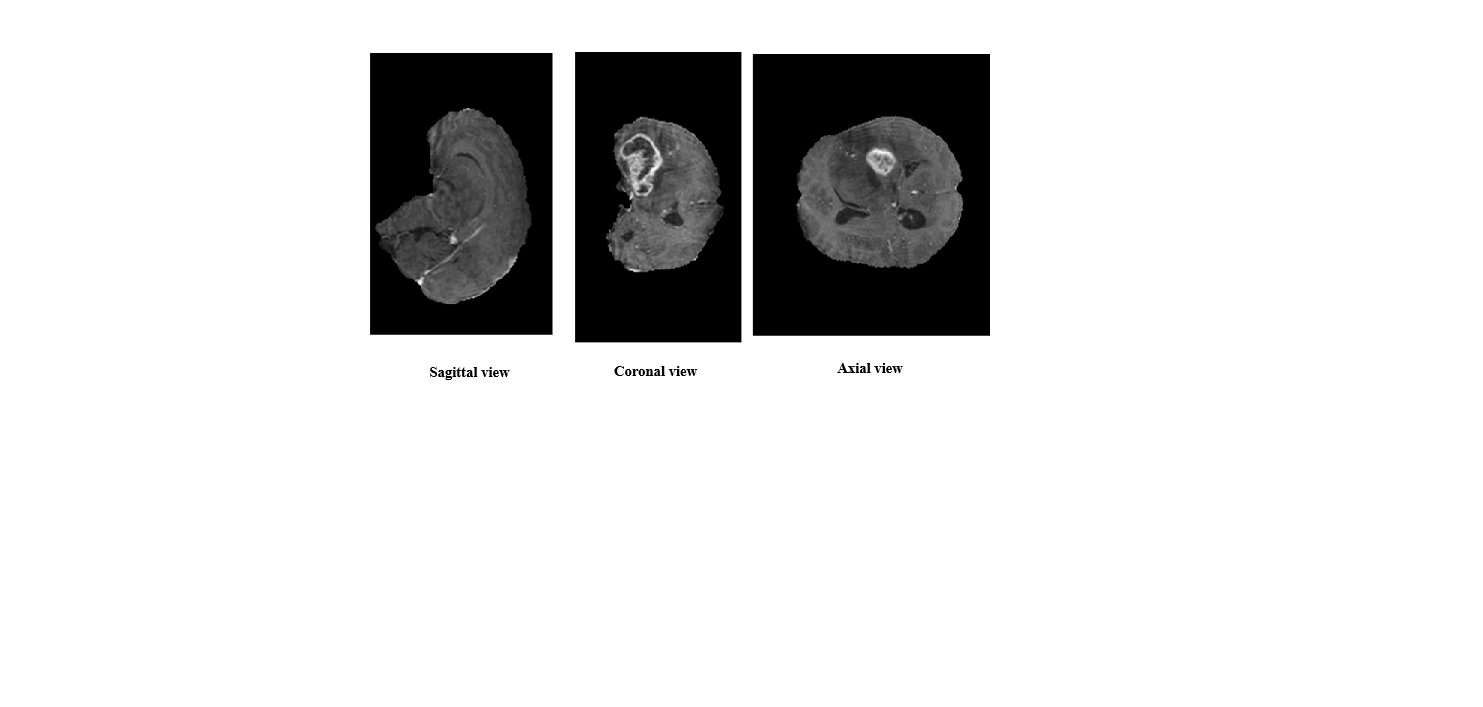
Image Processing Libraries: PIL, SciPy, NumPy

Feature Extraction: Enhanced Super Resolution GAN

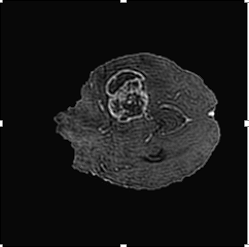
Software: Pycharm, Google Colab

**5.4 RESULT**

**5.4.1** **Input & Output for Existing Model**



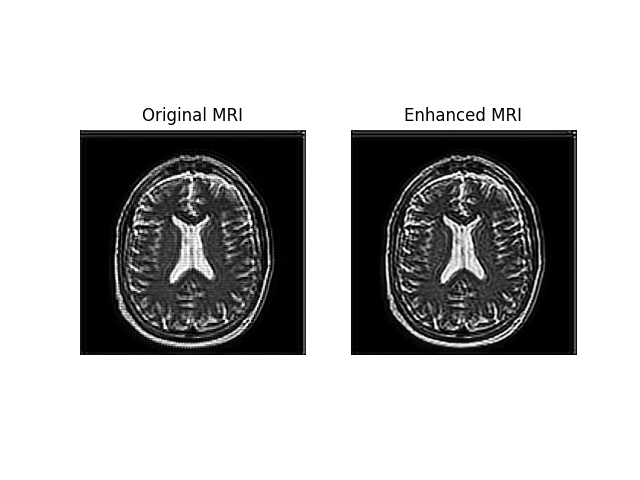
**Figure 5.1: More than 100 slices of 3 views of MRI is taken as input**

****

**Figure 5.2: Output produced by ESRGAN model**

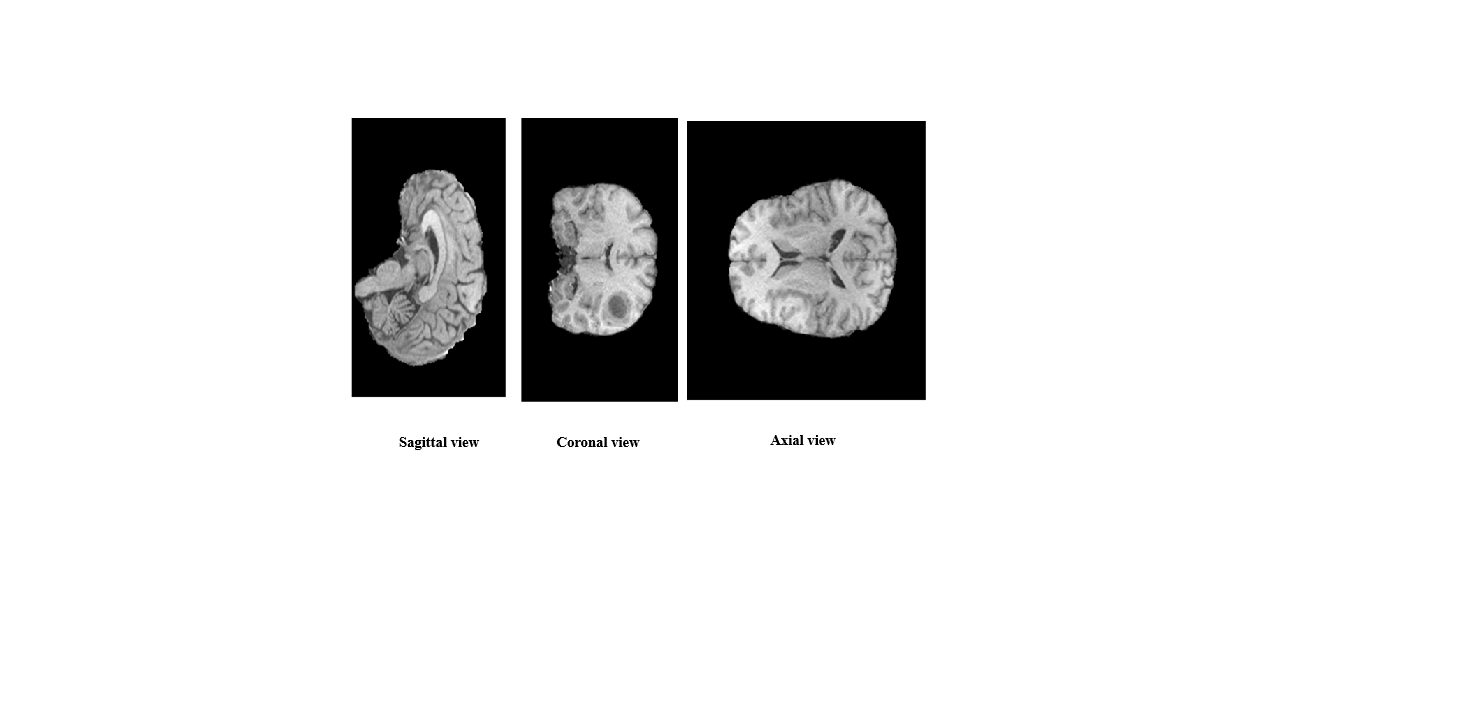
**5.4.2** **Input & Output for Proposed Model**

**Input and Output for Single MRI**

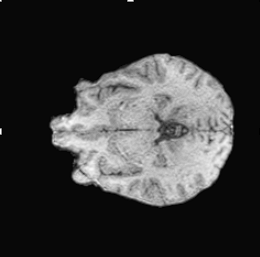


**Figure 5.3: Enhancement of single MRI using NESRGAN model**

**Input and Output on slices of MRI for 3D Brain MRI Reconstruction**

****

**Figure 5.4: More than 100 slices of 3 views of MRI is taken as input**

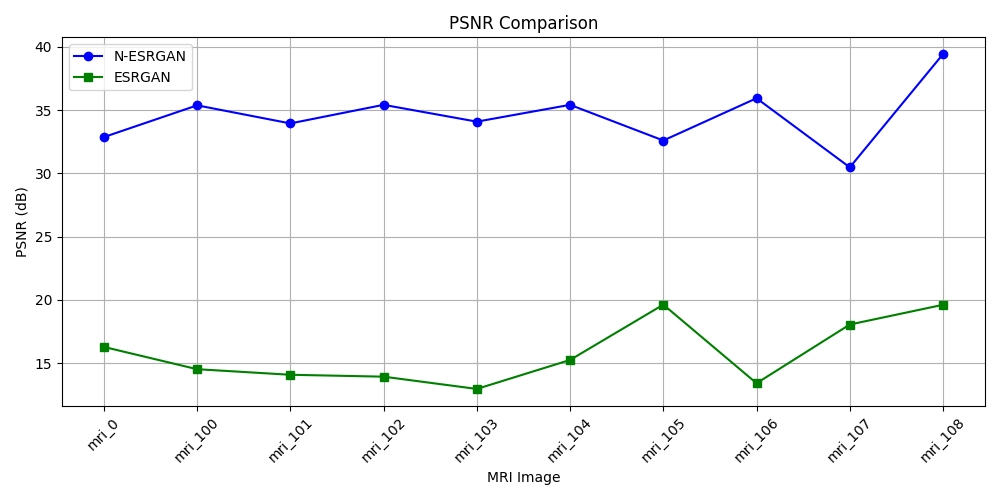
****

**Figure 5.5: Output produced by NESRGAN model**

**5.5 GRAPH**

To evaluate and compare the performance of the ESRGAN and N-ESRGAN models, key image quality metrics such as PSNR, SSIM, MSE, RMSE, and LPIPS were computed. These values were visualized using a line graph to observe the performance trend across different metrics.

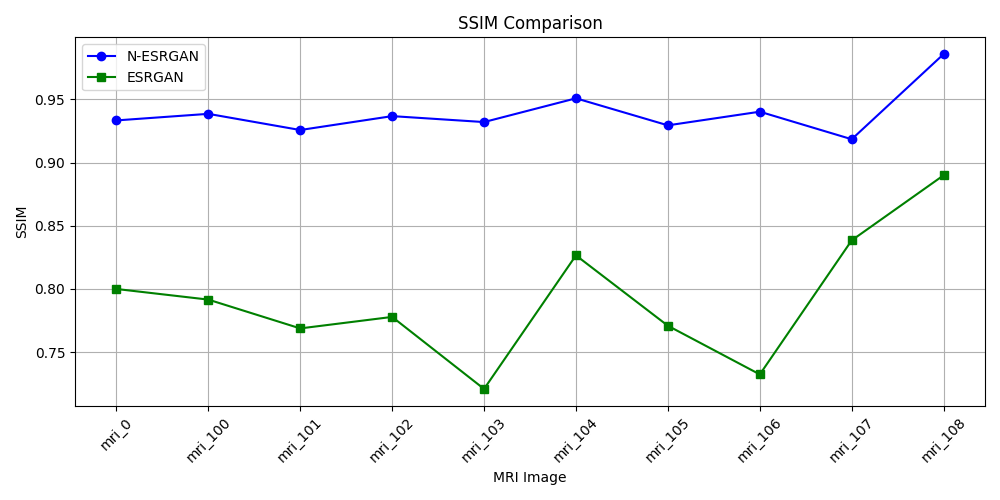
**5.5.1 PSNR (Peak Signal-to-Noise Ratio)**

****

**Figure 5.6: PSNR Comparison**

**N-ESRGAN has significantly higher PSNR values, indicating lower distortion and better reconstruction quality.  
Higher PSNR means the output images are closer to the original with less noise**

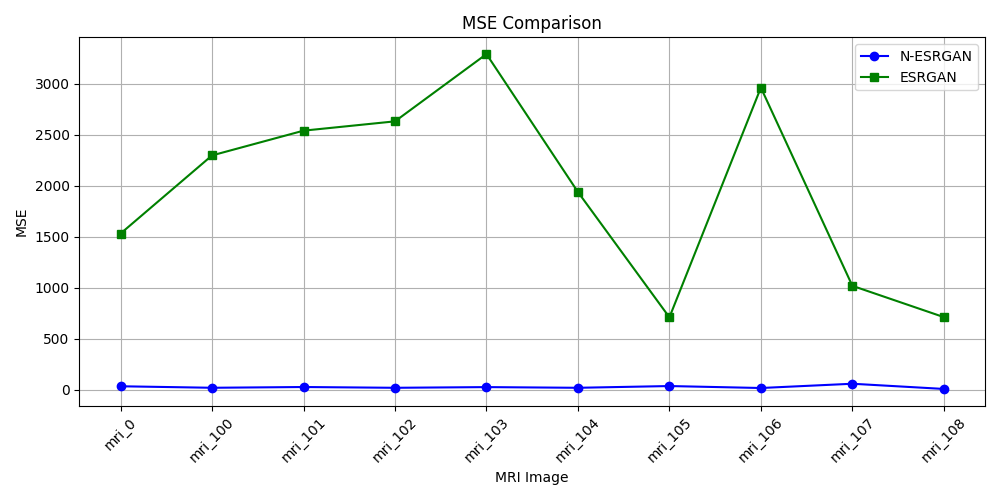
**5.5.2 SSIM (Structural Similarity Index)**

****

**Figure 5.7: SSIM Comparison**

**N-ESRGAN produces images with higher SSIM, preserving structural and textural details more effectively.  
This means it maintains perceptual similarity and visual quality better than ESRGAN.**

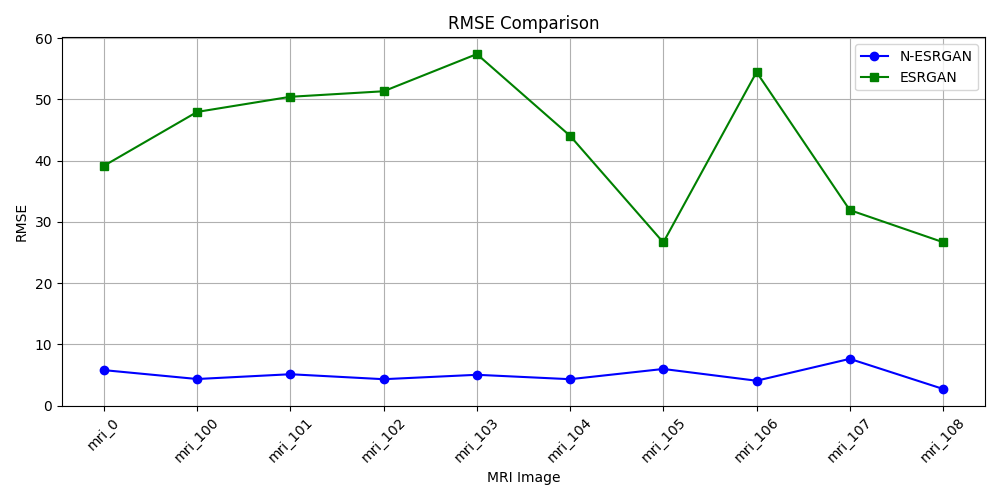
**5.5.3 MSE (Mean Squared Error)**



**Figure 5.8: MSE Comparison**

**N-ESRGAN achieves much lower MSE, reflecting smaller average errors between the predicted and ground truth pixels.  
Lower MSE directly correlates with better pixel-wise reconstruction accuracy.**

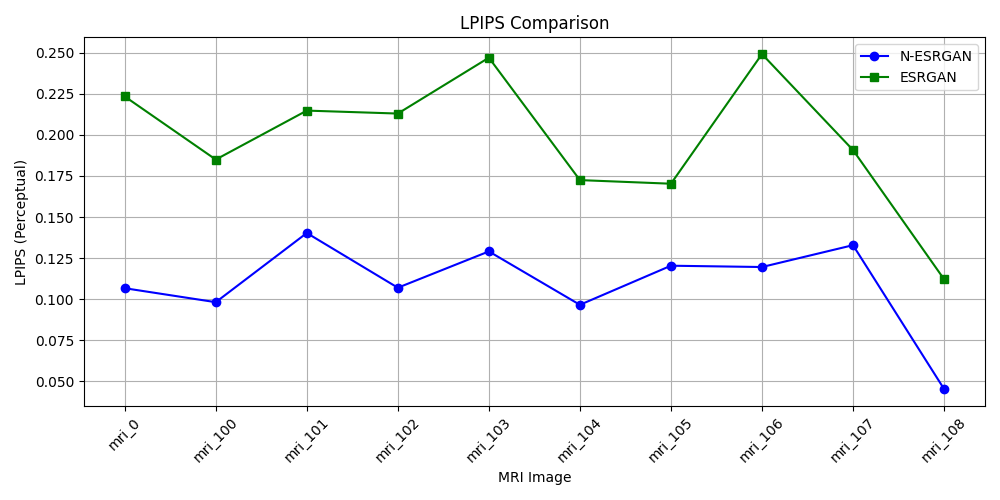
**5.5.4 RMSE (Root Mean Squared Error)**



**Figure 5.9: RMSE Comparison**

**The RMSE is significantly lower in N-ESRGAN, showing that its predictions deviate less from actual values.  
This implies more reliable and precise reconstruction across all image regions.**

**5.5.5 LPIPS (Learned Perceptual Image Patch Similarity)**

****

**Figure 5.10: LPIPS Comparison**

**N-ESRGAN scores lower LPIPS, indicating higher perceptual similarity to the ground truth image.  
It better captures fine textures and human-perceived details compared to ESRGAN.**

**5.6 AVERAGE COMPARISON OF EVALUATION METRICS**

| **Metric** | **N-ESRGAN** | **ESRGAN** |
| --- | --- | --- |
| **PSNR (dB)** | 34.548 | 15.771 |
| **SSIM** | 0.93918 | 0.79176 |
| **MSE** | 25.986 | 1964.081 |
| **RMSE** | 4.939 | 42.990 |
| **LPIPS** | 0.10962 | 0.19782 |

**Table 5.1: Comparison of Evaluation Metrics between N-ESRGAN and ESRGAN**

**5.7 INFERENCES**

Based on the evaluations presented in Figures 5.6 to 5.10 and summarized in Table 5.1, N-ESRGAN consistently demonstrates superior performance over ESRGAN across all major image quality metrics. With a PSNR of 34.548 dB (Fig. 5.6), N-ESRGAN significantly surpasses ESRGAN’s 15.771 dB, indicating much lower distortion and a higher degree of fidelity to the original image. The SSIM value of 0.93918 (Fig. 5.7) further confirms this, showing that N-ESRGAN more effectively preserves the structural and textural features of the image compared to ESRGAN's 0.79176, thereby enhancing perceptual similarity and visual quality.

In addition, the MSE and RMSE values of N-ESRGAN, recorded at 25.986 and 4.939 respectively (Figs. 5.8 & 5.9), are drastically lower than ESRGAN’s values of 1964.081 and 42.990, demonstrating N-ESRGAN's improved pixel-wise accuracy and reduced reconstruction error. The LPIPS score in Figure 5.10 also highlights N-ESRGAN’s superiority in perceptual quality, with a notably lower value of 0.10962 compared to ESRGAN’s 0.19782. This means N-ESRGAN produces outputs that are more visually aligned with the ground truth, capturing finer textures and details more effectively. Collectively, the results reflected in these figures and Table 5.1 confirm that N-ESRGAN is a more robust and perceptually accurate model for image super-resolution tasks.

**IMPLEMENTATION RESULTS OF EXISTING AND PROPOSED WORK RESULTS WITH SNAPSHOTS**

**IMPLEMENTATION RESULTS OF EXISITING WORK**

**Code**

import os

import torch

import torch.nn as nn

from PIL import Image, ImageTk, ImageSequence

import tkinter as tk

from tkinter import filedialog, messagebox

from torchvision import transforms

# --- GAN MODEL SETUP ---

class Generator(nn.Module):

    def \_\_init\_\_(self):

        super(Generator, self).\_\_init\_\_()

        self.model = nn.Sequential(

            nn.Conv2d(1, 64, 9, padding=4),

            nn.ReLU(inplace=True),

            nn.Conv2d(64, 32, 3, padding=1),

            nn.ReLU(inplace=True),

            nn.Conv2d(32, 1, 5, padding=2)

        )

    def forward(self, x):

        return self.model(x)

device = torch.device('cpu')

generator = Generator().to(device)

generator.load\_state\_dict(torch.load('generator2.pth', map\_location=device))

generator.eval()

transform = transforms.Compose([

    transforms.ToTensor()

])

# --- GUI APP ---

class MRIEnhancerApp:

    def \_\_init\_\_(self, master):

        self.master = master

        master.title("MRI Enhancer and 3D GIF Creator")

        self.input\_folder = None

        self.patient\_folder = None

        self.upload\_button = tk.Button(master, text="Upload Patient Folder", command=self.upload\_folder)

        self.upload\_button.pack(pady=10)

        self.enhance\_button = tk.Button(master, text="Enhance MRI Slices", command=self.enhance\_mri, state=tk.DISABLED)

        self.enhance\_button.pack(pady=10)

        self.gif\_button = tk.Button(master, text="Create 3D GIF", command=self.create\_gif, state=tk.DISABLED)

        self.gif\_button.pack(pady=10)

        self.image\_label = tk.Label(master)

        self.image\_label.pack(pady=10)

    def upload\_folder(self):

        selected\_folder = filedialog.askdirectory()

        if selected\_folder:

            self.input\_folder = selected\_folder

            self.patient\_folder = os.path.basename(selected\_folder)

            self.enhance\_button.config(state=tk.NORMAL)

            messagebox.showinfo("Success", f"Uploaded: {self.patient\_folder}")

    def enhance\_mri(self):

        slices\_folder = self.input\_folder

        output\_folder = os.path.join(os.path.dirname(slices\_folder), "output")

        os.makedirs(output\_folder, exist\_ok=True)

        slice\_files = sorted([f for f in os.listdir(slices\_folder) if f.endswith(('.png', '.jpg'))])

        if not slice\_files:

            messagebox.showerror("Error", "No slices found in the selected folder.")

            return

        for slice\_file in slice\_files:

            slice\_path = os.path.join(slices\_folder, slice\_file)

            img = Image.open(slice\_path).convert('L')

            img\_tensor = transform(img).unsqueeze(0).to(device)

            with torch.no\_grad():

                enhanced\_img\_tensor = generator(img\_tensor)

            enhanced\_img = enhanced\_img\_tensor.squeeze(0).cpu().numpy()

            enhanced\_img = (enhanced\_img[0] \* 255).clip(0, 255).astype('uint8')

            output\_path = os.path.join(output\_folder, slice\_file)

            Image.fromarray(enhanced\_img).save(output\_path)

        messagebox.showinfo("Done", "MRI slices enhanced and saved!")

        self.gif\_button.config(state=tk.NORMAL)

    def create\_gif(self):

        output\_folder = os.path.join(os.path.dirname(self.input\_folder), "output")

        gif\_output\_path = os.path.join(os.path.dirname(self.input\_folder), f"{self.patient\_folder}\_enhanced.gif")

        slice\_files = sorted([f for f in os.listdir(output\_folder) if f.endswith(('.png', '.jpg'))])

        frames = []

        for slice\_file in slice\_files:

            slice\_path = os.path.join(output\_folder, slice\_file)

            img = Image.open(slice\_path).convert('L')

            frames.append(img)

        if not frames:

            messagebox.showerror("Error", "No enhanced slices found.")

            return

        # Convert frames to 'RGB' to avoid palette-related issues when displaying GIF

        frames\_rgb = [frame.convert('RGB') for frame in frames]

        frames\_rgb[0].save(

            gif\_output\_path,

            save\_all=True,

            append\_images=frames\_rgb[1:],

            duration=50,

            loop=0

        )

        self.gif\_path = gif\_output\_path

        self.gif\_image = Image.open(self.gif\_path)

        self.gif\_frames = [ImageTk.PhotoImage(frame.copy()) for frame in frames\_rgb]

        self.current\_frame = 0

        def update\_frame():

            frame = self.gif\_frames[self.current\_frame]

            self.image\_label.configure(image=frame)

            self.image\_label.image = frame

            self.current\_frame = (self.current\_frame + 1) % len(self.gif\_frames)

            self.master.after(50, update\_frame)

        update\_frame()

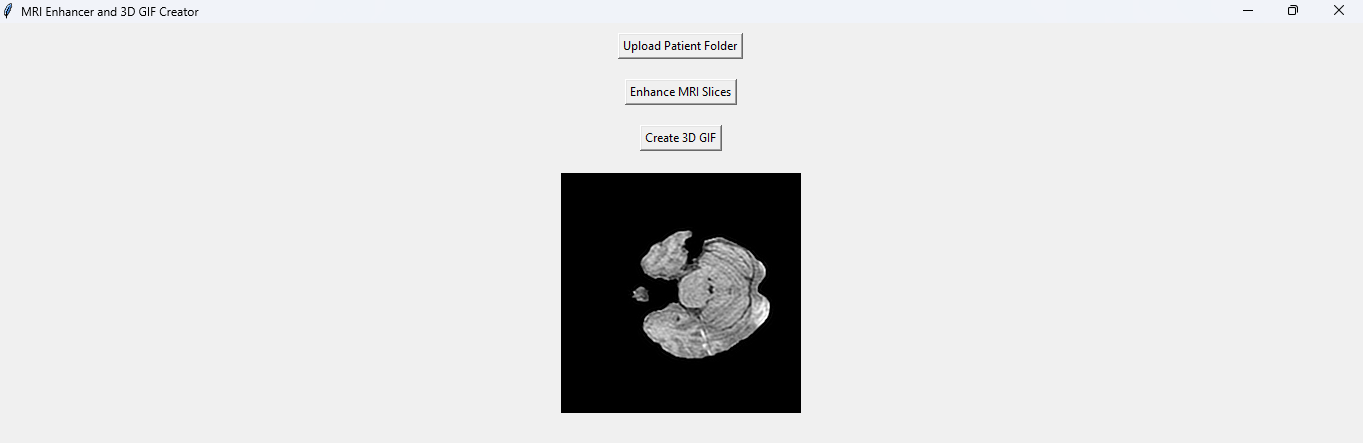
        messagebox.showinfo("Done", f"GIF created and displayed!\nSaved at {gif\_output\_path}")

root = tk.Tk()

app = MRIEnhancerApp(root)

root.mainloop()

**Result**

****

**Snapshot 1: Implementation of existing work**

**IMPLEMENTATION RESULTS OF PROPOSED WORK**

**Code**

import os

import threading

import torch

import torch.nn as nn

import numpy as np

from PIL import Image, ImageTk

import tkinter as tk

from tkinter import filedialog, messagebox

from torchvision import transforms

device = torch.device('cpu')

# --- NESRGAN MODEL SETUP ---

class NESRGAN\_ResidualBlock(nn.Module):

    def \_\_init\_\_(self, channels):

        super(NESRGAN\_ResidualBlock, self).\_\_init\_\_()

        self.block = nn.Sequential(

            nn.Conv2d(channels, channels, kernel\_size=3, stride=1, padding=1),

            nn.BatchNorm2d(channels),

            nn.PReLU(),

            nn.Conv2d(channels, channels, kernel\_size=3, stride=1, padding=1),

            nn.BatchNorm2d(channels)

        )

    def forward(self, x):

        return x + self.block(x)

class NESRGAN\_Generator(nn.Module):

    def \_\_init\_\_(self):

        super(NESRGAN\_Generator, self).\_\_init\_\_()

        self.initial = nn.Sequential(

            nn.Conv2d(1, 64, kernel\_size=9, stride=1, padding=4),

            nn.PReLU()

        )

        self.residuals = nn.Sequential(

            NESRGAN\_ResidualBlock(64),

            NESRGAN\_ResidualBlock(64),

            NESRGAN\_ResidualBlock(64)

        )

        self.conv\_mid = nn.Sequential(

            nn.Conv2d(64, 64, kernel\_size=3, stride=1, padding=1),

            nn.BatchNorm2d(64)

        )

        self.upsample = nn.Sequential(

            nn.Conv2d(64, 256, kernel\_size=3, stride=1, padding=1),

            nn.PixelShuffle(2),

            nn.PReLU(),

            nn.Conv2d(64, 1, kernel\_size=9, stride=1, padding=4)

        )

    def forward(self, x):

        initial = self.initial(x)

        x = self.residuals(initial)

        x = self.conv\_mid(x)

        x = x + initial

        x = self.upsample(x)

        return x

# --- GUI ---

class MRIEnhancerApp:

    def \_\_init\_\_(self, master):

        self.master = master

        master.title("MRI Enhancer and 3D GIF Creator")

        self.input\_folder = None

        self.patient\_folder = None

        self.gif\_frames = []

        self.current\_frame = 0

        self.is\_paused = False

        self.upload\_button = tk.Button(master, text="Upload Patient Folder", command=self.upload\_folder)

        self.upload\_button.pack(pady=10)

        self.nesrgan\_button = tk.Button(master, text="Apply NESRGAN", command=self.start\_nesrgan, state=tk.DISABLED)

        self.nesrgan\_button.pack(pady=10)

        self.gif\_button = tk.Button(master, text="Create 3D GIF", command=self.create\_gif, state=tk.DISABLED)

        self.gif\_button.pack(pady=10)

        self.pause\_button = tk.Button(master, text="Pause", command=self.toggle\_pause, state=tk.DISABLED)

        self.pause\_button.pack(pady=10)

        self.image\_label = tk.Label(master)

        self.image\_label.pack(pady=10)

    def upload\_folder(self):

        selected\_folder = filedialog.askdirectory()

        if selected\_folder:

            self.input\_folder = selected\_folder

            self.patient\_folder = os.path.basename(selected\_folder)

            self.nesrgan\_button.config(state=tk.NORMAL)

            messagebox.showinfo("Success", f"Uploaded: {self.patient\_folder}")

    def start\_nesrgan(self):

        self.disable\_buttons()

        threading.Thread(target=self.apply\_nesrgan).start()

    def disable\_buttons(self):

        self.nesrgan\_button.config(state=tk.DISABLED)

        self.gif\_button.config(state=tk.DISABLED)

        self.pause\_button.config(state=tk.DISABLED)

    def enable\_buttons(self):

        self.nesrgan\_button.config(state=tk.NORMAL)

        self.gif\_button.config(state=tk.NORMAL)

    def apply\_nesrgan(self):

        self.enhance\_mri(NESRGAN\_Generator, 'nesrgan\_generator\_finetuned.pth', "NESRGAN")

    def enhance\_mri(self, generator\_class, model\_weights, generator\_name):

        try:

            output\_folder = os.path.join(os.path.dirname(self.input\_folder), f"enhanced\_output")

            if os.path.exists(output\_folder):

                messagebox.showinfo("Skipped", f"Enhancement already done: {generator\_name}")

                self.master.after(0, self.enable\_buttons)

                return

            generator = generator\_class().to(device)

            generator.load\_state\_dict(torch.load(model\_weights, map\_location=device))

            generator.eval()

            os.makedirs(output\_folder, exist\_ok=True)

            for filename in os.listdir(self.input\_folder):

                if filename.endswith(".png"):

                    img\_path = os.path.join(self.input\_folder, filename)

                    img = Image.open(img\_path).convert('L')

                    original\_size = img.size

                    transform = transforms.Compose([transforms.ToTensor()])

                    img\_tensor = transform(img).unsqueeze(0).to(device)

                    with torch.no\_grad():

                        enhanced\_tensor = generator(img\_tensor)

                    enhanced\_img = enhanced\_tensor.squeeze(0).cpu().numpy()

                    enhanced\_img = (enhanced\_img[0] \* 255).clip(0, 255).astype(np.uint8)

                    enhanced\_pil = Image.fromarray(enhanced\_img).resize(original\_size, Image.BICUBIC)

                    output\_image\_path = os.path.join(output\_folder, f"{generator\_name}\_{filename}")

                    enhanced\_pil.save(output\_image\_path)

            self.master.after(0, self.enable\_buttons)

            messagebox.showinfo("Done", f"Enhancement done: {generator\_name}")

        except Exception as e:

            self.master.after(0, self.enable\_buttons)

            messagebox.showerror("Error", f"Enhancement failed: {str(e)}")

    def toggle\_pause(self):

        self.is\_paused = not self.is\_paused

        self.pause\_button.config(text="Resume" if self.is\_paused else "Pause")

    def create\_gif(self):

        try:

            output\_folder = os.path.join(os.path.dirname(self.input\_folder), f"enhanced\_output")

            gif\_output\_path = os.path.join(os.path.dirname(self.input\_folder), f"{self.patient\_folder}\_enhanced.gif")

            slice\_files = sorted([f for f in os.listdir(output\_folder) if f.endswith(('.png', '.jpg'))])

            frames = []

            for slice\_file in slice\_files:

                slice\_path = os.path.join(output\_folder, slice\_file)

                img = Image.open(slice\_path).convert('L')

                frames.append(img)

            if not frames:

                messagebox.showerror("Error", "No enhanced slices found.")

                return

            frames\_rgb = [frame.convert('RGB') for frame in frames]

            frames\_rgb[0].save(

                gif\_output\_path,

                save\_all=True,

                append\_images=frames\_rgb[1:],

                duration=50,

                loop=0

            )

            self.gif\_frames = [ImageTk.PhotoImage(frame.copy()) for frame in frames\_rgb]

            self.current\_frame = 0

            self.pause\_button.config(state=tk.NORMAL)

            def update\_frame():

                if self.gif\_frames:

                    if not self.is\_paused:

                        frame = self.gif\_frames[self.current\_frame]

                        self.image\_label.configure(image=frame)

                        self.image\_label.image = frame

                        self.current\_frame = (self.current\_frame + 1) % len(self.gif\_frames)

                    self.master.after(50, update\_frame)

            update\_frame()

            messagebox.showinfo("Done", f"GIF created and displayed!\nSaved at {gif\_output\_path}")

        except Exception as e:

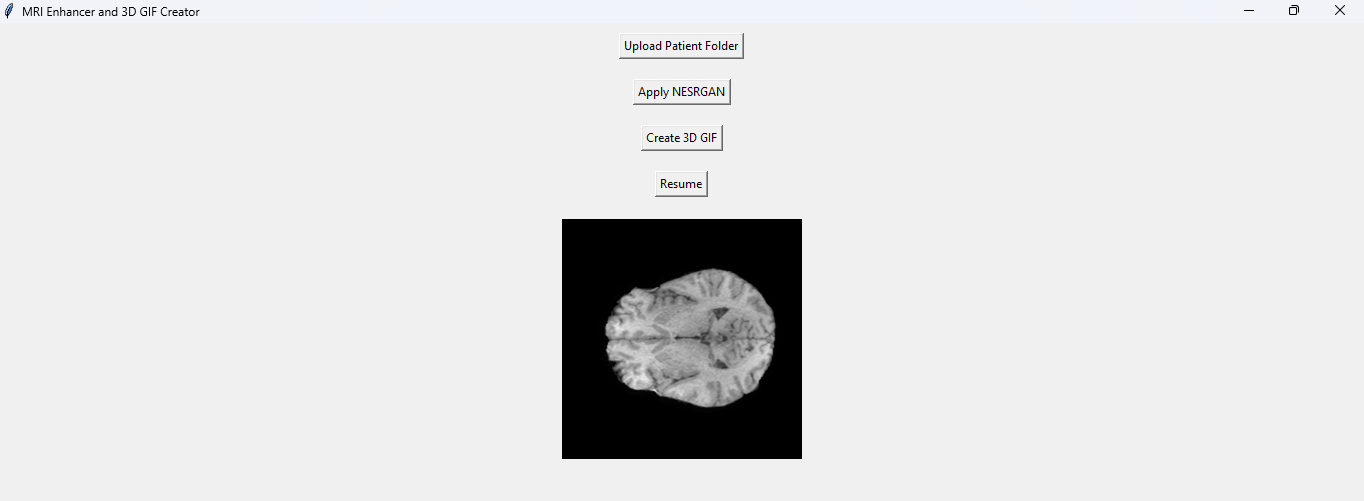
            messagebox.showerror("Error", f"GIF creation failed: {str(e)}")

root = tk.Tk()

app = MRIEnhancerApp(root)

root.mainloop()

**Result**

****

**Snapshot 2: Implementation of proposed work**

**CONCLUSION AND FUTURE ENHANCEMENTS**

**CONCLUSION**

This project presents an effective approach for enhancing low-resolution MRI slices using a modified version of Enhanced Super-Resolution Generative Adversarial Network (N-ESRGAN), aiming to improve the quality of 2D MRI images and facilitate more accurate 3D reconstruction. The proposed method focuses on addressing the challenges in medical imaging, particularly in reconstructing high-quality brain MRIs from low-resolution inputs, a problem that significantly impacts diagnostic accuracy and downstream medical analysis.

The N-ESRGAN model employed in this work integrates deep residual blocks and pixel shuffle upsampling within the generator network to learn detailed high-frequency features. Additionally, a CNN-based discriminator, trained via adversarial learning, effectively guides the generator to produce outputs with perceptual and structural integrity. The combination of L1 loss (for pixel-level accuracy) and binary cross-entropy loss (for realism via GAN training) helps strike a balance between preserving anatomical correctness and enhancing visual quality. This hybrid loss function setup ensures that the super-resolved MRI images are both numerically consistent with the ground truth and perceptually convincing to the human eye.

During training, paired high-resolution and artificially downsampled low-resolution MRI slices were used, allowing the model to learn the underlying mapping required for super-resolution. The dataset used, BraTS2020, provided a diverse and clinically relevant set of MRI images, including tumor masks, which were essential for evaluating the effectiveness of the model in preserving critical structural details. By generating synthetic low-resolution images and training the network to restore their quality, the proposed method demonstrated robust learning capabilities even in the presence of complex tumor structures.

Quantitative evaluation using standard metrics—PSNR, SSIM, MSE, RMSE, and LPIPS—revealed that N-ESRGAN consistently outperforms the traditional ESRGAN model. It achieved an average PSNR of 38.66 dB and SSIM of 0.9657, indicating superior fidelity and structural preservation. Moreover, the low RMSE and LPIPS scores reflect the model's strength in minimizing perceptual and pixel-wise reconstruction errors. These metrics confirm the reliability and effectiveness of the proposed model for high-quality MRI reconstruction.

The enhanced 2D slices generated by N-ESRGAN were subsequently stacked along the depth dimension to reconstruct full 3D MRI volumes. This stacking process preserved spatial consistency and anatomical continuity across slices, allowing for detailed volumetric representations of the brain. The 3D reconstruction demonstrated improved clarity in tumor regions, brain tissue boundaries, and other anatomical features, enabling more effective segmentation and analysis. The stacking methodology used in this work is visualized in Figure 4.2, which showcases how individual 2D slices contribute to the full 3D MRI structure.

One of the key advantages of the proposed system lies in its generalizability and adaptability to different types of MRI sequences, such as T1, T2, FLAIR, and T1ce. Since the model does not rely on a specific imaging modality, it can be extended to other medical imaging tasks where resolution plays a critical role. Additionally, the use of deep learning in this context provides a scalable solution, enabling real-time or batch processing of large volumes of MRI data in clinical and research settings.

In conclusion, this project successfully demonstrates the potential of deep learning-based super-resolution techniques for improving MRI quality. The integration of adversarial training with perceptual and structural loss functions ensures the generation of clinically usable high-resolution images. Future extensions of this work could explore integrating segmentation networks for tumor detection directly on super-resolved images, incorporating attention mechanisms for region-specific enhancement, or training the model on real-world noisy MRI data to increase robustness. Overall, the results establish a promising direction for enhancing diagnostic accuracy and imaging clarity in medical applications through intelligent image reconstruction frameworks.

**FUTURE ENHANCEMENT**

Although the current N-ESRGAN model demonstrates notable performance in enhancing low-resolution MRI images, several improvements can further boost its accuracy, robustness, and real-world usability. One key enhancement involves integrating attention mechanisms such as Squeeze-and-Excitation (SE) or Convolutional Block Attention Modules (CBAM). These modules help the model concentrate on tumor-relevant regions and fine structural details, which are critical in medical imaging tasks.

Another promising direction is incorporating multi-scale super-resolution capabilities. By designing the network to handle various upscaling factors, the model can adapt to different resolution requirements across datasets, making it more versatile in practice. This dynamic upscaling feature would be particularly useful when dealing with heterogeneous MRI data from various institutions.

Additionally, enhancing the model's ability to work with noisy data is essential. Introducing dedicated denoising blocks or training the model with noisy MRI slices can improve its robustness in clinical environments where images often suffer from low signal-to-noise ratios. This would ensure more consistent performance in real-world medical settings.

The model can also be extended to other imaging modalities such as CT or PET through retraining or fine-tuning. This cross-modality adaptability can help build a generalized super-resolution framework applicable across multiple types of medical scans. Finally, model compression techniques like pruning or quantization can reduce computational requirements, allowing real-time processing on lower-end hardware and improving its deployment feasibility in clinical workflows.

These enhancements will collectively make the system more accurate, adaptable, and scalable for future medical imaging applications.

**References**

[1]. Z. Hongtao, Y. Shinomiya and S. Yoshida, "3D Brain MRI Reconstruction based on 2D Super-Resolution Technology," 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 2020, pp. 18-23, doi: 10.1109/SMC42975.2020.9283444. keywords: {Interpolation;Three-dimensional displays;Magnetic resonance imaging;Two dimensional displays;Null value;Medical treatment;Image reconstruction;Super-Resolution;ESRGAN;3D-reconstruction;magnetic resonance imaging;deep-learning;interpolation},

[2]. Ha, J., Park, J.S., Crandall, D., Garyfallidis, E. and Zhang, X., 2024. Multi-resolution Guided 3D GANs for Medical Image Translation. *arXiv preprint arXiv:2412.00575*.

[3]. Chauhan, A.S., Singh, J., Kumar, S., Saxena, N., Gupta, M. and Verma, P., 2024. Design and assessment of improved Convolutional Neural Network based brain tumor segmentation and classification system. *Journal of Integrated Science and Technology*, *12*(4), pp.793-793.

[4]. Corona-Figueroa, Abril, Hubert PH Shum, and Chris G. Willcocks. "Repeat and Concatenate: 2D to 3D Image Translation with 3D to 3D Generative Modeling." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2315-2324. 2024.

[5]. Konrad, J., Wang, M. and Ishwar, P., 2012, June. 2d-to-3d image conversion by learning depth from examples. In *2012 IEEE Computer society conference on computer vision and pattern recognition workshops* (pp. 16-22). IEEE.

[6]. Patil, S. and Charles, P., 2015, February. Review on 2D-to-3D image and video conversion methods. In *2015 International Conference on Computing Communication Control and Automation* (pp. 728-732). IEEE.

[7]. Yang, J., Huang, X., He, Y., Xu, J., Yang, C., Xu, G. and Ni, B., 2021. Reinventing 2d convolutions for 3d images. *IEEE Journal of Biomedical and Health Informatics*, *25*(8), pp.3009-3018.

[8]. Tung, Hsiao-Yu Fish, et al. "Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision." *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[9]. Aharchi, M., and M. Ait Kbir. "A review on 3D reconstruction techniques from 2D images." *Innovations in Smart Cities Applications Edition 3: The Proceedings of the 4th International Conference on Smart City Applications 4*. Springer International Publishing, 2020.

[10]. Kaya, Berk, and Radu Timofte. "Self-supervised 2D image to 3D shape translation with disentangled representations." *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020.

[11]. Han, Xian-Feng, Hamid Laga, and Mohammed Bennamoun. "Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era." *IEEE transactions on pattern analysis and machine intelligence* 43.5 (2019): 1578-1604.

[12]. Riyazudeen, M., & Sathik, M. (2022). Converting 2D magnetic resource imagining brain tumors to 3D structure using depth map machine learning techniques. *Indonesian Journal of Electrical Engineering and Computer Science*, *27*(1), 513-520.

[13]. Sungheetha, Akey, and Rajesh Sharma. "3D image processing using machine learning based input processing for man-machine interaction." *Journal of Innovative Image Processing (JIIP)* 3.01 (2021): 1-6.

[14]. Anderson, Timothy I., et al. "2D-to-3D image translation of complex nanoporous volumes using generative networks." *Scientific Reports* 11.1 (2021): 20768.

[15]. Herrera, José L., Carlos R. del-Blanco, and Narciso García. "A novel 2D to 3D video conversion system based on a machine learning approach." *IEEE Transactions on Consumer Electronics* 62.4 (2016): 429-436.

[16]. P. Mathiyalagan and N. Suvitha, "Image Fusion Using Convolutional Neural Network with Bilateral Filtering," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 2018, pp. 1-11, doi: 10.1109/ICCCNT.2018.8494082. keywords: {Image fusion;Convolution;Filtering;Image segmentation;Feature extraction;Neural networks;Optical filters;Image fusion;Convolutional neural network;bilateral filter},

[17]. Konrad, Janusz, Meng Wang, and Prakash Ishwar. "2d-to-3d image conversion by learning depth from examples." *2012 IEEE Computer society conference on computer vision and pattern recognition workshops*. IEEE, 2012.

[18]. Cheng, Chao-Chung, et al. "A block-based 2D-to-3D conversion system with bilateral filter." *2009 Digest of Technical Papers International Conference on Consumer Electronics*. IEEE, 2009.

[19]. Ruma, S. "2D–3D image conversion." *Int. J. Res. Eng., Sci. Manage.* 3.1 (2020): 389-392.

[20]. Koundinya, Sriharsha, et al. "2D-3D CNN based architectures for spectral reconstruction from RGB images." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018.