MRI RECONSTRUCTION

A PROJECT REPORT

Submitted by

MADHESH RAJ S (Reg. No. 201804050)

SIVA PRASATH S (Reg. No. 201804093)

SUBAMAARLESHAR B (Reg. No. 201804102)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENTOFCOMPUTER SCIENCE ANDENGINEERING

MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI

(An Autonomous Institution affiliated to Anna University Chennai)

April 2022

BONAFIDE CERTIFICATE

Certified that this project report **"MRI Reconstruction"** is the bonafide work of **Mr. S. MADHESH RAJ (Reg. No.:201804050), Mr. S. SIVA PRASATH (Reg. No.:201804093), Mr. B. SUBAMAARLESHAR (Reg. No.: 201804102)** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project or dissertation in the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

Internal Guide

Mr.C.Balasubramanian, M.Tech.,
Assistant Professor (Sr.Grade)
Dept. of Computer Science and Engg.
Mepco Schlenk Engg. College
(Autonomous), Sivakasi.

Head of the Department

Dr.J.Raja Sekar M.E., Ph.D.,
Professor and Head
Dept. of Computer Science and Engg.
Mepco Schlenk Engg. College
(Autonomous), Sivakasi.

Submitted for Viva-Voce Examination held at **Mepco Schlenk Engineering College (Autonomous), Sivakasi** on _____ / / 20 __ **.**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

MR Image reconstruction using deep learning has proved in reducing the MRI acquisition time and reducing the computational complexity compared to the other analytical method. The challenges in this method is using large training dataset in larger neural networks. In recent methods, short connections between layers outperforms deeper neural networks. In this work, such short connections called as Densely Connected Residual Blocks (DCR) is used. DCR blocks concatenates the various feature maps that are extracted form the previous layers, thus increasing the residual block's efficiency. The qualitative and quantitative results shows Deep learning with DCR block integrated improves the reconstruction performance significantly. The model also denoises the existing noise in the image. So the reconstructed image also denoises the MR image if noise exist.

ACKNOWLEDGEMENT

First and foremost, we thank the **LORD ALMIGHTY** for his abundant blessings that is showered upon our past, present and future successful endeavours.

At the appellation, we would like to extend our sincere gratitude to our College management and Principal **Dr.S.ARIVAZHAGAN M.E., Ph.D.,** for providing sufficient working environment such as systems and library facilities. We also thank him very much for providing us with adequate lab facilities, which enabled us to complete out project.

We would like to extend our heartfelt gratitude to **Dr.Raja Sekar M.E., Ph.D.,** Senior Professor and Head, Department Of Computer Science and Engineering, Mepco Schlenk Engineering College for giving us the golden opportunity to undertake a project of this nature and for his most valuable guidance given at every phase of our work.

We thank our project coordinator **Mrs.G.Ananthi, M.E.,** Assistant Professor (Sr. Grade), Department Of Computer Science and Engineering, Mepco Schlenk Engineering College for being our Project Coordinator and directing us throughout our project.

We would also like to extend our gratitude and sincere thanks to **Mr.C.Balasubramanian M.Tech.,** Assistant Professor (Sr. Grade), Department Of Computer Science and Engineering, Mepco Schlenk Engineering College for being our Project Guide and for his moral support and suggestions.

Last but not least, we extend our indebtedness towards our beloved **family and our friends** for their support which made the project a successful one.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.No.	ABBREVIATION		
1	DCR	-	Densely Connected Residual
2	CNN	-	Convolution Neural Network
3	PSNR	-	Peak Signal to Noise Ratio
4	MRI	-	Magnetic Resonance Imaging

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Magnetic resonance image(MRI) is used to extract images of soft tissues of human body. It is used to analyze the human organs without the need for surgery. Generally MRI images are used in diagnostic characterization of the internal delicate organs. An average MR Image costs around 3000 to 5000 rupees. Based on the area of scanning and no of images, the time need for taking a scan and processing may vary.

1.1 PROBLEM DESCRIPTION:

MRI machines have powerful magnets which produce a strong magnetic field that forces the internal body molecules to align with that field. Radio frequency is pulsed into the patients and turned off. MRI sensors detect the energy released as the molecules align with the magnetic field. Based on the energy emitted by the molecules the raw data is taken by the sensors. With that raw data image processing is done to convert the raw data into image. Many image processing techniques are in use, but the computational complexity and the time required for processing is high.

1.2 PURPOSE OF THE PROJECT:

MRI is time consuming process. Compressed Sensing MRI is recently used technology to speed up the image reconstruction process by undersampling the k space data from the MR machines. But it suffers from low image quality and serious aliasing artifacts. The proposed method uses Deep learning for the reconstruction of the image. The model that is proposed produces high quality image with low processing time and automatically reduces the noise from the data. The proposed method uses skip connection which is proved more efficient than the deep neural networks. Two models are used for reconstruction of the MR Image – CNN with Densely Connected Residual Block and U-Net with Densely Connected Residual Block.

1.3 OBJECTIVES OF THE PROJECT:

- To use Convolutional Neural Network(CNN) with Densely Connected Residual (DCR) blocks to build a model for reconstruction of a image.
- To use U-Net with Densely Connected Residual(DCR) blocks to build a model for reconstruction of a image.
- To produce high quality reconstructed images with less noise for subsequent image processing as well as clinical diagnosis.

1.4 OUTCOMES OF THE PROJECT:

- Convolutional Neural Network with DCR block is built.
- U-Net with DCR block is built.
- High quality reconstructed images with less noise and superior visual quality is produced.

1.5 SCOPE OF THE PROJECT:

The proposed project can reconstruct images from the existing image that is extracted from the MRI machine. The reconstruct image have higher quality and less noise.

1.6 REPORT OVERVIEW

CHAPTER 2 presents the detailed study of the related work.

CHAPTER 3 presents the description about the system.

CHAPTER 4 presents the detailed study of the system design.

CHAPTER 5 presents the detailed study of the system implementation.

CHAPTER 6 deals with the results and discussion.

CHAPTER 7 presents the conclusion and future enhancement.

CHAPTER 2

LITERATURE SURVEY

2.1 3D PBV-NET: AN AUTOMATED PROSTATE MRI DATA SEGMENTATION METHOD:

The precise and automated segmentation of the prostate gland in MRI data is critical for prostate cancer diagnosis and therapy planning. Despite the fact that several automated segmentation systems are available, Despite the development of new methodologies, such as deep learning-based approaches, segmentation performance remains poor due to a number of factors. Variability in picture appearance, anisotropic spatial resolution, and imaging interference This investigation provides an improved 3D precision bicubic interpolation-based automated prostate MRI data segmentation algorithm. A virtual network (V-Net) is a network that is not physically connected. Because of the low-frequency components of the prostate gland, the bicubic Interpolation is used to preprocess the MRI data. A 3D PBV-Net is built on this base to perform prostate surgery MRI data segmentation.

To show the effectiveness of the method, they evaluate the proposed 3D PBV-Net using two clinical prostate MRI data datasets, PROMISE 12 and TPHOH, with manual delineations available as the ground truth. Their technique gives strong segmentation results on the PROMISE 12 and TPHOH datasets, with average accuracy of 97.65 percent and 98.29 percent, respectively, Dice metrics of 0.9613 and 0.9765, Hausdorff distances of 3.120 mm and 0.9382 mm, and average boundary distances of 1.708 and 0.7950. Their approach has improved the quality of automated prostate MRI data segmentation and looks to be capable of achieving telehealth application accuracy criteria.

Prostate cancer is one of the diseases with a wide range of tissue patterns and a high morbidity and mortality rate. The most effective non-invasive diagnostic technique for prostate cancer is magnetic resonance imaging, which provides strong soft-tissue contrast.It's also one of the most precise prostate imaging methods, allowing clinicians to view the disease in a more intuitive and comprehensive manner.Automatic segmentation of the prostate gland in a medical imaging dataset is usually recognised as a tough challenge for radiation therapy. Anatomical atlas registration is used in traditional prostate segmentation approaches. Furthermore, due to technical limitations, medical institutions often rely on manual segmentation of prostate MRI data. The entire process takes a long

time and is inaccurate. The manual segmentation technique can no longer meet the clinical expectations as the number of patients grows. As a result, a system that can automatically and effectively separate prostate MRI data is in high demand. More than half of WHO member countries have developed a national telehealth strategy. The usage of remote radiological image diagnostics, as well as the general development of telehealth, has increased the bar for image processing requirements. In the diagnosis of coronavirus pandemics and elder health, for example, wearable sensors and telehealth technologies have been used. Telehealth can efficiently organise and offer emergency medical treatment in remote locations. Furthermore, by exchanging local medical information and patient diagnostic images, it can break down geographical borders. In the context of MRI data diagnosis in telehealth treatment, medical image segmentation is still a massive undertaking. Clinicians may use three-level coarse-to-fine segmentation to separate and extract sections or regions of interest, boosting the speed and precision of sickness observations. Deep neural networks-based algorithms have been increasingly employed in medical image segmentation as processing capability has increased, replacing older image segmentation methods. The bulk of these deep learning-based techniques, on the other hand, have the drawback of depending too heavily on manual segmentation and being unable to directly handle whole 3D medical images. Before training deep learning models, existing methods frequently utilise dimensionality reduction or image slicing to convert 3D data into 2D slices. This is not conducive to accurate 3D image segmentation. As a result, it is critical to provide a more effective method for acquiring more in-depth information in order to better meet the clinical diagnosis objectives.

Their work has four key contributions, they are Using bicubic interpolation and a superior 3D V-NET, they developed an automated prostate MRI data segmentation method (3D PBV-Net). Three-dimensional convolution is utilised to extend the pixel information of the prostate MRI data in order to quickly, accurately, and effectively separate the prostate regions. Manual segmentation, a time-consuming labelling process, is also being utilised less. They tested their 3D PBV-Net on the publicly accessible PROMISE 12 dataset as well as a private dataset (acquired from the hospital). They were able to get high segmentation accuracy of 0.9613 and 0.9765 using the Dice metric (DM). Based on the segmentation findings, the delineated results are 3D rebuilt and displayed. The visualisation can provide clinicians with an intuitive diagnostic basis in telemedicine, allowing them to enhance patient care, treatment planning, and prognosis.

The recommended technique is evaluated on two clinical prostate MRI datasets that are used to train, measure, and objectively compare the performance of utilized5zed prostate MRI data segmentation algorithms. The dataset utilized initially is the prostate MRI data segmentation challenge. A set of 100 representative prostate MRI data is published on the challenge website (http://promise12.grandchallenge.org/). 50 sets of training data, 30 sets of test data, and 20 sets of challenge data are included in PROMISE 12. The Third Peoples Hospital also provides a second in-house private dataset . To test the model, 106 clinical prostate MRI data with hand segmentation labels are given.

To have a better grasp of the PROMISE 12 outcomes, they compared the results of prostate MRI data segmentation with previous research. By contrasting the results of several algorithms. They observed that their technique had some advantages using a DM (Dice Metric) and a 95 percent HD (Hausdorff distance) indicator. It shows that the target region of our prediction has a significant degree of overlap with the usual manual segmentation. Their technique, however, results in poor ABD(Average Boundary Distance) outcomes. Their model performed well on this clinical dataset.

In this they used the most prevalent neural networks and deep learning approaches to achieve autonomous prostate MRI data segmentation. Before training and optimising the network, the bulk of previously recommended deep learning algorithms convert 3D MRI data into 2D image slices. This takes a long time to train the model and does not correlate to the stereoscopic features of the 3D slice. The input layer of our proposed 3D PBV-Net is a 3D matrix, with 1 feature mappings in the original input, which can speed up training and improve efficiency. They also added several picture preprocessing methods, such as the bicubic interpolation algorithm and image rotation-based augmentation, on this foundation. We also put the model through its paces on TPHOH(Third Peoples Hospital of Hangzhou) and PROMISE 12, demonstrating its dependability.The results of the experiments reveal that the prostate MRI data segmentation framework based on our 3D PBV-Net has considerable benefits over existing techniques, such as greater accuracy and faster segmentation time. They expect more intuitive and accurate prostate MRI data segmentation findings to be supplied for telemedicine now that the correct segmentation results have been attained.

2.2 BENCHMARKING DEEP NETS MRI RECONSTRUCTION MODELS ON THE FASTMRI PUBLICLY AVAILABLE DATASET:

The MRI reconstruction sector lacked a proper data set that allowed for major outcomes on real raw data (i.e. complex valued), particularly when it came to deep learning (DL) algorithms, which require far more data than typical Compressed Sensing (CS) reconstruction. The fastMRI data set currently fills this gap, and it is required to test recent DL models against this benchmark. Furthermore, because these networks are developed in various frameworks and repositories (if publicly available), it is necessary to have a standard tool that is publicly accessible and allows for a reproducible benchmark of the various techniques as well as the convenience of creating new models. We propose a tool that allows alternative reconstruction deep learning models to be benchmarked.

The fastMRI dataset, which was released this year, filled a gap for a universal benchmark dataset for magnetic resonance imaging (MRI) reconstruction. Many neural network-based approaches to MRI reconstruction have not been tested against this new dataset. This paper presents the findings of these approaches in the single-coil component of the fastMRI data set, as well as a code that allows for repeatable results, easy benchmarking, and serves as a foundation for the development of other comparable networks: fastmri reproducible benchmark. This code is largely developed in Keras, but some networks have undocumented PyTorch variants. Other studies included benchmarks for a variety of deep learning (DL) models for MRI reconstruction, although they had certain flaws. The quantitative results for RefineGAN, for example, are for real-valued data (i.e. picture magnitude retrospectively under-sampled in k-space), while the qualitative results are for complex-valued data. However, the dataset only has 20 volumes in total, which is quite limiting. In comparison, the training and validation sets of the fastMRI dataset have 1172 volumes. The public dataset utilised by KIKI-net is real-valued, whereas the complex-valued dataset is proprietary. As a result, the fastMRI data set provides a once-in-a-lifetime opportunity to test various reconstruction network topologies in a real-world setting, i.e. using raw complex-valued k-space measurements.

The networks utilised in this benchmark have certain flaws that can be addressed. To begin with, the training time is prohibitive, as 300,000 steps of batch size 1 can take up to two days for the KIKI-net. The iteration cycles become extremely slow as a result of this. Second, there is no apparent reason for utilising CNN in the

k-space. When there is translation invariance, CNNs thrive, which is not the case in the k-space. Lower frequencies should never be treated in the same manner that higher frequencies are. Finally, by learning a portion of the loss function, Generative Adversarial Networks (GANs) may be able to assist in the improvement of those networks. A GAN is employed to improve the reconstruction of a U-net in the work introducing RefineGAN, but there is no reason why GANs couldn't be added to the end of the other networks provided in this study. With the inclusion of an extra perceptual loss computed with a pre-trained VGG network, the work introducing DAGAN does almost the same. Finally, the study did not look at the handling of complex values. Indeed, networks always concatenate the real and imaginary components, resulting in a 2-channel real-valued image or k-space from a complex-valued image or k-space. Other research, such as the one presenting deep residual learning, take advantage of the phase and magnitude of complex numbers in two different ways.

2.3 DOMAIN KNOWLEDGE AUGMENTATION OF PARALLEL MR IMAGE RECONSTRUCTION USING DEEP LEARNING:

A deep learning (DL) strategy for fast magnetic resonance (MR) imaging is provided, which uses parallel MR imaging domain knowledge to supplement DL networks for accurate and stable image reconstruction. A unique loss function combining mean absolute error, structural similarity, and sobel edge loss is used in the proposed DL approach. As inputs to the network, the DL model accepts both actual data and pictures rebuilt using the parallel imaging approach. The suggested technique was compared to state-of-the-art parallel imaging and deep learning algorithms in terms of accuracy utilising two anatomical areas and six MRI contrasts. The proposed technique outperformed the other methods by a large margin. The addition of domain knowledge significantly regularises the DL-based picture reconstruction and delivers accurate and stable image enhancement, according to extensive validation on large datasets.

Magnetic resonance imaging (MRI) generates high-resolution pictures of anatomical areas at the cost of lengthy scan periods. Shorter scan periods and higher-resolution pictures are ideal, but in order to acquire diagnostic-quality images, trade-offs are frequently required. For better speed, some approaches include a data consistency term. Other DL approaches recreate a fully sampled picture using both the

undersaturated k-space and related image. All of the aforementioned deep learning techniques for image reconstruction try to use a learned deep learning model to replace the compressed sensing reconstruction optimization issue. When solving the compressed sensing reconstruction issue using a taught deep learning model, picture quality and scan time were significantly improved. In this research, they proposed a solution to the two issues with conventional deep learning techniques. They propose a deep learning-based image reconstruction method that uses the parallel imaging statistical equation to supplement the deep learning model's inputs explicitly.To begin, their strategy reduces leftover parallel imaging defects and noise, making deep learning reconstruction more intelligible. GRAPPA reconstructed photos with noise amplification aliasing artefacts are used as input to the deep learning network at a fast production factor in their proposed techniques, nicknamed 'Grappa DL' and 'GrappaZf DL.' The GRAPPA rebuilt image augments the DL reconstruction by utilising parallel imaging's domain knowledge. To boost resolution and structural information, they designed a unique loss function consisting of mean absolute error, variance structural similarity, and sobel edge loss for training the DL networks. GRAPPA reconstruction methods and the commonly used variable density undersampling deep learning reconstruction were compared to the recommended methodologies. The work contributes significantly by developing a deep learning MRI reconstruction method that incorporates domain knowledge from conventional parallel imaging reconstruction to augment deep learning image reconstruction; the proposed method used an equidistant undersampling scheme that is readily available in MRI scanners, avoiding modifications to existing pulse sequences for routine use.

In this research, they provided a method for adding domain knowledge into MRI in order to improve the performance and stability of DL-based image reconstruction. They demonstrated how existing information from parallel imaging methods may be leveraged to boost the effectiveness of DL encoder-decoder systems. Parallel imaging drives deep learning MR image reconstruction, leading in higher picture quality in terms of SSIM, PSNR, and NMSE. According to experimental findings using various contrasts from brain and knee imaging, the recommended procedures reduced spurious features occasionally created in DL rebuilt images. The incorporation of domain information to the image reconstruction inverse problem has the potential to occur in a more precise and valid solution.

2.4 DENSELY CONEECTED HIERARCHICAL NETWORK FOR IMAGE DENOISING:

Deep convolutional neural networks have recently been used in a number of image processing studies and have shown to significantly enhance performance. We provide a densely connected hierarchical image denoising network (DHDN) that outperforms state-of-the-art image denoising systems in this paper. Our suggested network enhances picture denoising performance by using the modified U-hierarchical Net's architecture, which allows us to employ a larger number of parameters than existing techniques. We also leverage dense connectivity and residual learning to induce feature reuse and overcome the vanishing-gradient problem in our convolution blocks and network. Finally, the model ensemble and self-ensemble approaches are successfully applied, allowing us to increase the performance of the proposed network. The suggested network's performance was validated when it placed second in the NTRIE 2019 real image denoising competition sRGB track and third in the raw-RGB track. Additional experimental results on additive white Gaussian noise removal show that the suggested network outperforms existing approaches, despite the fact that the proposed network only has a single set of training parameters to handle a wide variety of noise levels.

External noises such as additive white Gaussian noise (AWGN), speckle noise, and impulse noise impair the image quality. Image denoising is a procedure that generates a high-quality image from a low-quality image. Because of its wide range of applications, such as medical image denoising, satellite image denoising, and compression noise denoising, image denoising is a key study topic in the image processing industry. Object detection and recognition in autonomous cars, among other applications, drew researchers' attention to image denoising. This is because it is critical to eliminate noise from a picture in order to improve object recognition performance. They have employed the modified U-hierarchical Net's architecture to enable our network to efficiently use limited memory. As a result, our model can use more parameters than traditional networks. They have used dense connectivity and residual learning to accurately remove noise from input photos and solve the vanishing-gradient problem using our new convolution blocks and network design. They have used the self-ensemble and model ensemble approaches to improve the objective and subjective quality of our proposed network's output images. With a single set of trained parameters, we train our model to handle a wide range of noise levels. They entirely overcame the constraint of traditional approaches because our network does not require input noise level information.

They have presented a denoising network with a hierarchical structure in this paper. Our proposed network efficiently employed limited memory by reducing the size of the feature maps by using the hierarchical structure of the modified U-Net. When upsampling the feature maps, they have used the sub-pixel interpolation method, which allowed our model to reliably and effectively interpolate the feature maps. Furthermore, using dense connectivity and residual learning, our suggested DCR block successfully eliminated noise from images and solved the vanishing-gradient problem. Finally, they used the self-ensemble approach and the model ensemble method to improve the suggested network's performance. As a result, our suggested network performed well in the NTIRE 2019 actual image denoising challenge, demonstrating its supremacy in the removal of nonspecific noise from images. Additional AWGN studies showed that our suggested network outperforms traditional approaches, managing a wide variety of noise levels with only a single set of learned parameters. When the suggested network was trained for a given noise level, it showed even better results.

2.5 K-SPACE DEEP LEARNING FOR ACCELERATED MRI:

The annihilating filter-based low-rank Hankel matrix technique (ALOHA) is a cutting-edge compressed sensing technique that directly interpolates missing k-space data using low-rank Hankel matrix completion. The duality of structured low-rankness in the k-space domain and image domain sparsity, which provides for compact signal analysis in the k-space domain, is the key to ALOHA's success. They described a completely data-driven deep learning strategy for k-space interpolation that is motivated by a recent mathematical breakthrough that connects convolutional neural networks to Hankel matrix decomposition via data-driven framelet basis. Their network may readily be expanded to non-Cartesian k-space trajectories by simply adding another regridding layer. According to extensive numerical experiments, the suggested deep learning approach consistently outperforms existing image-domain deep learning algorithms.

Many researchers have looked at deep learning algorithms for MR reconstruction difficulties and have shown that they can improve performance significantly. Using completely linked layers, Manifold Approximation (AUTOMAP) even tries to estimate the Fourier transform itself. All of these ground-breaking studies have consistently exhibited higher reconstruction performance over compressed sensing techniques while requiring much less computer complexity at run-time.

Although an end-to-end recovery strategy such as AUTOMAP [10] may recover a picture without ever interpolating the missing k-space samples, it only works for images of adequate size since completely linked layers demand a lot of memory.As a result, the majority of popular deep learning MR reconstruction techniques involve image domain post-processing or iterative updates between the k-space and the image domain via a cascaded network. One of the key goals of this research is to show that the aforementioned methodologies aren't the only ones that can be used for MR deep learning; there is another method that works well. The suggested deep learning technique directly interpolates missing k-space data, allowing for correct reconstruction by simply using the interpolated k-space data's Fourier transform. Our network, unlike AUTOMAP, is created as a convolutional neural network (CNN) without the necessity for a fully connected layer, resulting in a reduced GPU memory demand for the proposed k-space deep learning. Indeed, the success of structured low-rank Hankel matrix techniques prompted the creation of k-space deep learning schemes, which use the duality of structured low-rankness in the k-space domain and picture domain sparsity to utilise succinct signal representation in the k-space domain. An encoder-decoder network may also be seen as a signal representation originating from Hankel matrix decomposition, according to a novel theory of deep convolutional framelets. Furthermore, theoretical and practical findings show that by simply stacking multi-coil k-space data along the channel direction as an input to feed into the neural network, the proposed framework may easily achieve multi-channel calibration-free k-space interpolation. Scan-specific robust artificial neural networks for k-space interpolation are a modern k-space neural network technique (RAKI). RAKI, unlike the proposed technique, incorporates scan-specific learning without training data, which necessitates recalculating neural network weights for each k-space input data.

This study revealed that totally data-driven k-space interpolation is achievable using k-space deep learning and the image domain loss function, which was inspired by a relationship between ALOHA and deep learning. The suggested k-space interpolation network outperforms the present image domain deep learning network on a variety of sample pathways. They believe that the proposed k-space interpolation framework will open up a new path of study for a variety of Fourier imaging problems because it is exceedingly successful and is backed up by novel theory.

2.6 MULTI-SCALE PREDICTION FOR FIRE DETECTION USING CONVOLUTIONAL NEURAL NETWORK:

Fire detection systems that are automated can help to decrease the loss of life and property by allowing for a quick and accurate response to fires. While visual techniques offer some advantages over sensor-based systems, image processing-based methods typically result in false alarms. Recent research on convolutional neural networks has solved these drawbacks, demonstrating exceptional performance in fire detection applications. Previous research, however, have only employed single-scale feature maps for fire image classification, which are insufficiently resilient to fires of different sizes in the images. To solve this problem, we present a multi-scale prediction framework that makes use of the feature maps generated by the highly stacked convolutional layers at all sizes. This paper proposes a feature-squeeze block to use feature maps of various scales in the final prediction. To efficiently utilise the information from the multi-scale prediction, the feature-squeeze block compresses the feature maps spatially and channel-wise. Extensive tests show that the suggested method outperforms convolutional neural network-based solutions that are currently in use. As a result of the experiment, the proposed approach has an F1-score of 97.89 percent and a false positive rate of 0.0227 percent in the average of several evaluations.

Color, shape, flashing, frequency, and dynamic textures are all captured by traditional image processing methods. To detect fire, the RGB, YUV, YCbCr, and CIELab colour spaces were used. In addition to colour information, motion data has been utilised. To identify fire from other objects, methods such as background subtraction and optical flow analysis were utilised, which boosted the usage of the flame's movement for detecting the fire. The use of various aspects of fire in probability-based machine learning approaches has also been proposed. Sensor-based methods have been widely used in the field, despite these studies on image processing-based methods. Because traditional image processing-based technologies were not robust for usage in the field. Several researches have recently used neural networks for fire detection because deep learning-based methods, particularly convolutional neural network (CNN)-based methods, have demonstrated remarkable performance in visual tasks. The frames received from the CCTV video are used as input to the CNN-based approach in the entire process of the fire detection system, and the prediction result is delivered to the alarm system. A sequential CNN model with convolutional, down-sampling, and fully linked layers is used in most known approaches. Deeply stacked

layers collect multiple information from the input photos, resulting in great fire detection performance while minimising false alarms. A fine-tuning technique was used to apply current CNN models, such as GoogLeNet and VGGNet, to fire detection tasks. Muhammad et al. used MobileNet for real-time fire detection, which has similar performance to other CNN models but has a lower computational cost. To capture flames of varying sizes in the input photos, they offer a multi-scale prediction approach. By responding robustly to varying resolutions resulting from the distance between the camera and the fire, the multi-scale prediction greatly reduces the false positive rate (FPR). For final prediction, they suggest an FS block that uses feature maps of various scales. To make the most of the information in the feature maps, the FS block compresses them spatially and channel-wise. They test the suggested method on a variety of data sets and show that it outperforms existing state-of-the-art methods. To assess the universality of our suggested strategy, we employ an unseen data set for training.

We suggested a methodology in this paper that improves the robustness of the existing CNN-based fire image classification model by focusing on the different sizes of fires in photos. We used multi-scale feature maps derived sequentially from deeply layered CNN to improve robustness. To extract feature information from multi-scale feature maps, the suggested FS block was used. Experiments on four fire image data sets with a variety of assessment criteria showed that the suggested method works well for photos with a variety of fire properties. On three validation data sets, the F1-score and FPR are 97.89 percent and 0.0227 percent, respectively, surpassing the second-best model by 2.97 percent and reducing by 2.4969 percent. On an unknown data set, the suggested method beat numerous state-of-the-art models, demonstrating the generalisation capacity of multi-scale prediction. The confidence score of the multi-scale feature maps and the size of the fire were found to be correlated in more studies. This showed that by referring to the multi-scale features complementarily, multiscale prediction helps to the robustness of the suggested strategy. This study adds to the field by illustrating how multi-scale prediction can help with fire detection robustness and generalisation. This approach, we think, will be useful in future research.

2.7 U-NET: CONVOLUTIONAL NETWORKS FOR BIOMEDICAL IMAGE SEGMENTATION:

Thousands of annotated training examples are required for effective deep network training, according to most experts. In this paper, they presented a network and training approach that mainly relies on data augmentation to make greater use of the annotated examples supplied. A symmetric expanding path is used to ensure exact localisation, while a contracting path is used to collect context. On the ISBI challenge, we demonstrate that such a network can be trained end-to-end from a minimal number of images and outperforms the prior best solution for segmenting neural structures in electron microscopic stacks. In addition, the network is speedy.

During the previous two years, deep convolutional networks have beaten the state of the art in various picture identification tasks. Convolutional networks have been around for a while, but their performance has been limited by the amount of the training sets available and the scale of the networks under consideration. On the ImageNet dataset, which includes one million training photos, Krizhevsky made his breakthrough through supervised training of a massive network with eight layers and millions of parameters. Since then, the networks have grown in size and complexity. Convolutional networks are commonly used for classification tasks, using a single class label as the output to an image. However, in many visual tasks, particularly in biomedical image processing, the intended output should contain localization, which means that each pixel should be allocated a class name. Furthermore, in biological jobs, thousands of training photos are frequently out of reach.Ciresan trained a network to predict the class label of each pixel by feeding it a local region (patch) around that pixel as input using a sliding-window setup. This network, first and foremost, has the potential to localise. Second, in terms of patches, the training data outnumbers the number of training photographs by a large margin. In this study, they use a more elegant architecture called a "fully convolutional network." They modified and altered the architecture such that it now requires fewer training photographs and provides more precise segmentations. The basic idea is to layer a typical contracting network with successive layers, substituting pooling operators with upsampling operators. As a result, the output resolution is improved by these layers. To localise, the contracting route's high resolution features are combined with the upsampled output. A succeeding convolution layer can learn to provide a more precise output based on this information. One major modification in their architecture is that the upsampling portion now has a large number of feature channels, allowing the network to send context information to

higher resolution layers. As a result, the expanding and contracting paths are essentially symmetrical, resulting in a u-shaped structure. The network only uses the valid component of each convolution, which means the segmentation map only includes pixels in case the entire context is visible in the input image.

In a number of biological segmentation applications, the u-net architecture performs wonderfully. It supplements data with elastic deformations and requires very few annotated photographs, with a training time of roughly 10 hours on an NVidia Titan GPU (6 GB).

2.8 A DEEP CASCADE OF CONVOLUTIONAL NEURAL NETWORKS FOR DYNAMIC MR IMAGE RECONSTRUCTION:

We present a system for reconstructing dynamic sequences of 2-D cardiac magnetic resonance (MR) pictures from undersampled data utilising a deep cascade of convolutional neural networks (CNNs) to speed up the data gathering process, which is inspired by current breakthroughs in deep learning. We focus on the case where data is collected through aggressive Cartesian undersampling. First, we show that the proposed method outperforms state-of-the-art 2-D compressed sensing approaches, such as dictionary learning-based MR image reconstruction, in terms of reconstruction error and speed when each 2-D picture frame is reconstructed independently. Second, we show that by combining convolution and data sharing techniques, CNNs can learn spatio-temporal correlations quickly while recreating the frames of the sequences together. We show that the suggested method consistently outperforms existing methods, maintaining anatomical structure more faithfully up to an 11-fold undersampling. Furthermore, reconstruction is quick: each complete dynamic sequence can be reconstructed in less than 10 seconds, and each image frame can be reconstructed in 23 milliseconds in the 2-D case, allowing for real-time applications.

Images can be recovered using sub-Nyquist sampling using Compressed Sensing (CS), given the following: The images must first be compressible, that is, they must have a sparse representation in some transform domain. Second, incoherence between the sampling and sparsity domains must be ensured in order to ensure that the reconstruction issue has a unique solution that is feasible. In reality, this is accomplished through random k-space subsampling, which results in aliasing patterns in the image domain that can be considered correlated noise. Images can therefore be recreated using nonlinear optimisation or iterative methods under such assumptions.

CS-MRI refers to a group of methods that apply CS to the MR reconstruction challenge. These methods, in general, use fixed sparsifying transforms, such as wavelet transformations. Adaptive sparse modelling, which aims to learn the ideal sparse representation directly from the data, has been a natural development of existing approaches, allowing for more flexible representations. This can be accomplished by employing techniques such as dictionary learning (DL). In this paper, we look at using Cartesian undersampling to reconstruct dynamic sequences of 2D cardiac MR images, as well as employing CNNs to rebuild each frame individually. In the image domain, we see the reconstruction challenge as a de-aliasing problem. Undersampled MR image reconstruction is difficult since the pictures often have a low signal-to-noise ratio, but high-quality reconstructions are frequently required for therapeutic purposes. We propose a deep network design that builds a cascade of CNNs to overcome this problem. This cascade network closely resembles DL-based approaches' iterative reconstruction, however our approach enables for end-to-end optimization of the reconstruction procedure. The proposed method is compared to Dictionary Learning MRI (DLMRI) for 2D reconstruction and to Dictionary Learning with Temporal Gradient (DLTG), kt Sparse and Low-Rank (kt-SLR), and Low-Rank Plus Sparse Matrix Decomposition (L+S) for dynamic reconstruction, which are the state-of-the-art compressed sensing and low-rank approaches, respectively. We show that, for aggressive undersampling rates, the suggested technique outperforms them in terms of reconstruction error and perceptual quality. Furthermore, thanks to GPU-accelerated libraries, images can be reconstructed quickly with this method. Each image can be reconstructed in roughly 23 milliseconds for 2D reconstruction, which is fast enough for real-time applications. Sequences may be reconstructed in 10s for the dynamic example, which is fairly quick for off-line reconstruction approaches.

The usefulness of CNNs for the issue of reconstructing undersampled cardiac MR image data was investigated in this study. The tests reveal that by utilising a network with interleaved data consistency steps, a model capable of reconstructing images can be created. Even though the needed sparsity cannot be actually accomplished in medical imaging, the CS and low rank framework provides a mathematical guarantee for signal recovery, making the method appealing in theory as well as in practise. Despite the fact that this is not the case with CNNs, we have demonstrated that a CNN-based strategy can outperform them empirically. Furthermore, the CNN technique proved capable of rebuilding most anatomical

structures more precisely based on learned priors at very aggressive undersampling rates, whereas traditional methods do not guarantee such behaviour. To summarise, while CNNs can only learn local representations that should not impact global structure, it is yet unclear how the CNN technique works when images contain pathologies or other more variable material. We conducted a cross-validation investigation to ensure that the network can handle data gathered using the same acquisition procedure that was previously unseen. On a larger dataset, generalisation properties must be carefully assessed. However, CNNs are flexible enough that application-specific priors can be added to their objective functions to prioritise maintaining features of interest in the reconstruction, assuming that such expert knowledge is accessible at the time of training. In clinical settings, segmentation and/or registration are frequently used in cardiac image analysis. Multi-task learning appears to be a potential method for enhancing the usability of CNN-based MR reconstructions.

CHAPTER 3

SYSTEM STUDY

This chapter deals with the detailed description of the proposed system and its difference over the existing system.

3.1 EXISTING SYSTEM:

A Compressed Sensing theory is used to speed up the Magnetic resonance imaging(MRI) process. Compressed sensing method undersamples the available raw data collecting from the sensors of the MRI machines. If the undersampled raw data is represented using proper sparse matrix representation, image can be reconstructed in high quality. Existing method uses sparse matrix representation for the reconstruction of images.

A Various algorithm such as Bayesian Group Sparse Representation(BGSR), Patch based non local operator(PANO), Modified-Group Sparsity Dictionary(M-GSOD), Graph-based Redundant wavelet transform(GBRWT) etc are used in sparse matrix representation. GBRWT is introduced to sparsely represent magnetic resonance images in iterative image reconstructions. Within this transformation, image patches is viewed as vertices and their differences as edges, and the shortest path on the graph minimizes the total difference of all image patches. Group Sparsity with an orthogonal dictionary(GSOD) is used to optimize the CS-MRI reconstruction. Non-convex optimization is generalized in soft thresholding to obtain the solution from the lower bound. In these algorithm various dictionaries are used to study the vital information from the patches of the images.

The disadvantages in these methods are

- Produces poor quality image.
- It uses iterative method for finding the optimized patches. The iterative method is slow .
- The computational cost of the iterative method is very high.
- Finding the optimized iterative solutions is very difficult.

To overcome the above mentioned disadvantages , the system is proposed.

3.2 PROPOSED SYSTEM:

The proposed system uses deep learning based image reconstruction. Deep learning have received attention as means for accelerating MRI reconstruction. Deep learning techniques have proved its significance in image processing. Deep learning is entirely different from the sparse matrix representation. Deep learning learns the parameters form the image data automatically without the help of human supervision. Deep learning can also learn non-linear transformation. Deep learning can automatically learn transformations between a noisy image to clean image. Out of the various deep learning models available, the proposed system uses 2 models along with densely connected residual block(DCR) to enhance the reconstructed image quality. Proposed system includes 2 deep learning models.

- Convolutional Neural Network (CNN) with DCR
- U – Net with DCR

3.2.1 CONVOLUTIONAL NEURAL NETWORK WITH DCR:

Standard Convolutional Neural Network consists of convolution layers stacked after other layers. Convolution is followed by activation layers and Normalization layers. Batch Normalization layer is used to normalize the output of the convolutional layer. Densely Connected Residual block is the novel method added to the existing method. Densely Connected Residual (DCR) block consist of 3 convolutional layers activated by ReLU activation function. The DCR block is connected as shown in the figure 1. The f represents the total no of kernels. The input and the output of the DCR block has the same number of feature maps as they have the same number of the kernel size. The input features and the output feature maps are added in the residual layer to get high quality reconstructed image.

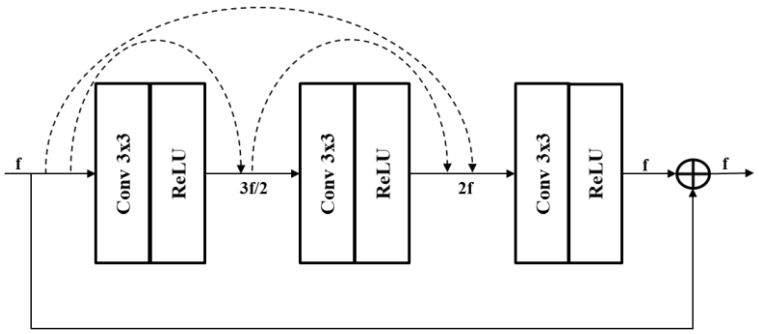

Figure 3.1. Densely connected Residual Block

3.2.2 U-NET WITH DCR:

The U-Net has succeeded in image processing domain. Some of the common application of the U-Net architecture is image segmentation. But U-Net can also be used for MR image reconstruction. The main key feature of the U-Net is its ability to extract features with different image scales using the downsampling and upsampling. The application of the DCR blocks enable to make the network deeper and to increase the total number of model parameters without having training difficulties. U-Net uses skip connections. Skip connections have proved to outperform long deeper neural networks. U-Net consists of 3 sections – encoder, bottleneck, decoder. The encoder extract feature from the image. These features may be size, color, structure, shape, position, dominant edge of the image, regions and textures. The output of the encoder consist of the complex salient features that are extracted from the images. All the features of the image are taken as vectors and represented as feature spaces. The bottleneck section further extract features from the downsized images. The decoder section takes the feature space and deconvolves the extracted features and reconstructs the image with available features that are extracted from the encoder and the bottleneck section.

CHAPTER 4

SYSTEM DESIGN

This chapter deals with the modules used in the project and explains the sequence in which data flows.

4.1 ARCHITECTURAL DESIGN:

Figure 4.1 shows the complete architecture of the proposed method. The architecture diagram shows the processing flow of the data and the output.

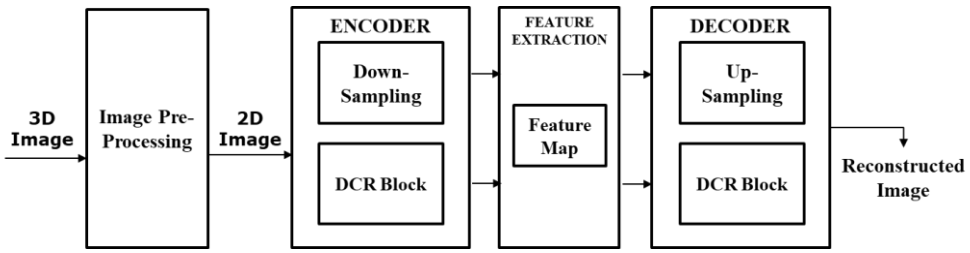

Figure 4.1. Densely connected Residual Block

The system has 4 modules. Each of the modules is discussed in detailed in the further sections.

The data is passed through the first module- Image Preprocessing. After the image is preprocessed the preprocessed image is passed to the two proposed deep network models CNN with DCR and U-Net with DCR. The dataset is splitted into training, testing and validating. The test images are passed through the model and the reconstructed image is obtained. The output reconstructed image is then compared with the other algorithms. The reconstructed image have less noise and high image quality.

4.2 LIST OF MODULES:

There are two models used in this work. Figure 4.2 and Figure 4.3 shows the architecture diagram for the CNN with DCR and U-Net with DCR. The modules for these two models are

Convolutional Neural Network with DCR

- Image Pre-Processing
- Densely Connected Residual (DCR) Blocks

U-Net with DCR

- Image Pre-Processing
- U-Net encoder
- Bottleneck
- U-Net Decoder

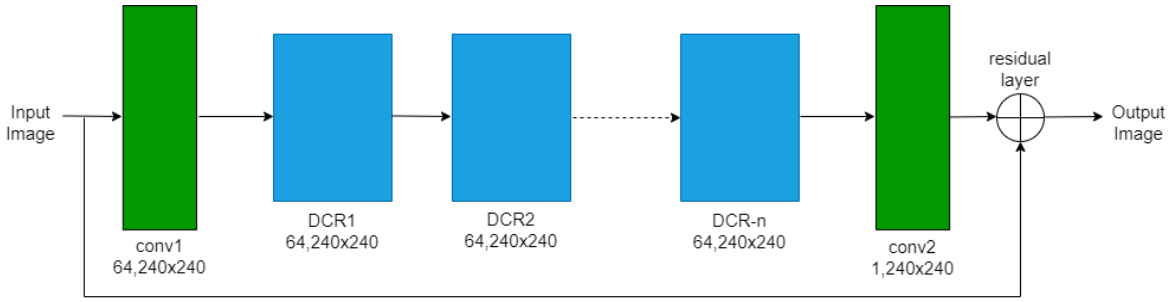

Figure 4.2: Architecture design of the CNN with Densely Connected Residual (DCR)

4.2.1 IMAGE PRE-PROCESSING:

Raw 3D images are unsuitable for training and testing. So the input image has to be preprocessed to transform the given 3D MRI dataset to suitable format for training, testing and validating. The 3D mri image is in the .nii or .nifti format. It is most common format used for multi dimensional files in neuro imaging. Each .nii image consists of voxels which consist of pixels with height, width and depth. Each

3D image consists of 220 voxels. To reduce the computational complexity and to increase the accuracy of the model only 51 center images are taken. After taking 51 clear central images, each pixel is normalized to change the range of pixel intensity to 0 to 1 from 0 – 255. After the preprocessing, the images are now suitable for model building.

4.2.2 DENSELY CONNECTED RESIDUAL BLOCK:

The Densely Connected Residual Block uses skip connection. Skip connection is more efficient than the deep neural networks. Each DCR consist of 3 Convolutional layers that are activated by the ReLU function. The DCR block uses concatenation to merge the features from the previous layers. The input and the output features are added in the residual layer at the last of the DCR block. The DCR block receives the feature maps as inputs and outputs feature maps of the same scale and same depth. The DCR block consists of three convolution layers and three activation layers with a dense connection and a residual connection. These connections provide a sufficient flow of information to extract various features of the MRI image. The gradient vanishing phenomenon that occurs due to long and deeper neural networks is prevented by the connection between the several different layers. This makes the training process of the MRI images more effectively and efficiently.

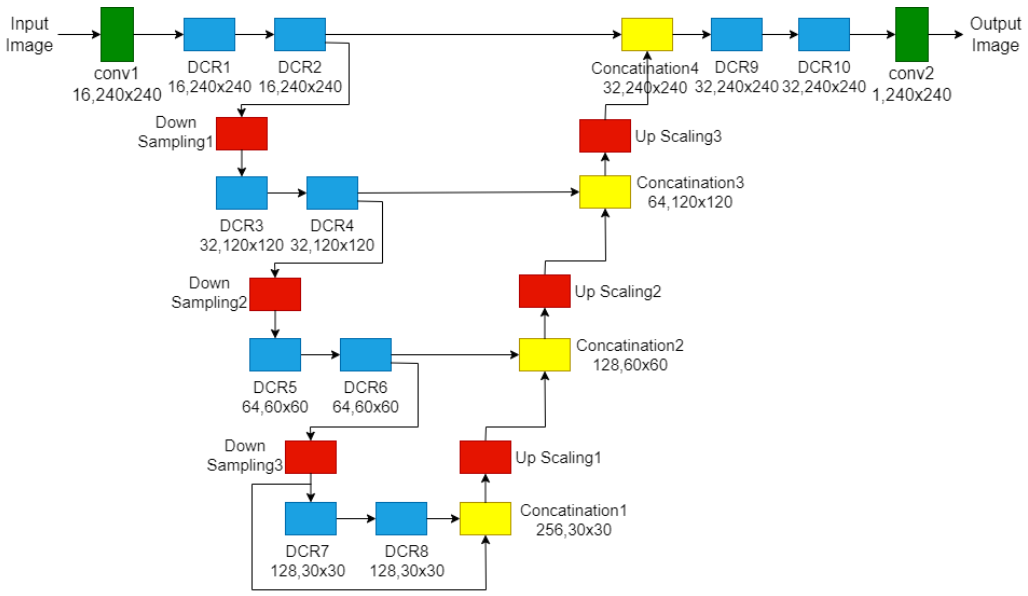

Figure 4.3: Architecture design of U-Net with DCR block

4.2.3 U-NET ENCODER:

Figure 4.3 shows the architecture design of the U-Net. The U-Net encoding section consist of 3 steps. Each steps consist of 2 DCR blocks followed by downsampling. The downsampling layer reduces the image size into half. The initial size of the image is 240x240. The final output of the encoding section is image of size 30x30. The input image is passed into the conv1 of kernel size 3x3 and 16 no of kernels. At each step the no of kernel is doubled and the size of the image is reduced into half. These encoders extract feature from the images. These features may be size, color, structure, shape, position, dominant edges of image, regions and textures. The output of the encoder consist of the complex salient features that are extracted from the images. All the features of the image are taken as vectors are represented as feature spaces.

4.2.4 U-NET BOTTLENECK:

The Bottleneck section consist of two DCR blocks and a concatenation block. The Concatenation layer concatenates the input features of the bottleneck section and the output of the DCR blocks. The concatenate layer concatenates the list of inputs. It takes list of features represented as tensor as input and returns a single tensor that is the concatenation of all inputs.

4.2.5 U-NET DECODER:

The decoder takes the feature space extracted from the encoder and reconstructs the image with available salient features. The decoder acts a transpose of the encoder. The decoder part deconvolves the feature map from the feature space. This is like transpose of the convolution layer. The decoder section merges the layer from the corresponding downsampling step. Concatenation operation is used in the merge layer. The decoder step is the output of the previous step concatenated with the feature map from the corresponding downsampling step. This is used to prevent the loss of salient features. In the last step of the decoding section 2 DCR block are used followed by the final convolution layer of kernel size 1x1. The output of the decoder is the reconstructed image that is provided by the final convolutional layer in the decoder. The output reconstructed image have high image quality and less noise.

CHAPTER 5

SYSTEM IMPLEMENTATION

This chapter deals with the implementation of the each module in the proposed system.

5.1 DATASET COLLECTION:

The 3T MR image dataset is taken from Neuro Imaging Tools and Resources Collaboratory (NITRC). NITRC is a open source environment which consist of various neuro imaging resources for computation. The dataset used here is taken from the NITRC. 36 T1 and DWI pictures obtained on a 3T MRI scanner, behavioral data for two computerized tasks, and three questionnaires make up Cortex. Due to incomplete or incorrect completion of questionnaires/computer activities, the behavioral data occasionally comprises a subset of the 36 individuals. MRI subject numbers are matched to subject numbers (e.g. 1244). (e.g. sc1244). The input is 3D dataset in the nii format. The dataset consists of 36 subjects and 3 Tesla MRI images. Each 3D nii file can be splitted into 220 2D images. These 36 3D images are used for deep learning training.

5.2 TRAINING AND TESTING:

The available dataset must be splitted to training and testing dataset in order to get test the model that is trained using the training dataset. The dataset is splitted in the ratio of 80:20 for training and testing data respectively. Table 5.1 shows the total number of images in the training and testing phases. 1472 images are used for training. 364 images are used in the testing and validation. The validation images are used to check the reconstruction of the image in each epoch. The images must be preprocessed before training and testing.

Subset Name	No of images
Training	1472
Testing/ Validation	364
Total images	1836

Table 5.1 : Train/ Test data splitting

5.3 IMAGE PRE-PROCESSING:

3D image data set is converted into 2D image using array slicing. The 3D image consist of 220 voxels. Each 220 voxels can be splitted to individual image. 220 images are extracted from each 3D image dataset. Each image is of size 240x240. After converting 3D image into 2D image only center 51 images are taken for the training and testing. 51 center core images have the clear image of the brain. Other 169 images have only small part of brain or no part of brain. So the central 51 images are taken from the 220 images using array slicing. Each pixel is represented in integer value of range between 0-255. The value 0 represents black and the value 255 represents white. The pixel must be converted to range of 0 to 1 to feed the image to the model that will be built. Pixel normalization is done to convert the values of each pixel with the range of 0 to 1. Then the images are passed to the training and testing phase. The images are represented in matrix format before passing the input to the models. The images are represented in matrix format because convolution function can be applied on the matrix. The image consist of 3 channel namely Red, Green and Blue.

5.4 MODEL BUILDING :

Two different models are used in the reconstruction of the image. Convolution layer is the main building block of the deep learning. It contains a set of filters also called as kernels and parameters that are learned in the entire training phase. The size of the filters may vary based on the application. Each filter applies a function called as convolution function with the image matrix and stores the output as feature maps or feature vectors. The convolution function F can be represented as show in the equation 5.1

$$F(x_i) = \lambda(W_i * x_i + b_i) \quad (5.1)$$

Where λ represents the convolution function, w represents the weight of the i^{th} layer and b denotes the bias of the i^{th} layer. The residual connection is done by adding the features that are extracted from the previous layers. The residual connection is done as shown in the equation 5.2

$$X_L = F_L (X_L - 1) + X_L - 1 \quad (5.2)$$

The concatenation operation is used to merge the previous layers. The L^{th} layer will input all the feature maps created by the previous convolutional layers. The concatenation of the layer will be represented as shown in the equation 5.3

$$X_L = F_L([X_0, X_1, X_2, \ldots, X_{L-1}]) \quad (5.3)$$

The parameters like growth rate or learning rate and weight for each parameter and each layer are updated using the optimizing function.

5.5 OPTIMIZATION :

The role of optimizer is to reduce the overall loss and improve the accuracy. It modifies the learning rate and the weight of the nodes. The optimizer used in this work is RMSprop(). RMSprop changes the learning rate based on the partial gradients of the image. Gradients provides the direction to reach the local minimum of the loss function. The calculation of mean squared partial derivative for one parameter is given by

$$s(t+1) = (s(t) * rho) + (f'(x(t))^2 * (1.0-rho)) \quad (5.4)$$

where s(t) is the average of the squared partial derivative for one parameter. Rho is the hyper parameter whose value is 0.9 and f'(x(t))^2 is the squared partial derivative of the current parameter. Based on the mean squared partial derivative the learning rate is changed dynamically for each parameter in each iteration as

$$learning\_rate(t+1) = learning\_rate / (1*10^{-8} + RMS(s(t+1))) \quad (5.5)$$

where RMS denotes the root mean square. With the learning rate the weight of the parameters are updated as

$$x(t+1) = x(t) - learning\_rate(t+1) * f'(x(t)) \quad (5.6)$$

where x(t) is the weight of a parameter.

5.6 ACTIVATION FUNCTION:

Activation function is used to calculate output from each node from the weighted sum of input. Two types of activation function are used for this network. One for the hidden layer and the another for the output layer. Rectified Linear Unit Activation(RELU) is used in the input hidden layer. If the input is positive the output will be same as the input. Otherwise the output will be zero. The ReLu is calculated as max(0.0,x). Sigmoid activation is used in the output layer. The output is calculated by the equation 5.7

$$f(x) = \frac{1}{1+e} - x \quad (5.7)$$

5.7 QUALITATIVE METRICS:

Activation Peak Signal to Noise Ratio(PSNR) is commonly used to measure the strength of signal in the image. PSNR can also be used to measure the quality of the image. PSNR is the ratio of the signal power in the signal to the noise in the signal. The PSNR is given by the equation

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAx}_i^2}{\text{MSE}} \right) \quad (5.8)$$

Where MSE stands for mean squared error and Max stands for the greatest pixel intensity. As the image pixels are normalized in preprocessing stage. The pixels range vary from 0 to 1. So the value of MAX = 1. A higher PSNR value represents more image quality.

CHAPTER 6
RESULTS AND DISCUSSIONS

6.1 OVERVIEW:

For MRI (Magnetic Resonance Imaging) reconstruction, we proposed a novel hybrid neural network combining CNN (Convolution Neural Network) with DCR (Densely Connected Residual) blocks and U-Net with DCR (Densely Connected Residual) blocks. The proposed network architecture is based on an encoder-decoder framework that accepts multiple continuous frames as input and reconstructs the image. Features on each input image is abstracted by Convolutional Neural Network Layers. The convolutional layers are sequentially connected to encode the features of all input images. The U-Net encoder outputs are then fed into the U-Net decoder for image upsampling and image reconstruction. Skip connections are used to merge the layer output features. Features are concatenated in the decoder layers.

For performance evaluation, datasets containing MRI images are created. When compared to baseline architectures that use a single image as input, the proposed architecture produced significantly better results, proving the effectiveness of using multiple continuous frames as input. Furthermore, the proposed models were tested on a dataset with extremely difficult MRI images such as noise were added in a MRI image and it also removes the noise and reconstructs the MRI image to ensure their robustness.

6.2 QUALITATIVE METRICS:

The qualitative metrics(PSNR-Peak Signal To Noise Ratio) obtained with each model is presented in this section

6.2.1 PERFORMANCE COMPARISON BETWEEN CNN, U-NET, CNN WITH DCR AND U-NET WITH DCR:

Performance Metrics	CNN	U-Net	CNN with DCR	U-Net with DCR
Reconstructed Image	24.75	30.46	31.64	**33.09**
Input Noisy Image	22.99	23.00	26.02	26.02
Reconstructed Noisy Image	23.19	26.59	29.20	**29.69**

Table 6.1 Performance Metrics Comparison between CNN, U-Net, CNN with DCR and U-Net with DCR

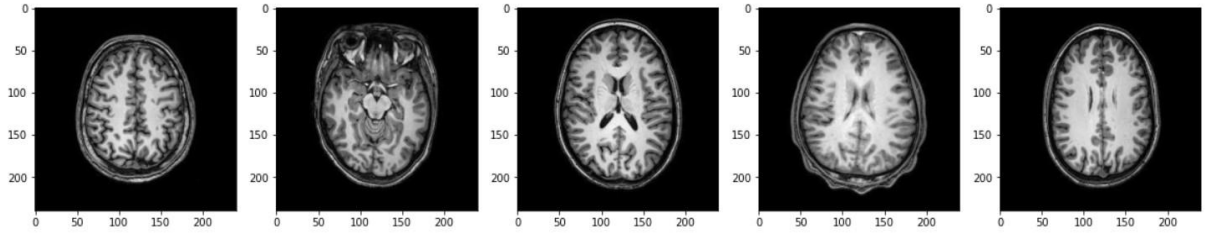

Figure 6.1.Input MRI images

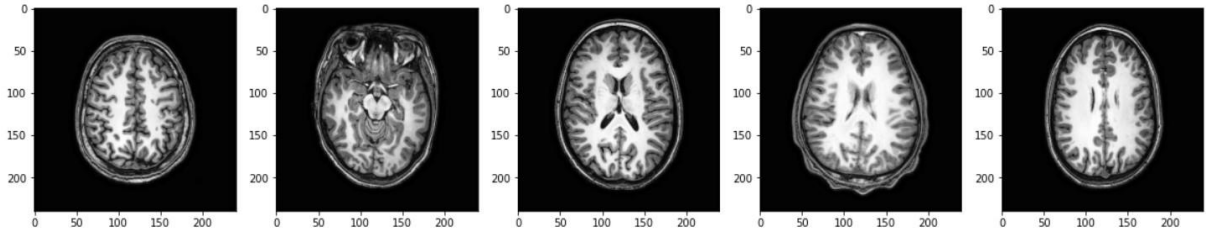

Figure 6.2.Output reconstructed image using CNN with DCR

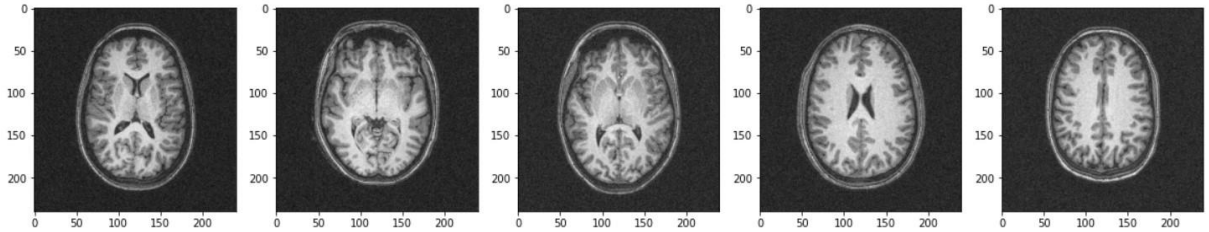

Figure 6.3.Input noisy image with gaussian noise (σ = 0.05)

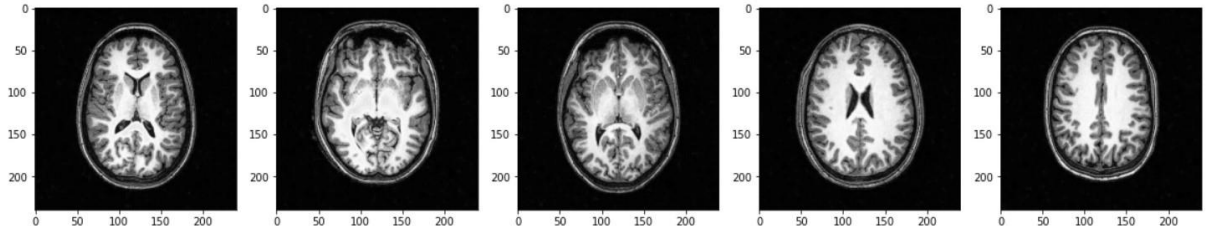

Figure 6.4.Output reconstructed image using CNN with DCR

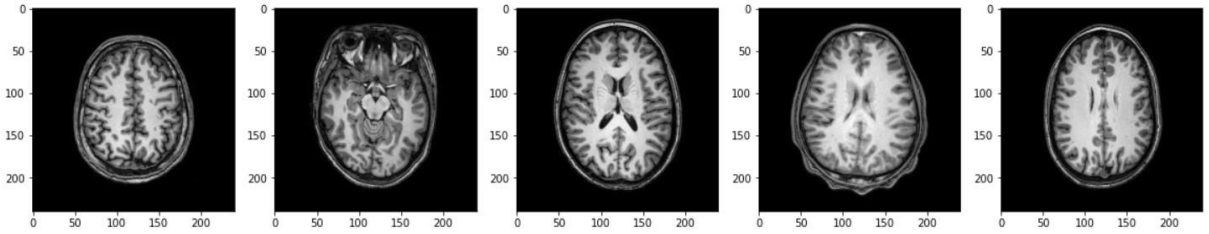

Figure 6.5.Input MRI images

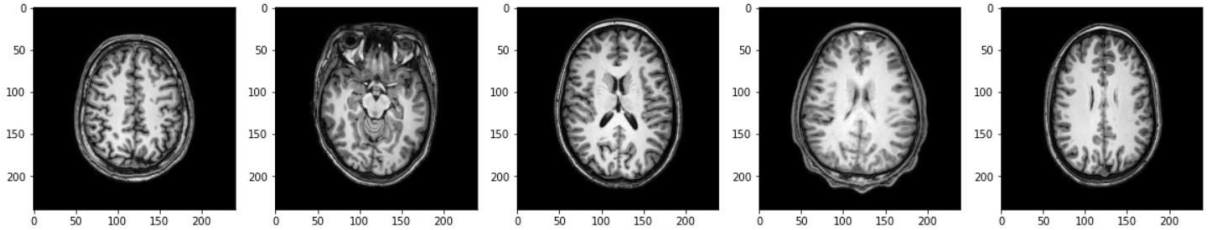

Figure 6.6.Output reconstructed image using U-Net with DCR

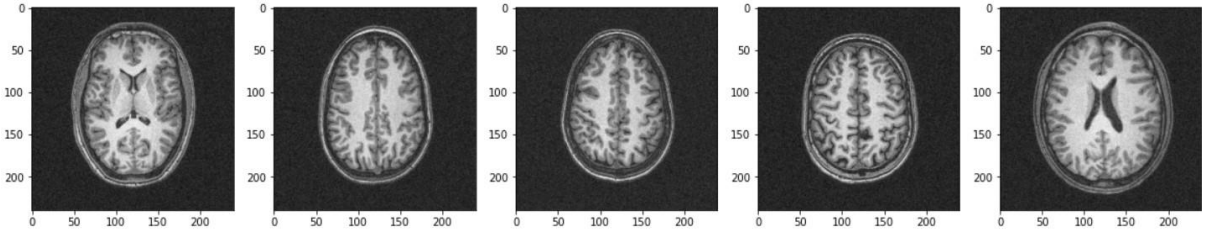

Figure 6.7.Input noisy image with gaussian noise (σ = 0.05)

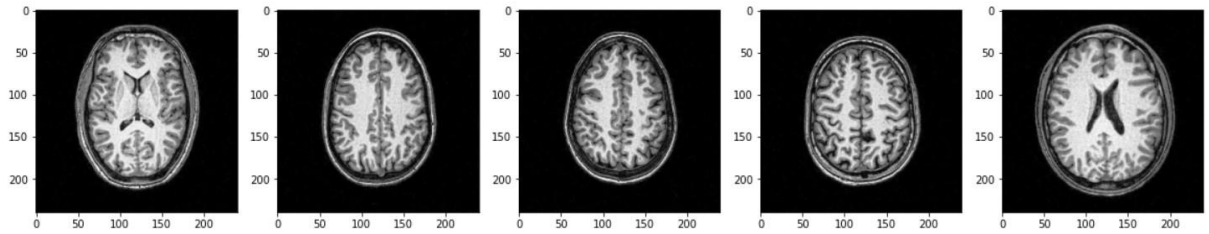

Figure 6.8.Output reconstructed image using U-Net with DCR

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

Other deep learning-based methods do not prevent the vanishing gradient problem, but the dense connection deep learning approach has demonstrated efficacy in a range of image processing problems for preventing the vanishing gradient problem and reducing unwanted noise in the image. We constructed an MR image reconstruction pipeline that includes densely linked structure in this work, which was motivated by dense connections in a variety of problems. To yield promising results, the DCR (Densely Connected Residual) blocks were integrated with two state-of-the-art deep learning frameworks, CNN (Convolution neural network) and U-Net. When compared to original structures with sparse connections, the MR image reconstruction findings reveal that performance is improved and noise in the MR image is reduced. The PSNR (Peak Signal to Noise Ratio) result shows that DCR (Densely Connected Residual) blocks might be utilised to rebuild MR network architecture without the information loss often associated with depth. The findings show that increasing the number of DCR (Densely Connected Residual) blocks enhances reconstruction performance. The effect of DCR(Densely Connected Residual) blocks is improved when compared to CNN(Convolution neural network)-based and U Net-based networks. PSNR values for densely linked CNN(Convolution neural network) structures with three DCR(Densely Connected Residual) blocks are significantly higher than those for the plain CNN(Convolution neural network); the plain CNN(Convolution neural network) PSNR value is 24.75, while the CNN(Convolution neural network) PSNR value is 31.64, a difference of 6.89dB. Similarly, the PSNR value for CNN(Convolution neural network) with three blocks is 29.68, with a 6.49dB increase, for a noisy MR reconstructed picture. Furthermore, the statistical significance of the improved outcomes is validated. Finally, the novel use of dense connections in the form of DCR (Densely Connected Residual) blocks has shown significant promise for use in MRI reconstruction and noise reduction.

7.2 FUTURE ENHANCEMENT

In future this work can be further improved with the help of different algorithms for improving the PSNR(Peak Signal To Noise Ratio)value. This work can be further developed by large datasets(2TB) and by using more techniques for improving the PSNR(Peak Signal To Noise Ratio)value and quality of the reconstructed image..

APPENDIX I

WORKING ENVIRONMENT

HARDWARE SPECIFICATION

- Processor : AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx, 2000 Mhz, 4 Core(s), 8 Logical Processor(s)
- Hard disk : 256 GB SSD
- RAM : 16 GB
- GPU : GTX 1050 Mobile 4GB VRAM

SOFTWARE SPECIFICATION

- Operating System : Windows 10
- Tool : Visual Studio Code, Anaconda, Jupyter Notebook
- Language used : Python 3.9 conda

APPENDIX II

CODING

CNNwithDCR.ipynb

```
import os
import cv2
from keras import layers
from keras.layers import
Input,Dense,Flatten,Dropout,merge,Reshape,Conv2D,MaxPooling2D,UpSampling2D,Conv2DTra
nspose,Dropout
from tensorflow.keras.layers import BatchNormalization
from keras.models import Model,Sequential,load_model
from keras.callbacks import ModelCheckpoint
from tensorflow.keras.optimizers import Adadelta, RMSprop,SGD,Adam
from keras import regularizers
from keras import backend as K
import numpy as np
import numpy.random as rng
from PIL import Image, ImageDraw, ImageFont
import nibabel as nib
from sklearn.model_selection import train_test_split
import math
import glob
import tensorflow as tf
from matplotlib import pyplot as plt
import imutils
import cv2 from PIL import Image
filelist = glob.glob('Dataset/*')
for f in range(len(ff)):
    a = nib.load(ff[f])
    a = a.get_fdata()
    a = a[:,78:129,:]
    for i in range(a.shape[1]):
        images.append((a[:,i,:]))
m = np.max(images)
mi = np.min(images)
images = (images-mi)/(m-mi)
from sklearn.model_selection import train_test_split
train_X,valid_X,train_ground,valid_ground =
train_test_split(images,images,test_size=0.2,random_state=42)
batch_size = 64
epochs = 25
inChannel = 3
x, y = 240, 240
input_img = Input(shape = (x, y, inChannel))
def DCRBlock(features,kernelsize):
    c1 = Conv2D(kernelsize*1.5,(3,3),activation='relu',padding='same')(features)
    c1 = BatchNormalization()(c1)

    conca1 = concatenate([features,c1])

    c2 = Conv2D(kernelsize*2,(3,3),activation='relu',padding='same')(conca1)
    c2 = BatchNormalization()(c2)

    conca2 = concatenate([features,conca1,c2])
```

```
    c3 = Conv2D(kernelsize,(3,3),activation='relu',padding='same')(conca2)
    c3 = BatchNormalization()(c3)

    ress = Add()([features,c3])

    return ress
def CNNwithDCR(input_img,kernelsize):
    conv1 = Conv2D(kernelsize, (3, 3), activation='relu', padding='same')(input_img)

    dcr1 = DCRBlock(conv1,kernelsize)

    dcr2 = DCRBlock(dcr1,kernelsize)

    dcr3 = DCRBlock(dcr2,kernelsize)

    dcr4 = DCRBlock(dcr3,kernelsize)

    dcr5 = DCRBlock(dcr4,kernelsize)

    dcr6 = DCRBlock(dcr5,kernelsize)

    dcr7 = DCRBlock(dcr6,kernelsize)

    dcr8 = DCRBlock(dcr7,kernelsize)

    convfinal = Conv2D(1,1,activation='sigmoid',padding='same')(dcr8)

    ress = Add()([input_img,convfinal])
    return convfinal
opt = RMSprop()
autoencoder = Model(input_img, CNNwithDCR(input_img,64))
autoencoder.compile(loss='mean_squared_error', optimizer = opt)
autoencoder.summary()
autoencoder_train = autoencoder.fit(train_X, train_ground,
batch_size=batch_size,epochs=150,verbose=1,validation_data=(valid_X, valid_ground))
autoencoder.save("mri_model",save_format="h5")
loss = autoencoder_train.history['loss']
val_loss = autoencoder_train.history['val_loss']
epochs = range(70)
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
pred = autoencoder.predict(valid_X)
plt.figure(figsize=(20, 4))
print("Test Images")
for i in range(5):
    plt.subplot(1, 5, i+1)
    plt.imshow(valid_ground[i, ..., 0], cmap='gray')
plt.show()
plt.figure(figsize=(20, 4))
print("Reconstruction of Test Images")
for i in range(5):
    plt.subplot(1, 5, i+1)
    plt.imshow(pred[i, ..., 0], cmap='gray')
```

```
plt.show()
[a,b,c,d]= np.shape(valid_X)
mean = 0
sigma = 0.03
gauss = np.random.normal(mean,sigma,(a,b,c,d))
noisy_images = valid_X + gauss
pred_noisy = autoencoder.predict(noisy_images)
plt.figure(figsize=(20, 4))
print("Noisy Test Images")
for i in range(5):
   plt.subplot(1, 5, i+1)
   plt.imshow(noisy_images[i, ..., 0], cmap='gray')
plt.show()
plt.figure(figsize=(20, 4))
print("Reconstruction of Noisy Test Images")
for i in range(5):
   plt.subplot(1, 5, i+1)
   plt.imshow(pred_noisy[i, ..., 0], cmap='gray')
plt.show()
mse = np.mean((noisy_images - valid_X)**2)
psnr_inputimage = 20 * math.log10(1/math.sqrt(mse))
print("PSNR of the input noisy image",psnr_inputimage)
mse1 = np.mean((pred_noisy-valid_X)**2)
psnr_outputimage = 20 * math.log10(1/math.sqrt(mse1))
print("PSNR of the output reconstructed image",psnr_outputimage)
```

U-NetwithDCR.ipynb

```
import os
import cv2
from keras import layers
from keras.layers import
Input,Dense,Flatten,Dropout,merge,Reshape,Conv2D,MaxPooling2D,UpSampling2D,Conv2DTra
nspose,Dropout
from tensorflow.keras.layers import BatchNormalization
from keras.models import Model,Sequential,load_model
from keras.callbacks import ModelCheckpoint
from tensorflow.keras.optimizers import Adadelta, RMSprop,SGD,Adam
from keras import regularizers
from keras import backend as K
import numpy as np
import scipy.misc
import numpy.random as rng
from PIL import Image, ImageDraw, ImageFont
from sklearn.utils import shuffle
import nibabel as nib
from sklearn.model_selection import train_test_split
import math
import glob
import tensorflow as tf
from matplotlib import pyplot as plt
from skimage import metrics
import imutils
import cv2 from PIL import Image
```

```
filelist = glob.glob('Dataset/*')
for f in range(len(ff)):
    a = nib.load(ff[f])
    a = a.get_fdata()
    a = a[:,78:129,:]
    for i in range(a.shape[1]):
        images.append((a[:,i,:]))
m = np.max(images)
mi = np.min(images)
images = (images-mi)/(m-mi)
from sklearn.model_selection import train_test_split
train_X,valid_X,train_ground,valid_ground =
train_test_split(images,images,test_size=0.2,random_state=42)
batch_size = 64
epochs = 25
inChannel = 3
x, y = 240, 240
input_img = Input(shape = (x, y, inChannel))
def DCRBlock(features,kernelsize):
    c1 = Conv2D(kernelsize*1.5,(3,3),activation='relu',padding='same')(features)
    c1 = BatchNormalization()(c1)

    conca1 = concatenate([features,c1])

    c2 = Conv2D(kernelsize*2,(3,3),activation='relu',padding='same')(conca1)
    c2 = BatchNormalization()(c2)

    conca2 = concatenate([features,conca1,c2])

    c3 = Conv2D(kernelsize,(3,3),activation='relu',padding='same')(conca2)
    c3 = BatchNormalization()(c3)

    ress = Add()([features,c3])

    return ress
def UNetWithDCR(input_img,kernelsize):

    conv1 = Conv2D(kernelsize, (3, 3), activation='relu', padding='same')(input_img)
    dcr1 = DCRBlock(conv1,kernelsize)
    m1 = MaxPooling2D()(dcr1)

    conv2 = Conv2D(kernelsize*2, (3, 3), activation='relu', padding='same')(m1)
    dcr2 = DCRBlock(conv2,kernelsize*2)
    m2 = MaxPooling2D()(dcr2)

    conv3 = Conv2D(kernelsize*3,(3,3), activation='relu', padding='same')(m2)
    dcr3 = DCRBlock(conv3,kernelsize*3)
    m3 = MaxPooling2D()(dcr3)

    u3 = UpSampling2D()(m3)
    c5 = Conv2D(kernelsize,(3,3),activation='relu',padding='same')(u3)
    c6 = Conv2D(kernelsize,(3,3),activation='relu',padding='same')(dcr3)
    con3 = concatenate([c5,c6])

    u2 = UpSampling2D()(con3)
    c3 = Conv2D(kernelsize,(3,3),activation='relu', padding='same')(u2)
    c4 = Conv2D(kernelsize,(3,3),activation='relu', padding='same')(dcr2)
    con2 = concatenate([c3,c4])
```

```
    u1 = UpSampling2D()(con2)
    c1 = Conv2D(kernelsize,(3,3),activation='relu', padding='same')(u1)
    c2 = Conv2D(kernelsize,(3,3),activation='relu', padding='same')(dcr1)
    con = concatenate([c1,c2])
    ddcr1 = DCRBlock(con,kernelsize*2)

    convfinal = Conv2D(1,1,activation='sigmoid',padding='same')(ddcr1)

    return convfinal
opt = RMSprop(learning_rate=1e-3)
autoencoder = Model(input_img, UNetWithDCR(input_img,16))
autoencoder.compile(loss='mean_squared_error', optimizer = opt)
autoencoder.summary()
autoencoder_train = autoencoder.fit(train_X, train_ground,
batch_size=16,epochs=15,verbose=1,validation_data=(valid_X, valid_ground))
pred = autoencoder.predict(valid_X)
lt.figure(figsize=(20, 4))
print("Test Images")
for i in range(5):
    plt.subplot(1, 5, i+1)
    plt.imshow(valid_ground[i, ..., 0], cmap='gray')
plt.show()
plt.figure(figsize=(20, 4))
print("Reconstruction of Test Images")
for i in range(5):
    plt.subplot(1, 5, i+1)
    plt.imshow(pred[i, ..., 0], cmap='gray')
plt.show()
[a,b,c,d]= np.shape(valid_X)
mean = 0
sigma = 0.03
gauss = np.random.normal(mean,sigma,(a,b,c,d))
noisy_images = valid_X + gauss
pred_noisy = autoencoder.predict(noisy_images)
plt.figure(figsize=(20, 4))
print("Noisy Test Images")
for i in range(5):
    plt.subplot(1, 5, i+1)
    plt.imshow(noisy_images[i, ..., 0], cmap='gray')
plt.show()
plt.figure(figsize=(20, 4))
print("Reconstruction of Noisy Test Images")
for i in range(5):
    plt.subplot(1, 5, i+1)
    plt.imshow(pred_noisy[i, ..., 0], cmap='gray')
plt.show()
mse = np.mean((noisy_images - valid_X)**2)
psnr_inputimage = 20 * math.log10(1/math.sqrt(mse))
print("PSNR of the input noisy image",psnr_inputimage)
mse1 = np.mean((pred_noisy-valid_X)**2)
psnr_outputimage = 20 * math.log10(1/math.sqrt(mse1))
print("PSNR of the output reconstructed image",psnr_outputimage)
```

REFERENCES

[1] Y. Jin, G. Yang, Y. Fang, R. Li, X. Xu, Y. Liu, X. Lai, "3D PBV-Net: an automated prostate MRI data segmentation method", Computer Biology Medicine,128 (2021),104160.

[2] Kamlesh Pawar,Gary F.Egan and Zhaolin Chen "Domain knowledge augmentation of parallel MR image reconstruction using deep learning", Science Direct Computerized Medical Imaging and Graphics Vol. 92, issue no. 5, pp:2168-2183,September 2021.

[3] Y. Han, L. Sunwoo, J.C. Ye, "k-space deep learning for accelerated MRI", IEEE Transcation Medical Imaging, 39 (2) (2019) 377–386.

[4] O. Ronneberger, P. Fischer, T. Brox, "U-Net: convolutional networks for biomedical image segmentation", in,Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer, Springer International Publishing, 2015, pp: 234–241.

[5] Z. Ramzi, P. Ciuciu, J.-L. Starck, "Benchmarking deep nets MRI reconstruction models on the fastMRI publicly available dataset", in: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), IEEE, 2020, pp: 1441–1445.

[6] J. Schlemper, J. Caballero, J.V. Hajnal, A.N. Price, D. Rueckert, "A deep cascade of convolutional neural networks for dynamic MR image reconstruction", IEEE Trans. Med. Imag. 37 (2) (2017) 491–503.

[7] Myeongho Jeon, Han-Soo Choi, Junho Lee, Myungjoo Kang," Multi-Scale Prediction For Fire Detection Using Convolutional Neural Network"in Springer Professional Vol. 92, issue no 5, pp:2168-2183, September 2021.

[8] B. Park, S. Yu, J. Jeong, "Densely connected hierarchical network for image denoising", in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.