Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1 : 2024
ISSN : **1906-9685**

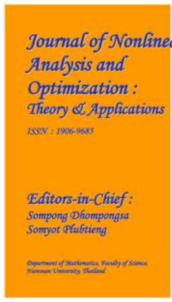

IMAGE QUALITY ASSESSMENT FOR MAGNETIC RESONANCE IMAGING USING CNN

Dr. R. SUNITHA Department of Electronics & Communication Engineering, NRI Institute of Technology, Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212
T. DEVI PRASANNA Department of Electronics & Communication Engineering, NRI Institute of Technology, Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212
T. SRAVANI Department of Electronics & Communication Engineering, NRI Institute of Technology, Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212
P. BHUVANESH RAMAN Department of Electronics & Communication Engineering, NRI Institute of Technology, Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212
V. SHANMUKHA SAI Department of Electronics & Communication Engineering, NRI Institute of Technology, Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212

Abstract:
Assessing quality of medical images is critical because the subsequent course of actions depend on it. Extensive use of clinical magnetic resonance (MR) imaging warrants a study in image indices used for MR images. The quality of MR images assumes particular significance in the determination of their reliability for diagnostics, response to therapies, synchronization across different imaging cycles, optimization of interventional imaging, and image restoration. The image quality assessment for MRI (Magnetic Resonance Imaging) using CNN (Convolutional Neural Network) can be considered as an objective method. Objective methods of image quality assessment aim to quantify image quality using computational algorithms and metrics without involving human perception directly. In the case of using CNNs for MRI image quality assessment, the network is trained on a dataset containing MRI images labelled as noisy and sharp datasets. However, it's important to note that the performance of the CNN model heavily depends on the quality of the training data and the chosen quality metrics.

Keywords:
Image Quality, Deep Learning, Metrics, Reconstruction Quality, Magnetic Resonance Imaging, Convolutional Neural Network.

1.Introduction

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above considerations are taken into account for developing the proposed system.A literature review is a body of text that aims to review the critical points of current knowledge including substantive findings as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources, and as such, do not report any new or original experimental work. Also, a literature review can be interpreted as a review of an abstract accomplishment. Most often associated with academic-oriented literature, such as a thesis, a literature review usually precedes a research proposal and results section. Its main goal is to situate the current study within the body of literature and to provide context for the particular reader.A literature survey on image quality assessment for Magnetic Resonance Imaging (MRI) typically involves reviewing existing research articles, papers, and studies that focus on methods, techniques, and algorithms for

evaluating the quality of MRI images. Researchers in this field aim to develop objective metrics and subjective evaluation methods to assess the clarity, resolution, contrast, and overall diagnostic value of MRI images[1-8].Some common topics covered in literature surveys on image quality assessment for MRI include:
Objective metrics for evaluating image sharpness, noise, artifacts, and contrast-to-noise ratio.
Subjective evaluation methods involving human observers or expert radiologists.
Comparison of different image processing techniques and algorithms for enhancing MRI image quality.
Validation studies to assess the accuracy and reliability of image quality assessment methods.
Here are some links to research papers and articles related to image quality assessment for MRI:
1. "Image quality assessment in magnetic resonance imaging" by S. S. Kruger et al. (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3841259/)
2. "Objective image quality assessment in MRI: a review" by A. R. Reeder et al. (https://pubmed.ncbi.nlm.nih.gov/22522758/)
3. "Quantitative image quality evaluation of MR images" by S. M. Smith et al. (https://www.sciencedirect.com/science/article/pii/S2213158217300491)

Table.1 Research paper titles with author names

S.No	Title	Author Name	Year	Technology
1.	Face Spoofing Detection From Single Images Using Micro-Texture Analysis	Jukka Maatta; Abdenour Hadid; Matti Pietikainen	2011	• Face, • Histograms, • Image resolution, • Gray-scale
2.	Fingerprint Liveness Detection Based on Quality Measures	Javier Galbally; Fernando Alonso-Fernandez; Julian Fierrez; Javier Ortega-Garcia	2010	• Fingerprint recognition, • Fingers, • Biometrics, • Image sensors, • Sensor phenomena and characterization, • Security, • Image matching,
3.	Predicting Iris Vulnerability to Direct Attacks Based on Quality Related Features	Marta Gomez-Barrero; Christian Rathgeb; Ulrich Scherhag; Christoph Busch	2018	• iris recognition; • image morphing; • face recognition
4.	Iris recognition using Gabor filters	Chumg-Chih T sai ; J. Taur ; C. Tao	2009	• Iris recognition • Fractal dimension • Gabor filter
5.	Iris Recognition Using Gabor Filters and the Fractal Dimension	P. Radu; K. Sirlantzis; W.G.J. Howells; S. Hoque; F. Deravi	2013	• Iris recognition, • Gabor filters, • Feature extraction, • Filter banks,

2.Proposed system

SYSTEM ARCHITECHTURE

System architecture refers to the high-level structure or design of a system, which encompasses its components, their interactions, and the principles governing their organization. It serves as a blueprint for building, integrating, and managing complex systems.

CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network, also known as CNN or ConvNet, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what colour each pixel should be. CNNs are a class of Deep Neural Networks that can recognize and classify particular features from images and are widely used for analyzing visual images. Their applications range from image and video recognition, image classification, medical image analysis, computer vision and natural language processing.

CNN has high accuracy, and because of the same, it is useful in image recognition. Image recognition has a wide range of uses in various industries such as medical image analysis, phone, security, recommendation systems, etc.

ARCHITECTURE OF CNN

There are two main parts to a CNN architecture

- A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction.
- The network of feature extraction consists of many pairs of convolutional or pooling layers.
- A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.
- This CNN model of feature extraction aims to reduce the number of features present in a dataset. It creates new features which summarises the existing features contained in an original set of features. There are many CNN layers as shown in the CNN architecture diagram.

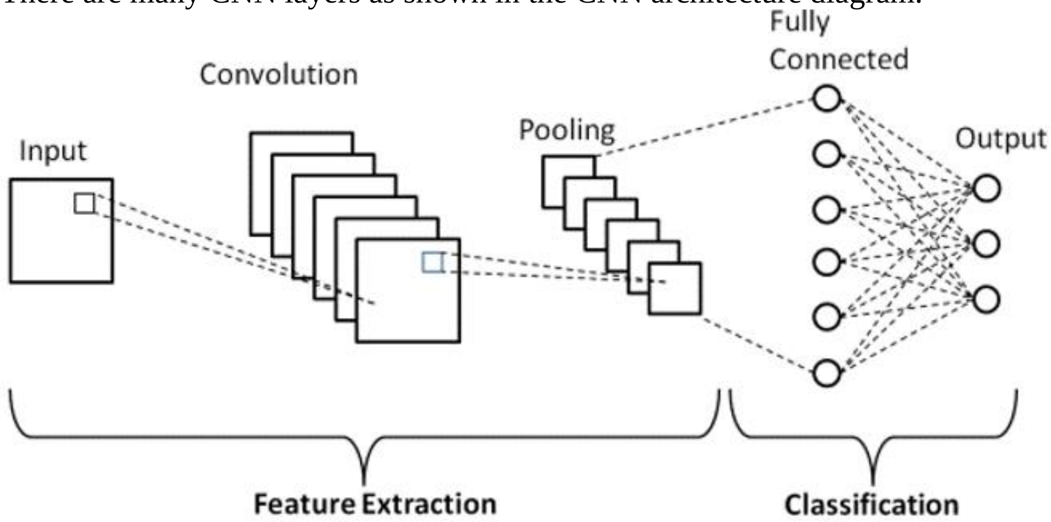

Figure.1.CNN Architecture

CONVOLUTION LAYERS

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below.

1. Convolutional Layer :

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size 3x3. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (3x3). The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

The convolution layer in CNN passes the result to the next layer once applying the convolution operation in the input. Convolutional layers in CNN benefit a lot as they ensure the spatial relationship between the pixels is intact.

2. Pooling Layer :

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map.

Depending upon method used, there are several types of Pooling operations. It basically summarises the features generated by a convolution layer.
In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.
This CNN model generalises the features extracted by the convolution layer, and helps the networks to recognise the features independently. With the help of this, the computations are also reduced in a network.
3. Fully Connected Layer :
The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.
In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place. The reason two layers are connected is that two fully connected layers will perform better than a single connected layer. These layers in CNN reduce the human supervision
4. Dropout :
Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data.
To overcome this problem, a dropout layer is utilised wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.
Dropout results in improving the performance of a machine learning model as it prevents overfitting by making the network simpler. It drops neurons from the neural networks during training.
5. Activation Functions :
Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network.
It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax, tanH and the Sigmoid functions. Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred an for a multi-class classification, generally softmax us used. In simple terms, activation functions in a CNN model determine whether a neuron should be activated or not. It decides whether the input to the work is important or not to predict using mathematical operations.
DENSE MODEL CNN
Densely Connected Convolutional Networks (DenseNet) is a feed-forward convolutional neural network (CNN) architecture that links each layer to every other layer. This allows the network to learn more effectively by reusing features, hence reducing the number of parameters and enhancing the gradient flow during training. In 2016, Gao Huang et al. presented the architecture in their DenseNet paper "Densely Connected Convolutional Networks".
DENSENET ARCHITECHTURE
DenseNet design is founded on a straightforward and basic principle: by concatenating the feature maps of all previous layers, a dense block allows each layer to access the features of all preceding levels. In classic CNNs, each layer only has access to the characteristics of the layer immediately before it.
The architecture of DenseNet is composed of transition layers and dense blocks. Each convolutional layer inside a dense block is linked to every other layer within the block. This is accomplished by connecting the output of each layer to the input of the next layer, producing a "shortcut" link. The transition layers minimize the size of the feature maps across dense blocks that lets the network to grow effectively.

Image classification, object recognition, and semantic segmentation are just some of the computer vision applications where the DenseNet architecture has been shown to reach state-of-the-art performance because of its ability to efficiently leverage feature reuse and decrease the number of parameters.

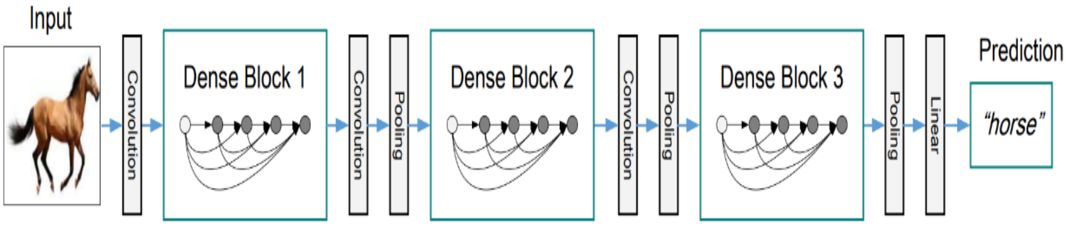

Figure.2. A deep Dense Net with three dense blocks. The layers between two adjacent blocks are referred transition layers and change feature-map sizes via convolution and pooling

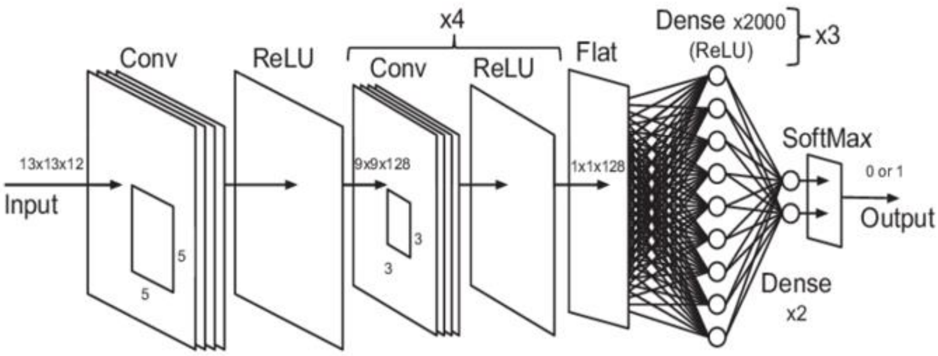

Figure.3. CNN architecture composed of 5 convolutional layers with 128 feature maps, 3 dense layers with 2000 neurons, and a final SoftMax dense layer with 2 neurons.

BENEFITS OF DENSENET

- Performance: As previously stated, DenseNet's state-of-the-art performance can be observed in a range of computer vision tasks including picture classification, object recognition, and semantic segmentation.
- Feature: DenseNet lets each layer access the features of all previous layers, optimizing the gradient flow during training and allows the network to acquire knowledge more effectively.
- Overfitting: The DenseNet design successfully tackles overfitting by lowering the number of parameters and enabling feature reuse, enhancing the model's capacity to generalize to unknown data.
- Vanishing Gradients: The DenseNet design mitigates the vanishing gradient issue by allowing gradients to flow across the whole network, allowing the training of deeper networks.
- Redundancy: The DenseNet design manages redundancy successfully by offering feature reuse and lowering the number of parameters, enhancing the model's capacity to generalize to unknown data.

APPLICATION OF DENSENET

DenseNet is a flexible architecture applicable to a variety of computer vision applications including picture classification, object identification, and semantic segmentation. Among the most prevalent uses of DenseNet are:

- NLP: Used in translation, sentiment analysis, and text generation.
- Generative Models: Used as a generator in generative models, such as generative adversarial networks (GANs), to produce new pictures.
- Object Detection: item recognition in photos and movies including automobiles, people, and buildings.
- Medical Image: To identify and categorize various types of cancers, lesions, and other anomalies.
- Audio: Implemented in audio processing applications including voice recognition, production, and audio synthesis.

- Image: Classifying photos into diverse categories such as wildlife, objects, and settings.
- Semantic Segmentation: Segment pictures into distinct areas such as sky, buildings, or roads.

Additionally, the DenseNet design is readily adaptable to different systems and various purposes.

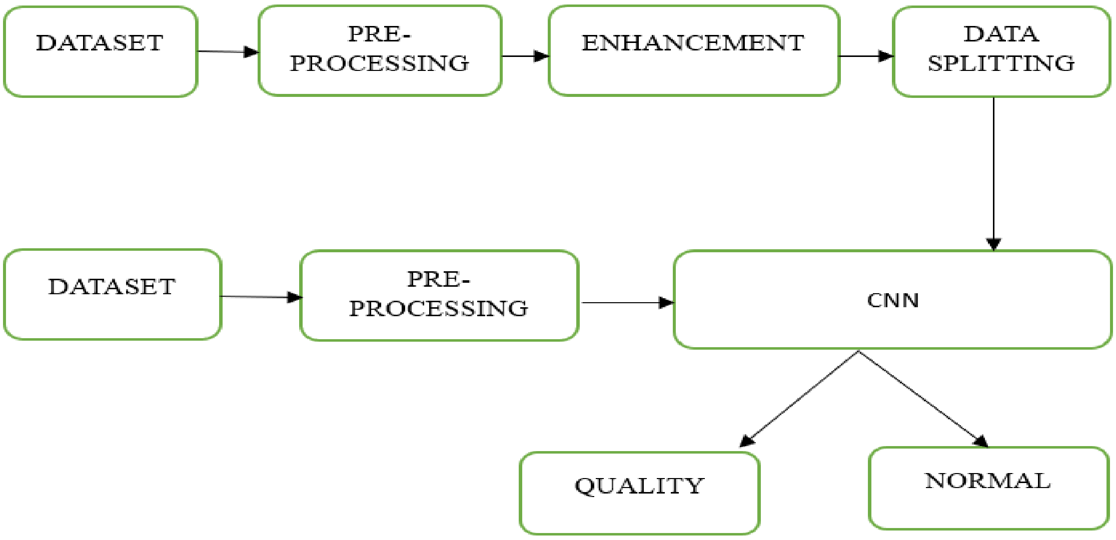

Figure.4. Block Diagram

IMPLEMENTATION MODULES

DATASET COLLECTION

Collecting data for a database involves the process of gathering, organizing, and storing information in a structured and accessible manner. Whether you are building a database for personal use, a business, or research, the following steps provide a general guideline for data collection. It will Gather the data you want to include in your database. This can involve various methods, depending on the nature of the data: Manual Entry: Data can be entered directly into the database by users. Import: Data can be imported from external sources, such as spreadsheets, CSV files, or other databases Data Capture: Automatically collect data through sensors, devices, or web scraping. Here in our project gathering input excel data from the online site to apply with models.

PRE-PROCESSING

Image Pre-processing is a common name for operations with images at the lowest level of abstraction. Its input and output are intensity images. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing.

Image restoration is the operation of taking a corrupted/noisy image and estimating the clean original image. Corruption may come in many forms such as motion blur, noise, and camera misfocus. Image restoration is different from image enhancement in that the latter is designed to emphasize features of the image that make the image more pleasing to the observer, but not necessarily to produce realistic data from a scientific point of view. Image enhancement techniques (like contrast stretching or de-blurring by a nearest neighbour procedure) provided by "Imaging packages" use no a priori model of the process that created the image. With image enhancement noise can be effectively be removed by sacrificing some resolution, but this is not acceptable in many applications. In a Fluorescence Microscope resolution in the z-direction is bad as it is. More advanced image processing techniques must be applied to recover the object. De-Convolution is an example of image restoration method. It is capable of: Increasing resolution, especially in the axial direction removing noise increasing contrast.

ENHANCEMENT

Image enhancement is a process of improving the visual appearance of an image to make it more suitable for a specific application or to make certain features more prominent. Image enhancement techniques aim to highlight relevant information, improve contrast, reduce noise, and enhance overall quality.

DATA SPLITING:

Data splitting is a crucial step in machine learning and data analysis. It involves dividing a dataset into multiple subsets for various purposes, such as model training, model evaluation, and testing. Proper

data splitting ensures that your models are trained and evaluated on different data, helping you assess their performance accurately and avoid issues like overfitting.
Training Set: The training set is the largest portion of the dataset and is used to train a machine learning model. The model learns patterns and relationships within this data. Typically, it comprises more percent of the dataset.
Validation Set: The validation set is used to fine-tune model parameters and to assess the model's performance during training. It helps prevent overfitting. The size is usually around some of the dataset.
Test Set: The test set is used to evaluate the final model's performance. The model should not have seen this data during training or validation. The test set is kept separate from the training and validation sets and is typically less percent of the dataset.

3.Results and Discussion
The total dataset consists of 808 MRI images, in which the total dataset is divided into training dataset and testing dataset, which are further splits as noisy images set (233 MRI images) and sharp images set (171 MRI images). The validation dataset for training the autoencoder model is specified using the validation_data argument in the model.fit() function call. The validation dataset consists of the same data used for training, which includes the blurry images (blurry_frames) and their corresponding clean images (clean_frames).
The encoder and decoder of the autoencoder consist of three convolutional layers each. Here's a breakdown of the filters used in each layer:
Encoder Layers:
The first convolutional layer uses 64 filters with a kernel size of 3x3 and a ReLU activation function.
The second convolutional layer uses 128 filters with a kernel size of 3
x3 and a ReLU activation function.
The third convolutional layer uses 256 filters with a kernel size of 3x3 and a ReLU activation function.
Decoder Layers:
The first deconvolutional layer (Conv2DTranspose) uses 256 filters with a kernel size of 3x3 and a ReLU activation function.
The second deconvolutional layer uses 128 filters with a kernel size of 3x3 and a ReLU activation function.
The third deconvolutional layer uses 64 filters with a kernel size of 3x3 and a ReLU active ation function.
Additionally, the final layer of the decoder uses 3 filters to reconstruct the three color channels (RGB) of the original image, with a kernel size of 3x3 and a sigmoid activation function.
The size of the filter (kernel size) used in the convolutional layers is specified as 3x3. This is a common choice for convolutional neural networks (CNNs) and is widely used in image processing tasks. each convolutional layer is created using the Conv2D function. The kernel_size parameter is set to (3, 3), indicating that a 3x3 filter is applied at each convolutional operation. This means that a 3x3 grid of pixels is used as the receptive field for each convolution operation.
Using a smaller filter size like 3x3 helps capture local patterns and features in the input images while keeping the number of parameters manageable. Larger filter sizes might capture more global patterns but can increase the computational cost and the risk of overfitting, especially in deeper networks.
The padding used in the convolutional layers is specified as 'same'.
The padding='same' parameter setting ensures that the spatial dimensions of the output feature maps from the convolutional layers are the same as the input feature maps. Specifically, 'same' padding adds padding to the input feature maps so that the output feature maps have the same spatial dimensions as the input, given the specified stride and kernel size. This padding strategy helps prevent information loss at the borders of the feature maps during convolution operations, particularly when using strides greater than 1.
To provide a comprehensive output with important parameter specifications, let's summarize the key parts of the code along with their specifications:

The encoder architecture comprises several layers, each with a specific configuration. These layers consist of 64, 128, and 256 filters respectively, with a kernel size of 3x3, a stride of 2, and ReLU activation function. Padding is set to 'same' to maintain the spatial dimensions of the input throughout the encoding process.

The decoder, on the other hand, mirrors the encoder architecture in terms of layer configuration but in reverse order. It begins with 256 filters, followed by 128 and 64. The kernel size, stride, activation function, and padding remain consistent with the encoder. The final output consists of 3 channels for RGB images, and the activation function used here is sigmoid.

There are two dense layers used:

Latent Vector Layer: This dense layer is named 'latent_vector' and is part of the encoder model. It takes the flattened output of the convolutional layers and produces the latent vector representation of the input image.

```
latent = Dense(latent_dim, name='latent_vector')(x)
```

Output Layer of the Decoder: Another dense layer is used in the decoder model to reshape the latent vector back to the shape that can be fed into the transposed convolutional layers.

```
x = Dense(shape[1]*shape[2]*shape[3])(latent_inputs)
```

These are the two instances of dense layers used in the provided code for image denoising using the autoencoder architecture.

During model compilation, the mean squared error (MSE) is chosen as the loss function, while the Adam optimizer with the default learning rate (typically 0.001) is employed. Accuracy is used as a metric to evaluate model performance.

Training is conducted over 10-50 epochs, with a batch size of 32. Following training, the final latent vectors are obtained from the encoder for a set of input images.

For evaluation purposes, three metrics are calculated: Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), and Contrast-to-Noise Ratio (CNR).

Data preprocessing involves loading and resizing images to dimensions of 128x128 pixels, followed by normalization of pixel values to the range [0, 1].

Finally, a random seed of 21 is set to ensure reproducibility throughout the execution of the code.

METRICS CALCULATION (PSNR, MSE, CNR)

Peak Signal-to-Noise Ratio (PSNR):

PSNR is a measure of image quality, indicating how much noise is present relative to the original signal. It's calculated by comparing the maximum possible pixel value (usually 255 for 8-bit images) to the Mean Squared Error (MSE) between the clean and noisy images. The result is expressed in decibels (dB), with higher values indicating higher image quality.

$$PSNR = 10 . \log_{10}(MAX^2/MSE)$$

Where:

MAX is the maximum possible pixel value (typically 255 for 8-bit images).

MSE is the Mean Squared Error between the clean and noisy images.

Mean Squared Error (MSE):

MSE measures the average squared difference between corresponding pixel values of the clean and noisy images. It provides a quantitative assessment of the reconstruction error, where lower MSE values indicate better image fidelity. The MSE is computed by summing the squared differences over all pixels and then averaging the result.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (I_{clean}(i) - I_{noisy}(i))^2$$

Where:

N is the total number of pixels in the image.

$I_{clean}(i)$ and $I_{noisy}(i)$ noisy are the pixel values at location i in the clean and noisy images, respectively.

Contrast-to-Noise Ratio (CNR):

CNR assesses the contrast of an image relative to the noise level present. It's calculated as the absolute difference between the mean pixel values of the clean and noisy images, divided by the square root of the average of their variances. CNR compares the signal strength (represented by the mean pixel values) to the combined noise level (estimated by the variances), with higher CNR values indicating better image contrast relative to noise.

$$\text{CNR} = |\mu_{clean} - \mu_{noisy}| / \sqrt{((\sigma_{clean})^2 - (\sigma_{noisy})^2)/2}$$

Where:

μ_{clean} and μ_{noisy} are the mean pixel values of the clean and noisy images, respectively.

σ_{clean} and σ_{noisy} are the standard deviations of pixel values in the clean and noisy images, respectively.

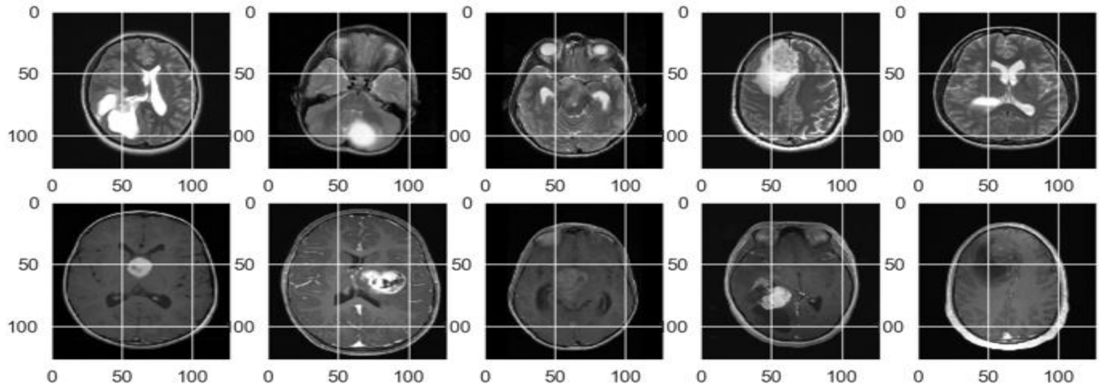

Figure.5.Five pairs of randomly selected clean and blurry images

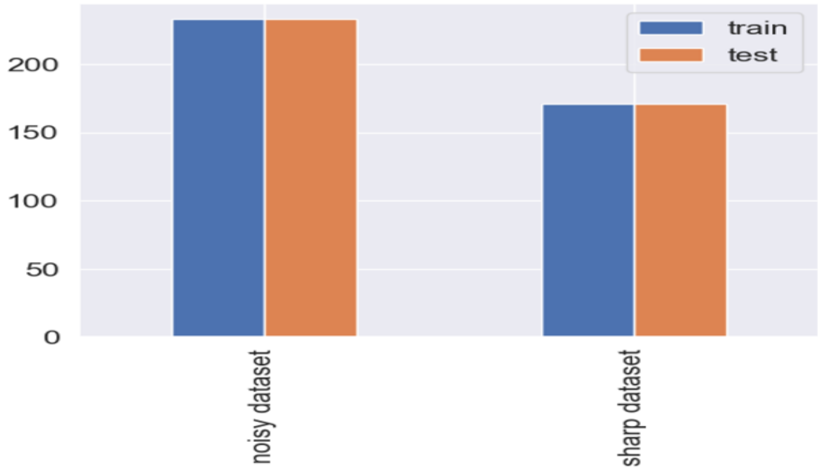

Figure.6. Bar plot shows the counts of each class in both the training and test datasets

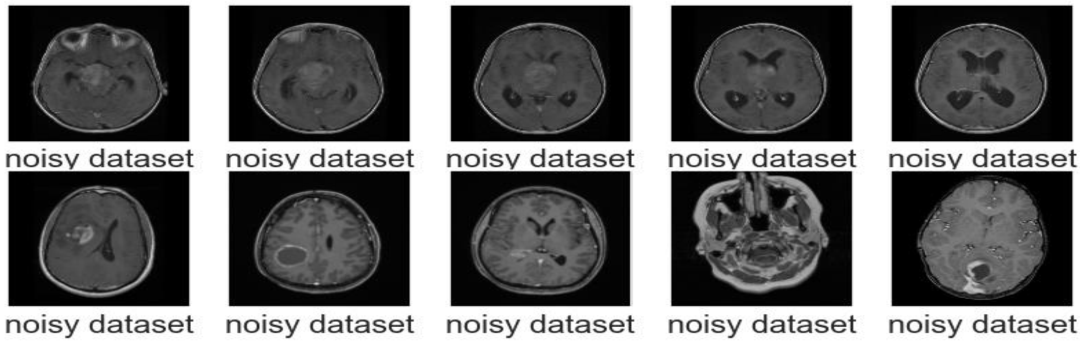

Figure.7.Images of Dataset

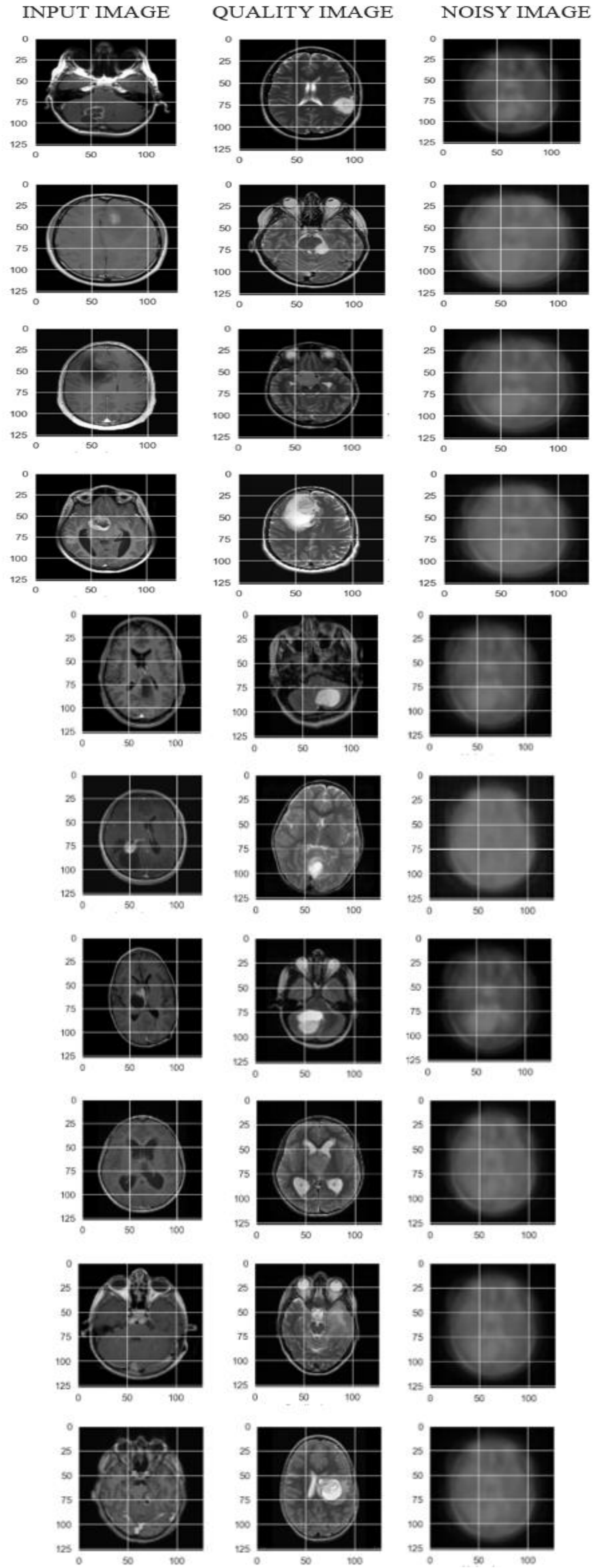
INPUT IMAGE
QUALITY IMAGE
NOISY IMAGE

Figure.8. Input, Quality & Noisy Images

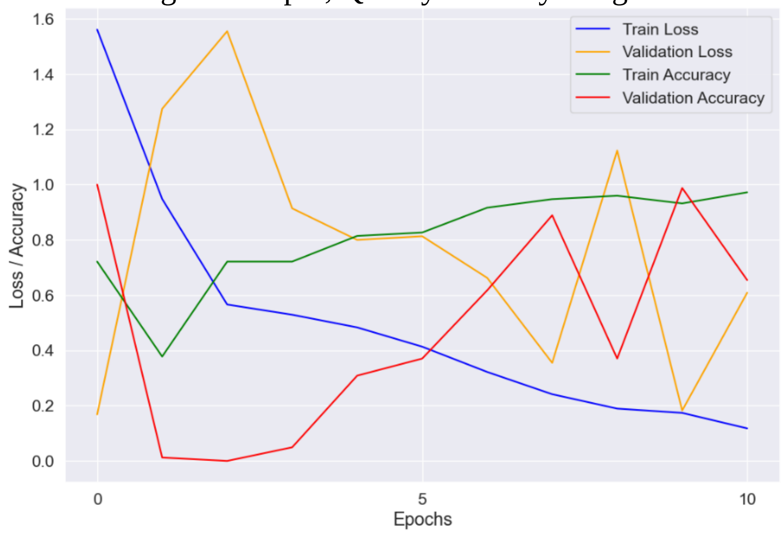

Figure.9.Plot for Training & Validation Loss / Accuracy

Table 2 Metric calculations for input ten images

	PSNR	MSE	CNR
0	15.307131	0.029464	0.153933
1	16.974710	0.020069	0.220428
2	18.056164	0.015645	0.211337
3	17.600845	0.017375	0.178192
4	17.189322	0.019102	0.085381
5	16.831254	0.020743	0.177787
6	15.418454	0.028718	0.347060
7	16.338552	0.023235	0.083741
8	16.050610	0.024828	0.280948
9	15.398601	0.028850	0.048845

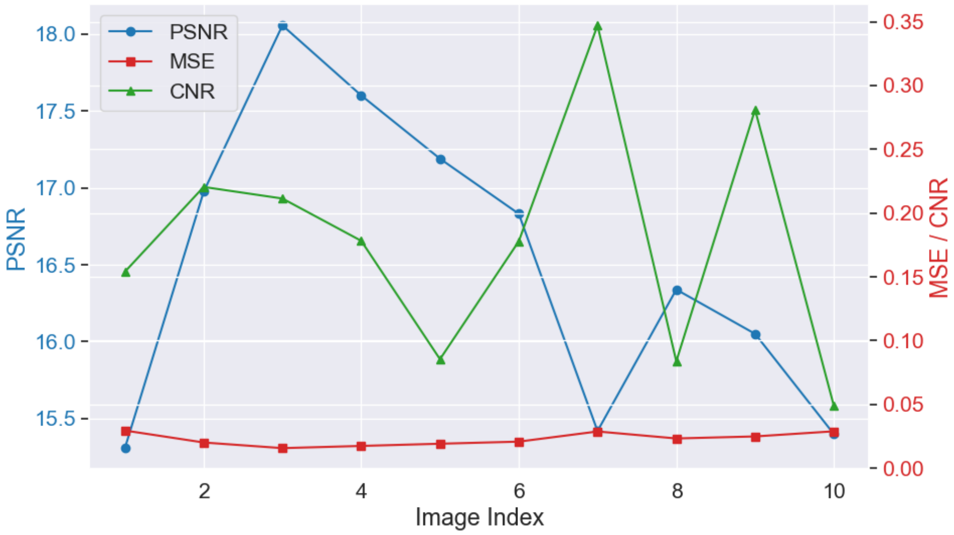

Figure.10.Plot for Comparison of PSNR, MSE, and CNR

4.Conclusion

The architecture of the proposed neural network forms an end-to-end learning mechanism that takes an image as input and directly outputs the quality score. Experimental results verify that the proposed method can yield a more accurate and reliable image quality estimation than existing methods. Additionally,enlarging the set of image samples will be useful for training a deeper neural network and hence further improving the performance of image quality assessment. CNN-based IQA methods can automate the process of assessing MRI image quality, reducing the need for manual inspection and saving time for radiologists and technicians. Traditional IQA methods often rely on subjective evaluations by human observers, which can be inconsistent. CNN-based IQA provides objective metrics for assessing image quality, leading to more reliable evaluations. CNNs are capable of learning complex features from large datasets, allowing them to achieve high accuracy in image quality assessment tasks. CNN-based IQA methods are often robust to variations in image acquisition settings, such as noise levels, artifacts, and resolution changes, making them suitable for different MRI scanners and protocols.

References

1. Mason, J. Rioux, S. E. Clarke, A. Costa, M. Schmidt, V. Keough, T. Huynh, and S. Beyea, ‘‘Comparison of objective image quality metrics to expert radiologists’ scoring of diagnostic quality of MR images,’’ IEEE Trans. Med. Imag., vol. 39, no. 4, pp. 1064–1072, Apr. 2020.
2. R. Li, G. Dai, Z. Wang, S. Yu, and Y. Xie, ‘‘Using signal-to-noise ratio to connect the quality assessment of natural and medical images,’’ in Proc. SPIE, vol. 10806, Aug. 2018, Art. no. 108064Q.
3. G. P. Renieblas, A. T. Nogués, A. M. González, N. Gómez-Leon, and E. G. del Castillo, ‘‘Structural similarity index family for image quality assessment in radiological images,’’ J. Med. Imag., vol. 4, no. 3, Jul. 2017, Art. no. 035501.
4. H. Rajagopal, ‘‘Subjective versus objective assessment for magnetic resonance images,’’ in Proc. 17th Int. Conf Communic Inf. Technol. Eng., vol. 9, 2015, pp. 1–6.
5. Z. Wang, ‘‘Applications of objective image quality assessment methods [applications corner],’’ IEEE Signal Process. Mag., vol. 28, no. 6, pp. 137–142, Nov. 2011.
6. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, ‘‘Image quality assessment: From error visibility to structural similarity,’’ IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.
7. Z. Wang and A. C. Bovik, ‘‘Modern image quality assessment,’’ Synth. Lect. Image, Video, Multimedia Proc., vol. 2, no. 1, pp. 1–156, 2006.
8. A. Mittal, A. K. Moorthy, and A. C. Bovik, ‘‘No-reference image quality assessment in the spatial domain,’’ IEEE Trans. Image Process., vol. 21, no. 12, pp. 4695–4708, Dec. 2012.