A Project report on

ENHANCING RICE QUALITY ASSESSMENT THROUGH INTEGRATED NEURAL NETWORKS: A SYNERGISTIC APPROACH WITH CNN AND CAEs

Submitted in partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

A. MEGHARSHINI	**(204G1A0557)**
S. ANEES FATHIMA	**(204G1A0512)**
V. GIRIVARDHAN	**(204G1A0532)**
M. ANITHA	**(214G5A0503)**

Under the Guidance of

Mr. P. Veera Prakash **M.Tech., (Ph.D)**

Assistant Professor

Department of Computer Science & Engineering

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
Rotarypuram Village, B K Samudram Mandal, Ananthapuramu - 515701

2023-2024

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE)

Rotarypuram Village, BK Samudram Mandal, Ananthapuramu-515701

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Certificate

This is to certify that the Project report entitled ENHANCING RICE QUALITY ASSESSMENT THROUGH INTEGRATED NEURAL NETWORKS: A SYNERGISTIC APPROACH WITH CNN AND CAEs is the bonafide work carried out by **A. Megharshini, S. Anees Fathima, V. Girivardhan, M. Anitha** bearing Roll Number **204G1A0557, 204G1A0512, 204G1A0532, 214G5A0503** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2023 - 2024.

Project Guide	**Head of the Department**
Mr. P. Veera Prakash M.Tech., (Ph.D)	Mr. P. Veera Prakash M.Tech., (Ph.D)
Assistant Professor	Assistant Professor

Date: **External Examiner**

Place: Rotarypuram

DECLARATION

We, Ms. A. Megharshini bearing reg no: 204G1A0557, Ms. S. Anees Fathima bearing reg no: 204G1A0512, Mr. V. Girivardhan bearing reg no: 204G1A0532, Ms. M. Anitha bearing reg no: 214G5A0503 students of **SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY**, Rotarypuram, hereby declare that the dissertation entitled **"ENHANCING RICE QUALITY ASSESSMENT THROUGH INTEGRATED NEURAL NETWORKS: A SYNERGISTIC APPROACH WITH CNN AND CAEs"** embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of **Mr. P. Veera Prakash,** Assistant Professor & Head Of the Department, Department of CSE and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

A. MEGHARSHINI	Reg no: 204G1A0557
S. ANEES FATHIMA	Reg no: 204G1A0512
V. GIRIVARDHAN	Reg no: 204G1A0532
M. ANITHA	Reg no: 214G5A0503

VISION & MISSION OF THE INSTITUTION

Vision:

To become a premier Educational Institution in India offering the best teaching and learning environment for our students that will enable them to become complete individuals with professional competency, human touch, ethical values, service motto, and a strong sense of responsibility towards environment and society at large.

Mission:

- Continually enhance the quality of physical infrastructure and human resources to evolve in to a center of excellence in engineering education.

- Provide comprehensive learning experiences that are conducive for the students to acquire professional competences, ethical values, life-long learning abilities and understanding of the technology, environment and society.

- Strengthen industry institute interactions to enable the students work on realistic problems and acquire the ability to face the ever changing requirements of the industry.

- Continually enhance the quality of the relationship between students and faculty which is a key to the development of an exciting and rewarding learning environment in the college.

VISION & MISSION OF THE DEPARTMENT OF CSE

Vision:

To evolve as a leading department by offering best comprehensive teaching and learning practices for students to be self-competent technocrats with professional ethics and social responsibilities.

Mission:

DM 1: Continuous enhancement of the teaching-learning practices to gain profound knowledge in theoretical & practical aspects of computer science applications.

DM 2: Administer training on emerging technologies and motivate the students to inculcate self-learning abilities, ethical values and social consciousness to become competent professionals.

DM 3: Perpetual elevation of Industry-Institute interactions to facilitate the students to work on real-time problems to serve the needs of the society.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our in debted gratitude to our Guide **Mr. P. Veera Prakash, Assistant Professor, Computer Science & Engineering,** who has guided us a lot and encouraged us in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep felt gratitude to **Mr. C. Lakshminatha Reddy, Assistant Professor and Mr. M. Narasimhulu, Assistant Professor, Project Coordinators** for their valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We are very much thankful to **Mr. P. Veera Prakash, Assistant Professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Bala Krishna, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, We thank all other faculty and non- teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey our gratitude to our families who fostered all the requirements and facilities that we need.

Project Associates

204G1A0557

204G1A0512

204G1A0532

214G5A0503

ABSTRACT

In the realm of food crops, rice stands as a pivotal staple, and ensuring high-quality rice is vital for consistent standards in the industry. Traditional methods for assessing rice quality lack precision, emphasizing the necessity for advanced techniques. The application of neural networks like MobileNet, specifically CNN (Convolutional Neural Networks), offers a promising solution to enhance accuracy and automation in classifying rice grain quality.

Additionally, incorporating the feature learning capabilities of Convolutional Autoencoders (CAEs) further refines the identification of nuanced variations indicative of rice quality. Our goal is to establish a robust framework that not only surpasses the limitations of individual techniques but also delivers optimized and unparalleled outcomes in the task of determining rice quality. This integrated approach aims to revolutionize the evaluation process, ensuring a more accurate and efficient assessment of rice quality for the benefit of human health.

Keywords: *Rice quality, Neural networks, CNN (Convolutional Neural Networks), MobileNet, Convolutional Autoencoders (CAEs), Accuracy, Automation, Classifying, Robust framework, Optimized outcomes.*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CAE	Convolutional Autoencoders
CNN	Convolutional Neural Networks
DL	Deep learning
MAE	Mean Absolute Error
MSE	Mean Square Error
PSNR	Peak Signal Noise Ratio
SSIM	Structural Similarity Index
KNN	K Nearest Neighbour
ReLU	Rectified Linear Unit

CHAPTER 1
INTRODUCTION

In the landscape of food production, the significance of rice as a primary staple necessitates a keen focus on maintaining and enhancing its quality standards. The traditional methods employed for evaluating rice quality often fall short in terms of precision and efficiency, prompting the exploration of more advanced and nuanced approaches. In response to this, our project introduces a novel methodology that harnesses the capabilities of Convolutional Autoencoders (CAEs) and Convolutional Neural Networks (CNN) for the detection and classification of rice quality.

The pivotal role of CNN lies in its adeptness at object identification, specifically tailored to discern and classify various characteristics associated with rice grains. Simultaneously, the integration of Convolutional Autoencoders contributes to the project's depth by facilitating the extraction of intricate features and patterns relevant to rice quality. This combination of CNN and CAEs creates a robust and comprehensive framework for the precise identification of subtle variations in rice quality.

Our project seeks not only to address the shortcomings of conventional methods but also to set a new standard for accuracy and efficiency in rice quality assessment. By capitalizing on the synergies between CAEs and CNN, MobileNet model we aim to surpass the limitations of individual techniques, offering a sophisticated and automated solution for the rice industry. This endeavor aligns with our broader goal of ensuring the consistency of high-quality rice products, thereby contributing to improved standards in the food sector and, ultimately, to the well-being of consumers.

1.1 Problem Statement

The Rice Industry faces a significant challenge in accurately and efficiently detecting quality attributes, such as broken grains, discolorations, and other anomalies within rice grains.

Traditional methods has lack precision and automation, hindering consistent quality assessment and decision-making in rice production processes.

Utilizing Convolutional Neural Networks (CNNs) in conjunction with appropriately curated datasets, we aim to achieve accurate and high-precision classification of rice grain quality attributes.

1.2 Objectives

To accomplish the project's purpose, the following particular objectives have been established.

i. To create a Convolutional Autoencoder (CAE)-based solution for rice quality detection, which aims to capture more complex, abstract features from data and offers better feature learning capabilities and potential for improved performance.

ii. To develop an automated rice quality detection system using Convolutional Neural Networks(CNN) to accurately identify and classify quality attributes within rice grains.

1.3 Machine Learning

Machine Learning is a field of inquiry to understanding and building methods that 'learn', that is, method that leverage data to improve performance on some set of tasks. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine Learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, agriculture, computer vision etc.

1.4 Deep Learning

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain from matching its ability allowing it to learn from large amounts of data. While neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy. There are Convolutional Autoencoders and Convolutional Neural Networks. These two CNNs and CAEs fall under the umbrella of deep learning, which encompasses a wide range of neural network architectures designed for various applications, such as image recognition, natural language processing, and more.

1.5 Introduction to CAE

Convolutional Autoencoders (CAEs) are a variant of autoencoders specifically designed for handling structured grid data, such as images. Unlike traditional autoencoders, which fully connect input and output layers, CAEs leverage convolutional layers for more effective feature extraction and spatial hierarchies. The encoder uses convolutional operations to downsample the input data, capturing hierarchical features. The decoder then employs transposed convolutions to reconstruct the original input from the learned representation. CAEs are widely employed in computer vision tasks, including image denoising, generation, and feature learning, due to their ability to effectively capture spatial relationships and patterns in visual data.

The CAE architecture consists of an encoder network that progressively reduces the spatial dimensions of the input image, capturing essential features. The decoder network then reconstructs the original image from this compressed representation. The use of convolutional layers allows the network to learn spatial hierarchies in a more adaptive manner compared to fully connected layers. This iterative process enhances the network's ability to capture and represent essential features within the images.

1.6 CAEs Essential Elements for Reconstruction

Convolutional Autoencoders (CAEs) primarily consist of two main components: the encoder and the decoder.

1. Encoder: The encoder in Convolutional Autoencoders (CAEs) utilizes convolutional layers to capture spatial hierarchies and extract features from input images. Employing filters and pooling layers, it downsamples spatial dimensions, reducing complexity and generating a compressed representation called the latent space or bottleneck layer.

2. Decoder: In Convolutional Autoencoders (CAEs), the decoder reconstructs the original input by upsampling the compressed representation. Employing transposed convolutional layers for upsampling, it increases spatial dimensions. The final decoder layer produces a reconstructed output, aiming for a close match to the initial input data.

In Convolutional Autoencoders (CAEs), as shown in Fig. 1.1, both the encoder and decoder utilize convolutional layers to capture spatial dependencies and recognize patterns in input data. The convolutional filters within these layers detect features such as edges and textures, forming a hierarchical arrangement that enables the network to learn progressively abstract representations from the input images.

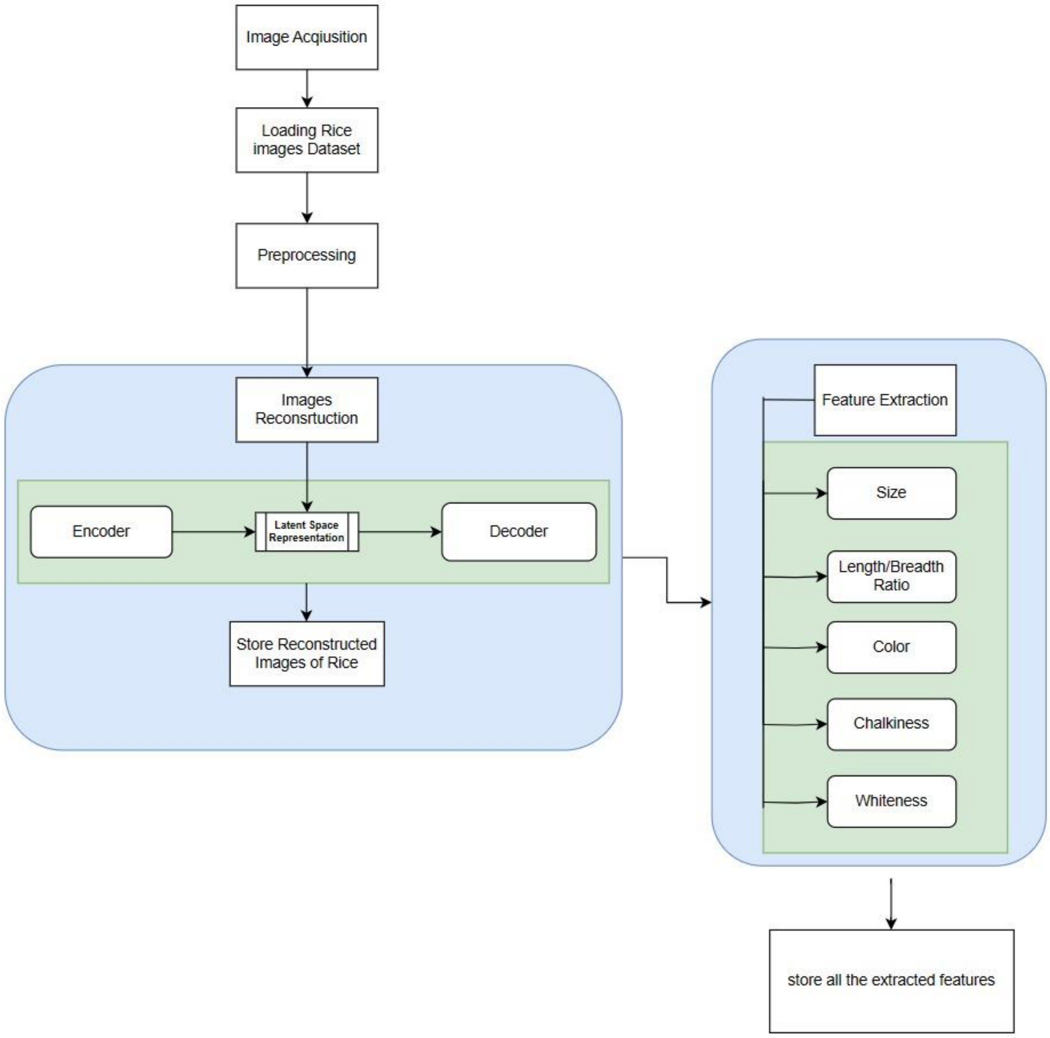

Fig. 1.1: Convolutional Autoencoders Reconstruction

CHAPTER 2

LITERATURE SURVEY

[1] Analysis on Feature Extraction and Classification of Rice Kernels for Myanmar Rice Using Image Processing Techniques

Authors: Thae Nu Wah, Pann Ei San, Thandar Hlaing

Summary: This research proposes an image processing system for rice grading and tests its effectiveness using Paw-San rice. KNN classifier is used for classification. Global Thresholding method is used in this work for image segmentation. Then, a series of measurements were done using image processing techniques on three classes of Paw-San rice in Myanmar. The real-field feature of Paw-San rice is percentage of broken rice contained in the batch. Also the classification of rice can be improved by using for using more distinct features. The results confirm that the feature extraction and classification of rice kernel based on image processing for Myanmar Rice. From this we get accuracy in their range of 83-100%.

Dataset: The dataset used for this study is 3 types of Paw-San Rice, they have taken data from google.

Methodology: K-NN classifiers used for classification, Flatbed Scan(FBS), Bwareaopen (one of the morphological operation method),Otsu Method, and Watershed transformation method are used for classification of rice grain images.

[2] Classification Model of Wheat Grain based on Autoencoder

Author: S. Wentao

Summary: In this study, Deep learning has been used in speech recognition, image search, image recognition. In the wheat particle database, this model is used by deep learning (DL) models like autoencoders to categorize and identify photos of damaged wheat. Deep Learning learning toolbox. It has 2 hidden layers, each of which has 100 nodes. Deep learning highlights the importance of feature learning, Each layer in the model is transformed by features, and finally gets more abstract features at the top layer, which solves artificial design problem levy consumption of time and resources. This experiment is based on the MATLAB R2015a operating platform, using a wheat particle recognition database, and using Matlab's. Experiments quantitative statistical results show that by adjusting the parameters of DL during the training of the autoencoder network, the training error of samples will be smaller, and the error of test samples will be reduced, which will improve correct classification and recognition of diseased wheat images.

Dataset: The data set contains 4 files, which are the images of the training samples, the labels of the training samples, the images of the test samples, and the labels of the test samples.

Methodology: FFNN(Feedforward Neural Network) and SAE (Stacked Autoencoder) Network**.**

[3] Deep Learning for Rice Quality Classification

Author: N. Hong Son and N. Thai-Nghe

Summary: In this work, two categories of rice (whole rice and broken rice) were recognized using image processing algorithms and machine learning techniques. This work proposes an approach for rice quality classification. In this approach, image processing algorithms and machine learning methods were used to recognize and classify two difference categories of rice (whole rice and broken rice) based on the rice's size of the national standard of rice quality evaluation, using Convolutional Neural Network (CNN). Experimental results for 2000 real images give 93.85% accuracy. The system also used Support Vector Machines method with HOG features and k-Nearest Neighbors methods in order to classify and compare the accuracy of those algorithms which show the results of 85.06% and 84.30% accuracy, respectively. These results show that rice quality evaluation and classification could be automatically done using Deep Learning approach.

Dataset: In this study the Datasets are self-collected images about the rice pattern of Loc Troi 20 breed and captured by 20.7 MP camera of Sony Z1 smartphone.

Methodology: Convolutional Neural Networks(CNN) and Support Vector Machine(SVM).

Extensions Proposed: As a future challenge ,we can try another edge detection technique that detects in less time, increasing the mentioned accuracy while decreasing analysis time.

[4] An automated inspection system for rice seed quality based on deep learning.

Author: Gao, Y., Wang, W., Zhang, W., & Gao, X.

Summary: This presents an automated inspection system for rice seed quality based on deep learning techniques. The study aims to address the limitations of traditional rice seed quality inspection methods, which are time-consuming, labour-intensive, and subjective. The proposed system uses deep learning algorithms to automatically classify rice seeds based on their appearance features. The authors collected a dataset of 18,000 rice seed images from different varieties and manually labelled them with their corresponding quality levels. The study aims to address the limitations of traditional rice seed quality inspection methods, which are time-consuming, labour-intensive, and subjective. The proposed system uses deep learning algorithms to automatically classify rice seeds based on their appearance features.

Dataset: The dataset used for this study is taken from google, kaggle.

Methodology: Convolutional Neural Networks(CNN) for finding the quality of the rice seeds.

[5] Rice quality analysis using deep learning techniques : A review

Author: Latha, M. N., & Nandhini, R

Summary: In this, discusses the importance of rice quality analysis in the food industry and the limitations of traditional methods . It then provides an overview of deep learning techniques, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs), and their applications in rice quality analysis. The authors discuss various aspects of rice quality analysis, including size and shape, texture, and composition, and review the deep learning-based methods that have been developed for each of these aspects. They also discuss the challenges and future directions for deep learning-based rice quality analysis, including the need for larger and more diverse datasets, more robust and interpretable models, and the integration of multiple modalities.

Dataset: The dataset used for this study is taken from google, kaggle.

Methodology: It then provides an overview of deep learning techniques, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs), and their applications in rice quality analysis.

[6] Machine vision based quality analysis of rice grains

Author: T. G. Devi, P. Neelamegam and S. Sudha

Summary: In this study, this algorithm used simple morphological images processing methods and it can be used to perform various images based analysis. In machine vision based testing, we take both physical (grain shape and size) and chemical characteristics (amylose content, gel consistency) for evaluation and grading of rice grains. Quality assessment is done by finding 1) the region of boundary and 2) the end points of each grain by measuring the length, breadth and diagonal size of grain. Majority of the operations used in this algorithm are based on Matlab commands. In an embedded imaging based system, images are acquired using CMOS/ CCD based image sensors. This algorithm is suitable to grade large number of grains efficiently and results were compared with manual work which provided the percentage of accuracy greater or much equally to the manual work.

Dataset: The dataset used for this study is taken from google.

Methodology: Get input image -> Binary conversion -> Edge detection -> Finding dimensions -> Sorting and labelling -> Grading.

Extensions Proposed: As a future challenge , the rice grains are identified with diverse varieties by extracting the features like length, shape, color and texture properties with different Machine learning and deep learning methodologies.

[7] Rice Quality Analysis Using Machine Learning

Author: Annadasu, P. and Jaisharma, K

Summary: In this work, it is applied to create a useful model that facilitates rice grain grading without requiring more effort. This paved the way for development of computerized vision in rice quality inspection. In the proposed method both image processing and machine learning techniques are clubbed to analyze and grade the quality of rice kernels with the help of Support Vector Machine (SVM) classifier in python platform. Quality of rice is determined based on the physical and chemical characteristics like area, length, width, moisture content, whiteness, milling degree etc,. Segmentation is acquired by many techniques like based segmentation, and segmentation by clustering. From the results obtained we can conclude that this system is efficient and cost-effective.

Dataset: In this study the Datasets are self-collected images contains multiple rice grains in a single image.

Methodology: The model used Support Vector Machine (SVM) for identifying the rice grains in the taken image.

[8] Comparison of CNN-based deep learning architectures for rice diseases classification

Author: Md Taimur Ahad, Yan Li, Bo Song, Touhid Bhuiyan

Summary: This study focuses on the performance comparison of convolutional neural network (CNN) architectures in detecting and localizing rice diseases. The researchers conducted a rice disease classification comparison of six CNN-based deep-learning architectures using a database of nine of the most epidemic rice diseases in Bangladesh.. They applied a transfer learning approach to DenseNet121, MobileNetV2, Resnet152V, Seresnext101, and an ensemble model called DEX to compare the six individual CNN networks, transfer learning, and ensemble techniques. Transfer learning can increase the accuracy by 17% from the results obtained by Seresnext101 in detecting and localizing rice leaf diseases.

Dataset: The dataset used for this study is taken from PlantVillage, google.

Methodology: It has used six CNN-based deep-learning architectures they are DenseNet121, Inceptionv3, MobileNetV2, resNext101, Resnet152V, and Seresnext101.

[9] Rice Leaf Disease Recognition using Local Threshold Based Segmentation and Deep CNN

Author: Anam Islam, Redoun Islam, S. M. Rafizul Haque, S.M. Mohidul Islam, Mohammad Ashik Iqbal Khan

Summary: The method uses local threshold-based segmentation and the Convolutional Neural Network (CNN) to segment disease-affected regions of rice leaves. The proposed method has been applied on three different datasets, including the one created by the authors, which consists of rice leaf images collected from the Bangladesh Rice Research Institute (BRRI). Three state-of-the-art CNN architectures, VGG, ResNet, and DenseNet, have been trained with these three datasets for classifying the diseases. This study used 4500 images for the experiment. The images were preprocessed by cropping manually and resized to reduce the time for training. The authors used transfer learning on AlexNet which is a relatively small scale and old CNN model. They used various image augmentation techniques on these collected images and managed to get test accuracy of 91.23%. The classification performance of the proposed method using the said three CNN architectures for the three datasets has been analyzed and compared.

Dataset: In this study the Datasets are self-collected images, contains three datasets of Rice Leaf blast, Bacterial Leaf Blight, Sheath Blight.

Methodology: In this, Convolutional Neural Network (CNN) VGG, ResNet, and DenseNet are used to identify the images of rice leaves which contains the disease.

[10] Convolutional neural network with transfer learning for rice type classification.

Author: Patel, Vaibhav.

Summary: This paper proposes a deep learning-based method for identifying rice types, reducing manual labor and error. Two methods are proposed: one uses a deep convolutional neural network (CNN) trained on segmented rice images, and the other uses a combination of a pretrained VGG16 network and the proposed method, using transfer learning for improved accuracy. We have proposed a transfer learning based approach for Basmati rice type classification. Though the rice type classification for Basmati rice group is a harder task than classification of rice types in general or rice groups, our proposed network with pretrained weights of VGG16 perform better than the other approaches. The approach can also classify rice grains as broken or fine. The architecture, pretrained on ImageNet data, significantly improves classification accuracy despite distinct rice images.

Dataset: The dataset used for this study is taken from google, contains 5 types of rice Basmati, Arborio, Jasmine, Ipsala, Karacadag.

Methodology: Deep convolutional neural network (CNN) and a pretrained VGG16 network is used for identifying the rice type of the given input images.

[11] CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction

Author: Jana, Debasish & Patil, Jayant & Herkal, Sudheendra & Nagarajaiah, Satish & Dueñas-Osorio, Leonardo.

Summary: In order to detect, classify, and reconstruct faults in sensor data, the study introduces a novel deep learning framework for linear systems with time-invariant parameters. The Convolutional Neural Network (CNN) in the framework is used to identify the type and existence of faults. For reconstruction, a set of separately trained Convolutional Autoencoder (CAE) networks is used for each type of fault. The models demonstrated strong performance for both simulated and experimental datasets with a single failure. The study empirically proves that the proposed framework performs better than other state-of-the-art techniques in terms of computational efficiency with comparable accuracy. Adoption of this framework in online structural health monitoring applications can lead to minimal disruption to monitoring processes, reduced downtime for structures and infrastructure, while simultaneously reducing uncertainty and improving the quality of sensor data for historical records. They achieved 100% accuracy in faulty sensor localization, over 98.7% accuracy in fault type classification, and over 98% accuracy in reconstruction.

Dataset: The dataset used for this study is taken from google, contains 50000 images of 5 types Normal, Missing, Random, Drift, Spiky.

Methodology: The model used Convolutional Autoencoder (CAE) and Convolutional Neural Network (CNN) to identify the type and existence of faults, this leads to minimal disruption to monitoring processes, reduced downtime for structures and infrastructure.

[12] Terahertz spectra reconstructed using convolutional denoising autoencoder for identification of rice grains infested with Sitophilus oryzae at different growth stages

Author: Pu, Hongbin, et al.

Summary: Convolutional denoising autoencoders, or AEs, have the potential to perform better when they are created by fusing the CNN model with automatic feature extraction and automatic parameter optimization. To get rid of contaminants from the surface and grains without rice, the grains were first soaked in water for thirty minutes. Additionally, the CNN model in deep learning (DL), a successful feature extraction technique, has been extensively employed to effectively extract information from THz spectrum data. In order to further reduce noise and obtain THz spectrum data, the CDAE along with AE and CNN was built. The AE, an unsupervised data reconstruction method, mainly consists of an encoder and a decoder, which can better represent the information of the original data with hidden layers. The CDAE mainly contained an encoder and a decoder with similar structure, which both have three convolutional layers with the convolutional kernel of the size 3×1 and step size 1. The number of convolutional kernels of the encoder increased gradually to 32, 64 and 128 respectively, while the number of convolutional kernels of the decoder gradually decreased to 128, 64 and 32 respectively. Mean square error (MSE) is a loss function that was employed to assess how well the CDAE performed in recreating the THz spectrum.

Dataset: In this study the Datasets are self-collected images, the dataset contains multiple rice grains in a image.

Methodology: Convolutional Autoencoders (CAE) and Convolutional Neural Network (CNN) it is applied to detect Bacterial Spot disease in peach plant, which is caused by a bacterium named Xanthomonas Campestris. This model can be used to detect other plant diseases, as well.

[13] Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network.

Author: Bedi, Punam, and Pushkar Gole.

Summary: Convolutional Neural Networks (CNNs) and Convolutional Autoencoders (CAEs) are two Deep Learning methods that are widely employed in computer vision applications because of their superior performance on picture data. These two methods extract different temporal and spatial elements from image data by using the convolution procedure. CNNs are utilized for input image classification into their appropriate classes, whereas CAEs are employed for effectively reducing an image's dimensionality. This research suggests a novel hybrid model based on CNN and CAE for automatic plant disease diagnosis. Convolutional Neural Network is a Deep Learning technique that uses convolution operation instead of simple matrix multiplication. As compared to other Deep Learning techniques, CNN deals with images most efficiently. Depicts the architecture of a typical CNN that contains one Input layer, one Output layer, a set of Convolutional layers, Pooling layers, and Fully Connected layers. A unique hybrid model is built to automatically detect plant illnesses, using fewer training parameters than current state-of-the-art systems reported in the literature. Two deep learning methods are used by this model: CNN and CAE.

Dataset: In this study the Datasets are self-collected images contains different leaf images of different plants.

Methodology: Convolutional Denoising Autoencoders (CDAE) and Convolutional Neural Network (CNN) model in deep learning (DL), a successful feature extraction technique, has been extensively employed to effectively extract information from THz spectrum data.

[14] Variety identification of single rice seed using hyperspectral imaging combined with convolutional neural network.

Author: Qiu, Zhengjun, et al.

Summary: Four types of rice seed were imaged hyperspectrally at two distinct spectral bands (380–1030 nm and 874–1734 nm). The spectral data were retrieved between 441 and 948 nm (Spectral range 1) and 975 and 1646 nm (Spectral range 2). Different numbers of training samples were used to build CNN, SVM, and K nearest neighbors (KNN) models. Models in the Spectral range 2 of KNN, SVM, and CNN outperformed those in the Spectral range 1 by a little margin.They modified the VGGNet architecture to handle inputs of one-dimensional spectra. Spectral curve patterns and picture patterns are similar in a few ways. VGGNet's modular design facilitates easy modification and extension, and it is chosen because to its outstanding performance in image classification tasks.The model's functionality increased.

Dataset: In this study the Datasets are self-collected images, the rice grain images are arranged in the form rectangle box and taken as input.

Methodology: Convolutional Neural Network (CNN), Support Vector Machine(SVM), and K nearest neighbors (KNN) are used to identify the variety of the single rice grain in the given input image.

[15] Deep-rice: Deep multi-sensor image recognition for grading rice.

Author: Wu, Yiqiang, et al.

Summary: DeepRice is based on a deep learning architecture. To be more precise, Deep-Rice uses a multi-view CNN architecture to extract discriminative features from several rice image views and then uses a modified softmax loss function to try and optimize the CNN parameters. Alongside this deep model, we also constructed a large-scale rice dataset, known as FIST-Rice, to serve as a foundational resource for food security research. Every sample is photographed in three distinct lighting scenarios. Using the FIST-Rice dataset, we compare the suggested Deep-Rice model with the AdaBoost and SVM techniques. We release the first multi-sensor rice grading dataset called FIST-Rice(Future Intelligent System Technology), which would greatly advance the field of rice grading. The outcomes of the experiment suggest that the Deep-Rice model performs better under various light intensity settings. Using a modified softmax, the deep network may learn angularly discriminative features.

Dataset: In this study the Datasets are self-collected images, they are captured using different cameras and using different angles. Every sample is photographed in three distinct lighting scenarios

Methodology: Convolutional Neural Network (CNN), and Support Vector Machine(SVM) used to assess the quality of rice grains.

[16] Automatic rice plant disease recognition and identification using convolutional neural network.

Author: Rathore, Narendra Pal Singh, and Lalji Prasad

Summary: Computer-aided methods are used to process results from disease detection quickly and accurately. Convolutional neural networks (CNNs), which automatically extract characteristics and classify images using fully linked networks, are used by the deep learning models. We have classified images using CNN architecture. We are utilizing input photos with the labels "healthy" and "leaf_blast" from the Kaggle dataset. In this study, 1000 samples of rice crops were used to train CNN using an RGB color model. We have used CNN model to process the images with the labeled class such as healthy & leaf blast. Weights are updated automatically in the training process of CNN that will be able to extract the features of the image. In our proposed architecture CNN model is having 2 layers. The suggested architectural design for differentiating between leaf-blast and healthy rice crops. OpenCV, Numpy, and Keras are the deep learning packages used in the Python development of this model. We employed a batch size of 32, which is a hyper-parameter in deep learning that may be changed. In this paper, we have trained a Sequential convolutional neural network and obtained a prediction accuracy of up to 99.61%.

Dataset: In this study the Datasets are collected from kaggle, it contains leaf_blast images.

Methodology: Convolutional Neural Network (CNN) is used to identify the leaft_blast images whether they are healthy or infected.

[17] Enhancing the Classification Accuracy of Rice Varieties by Using Convolutional Neural Networks.

Author: Tran-Thi-Kim, Nga, et al

Summary: This study uses Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) models to classify 17 rice grain kinds that are commonly cultivated in Vietnam. Seventeen rice varieties commonly planted in Vietnam were used in this study include DT8, HT1, IR4625, IR50404, IR6976, ML48, MO6162, OM108, OM3673, OM4218, OM429, OM4900, OM5451, OM6976, OM8108, OMCS2012, and RVT. There were five rice grains in each image. Each variety was represented by 200 images, from which 100 images were used for training, with the other 100 for testing. The images for training and testing were fixed for all experiments in this study. Pre-trained VGG16 and Resnet50 models serve as the foundation for the two CNN models—modified VGG16 and modified ResNet50. The CNN models were fed the image dataset, while the ANN was fed the feature dataset. The findings demonstrate that the classification accuracy of the 17 types of rice was considerably increased by the updated VGG16 and ResNet50 models. The findings demonstrated that, in comparison to the SVM, the classification accuracy of a CNN model was much higher. Using the feature dataset as a model, the ANN produced a 92.82% classification accuracy.

Dataset: In this study the Datasets are self-collected images, the rice grain images were scanned at a resolution of 2400 dpi. There were five rice grains in each image.

Methodology: Convolutional Neural Network (CNN), Artificial Neural Network(ANN) is used for rice variety classification.

CHAPTER 3
METHODOLOGY

Convolutional Autoencoders (CAEs) are a variant of autoencoders specifically designed for handling structured grid data, such as images. Unlike traditional autoencoders, which fully connect input and output layers, CAEs leverage convolutional layers for more effective feature extraction and spatial hierarchies. The encoder uses convolutional operations to downsample the input data, capturing hierarchical features. The decoder then employs transposed convolutions to reconstruct the original input from the learned representation. CAEs are widely employed in computer vision tasks, including image denoising, generation, and feature learning, due to their ability to effectively capture spatial relationships and patterns in visual data.

3.1 Loading Rice Images Dataset

Loading the Rice Images Dataset involves retrieving a collection of images representing various aspects of rice grains. Researchers typically employ data loading techniques using programming languages like Python and libraries such as TensorFlow or PyTorch. This dataset is essential for training and evaluating machine learning models, especially those designed for tasks such as quality assessment or classification in the context of rice grain analysis.

3.2 Preprocessing

Preprocessing is a crucial step in data preparation for machine learning tasks. It involves techniques such as normalization, data augmentation, and handling missing values to enhance the quality and consistency of the dataset. These preprocessing methods contribute to improved model performance by mitigating noise, ensuring uniformity, and facilitating better generalization during training.

3.3 Image Reconstruction

In CAEs, image reconstruction involves encoding input images into a latent representation and then decoding them back into reconstructed images. The objective is to minimize the reconstruction error, ensuring that the decoded images closely match the original input, thus capturing meaningful features in the process.

Encoder: In image reconstruction using CAEs, the encoder is responsible for transforming input images into a compressed latent representation. This encoded representation contains essential features extracted from the input, facilitating efficient storage and subsequent reconstruction during the decoding phase. The encoder's role is crucial in capturing meaningful information for faithful image representation.

Decoder: In image reconstruction using autoencoders, the decoder reconstructs images from their compressed latent representations, restoring them to a format similar to the input. The decoder plays a crucial role in generating faithful reconstructions by translating the learned features from the latent space back into the original data space.

3.4 Feature Extraction

3.4.1 Size: The size of feature extraction is a crucial factor impacting both model performance and computational efficiency in machine learning as shown in Table 3.1, a compact feature representation is essential for reducing memory requirements and speeding up computations during training and inference. Efficient feature extraction not only enhances the scalability of models but also facilitates deployment on resource-constrained devices, contributing to the overall effectiveness of the machine learning system.

3.4.2 L/B Ratio: The aspect ratio (l/b ratio) in feature extraction plays a pivotal role in maintaining a balance between capturing long-range and short-range patterns, thereby influencing the model's capability to learn diverse spatial information as shown in Table 3.1, a higher aspect ratio tends to emphasize long-range dependencies, enabling the model to grasp broader contextual relationships, while a lower aspect ratio focuses on finer details and short-range patterns.

3.4.3 Color: The mean color in feature extraction serves as a representative measure of the average RGB color values within an image, offering valuable insights into color-based information for machine learning tasks as shown in Table 3.1, analyzing the mean color enables the model to discern dominant hues and variations, facilitating tasks such as image classification and segmentation based on color characteristics. This feature provides a concise summary of the image's color distribution, enhancing the model's ability to understand and interpret visual content with a focus on color-related patterns.

3.4.4 Chalkiness: Chalkiness pertains to the degree of lightness or whiteness in an image, a parameter commonly employed in image analysis to characterize texture or material properties as shown in Table 3.1, chalkiness measurement allows for the quantification of textural features related to visual appearance. By incorporating chalkiness as a feature, machine learning models can effectively discern and classify variations in material texture, providing valuable insights for tasks such as quality assessment in crops or food products.

3.4.5 Whiteness: Whiteness signifies the degree of lightness or brightness within an image, typically quantified using metrics such as luminance or color intensity as shown in Table 3.1, this parameter holds significance across diverse applications, including image analysis and quality assessment. Assessing whiteness aids in characterizing the overall brightness of visual content, providing a valuable feature for machine learning models to discern and interpret variations in illumination and color intensity in tasks such as image recognition and quality control.

Table 3.1: Features of Reconstructed images

S. No	Size	Shape(L/B ratio)	Chalkiness	Whiteness	Mean color (BGR)
1	1790.5	1.942	0.091	24.79	(25.4,24.7,24.7)
2	1685.5	0.666	0.079	24.79	(23.2,22.5,22.6)
3	2025.0	0.796	0.122	29.24	(29.9,29.1,29.1)
4	1870.0	1.169	0.095	25.99	(26.8,25.9,25.8)
5	1872.5	1.037	0.052	23.54	(24.0,23.5,23.4)
6	1579.0	0.767	0.076	22.20	(22.4,22.1,22.1)
7	1509.5	1.875	0.075	21.05	(21.1,21.1,21.1)
8	1703.5	1.340	0.060	21.86	(22.2,21.7,21.8)
9	1943.9	0.057	0.075	25.77	(26.4,25.6,25.7)
10	1760.0	0.730	0.087	24.95	(25.5,24.8,24.9)
11	1748.5	0.646	0.090	24.48	(24.4,24.4,24.4)
12	2082.5	1.850	0.102	33.66	(33.6,33.6,33.6)
13	1855.0	1.131	0.051	24.06	(24.0,24.0,24.0)
14	2027.0	1.055	0.057	25.97	(27.0,26.9,26.9)
15	1783.0	1.842	0.061	24.02	(24.4,23.9,24.2)

Training and Testing datasets were created from the data after the reconstructed images were gathered and features were retrieved using convolutional Autoencoders. Convolutional Neural Networks (CNN) and MobileNet are the algorithms we utilized to train the model. For training and testing sets, arrange image data into the proper directories. Here, we utilized an unlabeled dataset consisting of 75,000 images divided into five varieties: Arborio, Basmati, Jasmine, Ipsala and Karacadag. Each rice variety comprises 15,000 images, and the dataset has been split into three classes - High, Low and Medium - for each variety. These classes are divided based on the features that were extracted from the CAEs. For the training and testing datasets, the data is also divided into three classes.

When loading and preprocessing your image data, you would typically ensure that all images are resized to these specified dimensions before being fed into the neural network for training. Image height and Image width, these dimensions are crucial because they determine the size of the input tensor that the neural network expects. The values are often chosen based on the characteristics of the dataset and the requirements of the neural network architecture.

3.5 Convolutional Neural Network

This Convolutional Neural Network architecture involves hyper parameters, plays a crucial role in shaping the behavior and expressive power of the convolutional layer. Choosing appropriate values for these parameters depends on the nature of the data, the complexity of the patterns to be learned, and the overall architecture of the neural network. Some of the hyper parameters are:

3.5.1 Conv2D Layer: The Conv2D layer serves as a foundational element in convolutional neural networks (CNNs), specializing in 2D convolutions tailored for image processing tasks. Operating on input data, this layer utilizes filters to extract spatial hierarchies and detect patterns within the visual input. By convolving the input across both dimensions, Conv2D plays a key role in capturing meaningful features, making it a cornerstone in the architecture of CNNs, especially well-suited for tasks involving image analysis and recognition.

3.5.2 Filters: The filters parameter is a critical aspect of Conv2D layers in a neural network, defining the number of individual filters or kernels that the layer will learn during training. Each filter specializes in detecting specific patterns or features within the input data, contributing to the network's ability to capture hierarchical representations of the input information.

3.5.3 Kernel Size: The kernel size parameter in a Conv2D layer specifies the dimensions of the convolutional window or filter applied to the input data.A smaller kernel size captures fine-grained details, while larger sizes help capture broader patterns, providing a crucial mechanism for controlling the scale and scope of feature extraction in convolutional neural networks for tasks like image processing and pattern recognition.

3.5.4 Activation Function: The activation parameter in a Conv2D layer defines the activation function applied element-wise to the output produced by the convolution operation. This function introduces non-linearity, enabling the neural network to model complex relationships and patterns within the input data. Common activation functions include Rectified Linear Unit (ReLU) for introducing non-linearity and improving convergence during training. The choice of activation function is crucial for shaping the model's capacity to capture and represent intricate features in tasks such as image recognition or object detection.

When building and training a neural network, tuning these hyper parameters often involves experimentation and iterative refinement to achieve optimal performance on the specific task at hand. Epochs refer to the number of times the entire training dataset is processed by the model. Each pass through the entire training dataset is called one epoch.

CHAPTER 4

PLANNING

4.1 Existing System:

The existing system performs the classification of rice grains into their respective divided classes based on the quality taken by the features of its images.

4.1.1 Disadvantages of the Existing System:

1. Results will be based on user input Image.
2. Image preprocessing steps performed were only suitable for Rice grain Images.

4.2 Proposed System:

In this project, we propose a novel approach that combines Convolutional Neural Networks (CNN) and Autoencoders to address the task of rice quality detection. We propose an innovative system that integrates Convolutional Neural Networks (CNN), MobileNet, and Autoencoders to optimize rice quality detection. The proposed system leverages the fusion of CNN, MobileNet, and Convolutional Autoencoder (CAE) methodologies. The CAE, implemented using TensorFlow or PyTorch, efficiently reconstructs rice images for feature extraction. Subsequently, the MobileNet and CNN models are employed for accurate image classification, capitalizing on MobileNet's lightweight architecture for efficiency. The novel combination of CAE and CNN, MobileNet aims to enhance accuracy and efficiency in rice quality assessment, holding promise for practical applications in the food industry, particularly for quality control and assurance.

Our application accepts the uploaded image of low resolution and then enhances and assures the correct detection of Rice grain quality based on reading its features and model training. It gives output the type of rice grain(high, low, medium) and classifies if given multiple images at the same time into its classes.

4.2.1 Advantages of Proposed System:

1. Performs detection of real-time images uploaded by users in our application based on trained data.
2. Classification of Rice grain quality (three classes high, low, and medium). Also, detect the disease after classification using the CNN algorithm.

3. Image pre-processing is performed in a generalized manner.
4. The difference between the manual division of classes of rice grain images and the results given by the Model is so low which indicates the model performance is accurate.

4.3 Convolutional Neural Network Model(Basic layers)

4.3.1 Pooling Layers: As shown in Fig. 4.1, Pooling layers in a Convolutional Neural Network (CNN) play a pivotal role in downsampling spatial dimensions while preserving essential features. Max pooling, a common technique, extracts the maximum value from a set of neighboring elements, effectively reducing the resolution of feature maps. This process aids in enhancing computational efficiency, decreasing model sensitivity to spatial translations, and promoting the extraction of dominant features for robust pattern recognition in tasks like image classification**.**

4.3.2 Convolution Layers: As shown in Fig. 4.1, Convolution layers in a Convolutional Neural Network (CNN) are fundamental components responsible for extracting features from input data. These layers apply learnable filters to input images, detecting patterns and spatial hierarchies. By leveraging local connectivity and weight sharing, convolution layers efficiently capture features, making them well-suited for tasks like image recognition. The application of multiple filters in parallel enables the CNN to learn diverse hierarchical representations, facilitating effective feature extraction and abstraction.

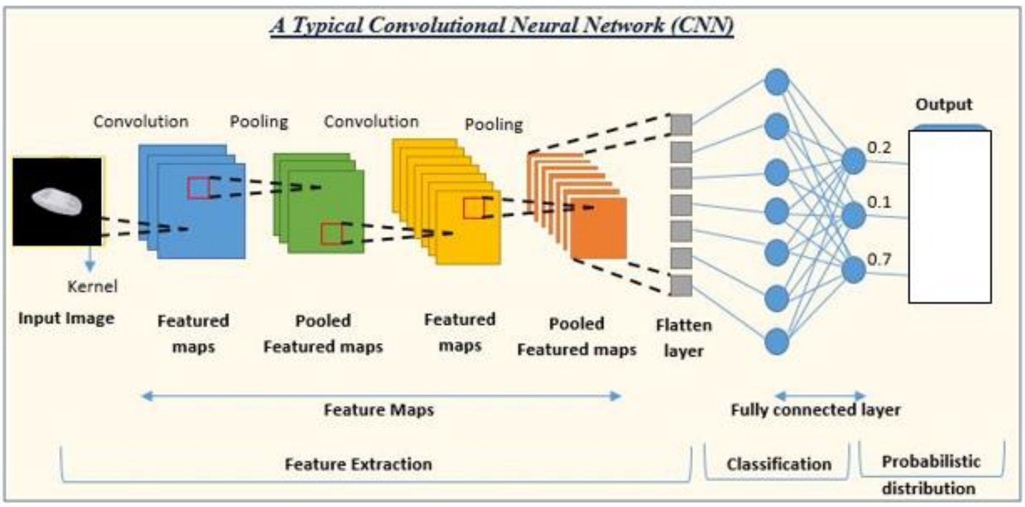

Fig. 4.1: CNN Model

4.4 Objectives of Planning:

4.4.1 Scope: This project is applicable for the Quality detection of Arborio, Basmati, Ipsala, and Jasmine Rice images and analyzing their Features. It is used by agricultural practitioners or farmers or anyone user can access our application.

4.4.2 Performance: To acquire an accuracy greater than 98% by using Convolutional Neural Networks(CNN), by increasing the training data of model and by giving more images until the line of overfit.

4.4.3 Cost :

- Mode = Organic
- Effort = 16 Person-Month Development
- Time = 4 Months
- Average Staff Required = 4 Persons

4.4.4 Time: On a whole this project takes about 4 months (16 person-months) of time period for development and testing.

Table 4.1: Modeling steps in design of each modules

Jan				Feb				Mar				Apr			
W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Elicit Requirements															
		Modularize Requirements													
				Develop and Integrate Modules											
				Business Modeling	Data Modeling		Process Modeling		Application Generation	Testing and Turnover					
												Test the final product and deliver			

As shown in Table 4.1, the timeline outlines modeling steps: requirements elicitation, modularization has done in month of January, module development, integration, modeling has done in month of February-March, and final testing, delivery has done in month of April.

4.5 Modules:

- Data Pre-processing
- Building CNN model
- Classification, Detection, and Comparison

4.6 Functional and Non-Functional Requirements:

4.6.1 Functional Requirements:

- Scan an image to give input to the system.
- Process and take input of the image, detect the Quality type and compare the results.
- Display the type quality of the Rice image.
- Show the percentage accuracy of classified images.

4.6.2 Non-Functional Requirements:

- Usability: The application is easy to use with very less complexity of interface.
- Availability: Available within the device.
- Scalability: This model can be trained for other Rice images.
- Performance: Performs well with no sort of failures.

4.7 System Requirements:

4.7.1 Software Requirements:

- Google Colaboratory
- Visual Studio App

4.7.2 Software Requirements:

- Processor: Minimum Intel Core I5
- RAM: Minimum 8 GB
- Hard Disk: Minimum 250 GB
- Operating System: Windows10

4.8 Test Case:

Input:

The input image, as shown in Fig. 4.2, is a real-time image used for testing purpose.

Fig. 4.2: Input Image

Output:

As shown in Fig. 4.3, the output image describes whether the given rice image is of high quality, low quality, or medium quality based on the training data. In addition, the output image includes a probability score, which indicates how accurately the model classified the quality of the rice image.

```
Result:  Quality is: High
1/1 [==============================] - 0s 23ms/step
Score: 0.9999977350234985
```

Fig. 4.3: Output Image

CHAPTER 5

DESIGN

5.1 Architecture Diagrams:

An architectural diagram is a visual representation that maps out the physical implementation for the components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element.

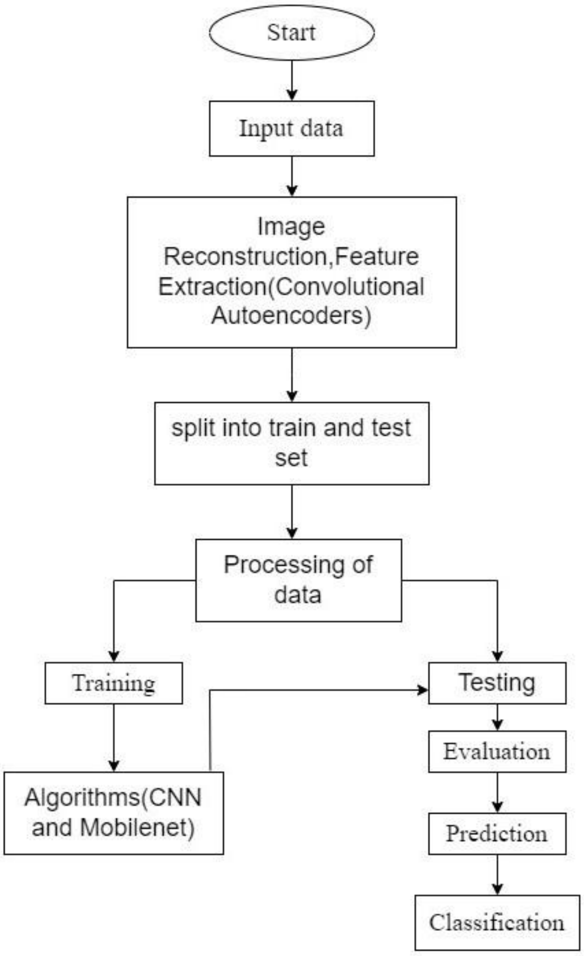

Fig. 5.1: Architecture of Proposed Method

In the proposed system, as shown in Fig. 5.1, the input data undergoes a series of preprocessing steps to prepare it for analysis. Initially, raw data, often in the form of images, is fed into the system. Subsequently, image reconstruction and feature extraction are performed using Convolutional Autoencoders (CAEs). This process enables the extraction of meaningful features from the input data, facilitating the subsequent analysis stages. Following feature extraction, the dataset is split into training and testing sets to ensure the robustness and generalization of the model.

Once the data is prepared, the training phase begins, wherein deep learning algorithms such as Convolutional Neural Networks (CNNs) and MobileNets are employed. These algorithms learn from the extracted features to recognize patterns and make predictions. During training, the model iteratively adjusts its parameters to minimize the difference between predicted and actual outputs. After training, the model undergoes testing to evaluate its performance on unseen data. This evaluation phase involves assessing the model's accuracy, precision, recall, and other relevant metrics. Finally, the trained model is ready for deployment, where it can be used for prediction and classification tasks on new input data.

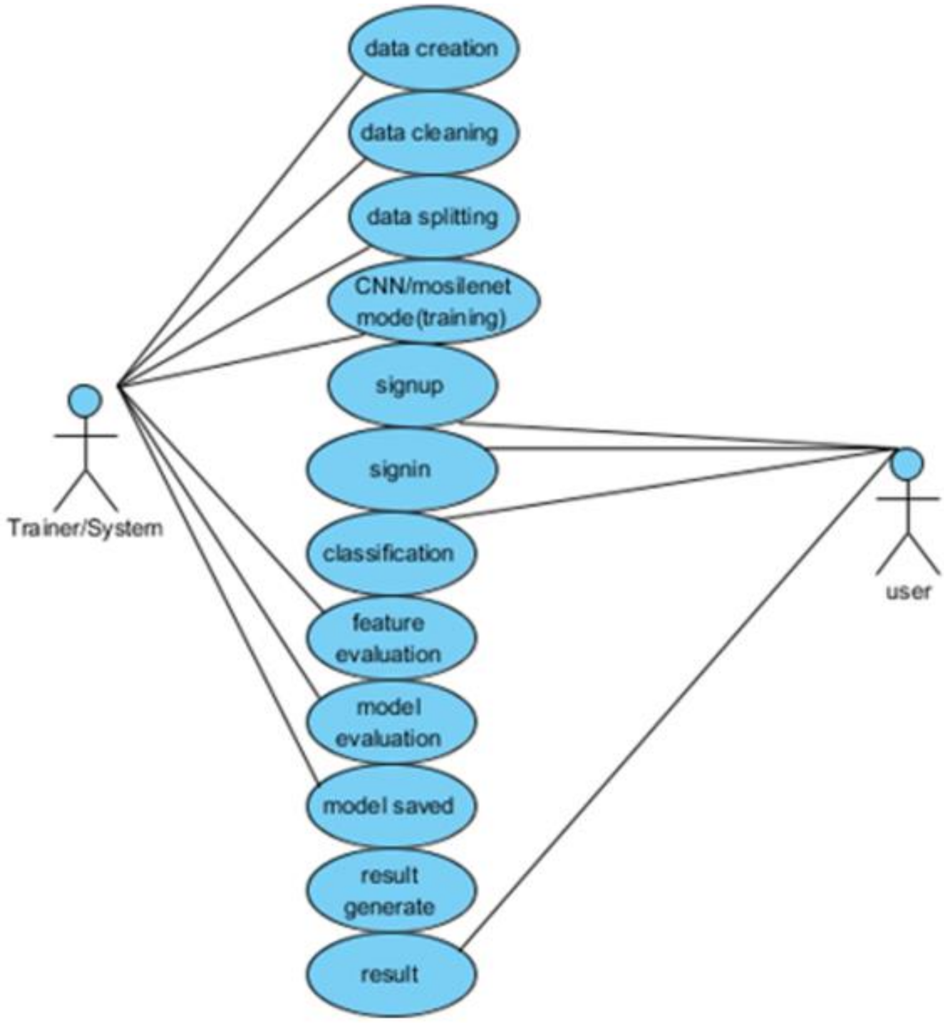

Fig. 5.2: Use case diagram

As shown in Fig. 5.2, illustrates a system's workflow. It encompasses data preprocessing, training utilizing CNN or MobileNet, user authentication, model evaluation, and result storage. Collaboration between trainers and users is central to the process.

5.2 Data Flow Diagrams:

The most generic and generally broadest reaching diagram you can make is the flow diagram. Itis a medium-high level diagram that shows all the pieces of a workflow. The audience for this type of diagram is generally technical. It may be used to pitch an idea to an architecture board or describe how a business process works to a developer.

The Data Flow Diagram is a crucial modeling tool (DFD). It models the system’s components. System processes, data, an external entity interacting with the system, and information flow within the system are all included in this list. In the Fig. 5.3 and 5.4, the DFD displays how information flows through the system and how it is transformed through a sequence of transformations. It’s a visual representation of how data goes from input to output and the transformations that take place along the way.

Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

5.2.1 Rules for creating DFD:

1. The name of the entity should be easy and understandable without any extra assistance.
2. The processes should be numbered or put in ordered list to be referred easily.
3. The DFD should maintain consistency across all the DFD levels.
4. A single DFD can have maximum processes up to 9 and minimum 3 processes.

5.2.2 Levels of DFD:

DFD uses hierarchy to maintain transparency thus multilevel DFD’s can be created. Levels of DFD are as follows:

- 0-level DFD
- 1-level DFD

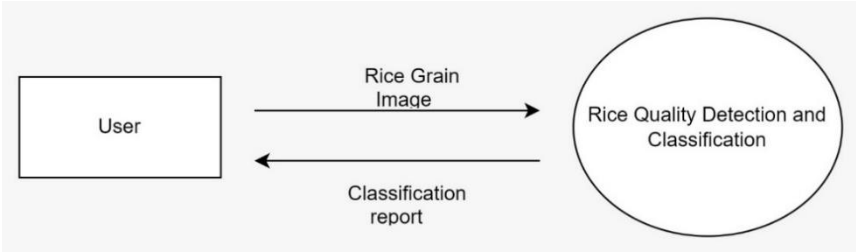

Fig. 5.3: DFD-0

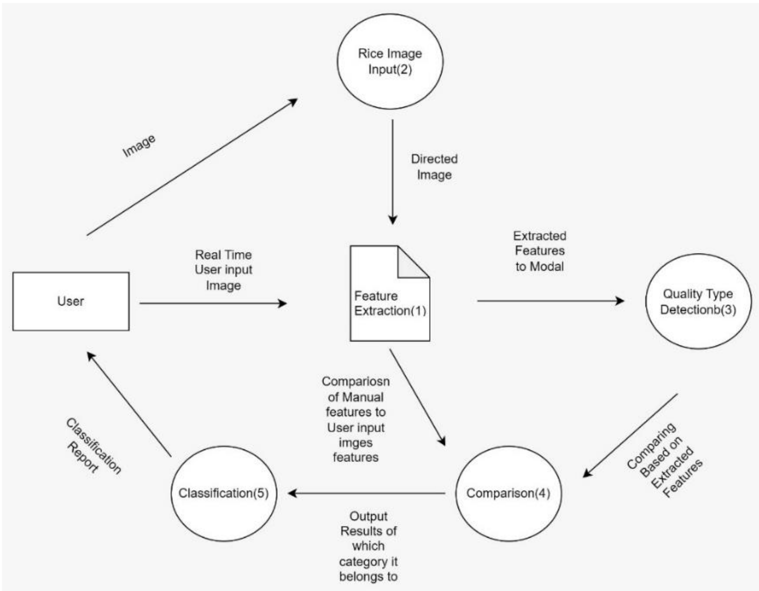

Fig. 5.4: DFD-1

5.3 User Interfaces Diagrams:

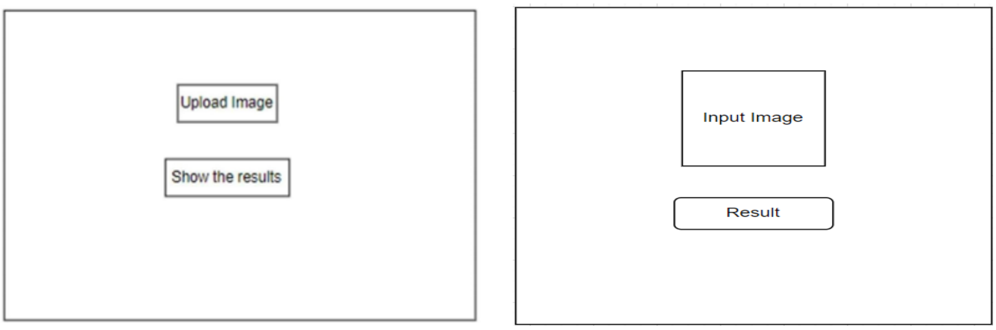

Fig. 5.5: User Interface Diagrams

As shown in Fig. 5.5, When a user uploads an image, the system promptly displays the rice grain quality, categorized as high, medium, or low.

CHAPTER 6
IMPLEMENTATION

6.1 Experimental Setup:

A 64-bit Windows system was used in the experiment. A high performance cpu was used for data preprocessing. Utilizing a GPU can significantly accelerate training times compared to using only a CPU. The CNN and CAE models were built with Keras, TensorFlow, Numpy, Matplotlib and various other frameworks and packages available in Python3 through Google Colaboratory environments.

6.2 Image Down sampling:

Images were acquired from Kaggle (a website for datasets). The datasets contains 5 types of rice(Arborio, Basmati, Ipsala, Jasmine, Karacadag) divided into 5 folders of 15000 images each. These images are given to CAE for image reconstruction. It consists of convolutional layers for encoding the input image into a latent representation and then decoding it back to the original image shape. The resolution of the reconstructed images is 256x256.The images were used for training and testing a Convolutional Neural Network model to assess the quality of rice.

6.3 Technologies and Libraries Used:

6.3.1 Python:

Python is used as a scripting language to write the entire codebase, from importing libraries to defining functions, loops, and conditional statements. Python's built-in data structures like lists, dictionaries, and tuples are extensively used for storing, manipulating, and iterating over data. Python's functions and modules are utilized for organizing code into reusable components. Functions are defined for tasks such as plotting confusion matrices, computing image quality metrics, and training models. Modules are used to organize related functions and classes, promoting code readability and maintainability.

6.3.2 Google Colab:

Google Colab is a cloud-based platform provided by Google that allows users to write and execute Python code in a Jupyter Notebook environment, directly in the browser. It provides free access to GPU and TPU hardware accelerators, which significantly speed up the training of deep learning models. This is especially useful for computationally intensive tasks like training convolutional neural networks (CNNs) and autoencoders. It allows easy access to data stored on Google Drive and provides built-in commands to upload and download data from the local file system. This makes it convenient to work with datasets, images, and other files required for the project. Google Colab comes pre-installed with popular Python libraries and packages, including TensorFlow, Keras, NumPy, Pandas, Matplotlib, and scikit-learn. This eliminates the need for manual installation and configuration, enabling users to start coding immediately. Google Colab seamlessly integrates with other Google services such as Google Drive, allowing users to store and share notebooks, datasets, and trained models. This facilitates collaboration among team members and simplifies project management.

6.3.3 Pandas:

'Pandas' is a popular data manipulation library in Python that provides data structures such as Data Frame and Series, making it easy to manipulate, analyze, and visualize data. It provides powerful tools for data cleaning, aggregation, filtering, and transformation, making it a go-to library for data analysis and data wrangling tasks.

6.3.4 Numpy:

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions. It is extensively used in the code for array manipulation, mathematical computations, and data handling.

6.3.5 CV2:

OpenCV is a library of programming functions mainly aimed at real-time computer vision. It is used in the code for image processing tasks such as loading images and manipulating them.

6.3.6 Keras and TensorFlow:

TensorFlow is an open-source deep learning framework developed by Google. Keras is an open-source neural network library written in Python. It acts as an interface for TensorFlow, making it easier to define and train deep learning model. These are used to define and build deep learning models. In the code, both Convolutional Neural Networks (CNNs) and Convolutional Autoencoders (CAEs) are constructed using TensorFlow's Keras API. The Sequential model and its various layers (Conv2D, MaxPool2D, Dense, GlobalAveragePooling2D, BatchNormalization, Dropout) are utilized to design the architectures of these models. TensorFlow provides optimization algorithms and training routines for training deep learning models. In the code, the compile() method configures the model for training by specifying the optimizer, loss function, and evaluation metrics. Additionally, the fit() method is used to train the models using training data generated by ImageDataGenerator.

6.3.7 Matplotlib:

Matplotlib is a plotting library for the Python programming language. It is used to visualize images, model training/validation metrics, and confusion matrices. Matplotlib is a widely used plotting library in Python that provides a flexible and comprehensive API for creating visualizations. It supports a wide range of plots, including line plots, scatter plots, bar plots, histograms, and more, making it suitable for data visualization and exploration tasks.

6.3.8 PIL (Python Imaging Library):

PIL is a library for opening, manipulating, and saving many different image file formats. In the code, PIL is used for loading and processing images before training the models.

6.3.9 Sklearn(scikit-learn):

Scikit-learn is a machine learning library for Python. It is used for computing metrics like confusion matrices and classification reports for evaluating model performance.

6.4 Work Flow:

6.4.1 Image Loading and Preprocessing:

The code imports necessary libraries such as PIL (Python Imaging Library) for image manipulation then images are loaded using the Image.open() function from PIL. After that image dimensions are retrieved to understand the size and channels of the images. A list of image filenames is generated from the folder path. Images are loaded, and basic information about their dimensions is printed. Images are displayed using Matplotlib.

6.4.2 CAE Implementation:

The libraries such as TensorFlow and Keras are imported for deep learning implementation. A CAE model is defined using the Keras Functional API. The model consists of convolutional layers for encoding and decoding. The model is compiled with an optimizer ('adam') and a loss function ('MSE').Image filenames are iterated through. Images are loaded and resized to a standard size of 128x128 pixels. Images are normalized (scaled to the range [0, 1]). The CAE model is trained on the input image for 50 epochs with a batch size of 1. The trained model is used to reconstruct the input image. Reconstructed images are saved in an output directory. Original and reconstructed images are displayed side by side using Matplotlib.

6.4.3 Evaluation Metrics:

Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are calculated to evaluate the quality of reconstructed images. MSE is computed using NumPy to quantify the difference between the original and reconstructed images. PSNR is computed using the Peak Signal Noise Ratio function from the skimage.metrics module.

6.4.4 Visualization:

MSE and PSNR values are displayed alongside the original and reconstructed images for visual comparison. This implementation theory outlines the steps involved in loading, preprocessing, training, and evaluating the CAE model for image reconstruction. It also includes visualizations to assess the quality of the reconstructed images.

6.4.5 Image Quality Evaluation:

Structural Similarity Index (SSIM) and Mean Absolute Error (MAE) are computed to further evaluate the quality of reconstructed images. SSIM is calculated using the structural similarity function from the skimage.metrics module. MAE is computed using NumPy to measure the average absolute difference between the original and reconstructed images. SSIM and MAE values are displayed alongside the original and reconstructed images for visual comparison.

6.4.6 CNN Model Training and Evaluation:

Utilizing TensorFlow and Keras, a Convolutional Neural Network (CNN) model is constructed for rice quality classification. The CNN model architecture consists of convolutional layers followed by max-pooling layers and fully connected layers. The model is compiled with the Adam optimizer and categorical cross-entropy loss function. Training and validation accuracy, loss, precision, and recall metrics are monitored and plotted over epochs.

6.4.7 Transfer Learning:

A pre-trained MobileNet model is employed for transfer learning to enhance the classification model. The MobileNet model is fine-tuned by adding additional layers and retraining on the rice quality dataset. Similar training and evaluation steps are performed, and model performance is analyzed through metrics.

6.5 Code Snippets:

```python
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
import os

# Define the path to the folder containing your images
folder_path = '/content/Rice_Image_Dataset/Basmati'

# Get a list of all image filenames in the folder
image_filenames = [filename for filename in os.listdir(folder_path) if filename.endswith('.jpg')]

# Define a directory to save the reconstructed images
output_dir = '/content/reconstructed_images1/'

# Create the output directory if it doesn't exist
os.makedirs(output_dir, exist_ok=True)  # This line creates the directory

# Define the CAE model
input_img = keras.Input(shape=(128, 128, 3))
x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(input_img)
x = layers.MaxPooling2D((2, 2), padding='same')(x)
x = layers.Conv2D(64, (3, 3), activation='relu', padding='same')(x)
x = layers.MaxPooling2D((2, 2), padding='same')(x)
encoded = layers.Conv2D(128, (3, 3), activation='relu', padding='same')(x)

x = layers.Conv2D(128, (3, 3), activation='relu', padding='same')(encoded)
x = layers.UpSampling2D((2, 2))(x)
x = layers.Conv2D(64, (3, 3), activation='relu', padding='same')(x)
x = layers.UpSampling2D((2, 2))(x)
decoded = layers.Conv2D(3, (3, 3), activation='sigmoid', padding='same')(x)
autoencoder = keras.Model(input_img, decoded)
autoencoder.compile(optimizer='adam', loss='mse')

# Loop through each image filename
for image_filename in image_filenames:
    # Combine the folder path and filename to get the complete image path
    image_path = os.path.join(folder_path, image_filename)
```

```python
    # Load and preprocess the image
    img = image.load_img(image_path, target_size=(128, 128))
    img = image.img_to_array(img)
    img = img / 255.0  # Normalize the image data

    # Train the CAE on the input image
    autoencoder.fit(np.expand_dims(img, axis=0), np.expand_dims(img, axis=0), epochs=50, batch_size=1)

    # Reconstruct the image using the trained CAE
    reconstructed_img = autoencoder.predict(np.expand_dims(img, axis=0))[0]

    # Save the reconstructed image
    reconstructed_filename = os.path.join(output_dir, f'reconstructed_{image_filename}')
    plt.imsave(reconstructed_filename, reconstructed_img)
```

Fig. 6.1: CAE Model

As shown in Fig. 6.1, the code lines represents the CAE model which contains encoder and decoder with their layers respectively with parameters .Here it is used to reconstruct the images for further feature extraction.

```python
def rcnn(image):
    region_proposals = selective_search(image)
    cnn = load_pretrained_cnn()  # This could be a model like VGG16, AlexNet, etc.
    features = [cnn(extract_region(image, proposal)) for proposal in region_proposals]
    svm_classifiers = load_svm_classifiers()  # Load pre-trained SVM classifiers for each class.
    bbox_regressors = load_bbox_regressors()  # Load pre-trained bounding box regressors.
    detections = []
    for feature in features:
        class_label = classify(svm_classifiers, feature)
        refined_bbox = refine_bbox(bbox_regressors, feature)
        detections.append((class_label, refined_bbox))
    final_detections = non_maximum_suppression(detections)

    return final_detections
```

Fig. 6.2: Defining CNN Model

As shown in Fig. 6.2, the code lines represents pre train model which comes before the main model . Using pre trained model helps main model to grasp the features effectively and trained well.

```python
model = Sequential()

model.add(Conv2D(filters = 32, kernel_size = (5,5),padding = 'Same',
                 activation ='relu', input_shape = (img_height,img_width,3)))
model.add(Conv2D(filters = 32, kernel_size = (5,5),padding = 'Same',
                 activation ='relu'))
model.add(MaxPool2D(pool_size=(2,2)))


model.add(Conv2D(filters = 64, kernel_size = (3,3),padding = 'Same',
                 activation ='relu'))
model.add(Conv2D(filters = 64, kernel_size = (3,3),padding = 'Same',
                 activation ='relu'))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))

model.add(Flatten())
model.add(Dense(256, activation = "relu"))

model.add(Dense(3, activation = "softmax"))

model.summary()
```

Fig. 6.3: CNN Sequential Main Model

As shown in Fig. 6.3, the code lines represents the Main CNN model (Sequential). Layers are included with their respective parameters and model is created.

```python
model.compile(optimizer="adam",loss="categorical_crossentropy",metrics=['accuracy',precision,recall])
hist=model.fit(train_generator,epochs=10,validation_data=test_generator)
```

Fig. 6.4: CNN Compilation Model

As shown in Fig. 6.4, the code lines represents the CAE model compilation.

```
fig, ax = plt.subplots(2,1)
ax[0].plot(hist.history['accuracy'], color='b', label="Training accuracy")
ax[0].plot(hist.history['val_accuracy'], color='r',label="Validation accuracy")
legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(hist.history['loss'], color='b', label="Training loss")
ax[1].plot(hist.history['val_loss'], color='r', label="validation loss")
legend = ax[1].legend(loc='best', shadow=True)
```

Fig. 6.5: CNN graph between Training and Validation accuracy

```
fig, ax = plt.subplots(2,1)
ax[0].plot(hist.history['precision'], color='b', label="Training precision")
ax[0].plot(hist.history['val_precision'], color='r',label="Validation precision")
legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(hist.history['loss'], color='b', label="Training loss")
ax[1].plot(hist.history['val_loss'], color='r', label="validation loss")
legend = ax[1].legend(loc='best', shadow=True)
```

Fig. 6.6: CNN graph between Training and Validation Precision

```
fig, ax = plt.subplots(2,1)
ax[0].plot(hist.history['recall'], color='b', label="Training recall")
ax[0].plot(hist.history['val_recall'], color='r',label="Validation recall")
legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(hist.history['loss'], color='b', label="Training loss")
ax[1].plot(hist.history['val_loss'], color='r', label="validation loss")
legend = ax[1].legend(loc='best', shadow=True)
```

Fig. 6.7: CNN graph between Training and Validation recall

As shown in Fig. 6.5, 6.6, and 6.7, the code lines represents the Classification metrics i.e., (Accuracy , Precision and Recall). These are graph plottings between Training images and validation images. This is for CNN Model Compilation model.

```
base_model = tf.keras.applications.MobileNet(input_shape=(img_height,img_width, 3), include_top=False,
                            weights='imagenet')
model8 = Sequential()
model8.add(base_model)
model8.add(GlobalAveragePooling2D())
model8.add(Dense(1024, activation='relu'))
model8.add(Dense(512, activation='relu'))
model8.add(Dense(256, activation='relu'))
model8.add(Dense(128, activation='relu'))
model8.add(Dense(64, activation='relu'))
model8.add(BatchNormalization())
model8.add(Dropout(0.2))
model8.add(Dense(3, activation='sigmoid'))
model8.summary()
```

Fig. 6.8: MobileNet Sequential Main Model

As shown in Fig. 6.8, the code lines represents the MobileNet model which contains their layers respectively with parameters . It is another main model which is especially used for mobile embedded applications and for classification tasks.

```
model8.compile(optimizer="adam",loss="categorical_crossentropy",metrics=['accuracy',precision,recall])
hist8=model8.fit(train_generator,epochs=50,validation_data=test_generator)
```

Fig. 6.9: MobileNet Compilation Model

```
fig, ax = plt.subplots(2,1)
ax[0].plot(hist8.history['accuracy'], color='b', label="Training accuracy")
ax[0].plot(hist8.history['val_accuracy'], color='r',label="Validation accuracy")
legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(hist8.history['loss'], color='b', label="Training loss")
ax[1].plot(hist8.history['val_loss'], color='r', label="validation loss",axes =ax[1])
legend = ax[1].legend(loc='best', shadow=True)
```

Fig. 6.10: MobileNet graph between Training and Validation accuracy

```
fig, ax = plt.subplots(2,1)
ax[0].plot(hist8.history['precision'], color='b', label="Training precision")
ax[0].plot(hist8.history['val_precision'], color='r',label="Validation precision")
legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(hist8.history['loss'], color='b', label="Training loss")
ax[1].plot(hist8.history['val_loss'], color='r', label="validation loss",axes =ax[1])
legend = ax[1].legend(loc='best', shadow=True)
```

Fig. 6.11: MobileNet graph between Training and Validation Precision

```
fig, ax = plt.subplots(2,1)
ax[0].plot(hist8.history['recall'], color='b', label="Training recall")
ax[0].plot(hist8.history['val_recall'], color='r',label="Validation recall")
legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(hist8.history['loss'], color='b', label="Training loss")
ax[1].plot(hist8.history['val_loss'], color='r', label="validation loss",axes =ax[1])
legend = ax[1].legend(loc='best', shadow=True)
```

Fig. 6.12: MobileNet graph between Training and Validation recall

As shown in Fig. 6.10, 6.11, and 6.12, the code lines represents the Classification metrics i.e., (Accuracy , Precision and Recall). These are graph plottings between Training images and validation images for this MobileNet Model.

CHAPTER 7

RESULTS

7.1 Comparison between Original and Reconstructed Images:

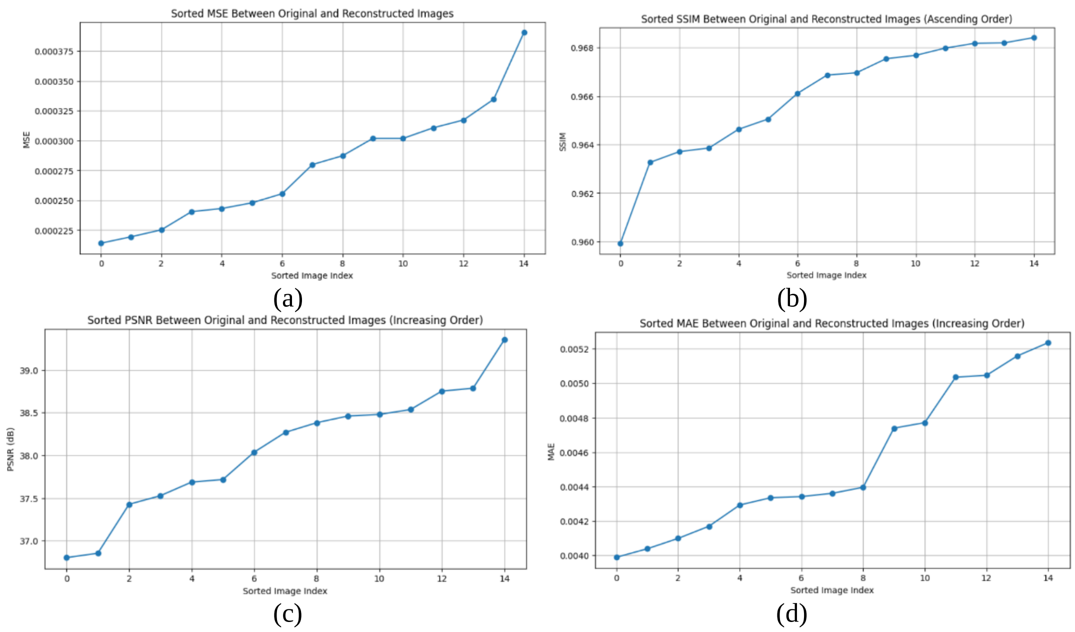

Fig. 7.1: The graphs (a), (b), (c), and (d) represent the MSE, SSIM, PSNR, and MAE respectively, indicating the differences between the original and reconstructed images.

7.1.1 MSE:

As shown in Fig. 7.1 (a), a popular metric for evaluating the average squared difference between comparable elements of two sets, such as projected and actual values, is mean squared error, or MSE. It is computed by averaging the squared differences between predicted and true values. MSE offers a measure of the average magnitude of errors, where higher values indicate greater overall error in prediction or reconstruction tasks.

7.1.2 SSIM:

As shown in Fig. 7.1 (b), a metric for comparing two images similarity is called the Structural Similarity Index (SSIM).It takes into account luminance, contrast, and structure, providing a more comprehensive measure of perceptual image quality than some traditional metrics. A higher SSIM value indicates greater similarity between the original and distorted images, with 1 representing perfect similarity.

7.1.3 PSNR:

As shown in Fig. 7.1 (c), a typical metric used in image and video compression is the Peak Signal-to-Noise Ratio (PSNR), which measures the quality of reconstructed signals. It will be calculated in Decibels. It calculates the ratio of the mean squared error of the original and reconstructed signals to the highest feasible signal value. Greater faithfulness and less distortion in the reconstruction are indicated by a higher PSNR.

7.1.4 MAE:

As shown in Fig. 7.1 (d), one statistic used to quantify the average size of deviations between expected and actual values is mean absolute error (MAE). By averaging the absolute disparities between the true and anticipated values, it is calculated. MAE offers a straightforward assessment of the average error magnitude, rendering it especially valuable in regression and prediction endeavours.

These graphs are used to calculate the loss between original and reconstructed images after performing the reconstruction using Convolutional Autoencoders.

7.2 Graphs:

7.2.1 CNN Graphs:

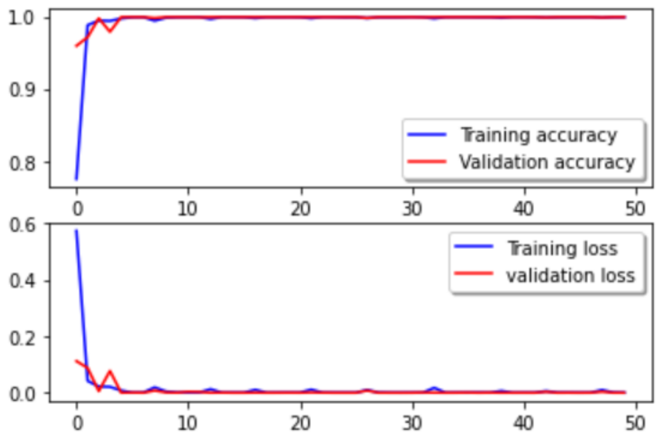

Fig. 7.2: Comparison of CNN accuracy

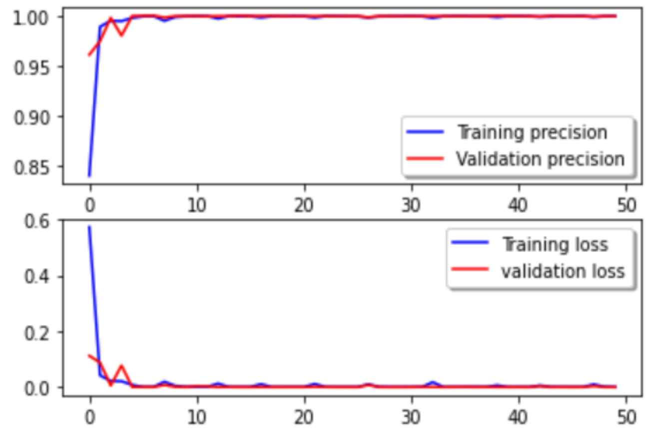

Fig. 7.3: Comparison of CNN precision

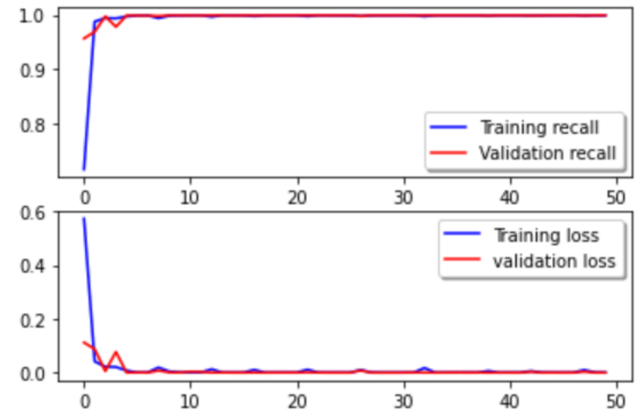

Fig. 7.4: Comparison of CNN recall

7.2.2 MobileNet Graphs:

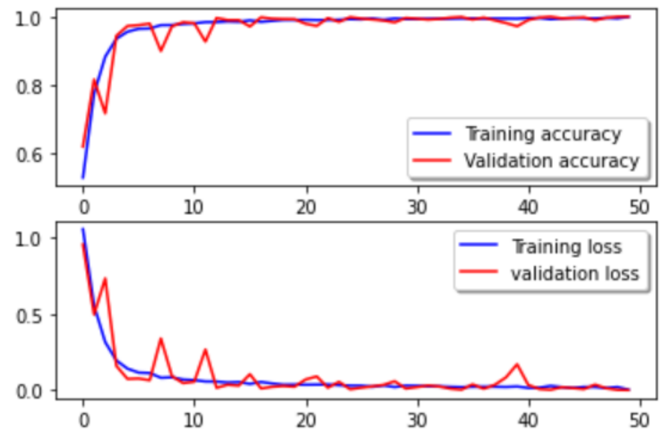

Fig. 7.5: Comparison of MobileNet accuracy

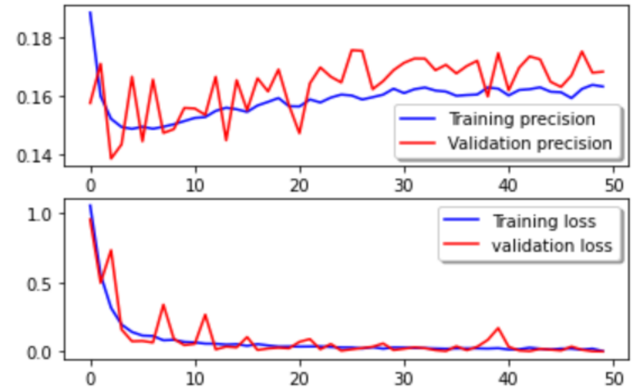

Fig. 7.6: Comparison of MobileNet precision

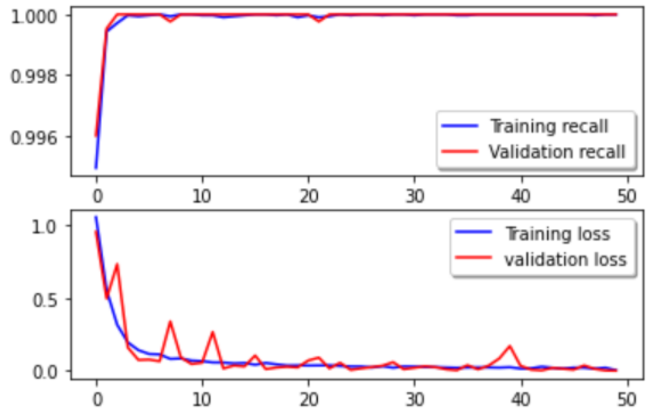

Fig. 7.7: Comparison of MobileNet recall

These graphs shows that how well the model is performing with CNN and MobileNet. It shows how well the model is trained on the given images as input and how well the model is performing on the given images.

In a neural network model, the optimizer is a mathematical algorithm that adjusts the weights and biases during training to minimize the chosen loss function, guiding the model towards better performance. The specified metrics during training provide performance measures for evaluating the model's classification abilities: Accuracy, Precision, Recall.

Accuracy: It is a vital metric in classification models, measures the ratio of correctly classified instances to total predictions during both training and testing. Calculated by dividing correct predictions by total predictions, higher accuracy values. As shown in Fig. 7.2 and 7.5, for training and testing data, denote superior model performance. These figures offer insights into the model's effectiveness in classification tasks.

Precision: It evaluates the model's performance on testing and training data. It represents the ratio of true positives to the sum of true positives and false positives, indicating the model's reliability in avoiding false positives. As shown in Fig. 7.3 and 7.6, depict precision values, highlighting the model's ability to accurately classify positives in both datasets and offering valuable insights into its overall performance.

Recall: This metric is crucial in evaluating its performance on both training and testing data. High recall values indicate the model's effectiveness in identifying actual positive instances and minimizing false negatives. As shown in Fig. 7.4 and 7.7, depicting recall values across datasets, offer valuable insights into the model's ability to accurately identify relevant instances, enhancing its overall performance.

7.3 Comparison of Feature result:

Here the graphs shows the difference between the features extracted from original and real time images and based on the features, the images are categorized into three classes High, Low, Medium.

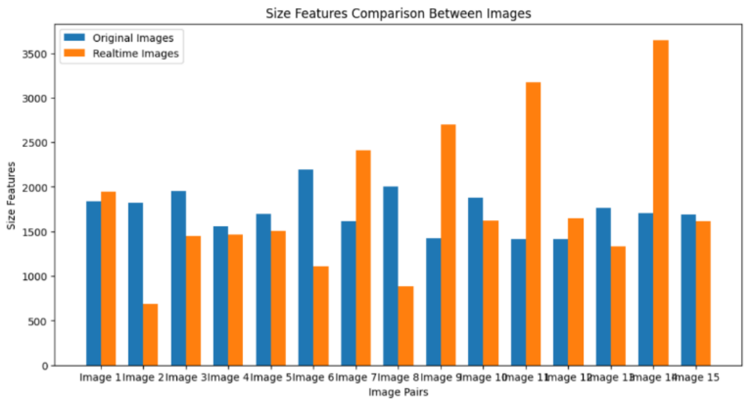

Fig. 7.8: Size Feature comparison between original and real time images

As shown in Fig. 7.8, this graph shows the size feature comparison. As it can be seen that the difference between orange bar and blue bar is low in most of the cases , hence we can say the model performance is good and giving accurate results. Here Blue Bar indicates the size features of original images and Orange bar indicates size features of Real time images .

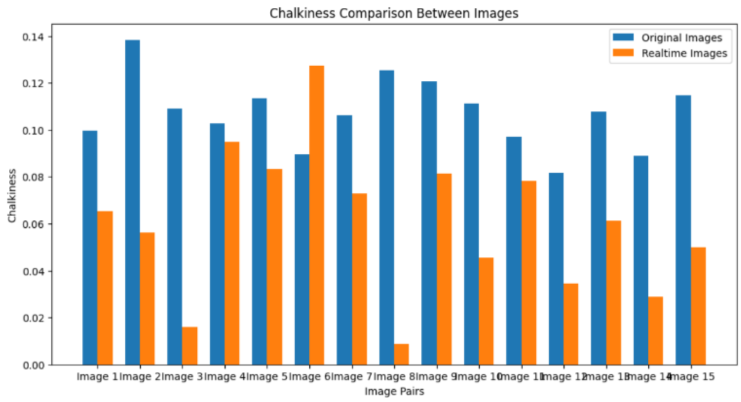

Fig. 7.9: Chalkiness Feature comparison between original and real time images

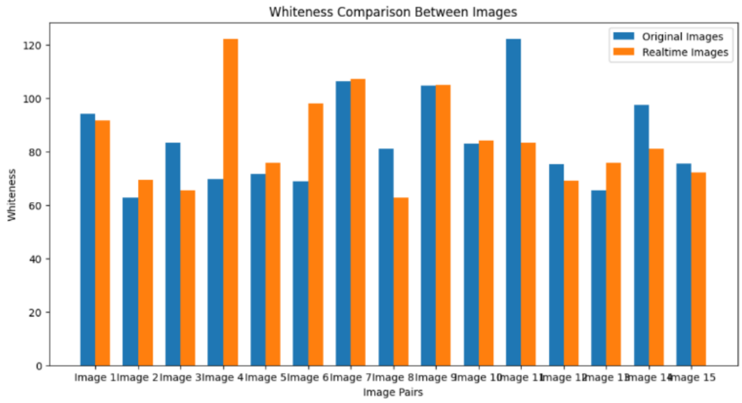

Fig. 7.10: Whiteness Feature comparison between original and real time images

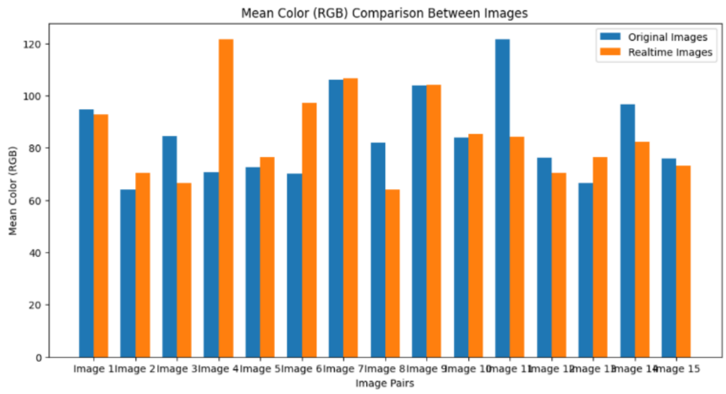

Fig. 7.11: Mean color Feature comparison between original and real time images

The graphs as shown in Fig. 7.8, 7.9, 7.10, and 7.11 illustrate the comparison of features. These graphs show that even when real-time images are provided to the model by the user, it can still produce and categorize the images based on how closely the features of the original and real-time images differ.

Fig. 7.12: Home page

The main page layout of the interface for our project is displayed as shown in Fig. 7.12, as seen in the images below, three primary modules have been built, each with corresponding buttons.

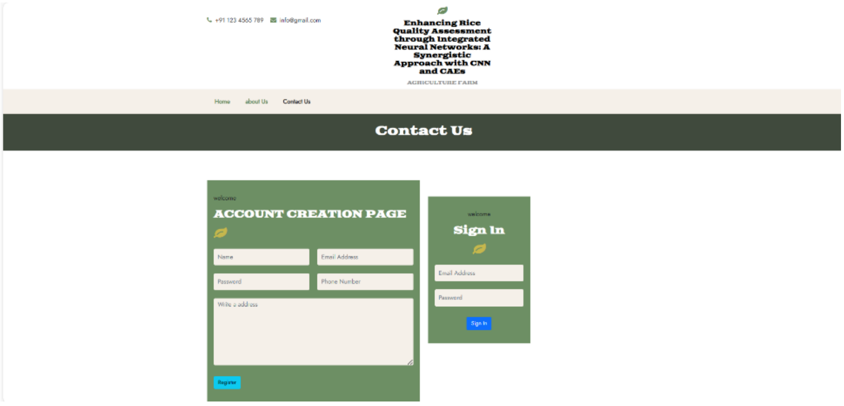

Fig. 7.13: Contact us Page

The contact page layout of our project's interface is displayed as shown in Fig. 7.13, as can be seen, this is the page where users register and establish an account using their email address and password. On the other side, this is the login page, where users who already have an account can log in using their credentials.

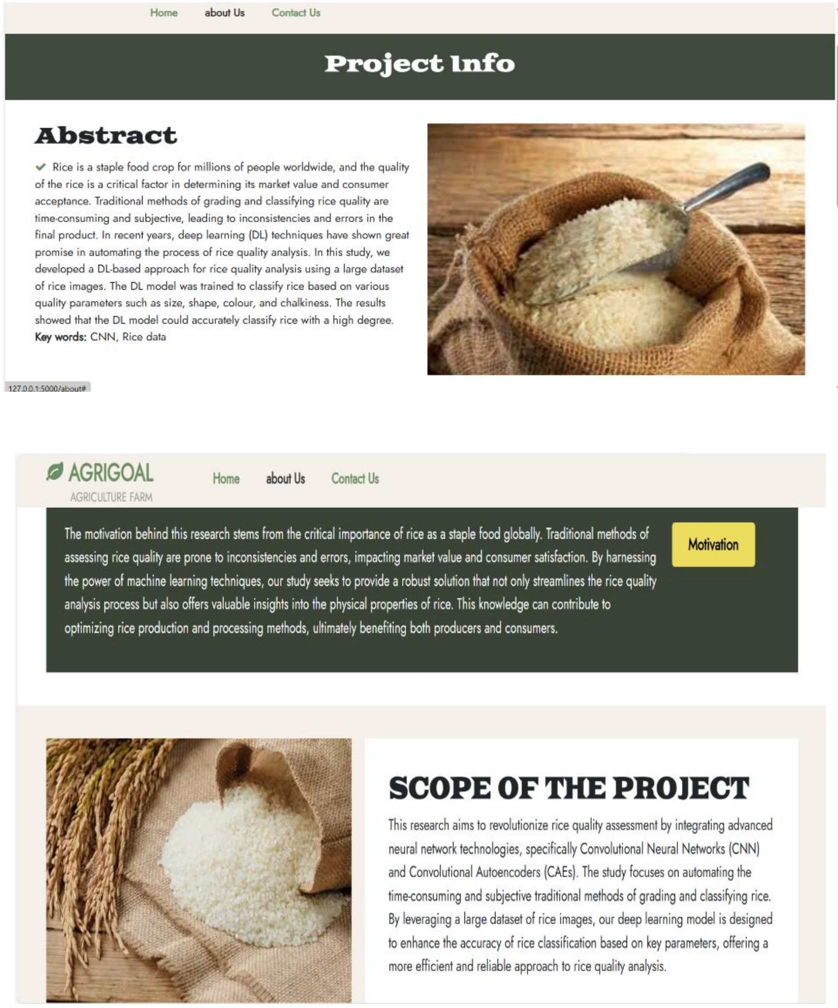

Fig. 7.14: About us Page

As shown in Fig. 7.14, the user is provided with an overview of our application and page. This includes a description and an illustration of the project's scope.

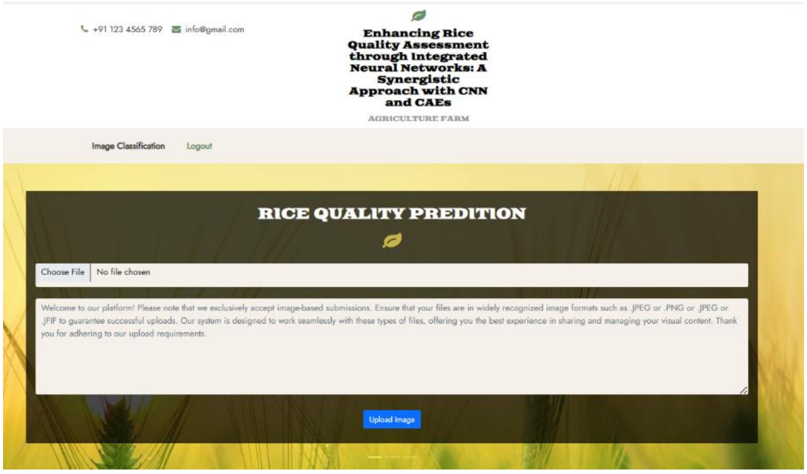

Fig. 7.15: Upload rice image

As shown in Fig. 7.15, user can upload their respective rice image and click the upload image button.

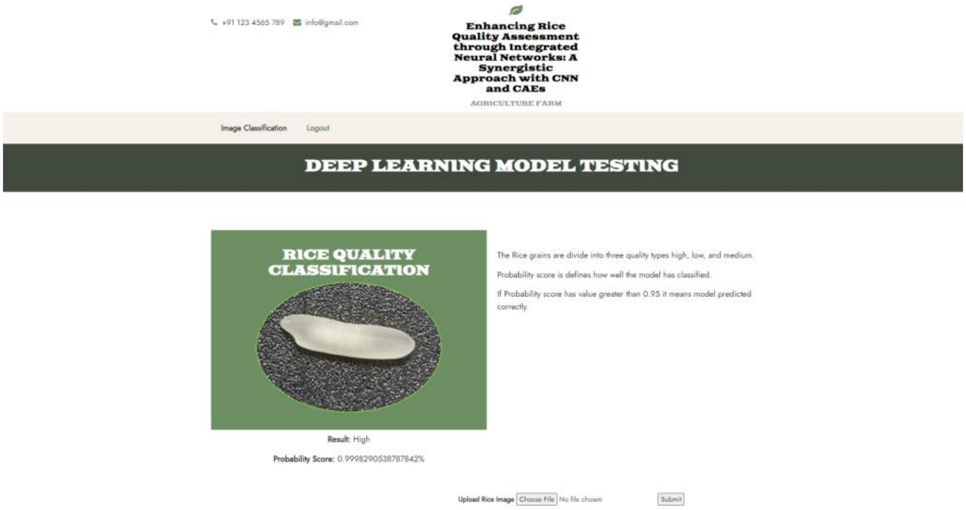

Fig. 7.16: Classification result

As shown in Fig. 7.16, where the rice image is classified into one of three classes. High, Low, or Medium. In order to show how accurately the image has been anticipated and identified, a probability score is also provided.

CONCLUSION AND FUTURE WORK

In culmination, our project's fusion of Sequential Convolutional Neural Networks (CNN) and Convolutional Autoencoders (CAEs) signifies a breakthrough in precision for rice grain quality assessment. By harmonizing CNN's adept object identification with CAEs' feature learning, the resulting framework exceeds the limitations of conventional methods. The model's fine-tuning, coupled with data augmentation, ensures robust adaptability and generalization, validated by impressive evaluation metrics.

As we deploy this cutting-edge technology into practical applications, it not only addresses existing challenges but also pioneers a new era of automated and elevated rice quality standards, crucial for global food security. This success is a testament to the collaborative efforts of neural network researchers, showcasing the impactful role of machine learning in revolutionizing the production and quality control of vital food crops like rice, ensuring both agricultural sustainability and consumer well-being.

Future enhancements to our integrated neural network approach for rice quality assessment could involve the incorporation of advanced sensor technologies and real-time data collection during the rice cultivation and processing stages. Implementing a more dynamic system that adapts to changing environmental conditions and crop variations could further improve the model's accuracy and robustness. Additionally, exploring the integration of explainable AI techniques can enhance the interpretability of the model's decisions, providing valuable insights for stakeholders. Continuous updates to the dataset and model architecture will ensure adaptability to emerging rice varieties and quality parameters.

REFERENCES

[1]. Thae Nu Wah, Pann Ei San, Thandar Hlaing (2018); "Analysis on Feature Extraction and Classification of Rice Kernels for Myanmar Rice Using Image Processing Techniques"; International Journal of Scientific and Research Publications (IJSRP) 8(8) (ISSN: 2250-3153).

[2]. S. Wentao, "Classification Model of Wheat Grain based on Autoencoder" 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2020, pp. 830-832.

[3]. N. Hong Son and N. Thai-Nghe, "Deep Learning for Rice Quality Classification", 2019 International Conference on Advanced Computing and Applications (ACOMP), Nha Trang, Vietnam, 2019, pp. 92-96.

[4]. Gao, Y., Wang, W., Zhang, W., & Gao, X. (2020). "An automated inspection system for rice seed quality based on deep learning", Journal of Food Process Engineering, 43(6), e13396.

[5]. Latha, M. N., & Nandhini, R. (2020). "Rice quality analysis using deep learning techniques: A review". Computers and Electronics in Agriculture, 174, 105404.

[6]. T. G. Devi, P. Neelamegam and S. Sudha, "Machine vision based quality analysis of rice grains", 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, India, 2017, pp. 1052-1055.

[7]. Annadasu, P. and Jaisharma, K., 2020. "Rice Quality Analysis Using Machine Learning", (No. 4761). EasyChair.

[8]. Md Taimur Ahad, Yan Li, Bo Song, Touhid Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification",Artificial Intelligence in Agriculture,Volume 9,2023, ISSN 2589-7217.

[9]. Anam Islam, Redoun Islam, S. M. Rafizul Haque, S.M. Mohidul Islam, Mohammad Ashik Iqbal Khan, "Rice Leaf Disease Recognition using Local Threshold Based Segmentation and Deep CNN", International Journal of Intelligent Systems and Applications(IJISA), Vol.13, No.5, pp.35-45, 2021.

[10]. Patel, Vaibhav. (2017). "Convolutional neural network with transfer learning for rice type classification".

[11]. Jana, Debasish & Patil, Jayant & Herkal, Sudheendra & Nagarajaiah, Satish & Dueñas-Osorio, Leonardo. (2022). "CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction". Mechanical Systems and Signal Processing. 169. 108723.

[12]. Pu, Hongbin, et al. "Terahertz spectra reconstructed using convolutional denoising autoencoder for identification of rice grains infested with Sitophilus oryzae at different growth stages". Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy (2024): 124015.

[13]. Bedi, Punam, and Pushkar Gole. "Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network". Artificial Intelligence in Agriculture 5 (2021): 90-101.

[14]. Qiu, Zhengjun, et al. "Variety identification of single rice seed using hyperspectral imaging combined with convolutional neural network". Applied Sciences 8.2 (2018): 212.

[15]. Wu, Yiqiang, et al. "Deep-rice: Deep multi-sensor image recognition for grading rice". 2018 IEEE International Conference on Information and Automation (ICIA). IEEE, 2018.

[16]. Rathore, Narendra Pal Singh, and Lalji Prasad. "Automatic rice plant disease recognition and identification using convolutional neural network". Journal of critical reviews 7.15 (2020): 6076-6086.

[17]. Tran-Thi-Kim, Nga, et al. "Enhancing the Classification Accuracy of Rice Varieties by Using Convolutional Neural Networks". International Journal of Electrical and Electronic Engineering & Telecommunications 12.2 (2023): 150-160.

Enhancing Rice Quality Assessment through Integrated Neural Networks: A Synergistic Approach with CNN and CAEs

Veera Prakash Pobbatti[1, a)] Megharshini Anke, Anees Fathima Shaik, Girivardhan Velupula, Anitha Myla[2,3,4,5 b)]

[1]*Assistant Professor, Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology, Anantapur, India.*
[2,3,4,5] *Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology*

[a)]*prakash.cse@srit.ac.in*,
[b)]*204g1a0557@srit.ac.in*
[c)]*204g1a0512@srit.ac.in*
[d)]*204g1a0532@srit.ac.in*
[e)]*214g5a0503@srit.ac.in*

Abstract. In the realm of food crops, rice stands as a pivotal staple, and ensuring high-quality rice is vital for consistent standards in the industry. Traditional methods for assessing rice quality lack precision, emphasizing the necessity for advanced techniques. The application of neural networks like MobileNet, specifically CNN (Convolutional Neural Networks), offers a promising solution to enhance accuracy and automation in classifying rice grain quality. Additionally, incorporating the feature learning capabilities of Convolutional Autoencoders (CAEs) further refines the identification of nuanced variations indicative of rice quality. Our goal is to establish a robust framework that not only surpasses the limitations of individual techniques but also delivers optimized and unparalleled outcomes in the task of determining rice quality. This integrated approach aims to revolutionize the evaluation process, ensuring a more accurate and efficient assessment of rice quality for the benefit of human health.

Keywords: Rice quality, Neural networks, CNN (Convolutional Neural Networks), MobileNet, Convolutional Autoencoders (CAEs), Accuracy, Automation, Classifying, Robust framework, Optimized outcomes.

INTRODUCTION

In the landscape of food production, the significance of rice as a primary staple necessitates a keen focus on maintaining and enhancing its quality standards. The traditional methods employed for evaluating rice quality often fall short in terms of precision and efficiency, prompting the exploration of more advanced and nuanced approaches. In response to this, our project introduces a novel methodology that harnesses the capabilities of Convolutional Autoencoders (CAEs) and Convolutional Neural Networks (CNN) to detect and classify rice quality.

Pivotal role of CNN lies in its adeptness at object identification, specifically tailored to discern and classify various characteristics associated with rice grains. Simultaneously, the integration of Convolutional Autoencoders contributes to the project's depth by facilitating the extraction of intricate features and patterns relevant to rice quality. This combination of CNN and CAEs creates a robust and comprehensive framework for the precise identification of subtle variations in rice quality.

Our project seeks not only to address the shortcomings of conventional methods but also to set a new standard for accuracy and efficiency in rice quality assessment. By capitalizing on the synergies between CAEs and CNN, MobileNet model we aim to surpass the limitations of individual techniques, offering a sophisticated and automated solution for the rice industry. This endeavor aligns with our broader goal of ensuring the consistency of high-quality rice products, thereby contributing to improved standards in the food sector and, ultimately, to the well-being of consumers.

LITERATURE SURVEY

Researchers dedicated their efforts to assessing the quality of rice grains.

Thae Nu Wah, Pann Ei San, Thandar Hlaing [1]. This research proposes an image processing system for rice grading and tests its effectiveness using Paw-San rice. KNN classifier is used for classification. This work uses the global thresholding method for image segmentation.

S. Wentao [2]. In this study, Deep learning has been used in speech recognition, image search, image recognition. In the wheat particle database, this model is used by deep learning (DL) models like autoencoders to categorize and identify photos of damaged wheat.

Wang, R., Jiang, Y., & Cao, F [3]. The paper introduces a deep learning approach aimed at identifying and categorizing rice seeds according to their visual characteristics. The study aims to mitigate the limitations associated with manual, subjective, and error-prone traditional techniques of rice seed identification. Automatic rice seed classification based on variety and quality is the goal of the suggested deep learning technique. A dataset of 8,000 photographs of rice seeds from various types was gathered by the authors, who then manually labeled each image with the variety and quality level that it matched.

N. Hong Son and N. Thai-Nghe [4]. In this work, image processing methods and machine learning approaches were used to identify two types of rice: whole rice and broken rice. CNN achieved 94.16% accuracy and KNN achieved 85.06% accuracy.

Gao, Y., Wang, W., Zhang, W., & Gao, X [5]. This presents an automated inspection system for rice seed quality based on deep learning techniques. The study seeks to overcome the constraints associated with conventional rice seed quality inspection methods, which are characterized by being time-consuming, labor-intensive, and subjective in nature. The proposed system employs deep learning algorithms to autonomously classify rice seeds according to their visual features. The authors curated a dataset comprising 18,000 rice seed images sourced from various varieties and meticulously annotated them with their respective quality levels.

Latha, M. N., & Nandhini, R [6]. In this section, the significance of rice quality analysis within the food industry is addressed, along with the drawbacks of conventional approaches. This is followed by a summary of deep learning methods, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs), and how each is used in applications related to rice quality assessments.

T. G. Devi, P. Neelamegam and S. Sudha [7]. In this study, the algorithm utilized straightforward morphological image processing methods, enabling it to conduct various image-based analyses. The majority of operations employed in this algorithm are founded on MATLAB commands. This algorithm is well-suited for efficiently grading many grains, and its results were compared with manual grading methods. The comparison showed that the algorithm achieved a percentage of accuracy equal to or greater than that of manual grading.

Annadasu, P. and Jaisharma, K [8]. In this work, it is applied to create a useful model that facilitates rice grain grading without requiring more effort. Segmentation is achieved through various techniques such as threshold-based segmentation and segmentation by clustering. Based on the results obtained, it can be concluded that this system is efficient and cost-effective.

Md Taimur Ahad, Yan Li, Bo Song, Touhid Bhuiyan [9]. This study concentrates on comparing performance of different convolutional neural network (CNN) architectures for detecting and pinpointing rice diseases. With the help of a database that included nine of the most common rice diseases in Bangladesh, the researchers compared 6 CNN-based deep learning architectures for studying rice diseases. They evaluated the performance of the six distinct CNN networks, transfer learning, and ensemble approaches using DenseNet121, MobileNetV2, ResNet152V, Seresnext101, and an ensemble model called DEX. When comparing the accuracy of rice leaf disease detection and localization with Seresnext101's results, transfer learning produced a significant increase, increasing accuracy by 17%.

Islam, Redoun Islam, S. M. Rafizul Haque, S.M. Mohidul Islam, Mohammad Ashik Iqbal Khan [10]. The approach employs local threshold-based segmentation combined with Convolutional Neural Network (CNN) to segment regions of rice leaves affected by disease. 3 different datasets have been used to test the suggested method; one of the datasets was developed by the authors and includes photographs of rice leaves collected from the Bangladesh Rice Research Institute (BRRI). These datasets were used to train three popular CNN architectures ResNet, VGG, and DenseNet to classify the disorders. The proposed methods classification performance has been evaluated and compared across three datasets, utilizing three CNN architectures for contrast.

Patel, Vaibhav [11]. In order to decrease manual effort and mistake, this research suggests a deep learning-based method for detecting different types of rice. Two approaches are suggested: the first makes use of a deep convolutional neural network (CNN) trained on segmented rice images, while the second combines the suggested approach with a pre-trained VGG16 network in order to improve accuracy. Additionally, the method can categorize rice grains as fine or broken. Despite diverse rice photos, the architecture considerably increases classification accuracy, having been pretrained on ImageNet data.

Jana, Debasish & Patil, Jayant & Herkal, Sudheendra & Nagarajaiah, Satish & Dueñas-Osorio, Leonardo [12]. To detect, classify, and reconstruct faults in sensor data, in order to address linear systems with time-invariant parameters, the study presents a novel deep learning algorithm. Within this framework, a Convolutional Neural Network (CNN) is employed to discern fault type and existence. For reconstruction purposes, a set of Convolutional Autoencoder (CAE) networks, separately trained for each fault type, is utilized. The models exhibited robust performance across both simulated and experimental datasets featuring a single failure. They achieved perfect accuracy in localizing faulty sensors, with fault type classification accuracy exceeding 98.7%, and reconstruction accuracy surpassing 98%.

Hongbin, et al [13]. Convolutional Denoising Autoencoders (CDAEs), when integrated with Convolutional Neural Network (CNN) models for automatic feature extraction and parameter optimization, have the potential to outperform traditional approaches. To eliminate surface contaminants and non-rice grains, the rice grains underwent a preliminary soaking in water for thirty minutes. Furthermore, the CNN model, a successful deep learning technique for feature extraction, was extensively utilized to effectively extract information from THz spectrum data. To further mitigate noise and enhance THz spectrum data, a CDAE, in conjunction with AE and CNN, was developed. Mean Square Error (MSE) served as the loss function to evaluate the performance of the CDAE in reconstructing the THz spectrum.

Bedi, Punam, and Pushkar Gole [14]. Due to their superior performance on image data, two Deep Learning techniques that are frequently used in computer vision applications are Convolutional Neural Networks (CNNs) and Convolutional Autoencoders (CAEs). The two methods extract different temporal and spatial elements from image data by using the convolution procedure. CNNs are utilized for input image classification into their appropriate classes, whereas CAEs are employed for effectively reducing an image's dimensionality. This study presents a novel hybrid model that combines convolutional neural networks (CNN) with convolutional autoencoders (CAE) to diagnose plant diseases automatically. In comparison to current state-of-the-art methods described in the literature, a unique hybrid model that uses less training parameters has been designed to automatically detect plant ailments. Two deep learning methods are used by this model: CNN and CAE.

Qiu, Zhengjun [15]. Four types of rice seeds were captured using hyperspectral imaging at 2 separate spectral bands: one spanning from 380 to 1030 nm and another from 874 to 1734 nm. The spectral data were obtained within two distinct ranges: the first range spanning from 441 to 948 nm (referred to as Spectral range 1), and the second range covering from 975 to 1646 nm (referred to as Spectral range 2). Different numbers of training samples were used to build CNN, SVM, and K nearest neighbours (KNN) models. The models utilizing Spectral range 2, including KNN, SVM, and CNN, demonstrated slightly superior performance compared to those utilizing Spectral range 1. They modified the VGGNet architecture to handle inputs of one-dimensional spectra. Spectral curve patterns and picture patterns are similar in a few ways. VGGNet's modular design facilitates easy modification and extension, and it is chosen because to its outstanding performance in image classification tasks. The model's functionality increased.

Wu, Yiqiang, et al [16]. Deep Rice is based on a deep learning architecture. To provide further detail, Deep-Rice employs a multi-view CNN architecture to extract discriminative features from multiple rice image views. It then utilizes a modified softmax loss function to optimize the CNN parameters effectively. In conjunction with this deep learning model, we created a comprehensive rice dataset called FIST-Rice, aiming to serve as a fundamental resource for food security research. Each sample in the dataset is photographed under three different lighting conditions. Using the FIST-Rice dataset, we conducted a comparative analysis between the proposed Deep-Rice model and traditional techniques such as AdaBoost and SVM. The outcomes of the experiment suggest that the Deep-Rice model performs better under various light intensity settings. Using a modified softmax, the deep network may learn angularly discriminative features.

Rathore, Narendra Pal Singh, and Lalji Prasad [17]. Computer-aided methods are used to process results from disease detection quickly and accurately. Convolutional neural networks (CNNs), which automatically extract characteristics and classify images using fully linked networks, are used by the deep learning models. We have classified images using CNN architecture. We are utilizing input photos with the labels "healthy" and "leaf_blast" from the Kaggle dataset. In this study, 1000 samples of rice crops were used to train CNN using an RGB color model. The suggested architectural design for differentiating between leaf-blast and healthy rice crops. OpenCV, Numpy, and Keras are the deep learning packages used in the Python development of this model. We employed a batch size of 32, which is a hyper-parameter in deep learning that may be changed.

Tran-Thi-Kim, Nga, et al [18]. This study employs (CNN) Convolutional Neural Network and (ANN) Artificial Neural Network models and classify 17 types of rice grains commonly cultivated in Vietnam. Pre-trained VGG16 and Resnet50 models serve as the foundation for the two CNN models-modified VGG16 and modified ResNet50. The CNN models were fed the image dataset, while the ANN was fed the feature dataset. The findings demonstrate that the classification accuracy of the 17 types of rice was considerably increased by the updated VGG16 and ResNet50 models. The findings demonstrated that, in comparison to the SVM, the classification accuracy of a CNN model was much higher. Using the feature dataset as a model, the ANN produced a 92.82% classification accuracy.

Murat Koklu, Ilkay Cinar, Yavuz Selim Taspinar [19]. Rice stands as one of the most extensively cultivated grain crops globally, boasting a diverse array of genetic varieties. These variants exhibit distinctions primarily in attributes such as colour, size, and shape. These distinctive characteristics among rice varieties can be utilized for classifying and assessing seed quality. In Turkey, five types of rice are commonly cultivated: Arborio, Basmati, Ipsala, Jasmine, and Karacadag. In this investigation, these types were used. Of the 75,000 grain pictures in the dataset, 15,000 come from each of these categories. Another dataset comprising 106 features was utilized alongside these images, encompassing 12 morphological, 4 shape-related, and 90 color features. Models were constructed using Artificial Neural Networks (ANNs).

PROPOSED SYSTEM

In this project, we propose a novel approach that combines Convolutional Neural Networks (CNN) and Autoencoders to address the task of rice quality detection. We propose an innovative system that integrates Convolutional Neural Networks (CNN), MobileNet, and Autoencoders to optimize rice quality detection. The proposed system leverages the fusion of CNN, MobileNet, and Convolutional Autoencoder (CAE) methodologies. The CAE, implemented using TensorFlow or PyTorch, efficiently reconstructs rice images for feature extraction. Subsequently, the MobileNet and CNN models are employed for accurate image classification, capitalizing on MobileNet's lightweight architecture for efficiency. This combined approach, excluding CNN, showcases innovation by streamlining the feature extraction and classification processes. This novel combination of Convolutional Autoencoder (CAE), Convolutional Neural Network (CNN), and MobileNet is intended to increase the precision and efficacy of rice quality assessment. It presents opportunities for real-world application in the food sector, particularly in activities related to quality assurance and control.

MATERIAL AND METHODOLOGY

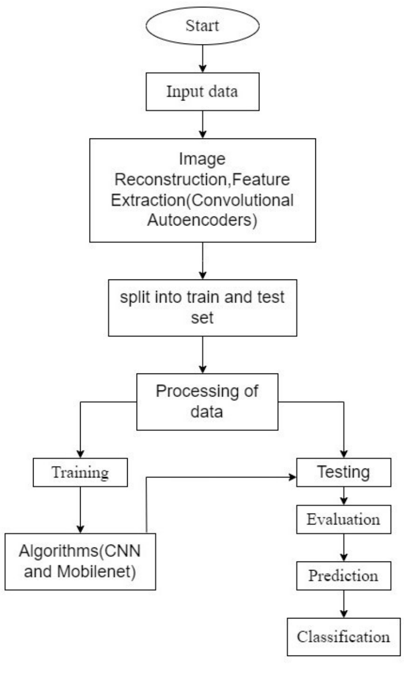

FIGURE 1: Block diagram for suggested approach

Methodology Overview

Image Acquisition

In convolutional autoencoders, image acquisition involves obtaining high-resolution, diverse datasets that undergo preprocessing for normalization and augmentation. The dataset composition, noise handling, and compatibility with convolutional layers are critical considerations to ensure effective feature learning and model performance.

Loading Rice Images Dataset

Loading the Rice Images Dataset involves retrieving a collection of images representing various aspects of rice grains. Researchers typically employ data loading techniques using programming languages like Python and libraries such as TensorFlow or PyTorch. This dataset is essential for training and evaluating machine learning models, especially those designed for tasks such as quality assessment or classification in the context of rice grain analysis.

Preprocessing

Preprocessing is a crucial step in data preparation for machine learning tasks. It involves techniques such as normalization, data augmentation, and handling missing values to enhance the quality and consistency of the dataset. These preprocessing methods contribute to improved model performance by mitigating noise, ensuring uniformity, and facilitating better generalization during training.

Image Reconstruction

In CAEs, image reconstruction involves encoding input images into a latent representation and then decoding them back into reconstructed images. The objective is to minimize the reconstruction error, ensuring that the decoded images closely match the original input, thus capturing meaningful features in the process.

- **Encoder:** In image reconstruction using CAEs, the encoder is responsible for transforming input images into a compressed latent representation. This encoded representation contains essential features extracted from the input, facilitating efficient storage and subsequent reconstruction during the decoding phase. The encoder's role is crucial in capturing meaningful information for faithful image representation.
- **Decoder:** In image reconstruction using autoencoders, the decoder reconstructs images from their compressed latent representations, restoring them to a format similar to the input. The decoder plays a crucial role in generating faithful reconstructions by translating the learned features from the latent space back into the original data space.

Feature Extraction

Size

The size of feature extraction is a critical factor influencing model performance and computational efficiency in machine learning.

L/B Ratio

The aspect ratio (l/b ratio) in feature extraction affects the balance between capturing long-range and short-range patterns, influencing the model's ability to learn diverse spatial information.

Color

The mean color in feature extraction provides a representative measure of the average color values of RGB in an image, contributing to color-based information analysis in machine learning tasks.

Chalkiness

Chalkiness in feature extraction refers to the degree of lightness or whiteness, often utilized in image analysis to characterize texture or material properties with applications in fields like agriculture or food processing.

$$\text{Chalkiness} = \frac{\text{Total number of pixels}}{\text{Number of pixels in chalky range}} \times 100$$

Whiteness

Whiteness in feature extraction represents the level of lightness or brightness in an image, often quantified through metrics such as luminance or color intensity, crucial for various applications like image analysis and quality assessment.

TABLE 1. Characteristics of reconstructed images

S. No	Size	Shape(L/B ratio)	Chalkiness	Whiteness	Mean color (BGR)
1	1790.5	1.942	0.091	24.79	(25.4,24.7,24.7)
2	1685.5	0.666	0.079	24.79	(23.2,22.5,22.6)
3	2025.0	0.796	0.122	29.24	(29.9,29.1,29.1)
4	1870.0	1.169	0.095	25.99	(26.8,25.9,25.8)
5	1872.5	1.037	0.052	23.54	(24.0,23.5,23.4)
6	1579.0	0.767	0.076	22.20	(22.4,22.1,22.1)
7	1509.5	1.875	0.075	21.05	(21.1,21.1,21.1)
8	1703.5	1.340	0.060	21.86	(22.2,21.7,21.8)
9	1943.9	0.057	0.075	25.77	(26.4,25.6,25.7)
10	1760.0	0.730	0.087	24.95	(25.5,24.8,24.9)
11	1748.5	0.646	0.090	24.48	(24.4,24.4,24.4)
12	2082.5	1.850	0.102	33.66	(33.6,33.6,33.6)
13	1855.0	1.131	0.051	24.06	(24.0,24.0,24.0)
14	2027.0	1.055	0.057	25.97	(27.0,26.9,26.9)
15	1783.0	1.842	0.061	24.02	(24.4,23.9,24.2)

Training and Testing datasets were created from the data after the reconstructed images were gathered and features were retrieved using convolutional Autoencoders. Convolutional Neural Networks (CNN) and MobileNet are the algorithms we utilized to train the model. For training and testing sets, arrange image data into the proper directories. Here, we utilized an unlabeled dataset consisting of 75,000 images divided into five varieties: Arborio, Basmati, Jasmine, Ipsala and Karacadag. Each rice variety comprises 15,000 images, and the dataset has been split into three classes - High, Low and Medium - for each variety. These classes are divided based on the features that were extracted from the CAEs. For the training and testing datasets, the data is also divided into three classes.

The following libraries are imported to create the model: NumPy, Pandas, Matplotlib, OpenCV, TensorFlow, Scikit-learn. ImageDataGenerator is a class provided by the TensorFlow Keras library for real-time data augmentation during model training. Several layer classes were imported from the TensorFlow Keras package. These layers are the basic components that make up neural network architectures: Dense, Dropout, Flatten, Conv2D, MaxPool2D, AveragePooling2D, BatchNormalization.

When loading and preprocessing your image data, you would typically ensure that all images are resized to these specified dimensions before being fed into the neural network for training. The dimensions of image height and width are critical as they dictate the size of the input tensor expected by the neural network. These values are typically selected based on the characteristics of the dataset and the specifications of the neural network architecture.

we employed a Sequential model as the foundational architecture for our neural network implementation. The Sequential model, a component of the TensorFlow Keras library, facilitated the structured and sequential assembly of various layers, allowing for the creation of a comprehensive neural network for our specific task. This approach provides a linear stack of layers, simplifying the representation of our model's architecture.

This architecture involves hyper parameters, plays a crucial role in shaping the behavior and expressive power of the convolutional layer. Choosing appropriate values for these parameters depends on the nature of the data, the complexity of the patterns to be learned, and the overall architecture of the neural network. Some of the hyper parameters are:

Conv2D Layer

The Conv2D layer is a fundamental building block in a convolutional neural network (CNN). It performs a 2D convolution on the input data, which is particularly well-suited for image processing tasks.

Filters

The parameter "filters" determines the number of filters, also referred to as kernels, that the layer will acquire. Each filter is responsible for identifying distinct patterns or features within the input data.

Kernel Size

The parameter "kernel size" dictates the dimensions of the convolutional window or filter.

Activation Function

The "activation" parameter defines the activation function applied elementwise to the output resulting from the convolution operation.

When building and training a neural network, tuning these hyper parameters often involves experimentation and iterative refinement to achieve optimal performance on the specific task at hand. Epochs denote the count of times the entire training dataset is iteratively processed by the model during the training phase. A single iteration through the complete training dataset is referred to as an epoch.

Adapt the epochs in accordance with the model, as taking too few or too many epochs results in either overfitting or underfitting.

In a neural network model, the optimizer is a mathematical algorithm that adjusts the weights and biases during training to minimize the chosen loss function, guiding the model towards better performance. During training, specific metrics are designated to assess the model's classification capabilities. These metrics include Accuracy, Precision, and Recall.

EXPERIMENTS AND RESULTS

Graphs that show the difference between the original and reconstructed pictures using MAE, MSE, PSNR, and SSIM.

MAE

One statistic used to quantify the average size of deviations between expected and actual values is mean absolute error (MAE). By averaging the absolute disparities between the true and anticipated values, it is calculated. MAE offers a straightforward assessment of the average error magnitude, rendering it especially valuable in regression and prediction endeavours.

MSE

A popular metric for evaluating the average squared difference between comparable elements of two sets, such as projected and actual values, is mean squared error, or MSE. It is computed by averaging the squared differences between predicted and true values. MSE offers a measure of the average magnitude of errors, where higher values indicate greater overall error in prediction or reconstruction tasks.

PSNR

A typical metric used in image and video compression is the Peak Signal-to-Noise Ratio (PSNR), which measures the quality of reconstructed signals. It will be calculated in Decibels. It calculates the ratio of the mean squared error of the original and reconstructed signals to the highest feasible signal value. Greater faithfulness and less distortion in the reconstruction are indicated by a higher PSNR.

SSIM

A metric for comparing two images similarity is called the Structural Similarity Index (SSIM).It takes into account luminance, contrast, and structure, providing a more comprehensive measure of perceptual image quality than some traditional metrics. A higher SSIM value indicates greater similarity between the original and distorted images, with 1 representing perfect similarity.

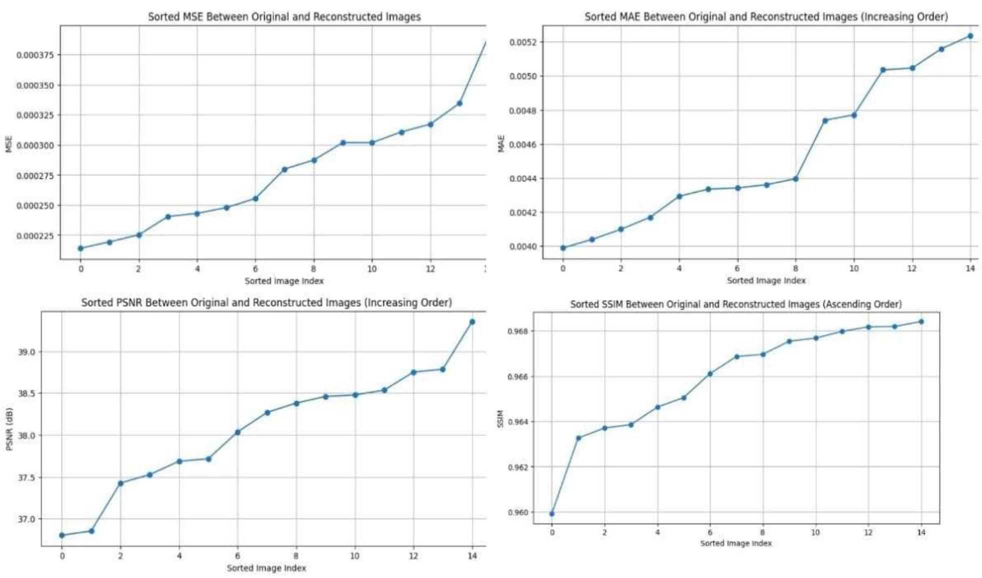

FIGURE 2: Difference between original and reconstructed images

CNN and MobileNet Graphs

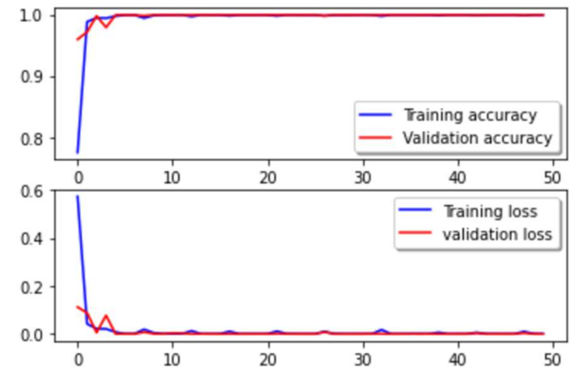

FIGURE 3: Testing accuracy for CNN

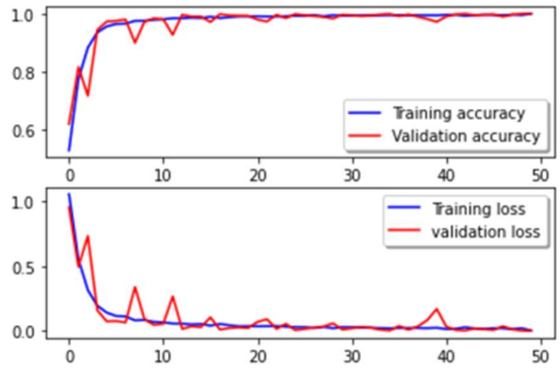

FIGURE 4: Testing accuracy for MobileNet

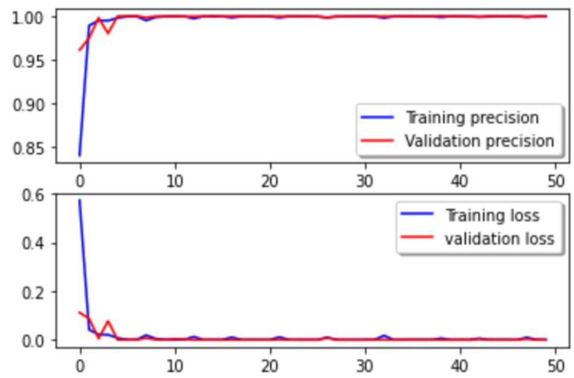

FIGURE 5: Testing precision for CNN

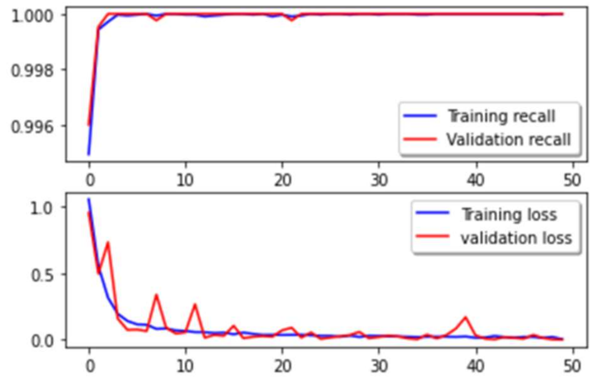

FIGURE 6: Testing precision foe MobileNet

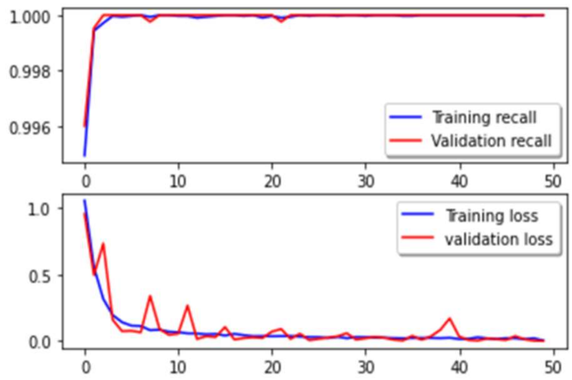

FIGURE 7: Testing recall for CNN

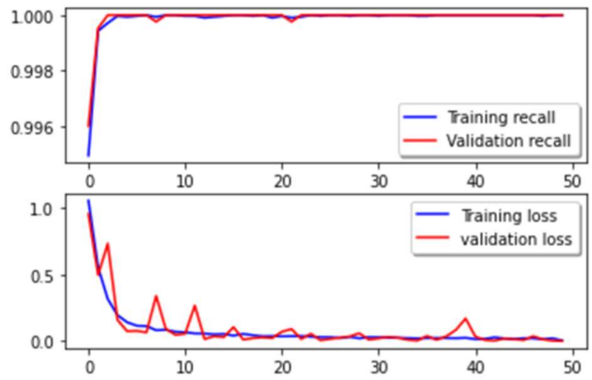

FIGURE 8: Testing recall for MobileNet

In terms of testing and training accuracy, the suggested CNN model achieved 99.89% and 99.92%, respectively, with testing and training losses of 0.02% and 0.03%. Furthermore, 99.46% precision and 98.85% recall were attained by the model.

Training and testing losses for the suggested MobileNet model are 0.40% and 0.51%, respectively. Training accuracy is 88.27%, and testing accuracy is 85.06%. It has also achieved 86.45% accuracy and 84.57% recall rate.

From the two models CNN and MobileNet we have highest accuracy, precision and recall for CNN Model.

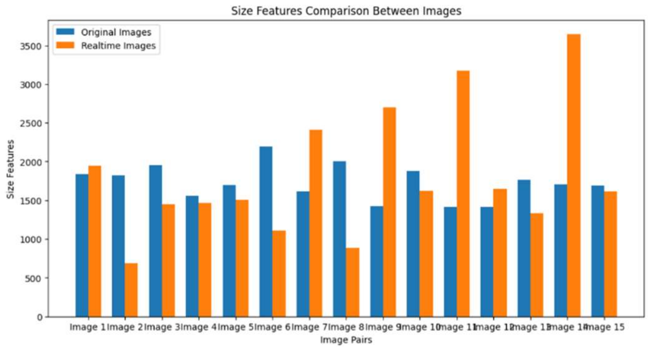

FIGURE 9: Comparison of size feature between original and real - time images

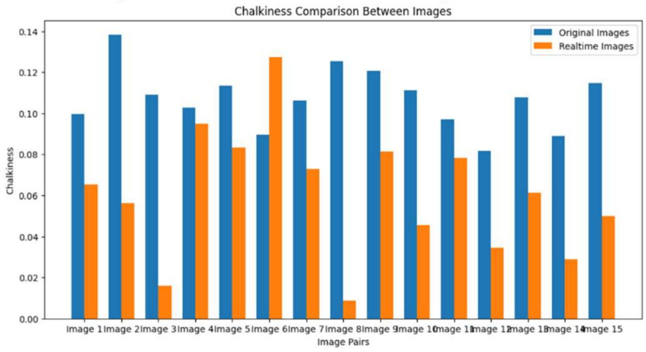

FIGURE 10: Comparison of chalkiness feature between original and real - time images

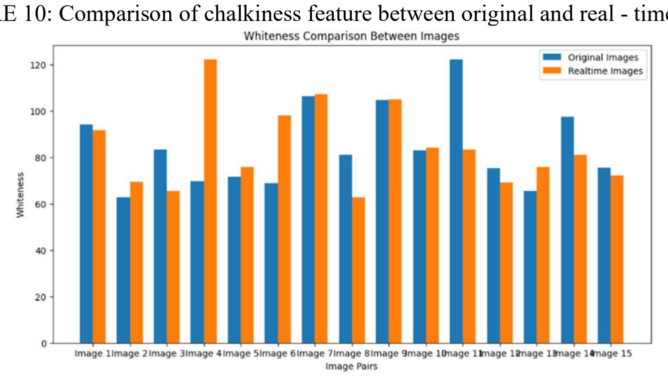

FIGURE 11: Comparison of whiteness feature between original and real - time images

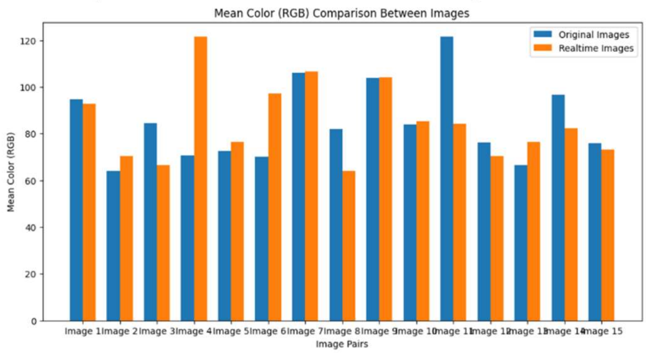

FIGURE 12. Comparison of mean color feature between original and real - time images

From the above graphs the given rice grains are divided into three classes High, Low, Medium based on the features of the images.

The application of deep learning techniques, including convolutional neural networks (CNN), DenseNet, and MobileNet, has demonstrated promising outcomes in the evaluation of rice quality. These methods have been employed to detect and categorize various parameters such as chalkiness, size, and shape, which are pivotal in assessing rice quality. Utilizing deep learning for rice quality analysis presents several advantages over traditional methods, including heightened accuracy, expedited analysis, and the capability to assess numerous samples concurrently. This can be particularly advantageous for rice breeding programs, where the assessment of a large volume of rice samples is essential for identifying desirable traits.

CONCLUSION

In culmination, our project's fusion of Sequential Convolutional Neural Networks (CNN) and Convolutional Autoencoders (CAEs) signifies a breakthrough in precision for rice grain quality assessment. By harmonizing CNN's adept object identification with CAEs' feature learning, the resulting framework exceeds the limitations of conventional methods. The model's fine-tuning, coupled with data augmentation, ensures robust adaptability and generalization, validated by impressive evaluation metrics.

As we deploy this cutting-edge technology into practical applications, it not only addresses existing challenges but also pioneers a new era of automated and elevated rice quality standards, crucial for global food security. This success is a testament to the collaborative efforts of neural network researchers, showcasing the impactful role of machine learning in revolutionizing the production and quality control of vital food crops like rice, ensuring both agricultural sustainability and consumer well-being.

FUTURE SCOPE

Future enhancements to our integrated neural network approach for rice quality assessment could involve the incorporation of advanced sensor technologies and real-time data collection during the rice cultivation and processing stages. Implementing a more dynamic system that adapts to changing environmental conditions and crop variations could further improve the model's accuracy and robustness. Furthermore, investigating the incorporation of explainable AI techniques can improve the model's decision-making interpretability, offering stakeholders insightful information. Continuous updates to the dataset and model architecture will ensure adaptability to emerging rice varieties and quality parameters.

ACKNOWLEDGMENTS

We extend our sincere gratitude to the contributors and collaborators involved in the development of this project aimed at enhancing rice quality assessment. Special thanks are extended to the teams and researchers who pioneered the utilization of neural networks, particularly Convolutional Neural Networks (CNNs) and Convolutional Autoencoders (CAEs), in the realm of food crop evaluation. Their innovative work forms the foundation of our approach, allowing us to leverage the power of machine learning for more accurate and efficient rice grain quality classification. This project would not have been possible without the collective efforts and expertise of those dedicated to advancing technology in agriculture and food industry applications.

REFERENCES

1. Thae Nu Wah, Pann Ei San, Thandar Hlaing (2018); *Analysis on Feature Extraction and Classification of Rice Kernels for Myanmar Rice Using Image Processing Techniques*; International Journal of Scientific and Research Publications (IJSRP) 8(8) (ISSN: 2250-3153).
2. S. Wentao, *Classification Model of Wheat Grain based on Autoencoder* 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2020, pp. 830-832
3. Wang, R., Jiang, Y., & Cao, F. (2019). *Identification and classification of rice seeds using deep convolutional neural networks*. Journal of Food Engineering, 263, 92-101.
4. N. Hong Son and N. Thai-Nghe, *Deep Learning for Rice Quality Classification*, 2019 International Conference on Advanced Computing and Applications (ACOMP), Nha Trang, Vietnam, 2019, pp. 92-96.

5. Gao, Y., Wang, W., Zhang, W., & Gao, X. (2020). *An automated inspection system for rice seed quality based on deep learning*. Journal of Food Process Engineering, 43(6), e13396.
6. Latha, M. N., & Nandhini, R. (2020). *Rice quality analysis using deep learning techniques: A review*. Computers and Electronics in Agriculture, 174, 105404.
7. T. G. Devi, P. Neelamegam and S. Sudha, *Machine vision based quality analysis of rice grains* 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, India, 2017, pp. 1052-1055.
8. Annadasu, P. and Jaisharma, K., 2020. *Rice Quality Analysis Using Machine Learning* (No. 4761). EasyChair.
9. Md Taimur Ahad, Yan Li, Bo Song, Touhid Bhuiyan, *Comparison of CNN-based deep learning architectures for rice diseases classification*, Artificial Intelligence in Agriculture,Volume 9,2023, ISSN 2589-7217.
10. Islam, Redoun Islam, S. M. Rafizul Haque, S.M. Mohidul Islam, Mohammad Ashik Iqbal Khan, *Rice Leaf Disease Recognition using Local Threshold Based Segmentation and Deep CNN*, International Journal of Intelligent Systems and Applications(IJISA), Vol.13, No.5, pp.35-45, 2021.
11. Patel, Vaibhav. (2017). *Convolutional neural network with transfer learning for rice type classification*.
12. Jana, Debasish & Patil, Jayant & Herkal, Sudheendra & Nagarajaiah, Satish & Dueñas-Osorio, Leonardo. (2022). *CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction*. Mechanical Systems and Signal Processing. 169. 108723.
13. Hongbin, et al. *Terahertz spectra reconstructed using convolutional denoising autoencoder for identification of rice grains infested with Sitophilus oryzae at different growth stages*. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy (2024): 124015.
14. Bedi, Punam, and Pushkar Gole. *Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network*. Artificial Intelligence in Agriculture 5 (2021): 90-101.
15. Qiu, Zhengjun, et al. *Variety identification of single rice seed using hyperspectral imaging combined with convolutional neural network*. Applied Sciences 8.2 (2018): 212.
16. Wu, Yiqiang, et al. *Deep-rice: Deep multi-sensor image recognition for grading rice*. 2018 IEEE International Conference on Information and Automation (ICIA). IEEE, 2018.
17. Prakash, P. V. ., & Srivenkatesh, M. . (2024). *Hybrid Deep Learning Algorithms for Predicting Nutrient Deficiencies in Paddy Crops using CNN and Super Resolution Generative Adversarial Neural Networks*. International Journal of Intelligent Systems and Applications in Engineering, 12(16s), 520–526.
18. Tran-Thi-Kim, Nga, et al. *Enhancing the Classification Accuracy of Rice Varieties by Using Convolutional Neural Networks*. International Journal of Electrical and Electronic Engineering & Telecommunications 12.2 (2023): 150-160.
19. Murat Koklu, Ilkay Cinar, Yavuz Selim Taspinar, *Classification of rice varieties with deep learning methods*,Computers and Electronics in Agriculture.

PARTICIPATION CERTIFICATE

AIP Conference Proceedings

CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360

ICIAET-24 Proceedings

International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24)

5^{th} & 6^{th} April 2024

Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Veera prakash pobbatti, of SRIT- Anantapur has participated and presented a paper entitled Enchancing Rice Quality Assessment through integrated Neural Networks: A synergistic Approach with CNN and CAEs with Paper ID: ICIAET-P172 in **ICIAET-2024.**

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT

turnitin
Official sponsor

AIP Conference Proceedings

CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360

ICIAET-24 Proceedings

International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24)

5th & 6th April 2024

Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Megharshini Anke, of SRIT - Anantapur has participated and presented a paper entitled Enchancing Rice Quality Assessment through Integrated Neural Networks: A Synergistic Approach with CNN and CAEs with Paper ID: ICIAET-P172 in **ICIAET-2024.**

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT

turnitin
Official sponsor

AIP Conference Proceedings

CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360

ICIAET-24 Proceedings

International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24)

5^{th} & 6^{th} April 2024

Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Anees Fathima Shaik, of SRIT- Anantapur has participated and presented a paper entitled Enchancing Rice Quality Assessment through Integrated Neural Networks: A Synergistic Approach with CNN and CAEs with Paper ID: ICIAET-P172 in **ICIAET-2024.**

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT

turnitin
Official sponsor

AIP Conference Proceedings

CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360

ICIAET-24 Proceedings

International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24)

5th & 6th April 2024

Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Anitha Myla, of SRIT- Anantapur has participated and presented a paper entitled Enchancing Rice Quality Assesment through Integrated Neural Networks : A synergistics Approach with CNN and CAES with Paper ID: ICIAET- P172 in **ICIAET-2024.**

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT

turnitin
Official sponsor

AIP Conference Proceedings

CHAITANYA BHARATHI
INSTITUTE OF TECHNOLOGY
(UGC- AUTONOMOUS)
PRODDATUR
Vidya Nagar, Proddatur, YSR Kadapa (Dist.), Andhra Pradesh 516360

ICIAET-24 Proceedings

International Conference on Innovative Approaches in Engineering & Technology (ICIAET-24)
5th & 6th April 2024

Organized by Department of Electrical & Electronics Engineering

Certificate of Appreciation

This certificate is awarded to Dr./Mr./Mrs./Miss. Giri Vardhan Velupula, of SRIT- Anantapur has participated and presented a paper entitled Enchancing Rice Quality Assement through Integrated Neural Networks: A Synergistics Approach with CNN and CAEs with Paper ID: ICIAET - P172 in ICIAET-2024.

Dr V Mahesh Kumar Reddy
Convenor & Organising Chair

Dr G Sreenivasula Reddy
Principal, CBIT

turnitin
Official sponsor