A NOVEL APPROACH TO ENHANCE CROP YIELD WITH SEED QUALITY ANALYSIS AND MACHINE LEARNING

A Project Report Submitted to

Jawaharlal Nehru Technological University Anantapur

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

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND SYSTEMS ENGINEERING

Submitted by

Batch No: CSSE 25-19

NAGELLA NAGA HARIKA	**21121A1581**
SHAIK ABDUL RAZAQ	**21121A15A1**
PALLAM GURUNATH	**21121A1583**
MANDAVA SASHANK	**18121A1561**

Under the Supervision of

Mrs. Ch Prathima, *M. Tech, Ph.D*
Assistant Professor
Department of Data Science

Department of Computer Science and Systems Engineering
SREE VIDYANIKETHAN ENGINEERING COLLEGE
(AUTONOMOUS)
Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA

2024 -2025

Department of Computer Science and Systems Engineering

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(AUTONOMOUS)

Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA

Certificate

This is to certify that, the Project work entitled

"A NOVEL APPROACH TO ENHANCE CROP YIELD WITH SEED QUALITY ANALYSIS AND MACHINE LEARNING"

is the bonafide work done by

NAGELLA NAGA HARIKA	**21121A1581**
SHAIK ABDUL RAZAQ	**21121A15A1**
PALLAM GURUNATH	**21121A1583**
MANDAVA SASHANK	**18121A1561**

in the Department of **Computer Science and Systems Engineering**, **Sree Vidyanikethan Engineering College (Autonomous), Sree Sainath Nagar, Tirupati** and is submitted to **Jawaharlal Nehru Technological University Anantapur, Anantapuramu** for partial fulfillment of the requirements of the award of B.Tech degree in Information Technology during the academic year 2024-2025. This work has been carried out under my supervision. The results of this project work have not been submitted to any university for the award of any degree or diploma.

Supervisor:

Mrs.Ch.Prathima
Assistant Professor
Dept. of Data Science
Sree Vidyanikethan Engineering College
Sree Sainath Nagar, Tirupati – 517 102

Head of the Dept.:

Dr. K. Gupta, *M.Tech., Ph.D.*
Professor & Head
Dept. of CSSE
Sree Vidyanikethan Engineering College
Sree Sainath Nagar, Tirupati – 517 102

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project report titled "**A Novel Approach To Enhance Crop Yield With Seed Quality Analysis And Machine Learning "** is the genuine work carried out by us, in **B.Tech(Computer Science and Systems Engineering)** degree course of **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** and has not been submitted to any other college or University for the award of any degree by us.We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of the students

1.

2.

3.

4.

ACKNOWLEDGEMENT

We are extremely thankful to our beloved Chairman and founder **Dr. M. Mohan Babu** who took keen interest to provide us the infrastructural facilities for carrying out the project work.

I am extremely thankful to our beloved Chief executive officer **Sri Vishnu Manchu** of Sree Vidyanikethan Educational Institutions who took keen interest in providing better academic facilities in the institution.

We are highly indebted to **Dr. Y. Dileep Kumar**, Principal of Sree Vidyanikethan Engineering College for his valuable support and guidance in all academic matters.

We are very much obliged to **Dr. K. Gupta**, Professor & Head, Department of CSSE, for providing us the guidance and encouragement in completion of this project.

We would like to express our indebtedness to the project coordinator, **Mr. P. Yogendra Prasad**, Assistant Professor, Department of CSSE for his valuable guidance during the course of project work.

We would like to express our deep sense of gratitude to **Mrs Ch Prathima**, Assistant Professor, Department of Data Science, for the constant support and invaluable guidance provided for the successful completion of the project.

We are also thankful to all the faculty members of the IT&CSSE Department, who have cooperated in carrying out our project. We would like to thank our parents and friends who have extended their help and encouragement either directly or indirectly in completion of our project work.

ABSTRACT

Seed quality is one of the important determinants of crop yield and food security through different types of seeds. It ranges from fruit seeds, such as apple and mango, to grain seeds like maize, soyabean and wheat. Seeds are susceptible to environmental and mechanical stresses associated with their production and processing, which negatively impact germination and plant growth quite often. There are many challenges associated with conventional assessment of seed quality. To overcome the problems discussed above, a predictive system of seed quality evaluation using machine learning is proposed. The model extracts features from seed images, assessing parameters like size, shape, color, and texture of the seeds in order to minimize the decision-making errors of humans while classifying the seeds according to their quality. The model also emphasizes the automation of seed segregation that adds to the efficiency and uniformity and also the profitability of business. The reason is because of the seed supply to fruit and grain markets amongst others has been made high quality. This holistic approach helps the farmer gain insights in real time, which forces farming to become sustainable and elevates resource use, thereby making agriculture into a more lucrative as well as sustainable venture.

***Keywords*:** *Seed Quality, Machine Learning, Image Classification, Automation, Sustainable Agriculture, Soyabean Seeds, Vegetable Seeds, Image Augmentation.*

TABLE OF CONTENTS

Title	Page No.
CERTIFICATE	i
DECLARATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
TABLE OF CONTENTS	v-vi
LIST OF FIGURES	vii
LIST OF TABLES	viii
CHAPTER 1 INTRODUCTION	1-2
CHAPTER 2 LITERATURE SURVEY	3-7
CHAPTER 3 SYSTEM ANALYSIS	8-12
3.1 Proposed System	8
CHAPTER 4 SYSTEM DESIGN AND IMPLEMENTATION	13-24
4.1 System Requirements	13
4.1.1 Hardware Requirements	13
4.1.2 Software Requirements	13
4.2 Design using UML Elements	14
4.2.1 Class Diagram	15
4.2.2 Use Case Diagram	16
4.2.3 Sequence Diagram	17
4.2.4 Activity Diagram	18
4.3 Project Implementation Details	19
4.3.1 Soya Bean Seeds	19
4.3.2 Preprocessing	20
4.3.3 Data Normalization	21
4.3.4 Feature Selection	21
4.3.5 Classification	22
CHAPTER 5 TESTING	25
5.1 Test Cases	25-27
CHAPTER 6 RESULT	27-31
CHAPTER 7 CONCLUSION AND FUTURE WORK	32

REFERENCES	33-34
APPENDIX	35-40
CERTIFICATES OF CONFERENCE	41-43
COLLEGE VISION & MISSION	44
DEPARTMENT VISION & MISSION	44
PROGRAM EDUCATIONAL OBJECTIVES	45
PROGRAM SPECIFIC OUTCOMES	46
PROGRAM OUTCOMES	46
COURSE OUTCOME EVALUATION SHEET/ CO-PO MAPPING	48

LIST OF FIGURES

Figure No.	Title	Page No.
Figure 3.1	The proposed system architecture	09
Figure 4.1	Class Diagram for Seed Classification	16
Figure 4.2	Use case diagram for Seed Classification System	17
Figure 4.3	Sequence Diagram for Seed Classification System	18
Figure 4.4	Activity Diagram for Seed Classification System	19
Figure 6.1	Feature Extraction of Images Into a Vector	27
Figure 6.2	Feature Extracts of Image into a Grey Scale	28
Figure 6.3	Confusion Matrix	29
Figure 6.4	Performance Comparison of Models	29

LIST OF TABLES

Table No.	Title	Page No.
4.1	Number of Samples and Percentages	20
5.1	Test Case 1	25
5.2	Test Case 2	25
5.3	Test Case 3	26
5.4	Test Case 4	26
5.5	Test Case 5	26

1. INTRODUCTION

Seed segregation into quality types is one of the basic processes in agriculture and directly affects the viability and marketability of crops. Good quality seeds are yielders of good crop while the low -quality seed results in poor germination rates and poor growth. Conventional seed separation techniques usually depend on visual examination and mechanical size grading-would often fall short in providing reliable quality assessments. Such methods are cumbersome and time-consuming and may result in unreliable quality assessments. Old conventional methods are not uniform hence, inefficient for modern crop production and competitive markets. The greatest potential in the improvement of seed segregation is through applying high image processing techniques and machine learning. It makes possible actual critical characteristics, such as seed size, shape, color, and even texture, so that such seeds can be analyzed and placed into the system with the right placement into quality categories. It will minimize the error and consistency of human choices, which will have only the best seeds selected for planting purposes. Deep learning is one of the subcategories in machine learning that has totally changed the whole process of seed sorting and has been quite high in terms of accuracy and efficiency in the classification process. Excellent sets of models in handling image data are deep learning structures, such as CNNs or Transfer learning architectures. They could determine important features without any human interference associated with form, texture and color. This led to accuracy and consistency in seed classification into quality classes which went beyond those that traditional techniques were possible to compare. With this, seed segregation will take a lesser amount of time and human errors lessened, it maximizes crop output. This is not only about making sure that only the best seeds will be held for planting but also enables consumer options to be increased. Seed separation will allow consumers to have a much wider variety of better-quality products since better crops are produced for them, giving the consumer choice in terms of size, color, and even texture. Such assistance makes it more competitive in the market and encourages the practice of sustainable agriculture as these benefits are obtained by farmers and consumers alike from system improvements. Also, keeping in view the parameters of availability of nutrients within the soils, region-specific rainfall information, region-specific customized crop recommendations can also be made by machine learning systems and it further enhances the yield. This mechanization of seed grading makes separation easier and contributes to green agriculture through savings on natural resources and minimal incidences of waste. They enhance profitability on farms through the quality as well as the speed of seed classification and ensure that quality seeds indeed make it to the market to indirectly benefit the producers and consumers. Indeed, in fact, this application of machine learning to seed separation is a super type of agricultural processing that contributes to a better sustainable landscape for agriculture.

2. LITERATURE SURVEY

[1] Erenstein, O., Chamberlin, J., & Sonder, K. (2021)

Enhancing Seed Quality Assessment in Global Agriculture through Machine Learning and Image Processing .

This study emphasizes the critical role of seed quality in ensuring sustainable agricultural output, particularly in maize and wheat production systems. According to Erenstein, Chamberlin, and Sonder [1], traditional seed testing and segregation methods often fall short in accuracy and scalability, especially when applied across diverse global farm settings. In response to these limitations, the authors advocate for the integration of modern technological advances—specifically machine learning (ML) and image processing techniques—to revolutionize the assessment and classification of seed quality. The research explores how these technologies can automate and enhance the identification of viable versus non-viable seeds, thereby improving overall crop yield and reducing the reliance on manual, labor-intensive practices. The authors suggest that deep learning models, such as convolutional neural networks (CNNs), can be trained to analyze high-resolution seed images and extract meaningful features that correlate with quality parameters like color, texture, and structural integrity. By implementing such models, agricultural stakeholders can not only increase the speed and accuracy of seed segregation processes but also ensure consistency across large-scale farming operations. The paper calls attention to the growing need for accessible, technology-driven tools in agriculture—tools that are capable of supporting remote or resource-constrained environments where traditional laboratory-based testing may not be feasible. Despite its promise, the application of ML in seed quality assessment comes with its own set of challenges, such as dataset limitations, model interpretability, and the need for domain-specific validation. Future work must address these issues by curating diverse and representative seed image datasets, improving model robustness under varying lighting and imaging conditions, and collaborating with agricultural experts to validate and refine predictive outputs.

Drawbacks

- Lack of comprehensive and diverse datasets may hinder model generalization across seed varieties.
- Variations in imaging conditions (e.g., lighting, camera resolution) may affect prediction accuracy.
- Model validation without domain expert input could lead to misclassification.
- Implementation in field environments may face logistical and technical barriers.

[2] Mariammal, G., Suruliandi, A., Raja, S. P., & Poongothai, E. (2021) Land Suitability Prediction Using Modified Recursive Feature Elimination and Machine Learning Classifiers.

uilding on the significance of precision in agricultural decision-making, Mariammal et al. [2] developed a machine learning framework that leverages a modified Recursive Feature Elimination (RFE) technique in conjunction with various classification algorithms to predict land suitability for crop cultivation. Their study focuses on the identification and analysis of critical soil and environmental characteristics—such as pH, moisture content, temperature, and organic matter—that influence agricultural productivity. By employing advanced feature selection, the authors were able to isolate the most impactful attributes from high-dimensional datasets, thereby enhancing both model interpretability and performance. The integration of RFE with classifiers like Support Vector Machines (SVM), Random Forests (RF), and Decision Trees (DT) demonstrated robust results in terms of predictive accuracy and generalization across diverse geographical regions. The implications of their findings extend beyond land evaluation to the broader context of agricultural sustainability. The authors underscore the importance of integrating such intelligent systems into crop planning workflows to support efficient resource allocation and strategic farming practices. Importantly, the study reinforces the urgent need for complementary advancements in seed quality detection—emphasizing that land suitability alone cannot guarantee high yields without reliable, high-quality seed inputs. As precision agriculture continues to evolve, combining land suitability modeling with effective seed quality assessment technologies could form a comprehensive strategy for achieving food security and environmental sustainability. Future research directions include the integration of remote sensing data, explainable AI methods, and cross-domain models to improve accuracy, scalability, and adoption in real-world scenarios.

Drawbacks

- Feature selection effectiveness may vary depending on dataset diversity and regional factors.
- Classifier performance is sensitive to parameter tuning and quality of input data.
- Predictive models may struggle to generalize across drastically different agro-climatic zones.
- Lack of integration with seed quality assessment tools limits full-cycle optimization

[3] de Medeiros, A. D., Lopes, R., Gomes-Junior, F. G., et al. (2020) Advanced Seed Quality Classification Using FT-NIR Spectroscopy and X-ray Imaging with Machine Learning

In a novel contribution to seed quality assessment, de Medeiros et al. [3] present an integrated machine learning framework that combines Fourier-transform near-infrared (FT-NIR) spectroscopy with X-ray imaging data to enhance the classification of seed viability. This dual-modality approach capitalizes on the strengths of both spectral and structural analysis, offering a more comprehensive evaluation of seed quality compared to traditional methods. By merging FT-NIR data, which provides insights into the chemical composition of seeds, with high-resolution X-ray images capturing internal structural integrity, the authors were able to train machine learning models capable of identifying subtle but significant indicators of seed health. The resulting classification system demonstrated superior accuracy and consistency across different seed types, showcasing the potential of multi-source data fusion in agricultural diagnostics. This technique addresses several longstanding challenges in seed quality testing, including non-destructive analysis, rapid throughput, and reduced subjectivity. The study's findings highlight how data-driven, sensor-integrated methodologies can significantly elevate seed scoring capabilities, ultimately supporting higher germination rates and better crop performance. The authors emphasize that such advancements not only streamline quality control in seed processing facilities but also contribute to global food security by ensuring that only the most viable seeds enter the production cycle. Future work may involve the use of deep learning for even finer feature extraction and real-time implementation in industrial settings.

Drawbacks

- The requirement for specialized FT-NIR and X-ray imaging equipment may limit scalability in resource-limited settings.
- Data preprocessing and fusion increase computational complexity.
- Model generalization across seed species may require extensive retraining and calibration.
- Real-time deployment needs optimization for speed and cost-effectiveness.

[4] Rehman, T. U., Mahmud, M. S., Chang, Y. K., Jin, J., & Shin, J. (2019) Applications of Statistical Machine Learning in Agricultural Machine Vision Systems for Seed Classification.

The Rehman et al. [4] provide an in-depth review of the current and emerging applications of statistical machine learning algorithms within agricultural machine vision systems. Their work particularly emphasizes the transformative role of image processing techniques in automating the classification and segregation of seeds based on visual cues such as shape, texture, and color. The study explores various supervised learning approaches—including Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Artificial Neural Networks (ANN)—that have been employed in conjunction with computer vision pipelines to analyze seed imagery. These methods enable the automated evaluation of seed quality, replacing traditional manual inspections that are often subjective and time-intensive. One of the study's key contributions is its forward-looking perspective on integrating these technologies into precision agriculture. The authors highlight how advancements in sensor technology, edge computing, and cloud-based analytics are converging to make real-time, in-field seed quality assessment increasingly viable. They also stress the importance of designing scalable and robust systems that can function reliably in diverse and often challenging agricultural environments. By focusing on visual-based classification, the research presents a compelling case for the broader adoption of machine learning in seed segregation workflows. Such integration can lead to enhanced consistency, reduced labor costs, and increased efficiency in agricultural operations.

Drawbacks:

- Variability in seed image quality due to lighting and background conditions can affect classification accuracy.
- Statistical models may require extensive retraining to adapt to new seed varieties or imaging settings.
- Real-time implementation challenges persist in low-resource environments lacking computational infrastructure.
- Potential trade-offs between model complexity and processing speed need careful optimization.

[5] Kiratiratanapruk, K., & Sinthupinyo, W. (2011)

Color and Texture-Based Maize Seed Classification Using Machine Vision Techniques.

Kiratiratanapruk and Sinthupinyo [5] present an early yet pivotal study that explores the feasibility of utilizing color and texture features in the automated classification of maize seeds through machine vision systems. Their research demonstrates that visual cues—specifically color distribution patterns and surface texture metrics—can effectively distinguish between high- and low-quality seeds without the need for invasive testing or manual inspection. By applying image processing techniques such as histogram analysis, edge detection, and texture feature extraction, the authors developed a classification system capable of evaluating seed quality with reasonable accuracy. Their findings provide a foundational framework for subsequent advancements in agricultural automation, emphasizing that even low-cost visual systems can significantly contribute to the efficiency and precision of seed grading. The study's relevance remains high in modern agricultural contexts, where demand for scalable and rapid quality assessment tools continues to grow. As the agricultural sector increasingly turns to digital solutions, the integration of machine vision for seed classification emerges as a practical and cost-effective approach, particularly in seed processing plants and breeding programs.

Drawbacks:

- Sensitivity to external lighting conditions may hinder consistency in feature extraction.
- Accuracy is dependent on precise calibration and preprocessing of seed images.
- The simplicity of early machine vision algorithms may limit classification performance compared to modern deep learning methods.
- Limited adaptability to other seed types without significant modification of feature parameters.

3. SYSTEM ANALYSIS

System analysis is an important activity that takes place when building a new system or changing the existing one. Analysis helps to understand the existing system and the requirements necessary for building the new system.

3.1 PROPOSED SYSTEM

The proposed system utilizes the DenseNet model for feature extraction to capture intricate seed characteristics and Inception V3 for classification, ensuring high accuracy, faster processing, and reliable seed quality assessment for improved agricultural outcomes.

- The Automates the seed quality assessment process, reducing reliance on time-consuming manual inspection.
- Enables faster decision-making for farmers regarding seed selection.
- Integrates deep learning techniques to achieve high accuracy in seed classification.
- Provides better insights into seed quality, helping farmers make informed choices.
- Speeds up the sorting and categorization of seeds for large-scale agricultural processes.
- Enhances the efficiency and reliability of seed quality checks.
- Promotes sustainable agriculture by ensuring only high-quality seeds are used.
- Supports food security through better crop outcomes and reduced wastage.
- Contributes to the mass production of quality crops, addressing challenges like climate change and resource scarcity.
- Optimized deep learning models improve system performance, aligning with modern agricultural needs.

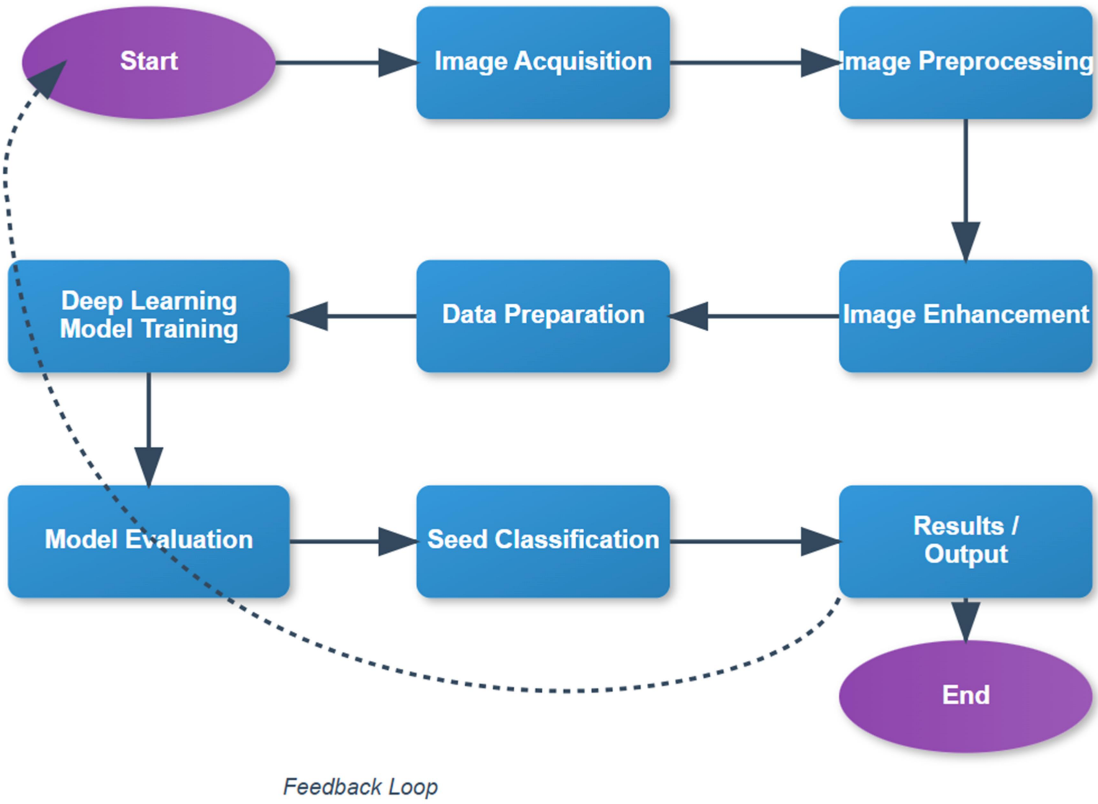


Fig 3.1 The proposed system architecture

I. Dataset

The dataset contains 5,513 single soybean seed images of five classes: intact, spotted, immature, broken, and skin-damaged, with more than 1,000 images in each class, sorted in accordance with the Standard of Soybean Classification (GB1352-2009). The images were taken by an industrial camera, originally with seeds in physical contact, then segmented into 227×227 pixel single seed images based on an image-processing algorithm, with a segmentation accuracy of more than 98%. This dataset is a strong basis for machine learning research on soybean seed classification and quality evaluation.

II. Data Preprocessing

A number of important preprocessing operations were performed to the data set in order to make the image quality and to use for training in the model. The cleaning process was done for erroneous images to the data set. Validity against a predefined list for approved extensions like JPEG, JPG, BMP, or PNG for image files was checked. Images that did not meet these extensions were found and deleted, as well as those that raised a problem while loading. Apart from this, to increase the diversified training dataset and for proper generalization of the model, many techniques based on image augmentation were used. produces variations of existing images in a variety of ways and is done by rotation and scaling, flipping, along with even changing the brightness at times. In this regard, it extends the set of samples for the training set. The pixel values of the image were then normalized to the range [0, 1] by performing each value divided by 255 to help the model converge better during training. Normal photos were reserved for the validation dataset with 20% set aside for the test and the rest kept at 80% for validation. These pretreatments were very important in allowing correct formatting and cleanliness on the dataset, enhancing the subsequent machine learning models. Data normalization is quite very essential to seed quality analysis and segregation because all features count equally in the performance of the system. Pixel values of images in the system are normalized to fall between 0 and 1, which improves the learning capability of the model about the data. This was achieved by carrying out a division of the raw pixel values, which range between 0 and 255, by 255.0 through the following formula:

III. Data Normalization

Data normalization is a crucial preprocessing step in this system, ensuring uniform pixel value distribution, stable model training, faster convergence, and improved accuracy. In the feature extraction phase, the ImageDataGenerator is used with the `rescale` parameter to normalize image pixel values by scaling them from the original range [0, 255] to [0, 1], allowing the model to learn efficiently without being affected by large input values. The transformation applied is , implemented as `datagen = ImageDataGenerator(rescale=1./255)`, which ensures that all images have a consistent range of pixel values, reducing computation complexity and improving model convergence. Additionally, during training, Batch Normalization layers are used in both the DenseNet and InceptionV3 classifiers to stabilize activations by normalizing inputs for each layer. This process involves calculating the mean and variance for each batch, normalizing data by subtracting the mean and dividing by the standard deviation, and applying scaling and shifting using learnable parameters (γ, β) to maintain model flexibility. This reduces internal covariate shift, speeds up training, and enhances the model's ability to generalize effectively.

$$\mathrm{Xnormalized} = \frac{\mathrm{Xmax} - \mathrm{Xmin}}{\mathrm{X} - \mathrm{Xmin}}$$

IV. Feature Extraction

Feature In this approach, the pre-trained DenseNet121 model, trained on the ImageNet dataset, was utilized as a feature extractor by removing its top classification layer and applying global average pooling to obtain fixed-dimensional feature vectors for each input image. The dataset was processed using an ImageDataGenerator for pixel normalization, and features were extracted in batches, with 4,000 samples for training and 1,000 for testing. These features served as input to a custom classifier, which consisted of multiple dense layers, starting with a 1024-unit layer with ReLU activation, followed by batch normalization and dropout for regularization. Additional dense layers with 512 and 256 units were included, each using ReLU activation and dropout, while the final layer utilized a softmax activation function to classify images into five categories. The model was compiled with the Adam optimizer and categorical crossentropy loss, achieving efficient training by leveraging DenseNet121's robust hierarchical feature representations while tailoring the classification layers to the dataset's specific requirements.

DenseNet (Densely Connected Convolutional Networks)

DenseNet (Densely Connected Convolutional Networks) is a deep learning architecture designed to improve information flow and gradient propagation through deep neural networks. Unlike traditional CNNs, where layers are connected sequentially, DenseNet establishes direct connections between each layer and every subsequent layer in a dense block. This structure enhances feature reuse, reduces the number of parameters, and mitigates the vanishing gradient problem. The primary components of DenseNet include dense blocks, where feature maps from all preceding layers are concatenated and passed forward, and transition layers, which use 1x1 convolutions and pooling to reduce the dimensionality of the network. The **DenseNet121** model, used in the proposed system, is a variant with 121 layers, offering an optimal balance between efficiency and accuracy for image classification tasks. In the project, DenseNet121 is employed as a feature extractor, leveraging its pre-trained convolutional layers to extract meaningful representations from seed images. The extracted features are then used as input for further classification using a custom-built model based on InceptionV3.

V. Classification

For accurate and efficient soybean seed quality assessment, our proposed model integrates Inception V3, a powerful deep convolutional neural network architecture known for its scalability and precision in image classification tasks. This approach ensures that subtle differences in seed characteristics are captured effectively, even under challenging visual conditions.

VI. Inception V3 for Soybean Seed Quality Classification

Inception V3 enhances traditional CNN architectures by employing factorized convolutions, auxiliary classifiers, and dimensionality reduction techniques to enable deep network training while maintaining computational efficiency. Its use of inception modules allows the network to extract multi-scale features, which is especially advantageous for analyzing the intricate texture and shape variations present in soybean seeds. In our system, Inception V3 is fine-tuned using a labeled dataset of soybean seed images categorized as good or bad. The model extracts rich hierarchical features—such as seed smoothness, cracks, discoloration, and structural uniformity—enabling robust classification even in the presence of noise or minor defects. By leveraging transfer learning and fine-tuning Inception V3 on the soybean dataset, the model achieves high accuracy and generalization. This architecture not only improves the overall performance but also contributes significantly to the automation and reliability of seed quality assessment, which is crucial for enhancing agricultural productivity.

4. SYSTEM DESIGN AND IMPLEMENTATION

The system design and implementation for soybean seed quality classification using Inception V3 is structured to accurately categorize seeds into five distinct classes: intact, immature, broken, skin-damaged, and spotted. The process begins with preprocessing, where each original image containing a 4x4 grid of seeds is segmented into 16 individual seed images. These images are resized to 299x299 pixels to meet the input requirements of the Inception V3 model and normalized for consistent pixel values. To improve the model's generalization ability, various image augmentation techniques such as rotation, brightness variation, and zooming are applied. Transfer learning is employed using the pre-trained Inception V3 architecture as a feature extractor, while the top layers are replaced with custom dense layers and a softmax activation function to enable multiclass classification. The model is compiled using the Adam optimizer with categorical cross entropy loss, suitable for handling multiple classes. To prevent overfitting, dropout and early stopping techniques are integrated. The dataset is split into 70% training, 20% validation, and 10% testing to ensure a balanced evaluation. This robust design ensures accurate and efficient classification of soybean seeds, making it suitable for practical agricultural quality control systems.

4.1 SYSTEM REQUIREMENTS

The idea of what is required for a proposed system is provided by the system requirements, which is crucial to the creation of any system.This addresses the necessary hardware and software components for the system.

4.1.1 HARDWARE REQUIREMENTS

Verifying the system requirements for hardware devices is equally crucial. The physical computer resources, sometimes referred to as hardware, are the most frequent set of needs specified by any operating system or software program.

These are the hardware specifications :

- RAM: Minimum 8 GB
- Storage: SSD or HDD above 500 GB
- Processor: Intel i5 or i7 processor
- Operating System: Windows 10/11, macOS, or Linux

4.1.2 SOFTWARE REQUIREMENTS

Software requirements pertain to specifying the minimum amount of software resources and prerequisites that must be installed on a computer in order for an application to work as best it can. Before installing the software, these prerequisites must be installed since they are typically not part of the software installation package.

The following are the requirements of software:

- Operating System: Windows 10/11, macOS, or Linux
- Programming Language: Python 3.7 or higher.
- Deep Learning Frameworks: TensorFlow 2.x or PyTorch 1.x
- Libraries and Dependencies: Keras , Sci-Kit learn , Numpy , Pandas , Matplotlib and OpenCv.
- IDE/Text Editor: Jupyter Notebook/Google Colab or VS Code.

4.2. DESIGNING UML ELEMENTS

The unification of object oriented modeling became possible as experience allowed evaluation of the various concepts proposed by existing methods. Based on the fact that differences between the various methods were becoming smaller, and that the method was did not move object-oriented technology forward any longer, Jim Rambaugh, Grady Booch and Jacobson adopted following goals.

1. To represent complete systems (instead if only software portion) using object oriented concepts.
2. To build a system with explicit coupling of concepts along with the executable part of code.
3. To take into account the scaling factors that are inherent to complex and critical systems.
4. To care a modeling language usable for both humans and machines.
5. Provide a formal basis for understanding the modeling language.
6. Integrate best practices.

Reasons to model:

1. To communicate the desired structure and behavior of the system.
2. To visualize and control the system’s architecture.
3. To better understand the system and expose opportunities for specification and reuse.

Introduction to UML:

UML is general purpose visual modeling language that will Specify, Visualize, Construct and

Document the artifacts of the software system.UML is the language used in the information technology industries versions of blue print. It is a method for describing the system architecture in details using this blue print. Building or maintaining a system and ensuring that it meets requirement changes becomes much easier. It is a method for describing the system architecture in details using this blue print. Model is a blue simplification of reality. A model provides the blue prints of a system. Models are constructed to enhance understanding of the system being developed. Complex systems are modeled because their entirety cannot be comprehended. Modelling helps us to visualize a system as it is or as desired. They give us a template that guides us in constructing system and documents the decisions made. The Unified Modelling Language (UML) is a visual language for the specifying, constructing and documenting the artifacts if the software-intensive systems.

Building blocks of UML

The vocabulary of the UML encompasses 3 kinds of building blocks.

1. Things
2. Relationships
3. Diagrams

- **Things:** Things are the data abstractions that are first class citizens in a model. Things are of four types like structural things, Behavioral things, Grouping things, Annotational things.
- **Relationships**: Relationships tie the things together. Relationships in the UML are Dependency, Association, Generalization, and Specialization.
- **Diagrams:** Diagrams in the UML are of 2 types. They are static diagrams, Dynamic diagrams. Static diagrams consist of Class diagram, Object diagram, and Component diagram. Dynamic diagram consists of use case diagram, sequence diagram, collaboration diagram, Activity diagram.

4.2.1 CLASS DIAGRAM

Class diagram is a static diagram and it is used to model the static view of a system. The static view describes the vocabulary of the system. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems.

It consists of classes such as

- ➢ Dataset
- ➢ Preprocessing

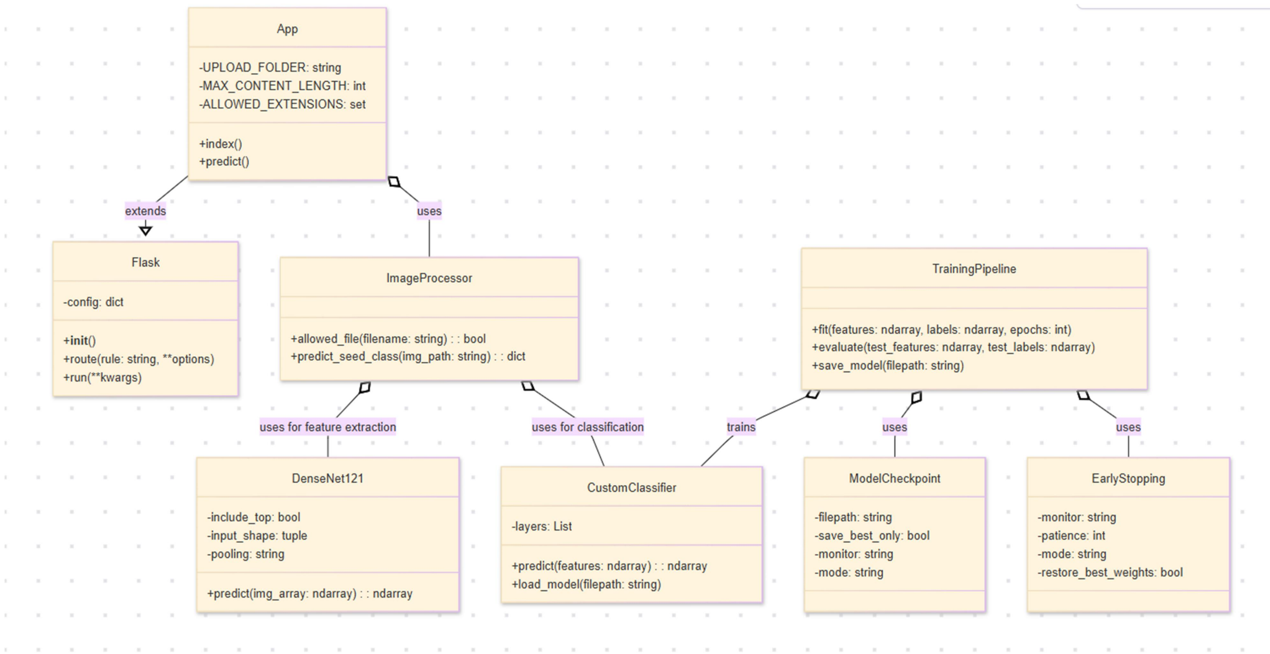


Fig 4.1 Class Diagram for Seed Classification System

4.2.2 USECASE DIAGRAM:

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases. Common Components include:

1. **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.
2. **System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.
3. **Goals:** The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

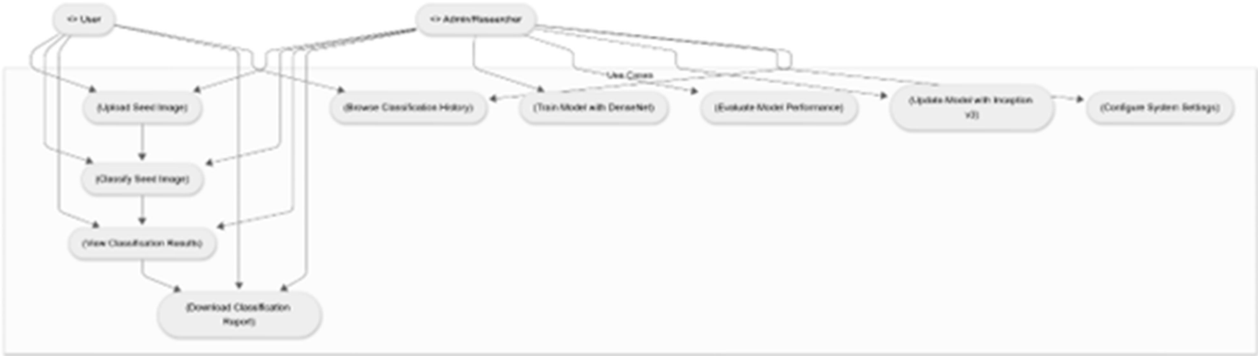


Fig 4.2 Use Case Diagram For Seed Classification System

4.2.3 SEQUENCE DIAGRAM:

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are sometimes known as event diagrams or event scenarios. Sequence diagrams are organized according to time. The time progresses as you go down the page. The objects involved in the operations are listed from left to right according to when they take part in the message sequence . A message is a specification of stimulus: i.e., it specifies the roles that the sender and receiver instances should conform to, as well as the procedure that will, when executed, dispatch a stimulus that conforms to the message.

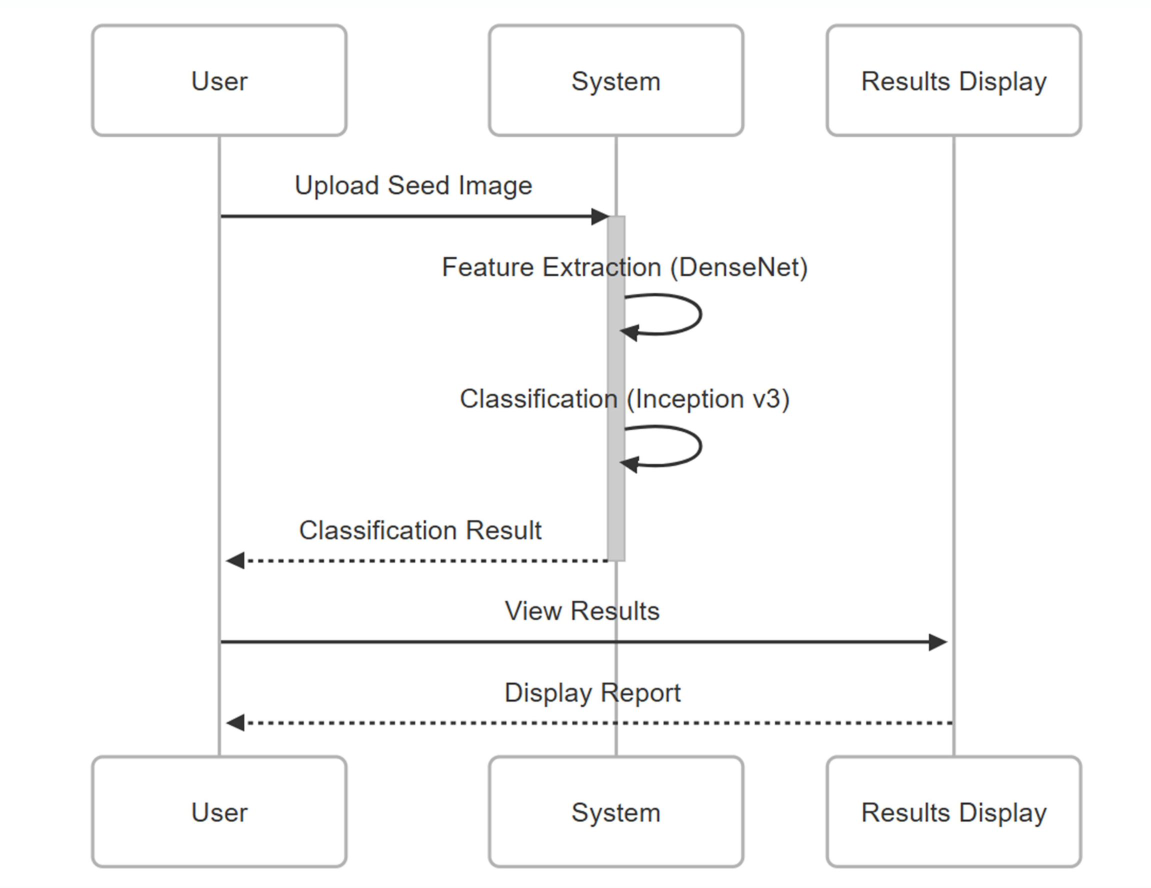


Fig 4.3. Sequence Diagram for Seed Classification System

4.2.4 ACTIVITY DIAGRAM:

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc. A swimlane activity diagram groups the activities into swimlanes columns that contain all of the activities which fit into the category represented by that swimlane. Swimlanes can represent many categories of information such as actors which perform the activities (i.e., role or department), the stage of the process in which the activity takes place, or whatever else the creator of the document feels should be emphasized and communicated by the swimlane diagram in fig 4.5.

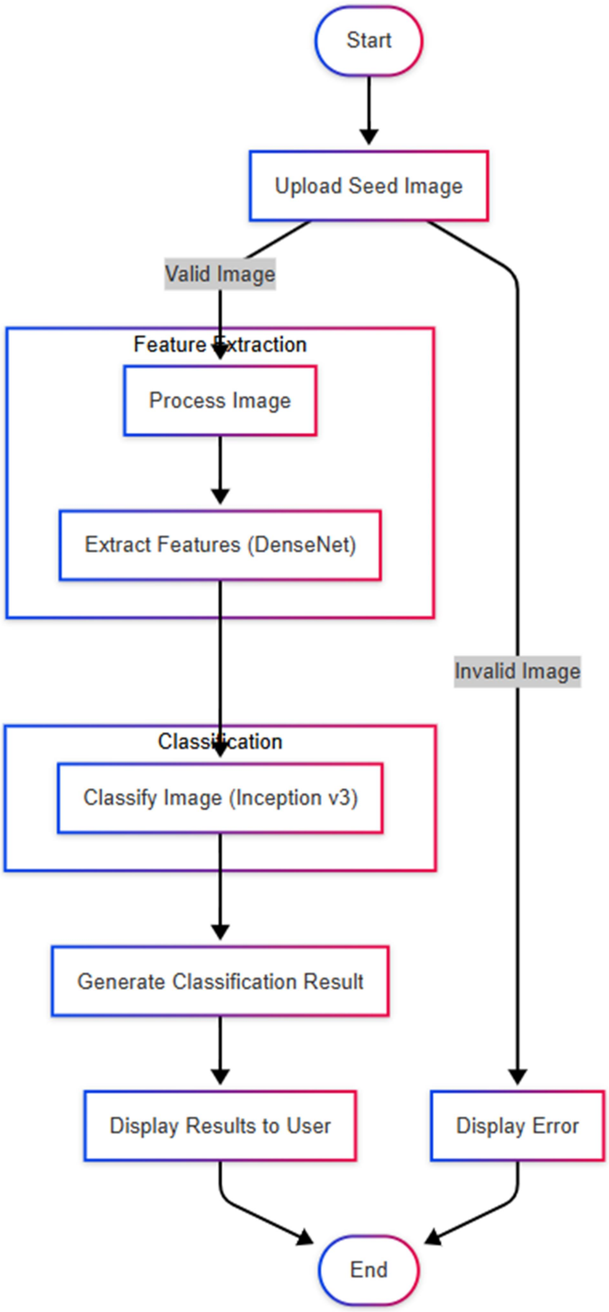


Fig 4.4. Activity Diagram for Seed Classification System

4.3. PROJECT IMPLEMENTATION DETAILS

4.3.1. Soya Bean Seeds

The dataset contains 5,513 single soybean seed images of five classes: intact, spotted, immature, broken, and skin-damaged, with more than 1,000 images in each class, sorted in accordance with the Standard of Soybean Classification (GB1352-2009). The images were taken by an industrial camera, originally with seeds in physical contact, then segmented into 227×227 pixel single seed images based on an image-processing algorithm, with a segmentation accuracy of more than 98%.

This dataset is a strong basis for machine learning research on soybean seed classification and quality evaluation.

Table 4.19: Number of samples and Percentage

Category	Number Of Samples	Percentage
Intact	241	24.29%
Spotted	107	10.79%
Immature	225	22.68%
Broken	201	20.26%
Skin-Damaged	226	22.78%
Total	1000	100%

4.3.2 PRE-PROCESSING

A number of important preprocessing operations were performed to the data set in order to make the image quality and to use for training in the model. The cleaning process was done for erroneous images to the data set. Validity against a predefined list for approved extensions like JPEG, JPG, BMP, or PNG for image files was checked. Images that did not meet these extensions were found and deleted, as well as those that raised a problem while loading. Apart from this, to increase the diversified training dataset and for proper generalization of the model, many techniques based on image augmentation were used. produces variations of existing images in a variety of ways and is done by rotation and scaling, flipping, along with even changing the brightness at times. In this regard, it extends the set of samples for the training set. The pixel values of the image were then normalized to the range [0, 1] by performing each value divided by 255 to help the model converge better during training. Normal photos were reserved for the validation dataset with 20% set aside for the test and the rest kept at 80% for validation. These pretreatments were very important in allowing correct formatting and cleanliness on the dataset, enhancing the subsequent machine learning models. Data normalization is quite very essential to seed quality analysis and segregation because all features count equally in the performance of the system. Pixel values of images in the system are normalized to fall between 0 and 1, which improves the learning capability of the model about the data. Steps included in Data Pre-processing are:

Step 1 : Import the libraries

Step 2 : Import the data-set

Step 3 : Check out the missing values

Step 4 : See the Categorical Values

Step 5 : Splitting the data-set into Training and Test Set

Step 6 : Feature Scaling

4.3.3. DATA NORMALIZATION:

Data normalization is essential to ensure that all features in a dataset contribute equally to the model's performance. Min-Max normalization scales features to a fixed range, typically [0,1], preserving the relationships among data points while ensuring consistency across different features. This method is particularly useful when dealing with varying data ranges and improving the stability of machine learning models.

$$\mathrm{Xnormalized} = \frac{\mathrm{Xmax} - \mathrm{Xmin}}{\mathrm{X} - \mathrm{Xmin}}$$

4.3.4. FEATURE SELECTION

by removing its top classification layer and applying global average pooling to obtain fixed-dimensional feature vectors for each input image. The dataset was processed using an ImageDataGenerator for pixel normalization, and features were extracted in batches, with 4,000 samples for training and 1,000 for testing. These features served as input to a custom classifier, which consisted of multiple dense layers, starting with a 1024-unit layer with ReLU activation, followed by batch normalization and dropout for regularization. Additional dense layers with 512 and 256 units were included, each using ReLU activation and dropout, while the final layer utilized a softmax activation function to classify images into five categories. The model was compiled with the Adam optimizer and categorical crossentropy loss, achieving efficient training by leveraging DenseNet121's robust

Workflow

1. *Initialize Acquisition of images*
2. *Get images*
3. *Image Preprocessing*
4. *Preprocessing Images - Enlargement - Normalize - Perform any other transformations*
5. *Data Preparation*
6. *Create the training, validation, and test datasets - Split Original Dataset - Augment training data if needed*
7. *Deep Learning Model Training*

8. Model init

9. For every training epoch:

10. Forward pass on training data

11. Compute Loss

12. Backpropagate and update model weights

13. Model evaluation on validation data

14. Model Evaluation

15. Evaluate model performance on test dataset - Calculate accuracy, F1-score, etc.

16. Seed Classification

17. Use a trained model to classify seed images - Passing images through the model - Obtain classification results

18. #Output Results

***19.** Display classification results*

4.3.5. CLASSIFICATION

The InceptionV3-based deep learning model offers a highly accurate approach for multi-class seed quality classification, effectively distinguishing between various categories such as 'Broken', 'Immature', 'Intact', 'Skin-Damaged', and 'Spotted' seeds. By utilizing DenseNet121 as a feature extractor and feeding these rich embeddings into a finely tuned InceptionV3 classifier, the model achieves superior generalization while maintaining robustness across diverse seed image samples. This combination benefits from DenseNet's compact and informative representations and InceptionV3's depth and efficiency in handling complex patterns. The model architecture incorporates Batch Normalization and Dropout layers, improving training stability and reducing overfitting. During training, callbacks like EarlyStopping and ModelCheckpoint help retain the best-performing version based on validation accuracy. The trained model achieves high precision and recall across all classes, as indicated by the classification report, making it a valuable component of seed quality assessment systems that require accurate, real-time performance under varying image conditions. By effectively capturing subtle visual cues, the InceptionV3 model supports intelligent agricultural decision-making processes within smart farming frameworks.

Pseudocode

BEGIN

1. INITIAL SETUP

- Define CLASS_NAMES = ['Broken', 'Immature', 'Intact', 'Skin-Damaged', 'Spotted']

- Set IMAGE_SIZE = (224, 224)

- Define UPLOAD_FOLDER for saving uploaded images

- Load pretrained DenseNet121 model (excluding top layers) with global average pooling

2. FEATURE EXTRACTION

FUNCTION extract_features(image_path):

- Load image and resize to IMAGE_SIZE

- Convert image to array and expand dimensions

- Preprocess image using preprocess_input (for DenseNet)

- Use DenseNet121 to extract features

- RETURN extracted features

3. BUILD INCEPTIONV3 CLASSIFIER

- Initialize a Sequential model

- Add Dense layer (1024 units, ReLU) with input shape matching DenseNet output

- Add BatchNormalization and Dropout(0.5)

- Add Dense layer (512 units, ReLU) + Dropout(0.5)

- Add Dense layer (256 units, ReLU) + Dropout(0.5)

- Add Dense output layer (5 units, Softmax activation)

- Compile the model with:

optimizer = 'adam'

loss = 'categorical_crossentropy'

metrics = ['accuracy']

4. TRAINING PHASE

- Load precomputed DenseNet features and labels: (train_features, train_labels, test_features, test_labels)

- Define callbacks:

- EarlyStopping (monitor='val_accuracy', patience=10)

- ModelCheckpoint (save best model)

- Train model on training data with validation data

- Save best model as 'inception_v3.keras'

5. PREDICTION PHASE

FUNCTION predict_seed_class(image_path):

- Extract features using extract_features(image_path)

- Load trained classifier (inception_v3.keras)

- Predict class probabilities

- Identify class with highest probability

- RETURN predicted class, confidence, and all class probabilities

6. API ENDPOINTS (Flask)

- '/' : Renders homepage (upload interface)

- '/predict' : Accepts image upload, calls predict_seed_class, returns JSON response

END

5. TESTING

The aim of the testing process is to identify all defects in a software product. It is not possible to guarantee that the software is error free. This is because of the fact that the input data domain of most software projects is very large. Testing provides a method for reducing defects in a system and enhancing the user's confidence in the developed system.

5.1 Test cases

Testing the input by checking whether the input data i.e. taking input from user. Checking whether operation has done successfully, checking performance, and user accessing has done by using manual testing. The table from 5.1 to 5.5 shows the testcases written on proposed system.

Table 5.1 Test Case-1

Test Case ID	TC01
Test Case Name	Data pre-processing
Test Case Description	Pre-processing of real world data
Input Parameters	Sample Image of the Seed
Requirements Required	VSCode
Expected Results	Pre-processed data
Actual Results	Pre-processed data
Test Result	Pass
Remarks	No Remarks

Table 5.2 Test Case-2

Test Case ID	TC02
Test Case Name	Data Normalization
Test Case Description	Scales the data between min and max
Input Parameters	Pre-Processed attributes
Requirements Required	VSCode
Expected Results	Normalized data
Actual Results	Normalized data
Test Result	Pass
Remarks	No Remarks

Table 5.3 Test Case-3

Test Case ID	TC03
Test Case Name	Dimensionality Reduction
Test Case Description	Selecting and extracting most important features into a numpy array.
Input Parameters	Normalized attributes
Requirements Required	VSCode
Expected Results	Dimensionality reduced dataset
Actual Results	Dimensionality reduced dataset
Test Result	Pass
Remarks	No Remarks

Table 5.4 Test Case-4

Test Case ID	TC04
Test Case Name	Classification
Test Case Description	Classification using classifiers
Input Parameters	Dimensionality reduced dataset
Requirements Required	VSCode
Expected Results	Accuracy obtained from classifiers
Actual Results	Accuracy obtained from classifiers
Test Result	Pass
Remarks	No Remarks

Table 5.5 Test Case-5

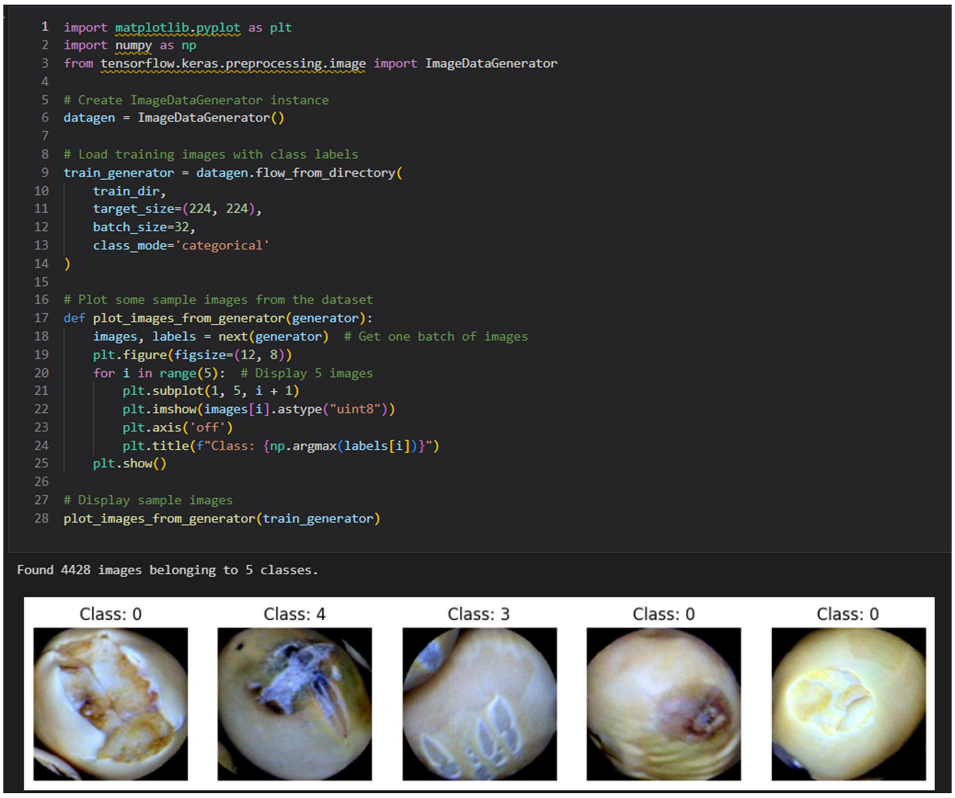
Test Case ID	TC05
Test Case Name	Detection Model
Test Case Description	Seed Classification
Input Parameters	Result obtained from classifiers
Requirements Required	VSCode
Expected Results	Classifies the Seed with respect to their class
Actual Results	Classifies the Seed with respect to their class

Test Result	Pass
Remarks	No Remarks

6. RESULT

Soybean Seed Classification Dataset

The dataset consists of 5,513 single soybean seed images across five categories: Intact, Spotted, Immature, Broken, and Skin-Damaged, with over 1,000 images in each class. All images are standardized to 227×227 pixels, having been segmented from larger images using an image processing algorithm with over 98% segmentation accuracy. The classification follows the Standard of Soybean Classification (GB1352-2009). The dataset provides a solid foundation for machine learning applications in agricultural quality control. The implemented InceptionV3-based classifier utilizes transfer learning by extracting 1024-dimensional feature vectors from DenseNet121, which are then processed through a sequential neural network with three dense layers (1024, 512, and 256 units), batch normalization, and dropout regularization. This architecture achieved strong performance across all five seed categories as evidenced by the confusion matrix, with particularly high accuracy in classifying Intact seeds.



```python
import matplotlib.pyplot as plt
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Create ImageDataGenerator instance
datagen = ImageDataGenerator()

# Load training images with class labels
train_generator = datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)

# Plot some sample images from the dataset
def plot_images_from_generator(generator):
    images, labels = next(generator)  # Get one batch of images
    plt.figure(figsize=(12, 8))
    for i in range(5):  # Display 5 images
        plt.subplot(1, 5, i + 1)
        plt.imshow(images[i].astype("uint8"))
        plt.axis('off')
        plt.title(f"Class: {np.argmax(labels[i])}")
    plt.show()

# Display sample images
plot_images_from_generator(train_generator)
```


Fig 6.1 Sample Images from the dataset retrieved randomly.

In this model, feature extraction is performed using the DenseNet121 architecture, which is pretrained on ImageNet. The top classification layers are removed, and only the convolutional base is used to extract deep, high-level features from input seed images. These features capture intricate patterns such as texture, shape, and color variations. The resulting 1024-dimensional feature vectors serve as rich input to the InceptionV3-based classifier, enabling more accurate and robust classification.

```python
# Display the first feature vector
print("First 10 extracted feature values for the first sample in the training set:")
print(densenet_train_features[0][:10])  # Display the first 10 values

print("\nShape of feature vector:", densenet_train_features[0].shape)

# Display multiple feature vectors
print("\nFirst 5 feature vectors with first 5 values:")
for i in range(5):
    print(f"Sample {i + 1}:", densenet_train_features[i][:5])

```

```
First 10 extracted feature values for the first sample in the training set:
[5.9016613e-05 5.5387835e-03 3.8996928e-03 2.4631540e-03 6.4646542e-02
 1.0872296e+00 1.3914286e-03 2.1080947e-03 1.3035554e-01 3.9736979e-04]

Shape of feature vector: (1024,)

First 5 feature vectors with first 5 values:
Sample 1: [5.9016613e-05 5.5387835e-03 3.8996928e-03 2.4631540e-03 6.4646542e-02]
Sample 2: [0.00015548 0.00486723 0.00626041 0.00155096 0.05498223]
Sample 3: [0.0001007  0.00571491 0.00519302 0.00214454 0.07052059]
Sample 4: [0.00022117 0.00697958 0.00369843 0.00204438 0.05035614]
Sample 5: [0.0001594  0.00452116 0.0031273  0.00156499 0.05542429]
```

Fig 6.2 Feature Extracts from the Images

The function display_extracted_features is used to **visualize the extracted feature vectors** from the DenseNet model. It selects the first few feature vectors (default is 5) and normalizes them to a 0–1 range for clearer visualization. Each normalized vector is reshaped into a square-like grid and displayed as a grayscale image using Matplotlib. This helps in understanding how the model perceives and encodes patterns from the input seed images into numerical representations used for classification.

```python
def display_extracted_features(features, num_features=5):
    plt.figure(figsize=(15, 5))

    # Display the first few feature maps
    for i in range(num_features):
        feature_map = features[i]

        # Normalize feature map for better visualization
        feature_map_min = feature_map.min()
        feature_map_max = feature_map.max()

        normalized_feature_map = (feature_map - feature_map_min) / (feature_map_max - feature_map_min)

        plt.subplot(1, num_features, i + 1)
        plt.imshow(normalized_feature_map.reshape((int(normalized_feature_map.shape[0] ** 0.5), -1)), cmap='gray')
        plt.axis('off')
        plt.title(f"Feature {i + 1}")
    plt.show()

# Display sample extracted features
display_extracted_features(densenet_train_features)
```

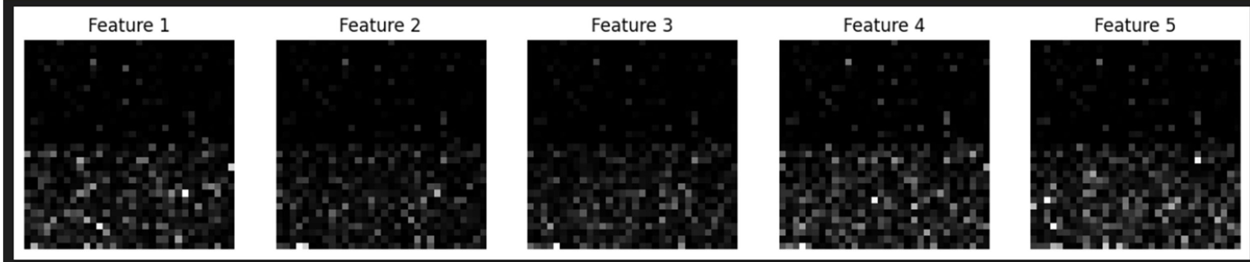


Fig 6.3. Features of the random Images in Grayscale.

```python
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

# Generate confusion matrix for InceptionV3 predictions
conf_matrix_inception = confusion_matrix(test_labels.argmax(axis=1), inception_pred_labels)

# Define class names (modify based on your dataset)
class_names = ['Broken', 'Immature', 'Intact', 'Skin-Damaged', 'Spotted']

# Visualize the confusion matrix
disp_inception = ConfusionMatrixDisplay(confusion_matrix=conf_matrix_inception, display_labels=class_names)
disp_inception.plot(cmap=plt.cm.Blues, xticks_rotation='vertical')

# Add title and labels
plt.title("Confusion Matrix for InceptionV3")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")

# Show the plot
plt.show()
```

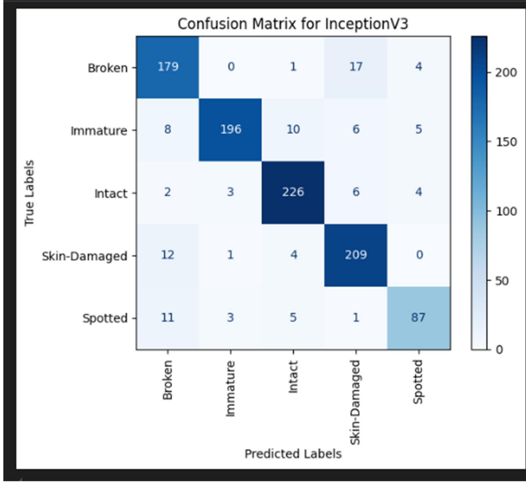


Fig 6.4. Confusion Matrix

The confusion matrix illustrates the classification performance of the InceptionV3 model on maize seed images across five categories: Broken, Immature, Intact, Skin-Damaged, and Spotted. Each row represents the actual class, while each column shows the predicted class, with the diagonal elements indicating correct predictions. For instance, the model accurately classified 179 Broken seeds, 196 Immature seeds, and 226 Intact seeds. Misclassifications are also visible—for example, 17 Broken seeds were wrongly predicted as Skin-Damaged. Overall, this matrix provides a comprehensive view of how well the model distinguishes between different seed types and highlights areas where the model may need improvement

CNN
• Accuracy: 81%
• Precision: 80%
• Recall: 80%
• F1-Score: 80%

DenseNet
• Accuracy: 88%
• Precision: 89%
• Recall: 88%
• F1-Score: 88%

EfficientNet
• Accuracy: 86%
• Precision: 87%
• Recall: 86%
• F1-Score: 89%

InceptionV3
• Accuracy: 92%
• Precision: 91%
• Recall: 90%
• F1-Score: 90%

MobileNetV2
• Accuracy: 89%
• Precision: 89%
• Recall: 88%
• F1-Score: 88%

NASNet
• Accuracy: 89%
• Precision: 88%
• Recall: 88%
• F1-Score: 88%

ResNet
• Accuracy: 88%
• Precision: 89%

• Recall: 88%
• F1-Score: 88%

VGG19
• Accuracy: 89%
• Precision: 89%
• Recall: 89%
• F1-Score: 89%

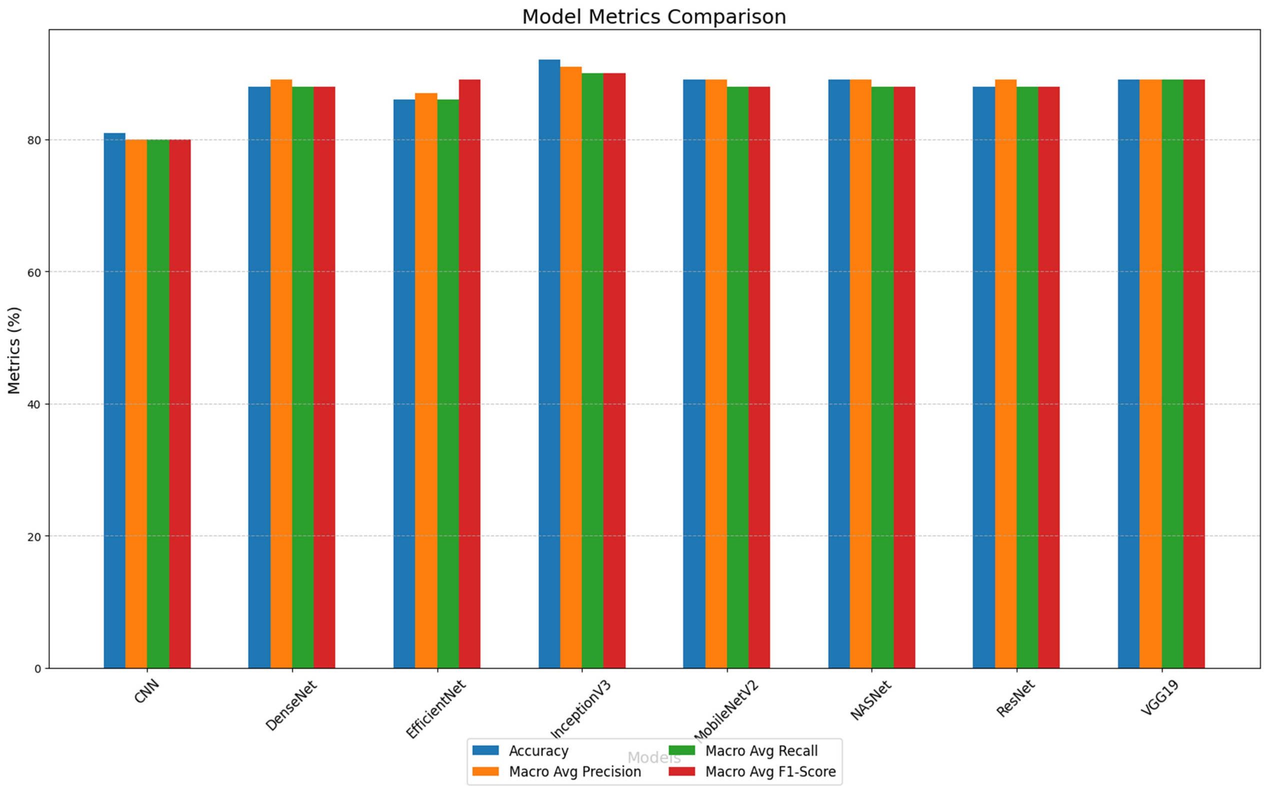


Fig 6.5. Performance Comparison Models

The bar chart presents a Performance Comparison of Deep Learning Models based on four key evaluation metrics: Accuracy, Precision, Recall, and F1-Score. The models compared include CNN, DenseNet , EfficientNet , InceptionV3, MobileNetV2, NASNet, ResNet, and VGG19. Among these, InceptionV3 outperformed others with the highest overall scores across all metrics, particularly achieving 92% accuracy and 90% F1-score, indicating its robustness in classification tasks. Models like DenseNet, MobileNetV2, and VGG19 also showed consistently high performance, making them strong alternatives depending on the complexity and resource requirements of the application.

7. CONCLUSION AND FUTURE WORK

Future research can be conducted on more complex machine learning methods, such as ensemble methods, or even more complicated processes of image processing, for further improvement in classification effectiveness. Moreover, the approach can be taken to other crop species, further increasing relevance and influence across the whole agriculture sector. In such cases, the IoT sensors would be connected with some kind of a machine learning algorithm that may process real-time data coming from the soil. Increasing productivity of agricultural practices is found along with agriculture health, climate, and crop growth. Creating user-friendly applications will allow farmers, particularly in developing regions, to apply machine learning methods for seed quality assessment. Collaboration with agronomists can improve models by infusing expert knowledge in feature selection and training, therefore improving the relevance of these features in real-world contexts. The continuous development of such technology will be critical to solving these issues of climate change and resource scarcity that plague modern agriculture, opening its doors to far more resilient and efficient farming systems. On a larger scale, this piece further inks the revolutionary potentiality of machine learning in agriculture, underlining incessant innovation and adaptation of techniques which look for increased food production to be aligned with sustainability within future generations.

REFERENCES

[1] O. Erenstein, J. Chamberlin, and K. Sonder, "Estimating the global number and distribution of maize and wheat farms," Global Food Security, vol. 30, Sep. 2021, Art. no. 100558.

[2] G. Mariammal, A. Suruliandi, S. P. Raja, and E. Poongothai, "Prediction of land suitability for crop cultivation based on soil and environmental characteristics using modified recursive feature elimination technique with various classifiers," IEEE Trans. Comput. Social Syst., vol. 8, no. 5, pp. 1132–1142, 2021.

[3] A. D. de Medeiros et al., "Machine learning for seed quality classification: An advanced approach using merger data from FT-NIR spectroscopy and x-ray imaging," Sensors, vol. 20, no. 15, Aug. 2020, doi: 10.3390/s20154319.

[4] T. U. Rehman, M. S. Mahmud, Y. K. Chang, J. Jin, and J. Shin, "Current and future applications of statistical machine learning algorithms for agricultural machine vision systems," Comput. Electron. Agricult., vol. 156, pp. 585–605, Jan. 2019.

[5] K. Kiratiratanapruk and W. Sinthupinyo, "Color and texture for corn seed classification by machine vision," in Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS), Dec. 2011, pp. 1–5.

[6] S. H. Ahammad et al., "Advanced model-based machine learning technique for early stage prediction of ankylosing spondylitis under timely analysis with featured textures," Multimedia Tools Appl., vol. 83, no. 26, pp. 68393–68413, 2024.

[7] J. V. Rao, P. Sarath, P. Prathyusha, O. Karthik, and S. Aastha, "Data augmentation and hyperparameter tuning boost food image recognition," in Proc. IEEE Int. Conf. Inf. Technol., Electron. Commun. Syst. (ICITEICS), Bangalore, India, 2024, pp. 1–5, doi: 10.1109/ICITEICS61368.2024.10625651.

[8] R. Swathi, K. Suneetha, I. Suneetha, B. V. Suresh Reddy, and S. K. Depuru, "Image capturing and deleting duplicate images through feature extraction using hashing techniques," Int. J. Eng. Trends Technol., vol. 72, no. 1, pp. 64–70, 2024, doi: 10.14445/22315381/IJETT-V7211P107.

[9] N. V. R. Goluguri and K. Suganya Devi, “Infectious diseases of rice plants classified using a deep learning-powered least squares support vector machine model,” Indian J. Comput. Sci. Eng., vol. 13, no. 5, pp. 1640–1659, 2022.

[10] S. Harshitha, P. Abhiram, S. Susanna, S. Vyshnavi, and B. C. Babu, “Analysing of image quality computation models through convolutional neural network,” in Proc. 2nd Int. Conf. Vision Towards Emerging Trends Commun. Netw. Technol. (VITECON), Vellore, India, 2023, pp. 1–7, doi: 10.1109/VITECON58111.2023.10157161.

APPENDIX

```
from tensorflow.keras.applications import DenseNet121
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, BatchNormalization
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import joblib
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
# Set paths
train_dir = 'E:/finaltrail/train'
test_dir = 'E:/finaltrail/test'
# Load pre-trained DenseNet121 model without the top layer (used for feature extraction)
densenet_model = DenseNet121(include_top=False, input_shape=(224, 224, 3), pooling='avg')

# Extract features function
def extract_features_with_model(model, directory, sample_count, batch_size=32):
    datagen = ImageDataGenerator(rescale=1./255)
    generator = datagen.flow_from_directory(
        directory,
        target_size=(224, 224),
        batch_size=batch_size,
        class_mode='categorical',
```

```
        shuffle=False
    )

    features = []
    labels = []
    for inputs_batch, labels_batch in generator:
        features_batch = model.predict(inputs_batch)
        features.append(features_batch)
        labels.append(labels_batch)
        if len(features) * batch_size >= sample_count:
            break

    features = np.vstack(features)[:sample_count]
    labels = np.vstack(labels)[:sample_count]
    return features, labels

# Extract features using DenseNet121
densenet_train_features, train_labels = extract_features_with_model(densenet_model, train_dir,
4000)
densenet_test_features, test_labels = extract_features_with_model(densenet_model, test_dir,
1000)

# Save extracted features
joblib.dump((densenet_train_features, train_labels, densenet_test_features, test_labels),
'densenet_features.joblib')
```

```
# Load features
densenet_train_features, train_labels, densenet_test_features, test_labels =
joblib.load('densenet_features.joblib')

# Define DenseNet121-based model with additional layers
densenet_classifier = Sequential([
    Dense(1024, activation='relu', input_shape=(1024,)),
    BatchNormalization(),
    Dropout(0.5),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(256, activation='relu'),
    Dense(5, activation='softmax')
])

densenet_classifier.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
# Callbacks for early stopping and model saving
densenet_checkpoint = ModelCheckpoint('densenet_best_model.keras', save_best_only=True,
monitor='val_accuracy', mode='max')
early_stop = EarlyStopping(monitor='val_accuracy', patience=10, mode='max',
restore_best_weights=True)

# Train the DenseNet model
```

```
densenet_history = densenet_classifier.fit(
    densenet_train_features, train_labels,
    epochs=50,
    validation_data=(densenet_test_features, test_labels),
    callbacks=[densenet_checkpoint, early_stop]
)
# Load extracted DenseNet features
densenet_train_features, train_labels, densenet_test_features, test_labels =
joblib.load('densenet_features.joblib')

# InceptionV3-based Model Architecture
inception_classifier = Sequential([
    Dense(1024, activation='relu', input_shape=(1024,)),  # Input shape matches DenseNet feature
size
    BatchNormalization(),
    Dropout(0.5),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(5, activation='softmax')  # Number of output classes
])

inception_classifier.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
```

```
# Callbacks for early stopping and model saving
inception_checkpoint = ModelCheckpoint('inception_v3.keras', save_best_only=True,
monitor='val_accuracy', mode='max')
early_stop = EarlyStopping(monitor='val_accuracy', patience=10, mode='max',
restore_best_weights=True)


# Train the InceptionV3 model using DenseNet features
inception_history = inception_classifier.fit(
    densenet_train_features, train_labels,
    epochs=50,
    validation_data=(densenet_test_features, test_labels),
    callbacks=[inception_checkpoint, early_stop]
)


# Plot InceptionV3 accuracy graphs
plt.figure(figsize=(10, 6))  # Larger figure size for better readability
plt.plot(inception_history.history['accuracy'], label='Train Accuracy', color='blue', linestyle='-',
marker='o')
plt.plot(inception_history.history['val_accuracy'], label='Validation Accuracy', color='green',
linestyle='--', marker='x')
plt.title('InceptionV3 Training and Validation Accuracy', fontsize=16)
plt.xlabel('Epochs', fontsize=14)
plt.ylabel('Accuracy', fontsize=14)
plt.legend(fontsize=12)
```

```
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()


# Classification Report for InceptionV3
inception_predictions = inception_classifier.predict(densenet_test_features)
inception_pred_labels = inception_predictions.argmax(axis=1)
print("InceptionV3 Classification Report:")
print(classification_report(test_labels.argmax(axis=1), inception_pred_labels,
target_names=['Broken', 'Immature', 'Intact', 'Skin-Damaged', 'Spotted']))
```

CERTIFICATES OF CONFERENCE

A Research paper entitled “A Novel Approach To Enhance Crop Yield Using Seed Quality Analysis And Machine Learning”authored by Ch Pratima, Nagella Naga Harika , Shaik Abdul Razaq, Pallam Gurunath, Mandava Shashank is accepted for presentation in the “3rd International Conference on Data Analytics, Smart Computing and Networks (IDASCN - 2024)” Organised by Department of Data Science & Computer Applications, Mohan Babu University (Erstwhile Sree Vidyanikethan Engineering College (Autonomous)) Tirupati-517102 during 24 - 26 October, 2024. The paper will be further indexed by Scopus database.

AIP Publishing

3rd International Conference on DATA ANALYTICS, SMART COMPUTING AND NETWORKS

IDASCN'2024

MBU MOHAN BABU UNIVERSITY

Organized by Department of Computer Applications/Data Science

⋆ Certificate of Appreciation ⋆

This is to certify that Mr./Ms./Dr. Naga Harika Nagella has participated and presented a Paper on *A Novel Approach To Enhance Crop Yield With Seed Quality Analysis And Machine Learning Approaches* in 3rd International Conference on "Data Analytics, Smart Computing and Networks (IDASCN-2024)" during 24th–26th October, 2024 organized by Department of Computer Applications/Data Science, Mohan Babu University, Tirupati-517 102, Andhra Pradesh, India.

Dr. B. M. Satish
Dean, School of Engineering and Technology

K. Ram
Dr. K. Ramani
Convener

AIP Publishing

3rd International Conference
on
DATA ANALYTICS, SMART COMPUTING AND NETWORKS
IDASCN'2024

MBU MOHAN BABU UNIVERSITY
DREAM. BELIEVE. ACHIEVE

Organized by
Department of
Computer Applications/Data Science

⋆ Certificate of Appreciation ⋆

This is to certify that **Mr./Ms./Dr.** ....**P Gurunath**........ has participated and presented a Paper on ***A Novel Approach To Enhance Crop Yield With Seed Quality Analysis And Machine Learning Approaches***........ in **3rd International Conference on "Data Analytics, Smart Computing and Networks (IDASCN-2024)"** during **24th –26th October, 2024** organized by Department of Computer Applications/Data Science, Mohan Babu University, Tirupati-517 102, Andhra Pradesh, India.

Dr. B. M. Satish
Dean, School of Engineering and Technology

Dr. K. Ramani
Convener

AIP Publishing

3rd International Conference
on
DATA ANALYTICS, SMART COMPUTING AND NETWORKS
IDASCN'2024

MBU MOHAN BABU UNIVERSITY
DREAM. BELIEVE. ACHIEVE

Organized by
Department of
Computer Applications/Data Science

⋆ Certificate of Appreciation ⋆

This is to certify that **Mr./Ms./Dr.** ....**M. Sashank**........ has participated and presented a Paper on ***A Novel Approach To Enhance Crop Yield With Seed Quality Analysis And Machine Learning Approaches***........ in **3rd International Conference on "Data Analytics, Smart Computing and Networks (IDASCN-2024)"** during **24th–26th October, 2024** organized by Department of Computer Applications/Data Science, Mohan Babu University, Tirupati-517 102, Andhra Pradesh, India.

Dr. B. M. Satish
Dean, School of Engineering and Technology

Dr. K. Ramani
Convener

AIP Publishing

3rd International Conference on

DATA ANALYTICS, SMART COMPUTING AND NETWORKS

IDASCN'2024

MBU MOHAN BABU UNIVERSITY

DREAM. BELIEVE. ACHIEVE

Organized by
Department of
Computer Applications/Data Science

★ Certificate of Appreciation ★

This is to certify that **Mr./Ms./Dr.** Shaik Abdul Razaq has participated and presented a Paper on ***A Novel Approach To Enhance Crop Yield With Seed Quality Analysis And Machine Learning Approaches*** in **3rd International Conference on "Data Analytics, Smart Computing and Networks (IDASCN-2024)"** during **24th–26th October, 2024** organized by Department of Computer Applications/Data Science, Mohan Babu University, Tirupati-517 102, Andhra Pradesh, India.

Dr. B. M. Satish
Dean, School of Engineering and Technology

K. Ram
Dr. K. Ramani
Convener

COLLEGE VISION & MISSION

VISION

To be one of the Nation's premier Engineering Colleges by achieving the highest order of excellence in Teaching and Research.

MISSION

Through multidimensional excellence, we value intellectual curiosity, pursuit of knowledge building and dissemination, academic freedom and integrity to enable the students to realize their potential. We promote technical mastery of Progressive Technologies, understanding their ramifications in the future society and nurture the next generation of skilled professionals to compete in an increasingly complex world, which requires practical and critical understanding of all aspects.

DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS ENGINEERING

VISION & MISSION

VISION

To become a centre of excellence in Computer Sciences and Systems Engineering through teaching, training, research and innovation to create quality engineering professionals who can solve the growing complex problems of the society.

MISSION

- Established with the cause of development of technical education in advanced computer sciences and engineering with applications to systems there by serving the society and nation.
- Transfer of Knowledge through contemporary curriculum and fostering faculty and student development.
- Create keen interest for research and innovation among students and faculty by understanding the needs of the society and industry.
- Skill development among diversity of students in technical domains and profession for development of systems and processes to meet the demands of the industry and research.
- Imbibing values and ethics in students for prospective and promising engineering profession and develop a sense of respect for all.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

After few years of graduation, the graduates of B. Tech (CSSE) will:

1. Demonstrate competencies in the Computer Science domain and Management with an ability to comprehend, analyze, design and create software systems for pursuing advanced studies in the areas of interest.
2. Evolve as entrepreneurs or be employed by acquiring required skill sets for developing computer systems and solutions in multi-disciplinary areas.
3. Exhibit progression and professional skill development in Computer programming and systems development with ethical attitude through life-long learning.

PROGRAM SPECIFIC OUTCOMES (PSOs)

On successful completion of the Program, the graduates of B. Tech (CSSE) program will be able to:

PSO1 Employ Systems Approach to model the solutions for real life problems, design and develop software systems by applying Modern Tools.

PSO2 Develop solutions using novel algorithms in High Performance Computing and Data Science.

PSO3 Use emerging technologies for providing security and privacy to design, deploy and manage network systems.

PROGRAM OUTCOMES (POs)

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (**Engineering knowledge**).

2. Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (**Problem analysis**).

3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations (**Design/development of solutions**).

4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (**Conduct investigations of complex problems**).

5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations (**Modern tool usage**).

6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice (**The engineer and society**).

7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development (**Environment and sustainability**).

8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (**Ethics**).

9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (**Individual and team work**).

10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write

effective reports and design documentation, make effective presentations, and give and receive clear instructions (**Communication**).

11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments (**Project management and finance**).

12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change (**Life-long learning**).

COURSE OUTCOMES (COs)

After successful completion of this course, the students will be able to:

CO1. Create/Design algorithms and software to solve complex Computer Science and allied problems using appropriate tools and techniques following relevant standards, codes, policies, regulations and latest developments.

CO2. Consider Society, Health, Safety, Environment, Sustainability, Economics and project management in solving complex Computer Science and allied problems.

CO3. Perform individually or in a team besides communicating effectively in written, oral and graphical forms on Computer Science and allied systems or processes.

Mapping of Course Outcomes with POs and PSOs:

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	-	-	3	-	-	-	3	3	3	3
CO2	-	-	-	-	-	3	3	-	-	-	3	-	3	3	3
CO3	-	-	-	-	-	-	-	-	3	3	-	-	3	3	3
Average	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Level of correlation of the course	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3