Rice Seed Quality Detection

A Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfilment for the Award of

Bachelor of Technology in *Computer Science Engineering*

Under the Supervision of

Dr. Santosh Varshney

Submitted by:

Janhavi Thakur(0827CS181096)

Joy Singh(0827CS181100)

Naman Sukhwani(0827CS181126)

Acropolis Institute of Technology & Research, Indore

July - Dec 2021

EXAMINER APPROVAL

The Project entitled "**Rice Seed Quality Detection**" submitted by Janhavi Thakur (0827CS181096), Joy Singh (0827CS181100) and Naman Sukhwani (0827CS181126) has been examined and is hereby approved towards partial fulfilment for the award of Bachelor of Technology degree in Computer Science discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner)
Date:

(External Examiner)
Date:

GUIDE RECOMMENDATION

This is to certify that the work embodied in this project entitled "**Rice Seed Quality Detection**" submitted by Janhavi Thakur (0827CS181096), Joy Singh (0827CS181100) and Naman Sukhwani (0827CS181126) is a satisfactory account of the bonafide work done under the supervision of **Dr. Santosh Varshney** is recommended towards partial fulfilment for the award of the Bachelor of Technology (Computer Science) degree by Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal.

(Project Guide)
Dr. Santosh Varshney

(Project Coordinator)
Prof. Praveen Bhanodia

STUDENTS UNDERTAKING

This is to certify that a project entitled "**Rice Seed Quality Detection**" has been developed by us under the supervision of **Dr. Santosh Varshney**. The whole responsibility of work done in this project is ours. The sole intention of this work is only for practical learning and research.

We further declare that to the best of our knowledge; this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work is found then we are liable for explanation to this.

Janhavi Thakur (0827CS181096)

Joy Singh (0827CS181100)

Naman Sukhwani (0827CS181126)

Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are a number of people without whom this project's work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Dr. Santosh Varshney**, AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks **to Dr Kamal Kumar Sethi**, HOD CS, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CS Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if we do not acknowledge the support and guidance received from **Dr S C Sharma**, Director, AITR, Indore whenever needed. We take the opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for the project to achieve our objectives.

We are grateful to our parents and family members who have always loved and supported us unconditionally. To all of them, we want to say “Thank you”, for being the best family that one could ever have and without whom none of this would have been possible.

Janhavi Thakur (0827CS181096), Joy Singh (0827CS181100), and Naman Sukhwani (0827CS181126)

Executive Summary

Rice Seed Quality Detection

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (MP), India for partial fulfilment of Bachelor of Technology in Computer Science branch under the sagacious guidance and vigilant supervision of **Dr. Santosh Varshney**.

The project is based on Deep Learning, which is a sub field of machine learning, concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. In the project, TensorFlow is used, which is an open-source software library created by Google for machine learning applications. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. It provides a user-friendly interface for seed quality detection and will also provide graphical representation of seeds based on their quality. This tool will also collect the seed quality data and show the graphical representation for the data according to the seed, quality and various other aspects and will analyse it. Open-source API that can be used by other e-commerce platforms for seed quality detection in real-time, so that the consumers are satisfied with the seed quality while purchasing online.

Key-Words: Quality factor of Seed, Convolutional neural network, Computer vision, Deep Learning

“Where the vision is one year, cultivate flowers;
Where the vision is ten years, cultivate trees;
Where the vision is eternity, cultivate people.”

- Oriental Saying

Table of Contents

Chapter 1. Introduction

Introduction

Every grower at some point has observed the effects of poor seed quality: slow germination, damping-off, poor standards, and mixed or genetically contaminated lots. Determining the seed quality, that is, their suitability for planting is very important for good quality crop.

Seeds are quite small in sizes and are difficult to differentiate based on their colour and size. Human testing procedures have their limitations, like most of these are time consuming, labour intensive and sometimes the results are not reproducible in actual field conditions. The seeds we intake in daily life is important part of the diet and we must ensure the quality of seeds to ensure the nutrient ratio. This is a major issue while we buy them from online stores, and this may also affect our health also. Using this tool, a good quality of seed can be determined easily.

1.1 Overview

Traditional seed classification is costly, time-consuming and requires sophisticated human expertise. Besides, computer vision based methods are still based on predefined morphological features that are often not transferable across different types of grains. In this paper, the feasibility of automated feature extraction for rice grain purity analysis has been demonstrated using a Convolutional Neural Network (CNN) based deep learning approach. Due to the lack of benchmark datasets, the paper defines a dataset with technician-verified, labelled images of different types of grains with a background of uniform illumination. Moreover, the application also proposes the architecture of a CNN for automated grain feature extraction. The performance of a classifier trained on these features is compared to classifiers trained on morphological features used by modern computer vision approaches. It is found that in this dataset, the proposed method can detect the presence of native and foreign grains in a given sample of rice grains with superior accuracy which is at least 25% better in case of a multiclass classification scenario.

The project is based on Image analysis is a state-of-the-art technique for seed quality testing. This tool provides vast usage in evaluation of various physiological and morphological characteristics of the seed with a more comprehensive

perception. It is based on the extraction of numerical data from a captured image for characteristics like colour, size, shape of seed and seedlings and their subsequent processing with the help of suitable computer software.

Assuring the rice seed quality is the biggest and difficult test for the massive rice exporting countries. India, Thailand, US and Vietnam are the topmost countries which exporting rice. Rice seeds purity decides the yield of the grains, inferiority seeds not only affects yield its reduce the market rate of the product. So the quality of the rice is the basic test for all the rice makers which lies in selecting the good seeds. When we are doing it manually it takes lots of time and there is no guarantee for the accuracy because manual testing is like what they feel and see. I n other way when considering the physical characteristics of rice seed like shape, size, colour, length and width etc, we can use computer systems to analyse the quality of the seed.

Seed is the establishment of any rice edit. It must be developed, collected, and prepared accurately for best yield and quality outcomes.

➔ **Factors to classify seed:**

- Varietal purity
- Seed Viability
- Moisture content

➔ **Varietal purity depends upon the following conditions:**

- Other mixed varieties
- Red rice seeds
- Germination capacity of the seed

Germination %=(No. of Seeds germinated)/ (No. of seeds taken for the sample)

- Weed seeds and other crop seeds

Weed %=(weight of weed) / (Weight of seeds taken for the sample)

- Unwanted materials like stones, dust, soil etc.

Unwanted materials %=(Weight of Unwanted materials) /(Weight of seeds taken for the sample)

- Moisture content

➔ **Seed viability calculated by the following factors:**

The suitability of seed in the field is dictated by its germination potential, force, and moisture level. Then to get the good seed is in the form of, Purchase confirmed seed that is unadulterated and named, Get rancher created great seed, select your own great seed.

➔ **Moisture content is measured by:**

The amount of water contained in the rice seed fix the moisture content of the rice seed. For good seed related to moisture content must be less than 14 percentages. By using resistance type moisture meter we can measure moisture content at the time of harvesting seeds, it gives results quickly but we can use it only for small samples. For large set of samples we need capacitive moisture meters, they are more expensive.

1.2 Purpose of the System

- Application/platform which will be able to detect the seed quality and classify them accordingly in real time.
- This tool can be used by the industries to classify seeds and their quality, this can also be used by the local/small businesses and can also be used by the ecommerce website to show the quality for the pulses they are selling online.
- This tool will also collect the seed quality data and show the graphical representation for the data according to the seed, quality and various other aspects and will analyse it.
- The objectives of the systems development and event management are:
- Collect the dataset of rice grain and pre-process it and apply machine learning algorithms. Generate seed quality data and show the graphical representation.
- Apply generated model on the rice grain and get quality and various other aspects and analyse it.

1.3 Background and Motivation

As the holy grail of computer vision research is to tell a story from a single image or a sequence of images, object detection and recognition has been studied for more than four decades. Significant efforts have been paid to develop representation schemes and algorithms aiming at recognizing generic objects in images taken under different imaging conditions (e.g., viewpoint, illumination, and occlusion).

Traditional seed classification is costly, time-consuming and requires sophisticated human expertise. Besides, computer vision based methods are still based on predefined morphological features that are often not transferable across different types of grains. In this paper, the feasibility of automated feature extraction for rice grain purity analysis has been demonstrated using a Convolutional Neural Network (CNN) based deep learning approach. Due to the

lack of benchmark datasets, the paper defines a dataset with technician-verified, labelled images of different types of grains with a background of uniform illumination.

Moreover, the project also proposes the architecture of a CNN for automated grain feature extraction. The performance of a classifier trained on these features is compared to classifiers trained on morphological features used by modern computer vision approaches.

A training and predicting pipeline are implemented to contrast performance of various popular classification algorithms and determine the best suited model.

1.4 Existing System

Title	Description	Result
Seed vigor Imaging System	Seed vigor has traditionally been evaluated by physiological, biochemical and stress tolerance tests. More recently, with the use of computerized image analysis, objective information has become accessible in a relatively short period of time, with less human interference. The aim of this study was to verify the efficiency of computerized seedling image analysis by Seed Vigor Imaging System (SVIS) to detect differences in vigor between carrot (*Daucus carota* L.) seed lots as compared to those provided by traditional vigor tests. Seeds from seven lots from the Brasilia cultivar were subjected to a germination test, first count of germination, speed of germination, accelerated aging with saline solution and seedling emergence; furthermore, a vigor index, growth index and uniformity index were determined by the Seed Vigor Imaging System (SVIS) during four evaluation periods. The results obtained by the computerized seedling analysis (vigor index and growth index) show that SVIS is efficient in assessing carrot seed vigor.	It is Old Quality detection strategy and usually available for medium amount of grains.

Matrox image processing board	Matrox Imaging smart cameras offer dust-proof, immersion-resistant, and extremely rugged construction. A choice of image sensors—combined with an efficient Intel® Atom® embedded processor—allow these model edge IoT devices to tackle a wide variety of applications using traditional machine vision as well as deep learning in tight spots and dirty industrial environments. Systems integrators, machine builders, and OEMs alike develop applications for these smart cameras by way of an intuitive flowchart-based integrated development environment (IDE), Matrox Design Assistant X.	This application only provides analysis for Lettuce and Sorgham.
ImageJ software	The focus of the paper is on the Identification and classification of seeds, grading of seeds, quality determination of seeds in seed science and food processing sectors are the essential role of these techniques. This survey provides a review of image analysis techniques and proposes a processing module for seed identification and classification. Mainly this review paper focuses on the quality control of rice which is the most used crop in the world on the basis of image processing methods.	It is Old Quality detection strategy and usually available for medium amount of grains.
Delta-T(winDIAS)	The focus of WinDIAS is optimised for the rapid analysis of area by colour difference, creating many applications in plant pathology and plant protection. Examples include: necrosis caused by fungi and bacteria, leaf tip burn and leaf spotting, nutrient deficiency symptoms, viral infection and leaf senescence.	It is Available for large amount of grains but can only used to analyse mustard and Oat.

Grain Identification and Quality Analysis using Image Processing based on Principal Component Analysis	This paper aims to Classification and quality analysis is done by comparing the sample image with database image. Canny edge detector is applied to detect the edges of rice grains. Eigen values and Eigen vectors are calculated on the basis of morphological features. Then by applying the PCA, different varieties of rice are classified by comparing the sample image with a database. Results obtained in terms of classification and quality analysis are 92.3% and 89.5% respectively. Proposed system can work well within minimum time and low cost.	In this paper aim is to analyse Flax and Lentils.

1.5 Problem Statement and Objectives:

Different types of foods are available in grain form, but rice is one of the important and most used cereal grains of Pakistan and all over the world. Quality inspection of rice grain is also important for both local as well as export purpose. It is necessary to propose an automatic solution to perform the quality analysis as well as to distinguish between different classes of rice. Seeds are quite small in sizes and are difficult to differentiate based on their colour and size.

Human testing procedures have their limitations, like most of these are time consuming, labour intensive and sometimes the results are not reproducible in actual field conditions. The seeds we intake in daily life is important part of the diet and we must ensure the quality of seeds to ensure the nutrient ratio. This is a major issue while we buy them from online stores, and this may also affect our health also. Using this tool, a good quality of seed can be determined easily.

Thus, the system implemented has the following objectives :

Objective 1: This tool can be used by the industries to classify seeds and their quality, this can also be used by the local/small businesses and can also be used by the ecommerce website to show the quality for the pulses they are selling online.

Objective 2: This tool will also collect the seed quality data and show the graphical representation for the data according to the seed, quality and various other aspects and will analyse it.

Objective 3: Application/platform which will be able to detect the seed quality and classify them accordingly in real time.

1.6 Scope of the project:

This application aims to provide a modern interface and tooling for seed quality analysis and classification. The seeds we intake in our day-to-day life are of good quality or not and have sufficient nutrients, for determining these parameters this application is made with modern techniques for classifying seed into good, medium, or bad quality.

Provides a user-friendly interface for seed quality detection and will also provide graphical representation of seeds based on their quality.

Open-source API that can be used by other e-commerce platforms for seed quality detection in real-time, so that the consumers are satisfied with the seed quality while purchasing online.

1.7 Team Organization:

Our team consist of three members-

- **Janhavi Thakur**– Contribution in the project was as a frontend developer and backend developer and along with that I also tried to understand the limitations of current system, I studied about the topic and its scope, gathered the requirements for the project. And also worked on documentation like UML, PPT, SRS and Model testing and accuracy.

- **Joy Singh**- Contribution in the project was a frontend and backend developer and along with that I learned different technologies to implement the project like TinnyML and TensorFlow. I also tried to understand the limitations of current system, I studied about the topic and its scope and worked on model implementation and documentation.

- **Naman Sukhwani** – Contribution in the project was as backend and frontend developer. While gathering the preliminary requirement and investigating with my team member, selected the right technology and architecture to implement our idea. And worked on model implementation and processes.

1.8 Report Structure:

The project **Rice Seed Quality Detection** is primarily concerned with the Seed Quality analysis in real-time and whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: System Analysis- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discuss the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 4: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2. System Analysis

System Analysis

2.1 Study of the System

A proper, precise, and modern solution for seed quality detection. An application that can provide sufficient data for analysis to classify seeds based on quality. This tool uses Image analysis that can be a significant system to monitor phases of seed quality in controlled environment and the changes associated with it can be assessed accurately, thus helps in seed viability using data analysis.

The project aim to be used by the industries to classify seeds and their quality, this can also be used by the local/small businesses and can also be used by the ecommerce website to show the quality for the pulses they are selling online.

The assessment of RGB index, size, and shape of each individual seed within a large seed sample may allow the development of non-destructive methods for sorting seed sub-samples with different quality. This tool will also collect the seed quality data and show the graphical representation for the data according to the seed, quality and various other aspects and will analyse it.

Finally, we will compare the results of proposed and baseline features with other machine learning algorithms. Findings of the analysis indicate the significance of the proposed features in Rice Seed Quality Detection.

This project we will develop using python, ML, TypeScript and NodeJS.

- First we will search and find the dataset from Kaggle or GitHub and download it to train the model.
- After downloading we will pre-process the data in which scraping the data for model creation , cleaning data, augmentation and erosion, etc.
- Then For model creation testing will be done with the help of CNN (Convolutional Neural Network), KNN and Random Forest we will train the dataset and generate models separately.
- Then we are going to develop frontend for our application using Angular.
- For the frontend purpose we will create a platform where there will be different logins for user and admin and the user will be able to detect the quality of seed grain and group of seed.
- These all-purpose we are using NodeJS and Flask as backend with the help of Express.JS framework and MongoDB is database, for training the model we are

using ML algorithms like CNN (Convolutional Neural Network), KNN(K-nearest Neighbour), Random Forest etc and for frontend TypeScript and NodeJS etc.

CNN(Convolutional Neural Network)

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

- Convolutional layer
- Pooling layer
- Fully-connected (FC) layer

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colours and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

Convolutional Layer

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

Pooling Layer

Pooling layers, also known as down sampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights.

Fully-Connected Layer

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

SYSTEM DEVELOPMENT LIFE CYCLE MODEL (SDLC MODEL) –

This is also called as Classic Life Cycle Model (or) Linear Sequential Model (or) Waterfall Method. This model has the following activities: -

1. System/Information Engineering and Modelling
2. Software Requirements Analysis
3. Systems Analysis and Design
4. Code Generation
5. Testing
6. Maintenance

1) System/Information Engineering and Modelling:

As software development is a large process, so work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. The view of this system is necessary when software must interface with other elements such as hardware, people and other resources. System is the very essential requirement for the existence of software in any entity. In some cases, for maximum output, the system should be re-engineered and spruced up. Once the ideal system is designed according to requirement, the development team studies the software requirement for the system.

2) Software Requirement Analysis:

Software Requirement Analysis is also known as feasibility study. In this requirement analysis phase, the development team visits the customer and studies their system requirement. They examine the need for possible software automation in the given software system. After feasibility study, the development team provides a document that holds the different specific recommendations for the candidate system. It also consists of personal assignment, costs of the system, project schedule and target dates.

The requirements analysis and information gathering process is intensified and focused especially on the software. To understand what type of the programs to be built, the system analyst must study the information domain for the software as well as understand requirement function, behaviour, performance and interfacing.

The main purpose of the requirement analysis phase is to find the need and to define the problem that needs to be solved.

3) System Analysis and Design:

In System Analysis and Design phase, the whole software development process, the overall software structure and its outlay are defined. In case of the client/server processing 14 technology, the number of tiers required for the package architecture, the database design, the data structure design etc. are all defined in this phase. After designing part, a software development model is created.

Analysis and Design are very important in the whole development cycle process. Any fault in the design phase could be very expensive to solve in the software development process. In this phase, the logical system of the product is developed.

4) Code Generation:

In Code generation phase, the design must be decoded into a machine-readable form. If the design of software product is done in a detailed manner, code generation can be achieved without much complication. For generation of code, programming tools like Compilers, Interpreters, and Debuggers are used. For coding purpose high level programming language php is used. The right programming language is chosen according to the type of application

5) Testing:

After code generation phase the software program testing begins. Different testing methods are available to detect the bugs that were committed during the previous phase. A number of testing tools and methods are available for testing purpose.

6) Maintenance:

Software will definitely go through change once when it is delivered to the customer. There are large numbers of reasons for the change. Change could happen due to some unpredicted input values into the system. In addition to this the changes in the system directly have an effect on the software operations. The software should be implemented to accommodate changes that could happen during the post development period.

2.2 Functional Requirements:

The major functional requirement of this project is to gather the dataset , cleaning, scraping, augmentation, erosion and pre-processing the dataset to train the model. Proper reviewing of tweets and to check proper functioning of model is important. The system should be able to analyse the quality of Seed.
We need to make sure that the functioning of the model is proper and the system should give satisfactory results.

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements: -

Following are the functional requirements on the system:

1. For detecting and analysing the quality of seed we need a proper data set which is implemented in real time.
2. A proper algorithm for analysing the data set.
3. The model should give a respective output and it should not deviate from the expected output.

2.3 Non-functional Requirements:

ACCURACY -The model should give be able to give accurate results.

RELIABILITY-The model is reliable and beneficial for the users.

MAINTAINABILITY- To maintain the model is important also to make changes when required.

Non functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy.

Non functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:-

¬ Product Requirements
¬ Basic Operational Requirements

➔ **Product Requirements:**

Platform Independency:
Standalone executables for embedded systems can be created so the algorithm developed using available products could be downloaded on the actual hardware and executed without any dependency to the development and modeling platform.

Correctness:
It followed a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.

Ease of Use:
Model Coder provides an interface which allows the user to interact in an easy manner.

Modularity:
The complete product is broken up into many modules and well-defined interfaces are developed to explore the benefit of flexibility of the product.

Robustness:
This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness. Non functional requirements are also called the qualities of a system.

These qualities can be divided into execution quality & evolution quality. Execution qualities are security & usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

Chapter 3. Implementation

Implementation

3.1 System Requirement Specification (SRS):

Software Requirement specifications (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easier comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an ecommerce website and so on) must provide, as well as states any required constraints by which the system must abide. SRS also functions as a blueprint for completing a project with as little cost growth as possible.

SRS is often referred to as the “parent” document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. Requirement is a condition or capability to which the system must conform.

Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number.

Role of SRS:

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development.

A good SRS should satisfy all the parties involved in the system.

Requirement Specifications:

The focus is on specifying what has been found giving analysis such as representation, specification languages and tools and checking the specifications are addressed during this activity. The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

3.2 Methodology:

This project we will develop using python, ML and web technology.

- First we will search and find the dataset from Kaggle or GitHub and download it to train the model.

- After downloading we will pre-process the data ,clean the data then with the help of naïve bayes, CNN (Convolutional Neural Network), KNN and Random Forest we will train the dataset and generate models separately and also we will apply different libraries like Keras, Tensorflow, MetplotLib, CV2, numpy, sklearn etc to check its accuracy.

- We will fetch the real time data and then we apply the generated model to these fetched images and check for the quality of seeds. Also we will compare different algorithms and check the quality of various seeds and select the best grain.

- For the frontend purpose we will create a platform where there will be different logins for
  user and admin and the user will be able to detect the quality of seed grain and group of seed.

- These all-purpose we are using NodeJS and Flask as backend with the help of Express.JS framework and MongoDB is database, for training the model we are using ML algorithms like CNN (Convolutional Neural Network), KNN(K-nearest Neighbour), Random Forest etc and for frontend TypeScript and NodeJS etc.

3.3 Techniques Used:

3.3.1 Deep Learning:

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms.

Learning can be supervised, semi-supervised or unsupervised. Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain.

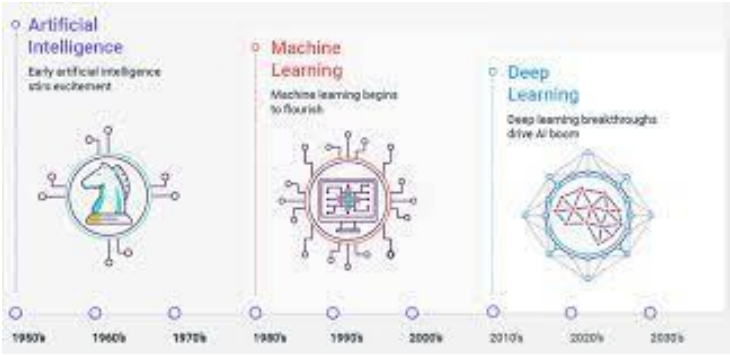


Figure 3-1: Deep Learning

Deep learning architectures such as deep neural networks, deepbelief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network

filtering, machine translation, bioinformatics and drug design, where they have produced results comparable to and in some cases superior to human experts.

3.3.2 Neural Networks :

In machine learning, a convolutional neural network (CNN, or Convolutional Neural Network) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery.

CNNs use a variation of multilayer perceptrons designed to require minimal pre-processing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared weights architecture and translation invariance characteristics.

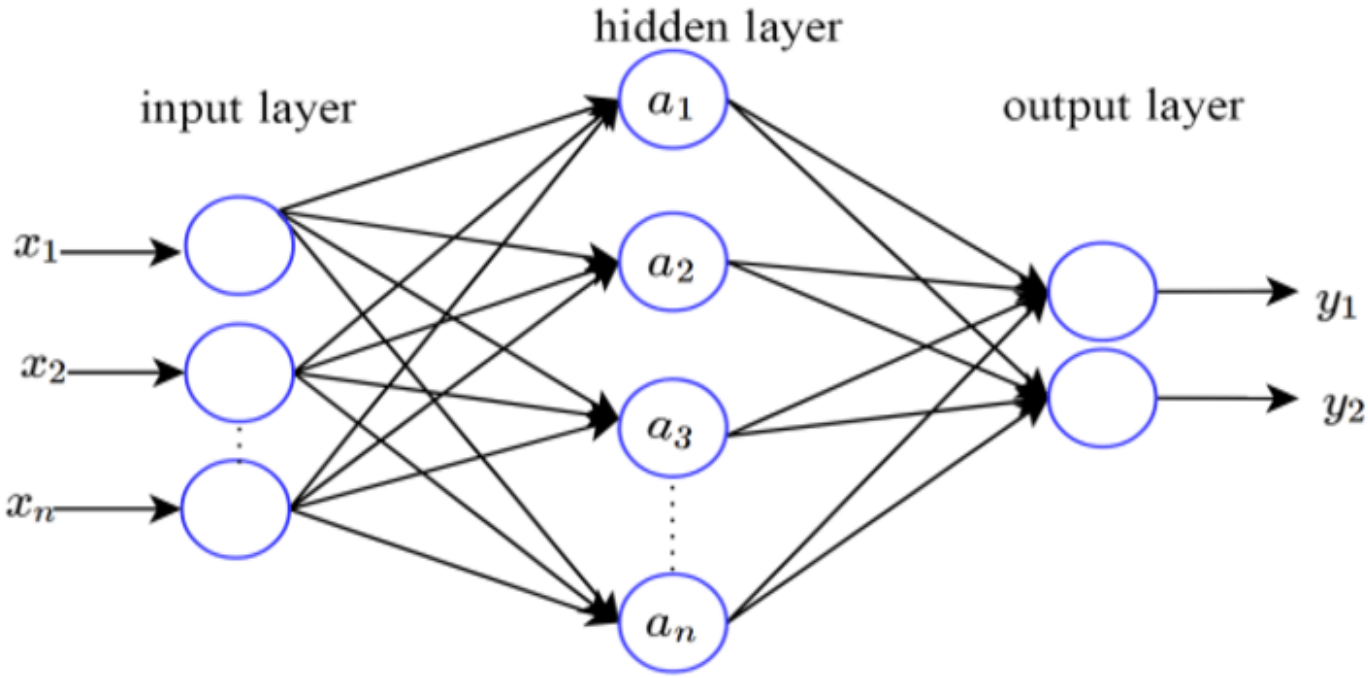


Figure 3-2: Neural Network

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in image and video recognition, recommender systems and natural language processing.

3.4 Tools Used

3.4.1 OpenCV:

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing.

Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

3.4.2 TensorFlow:

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

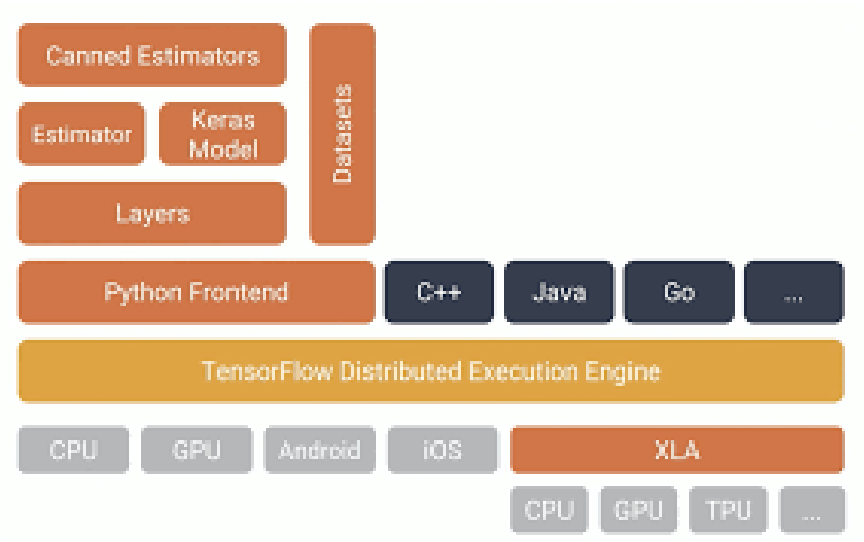


Figure 3-3: TensorFlow Architecture

TensorFlow **allows developers to create dataflow graphs**—structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor.

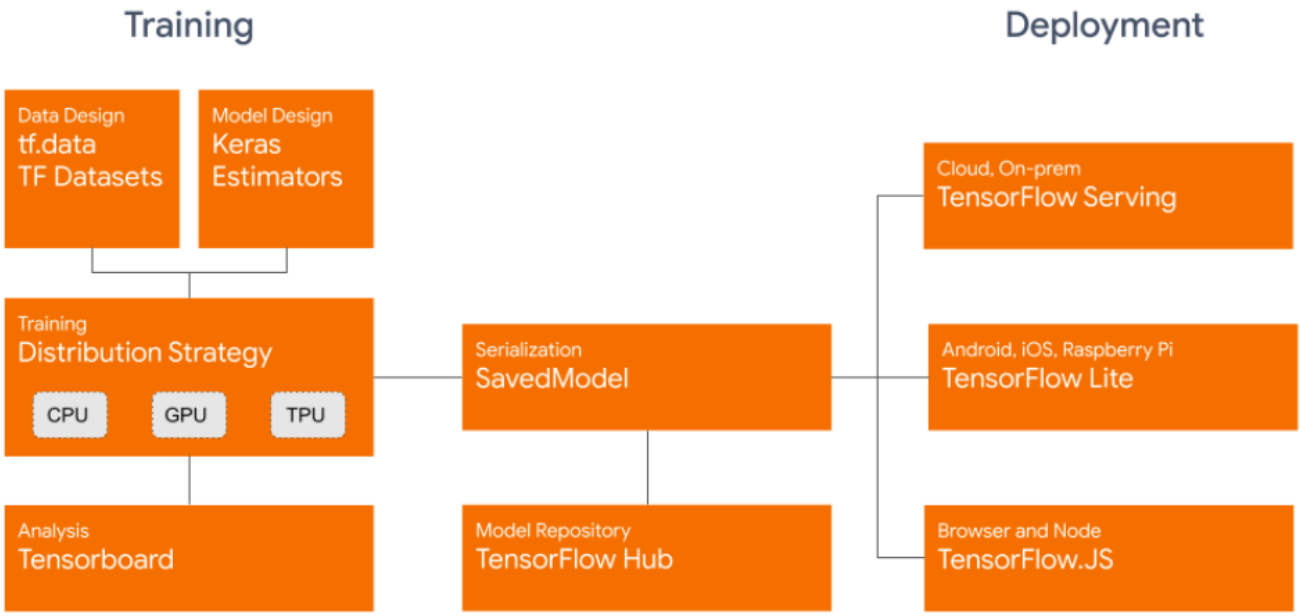


Figure 3-4: TensorFlow Working

3.5 Machine Learning Libraries:

3.5.1 Keras:

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Up until version 2.3, Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, Theano, and PlaidML.

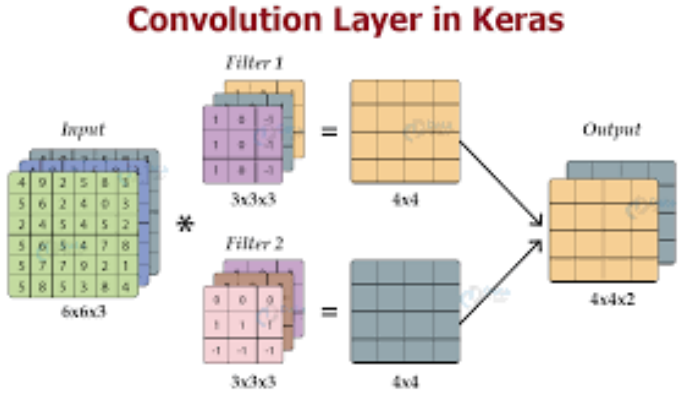


Figure 3-5: Keras CNN Layers and working

3.5.2 NumPy:

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

3.5.3 : Matplotlib:

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

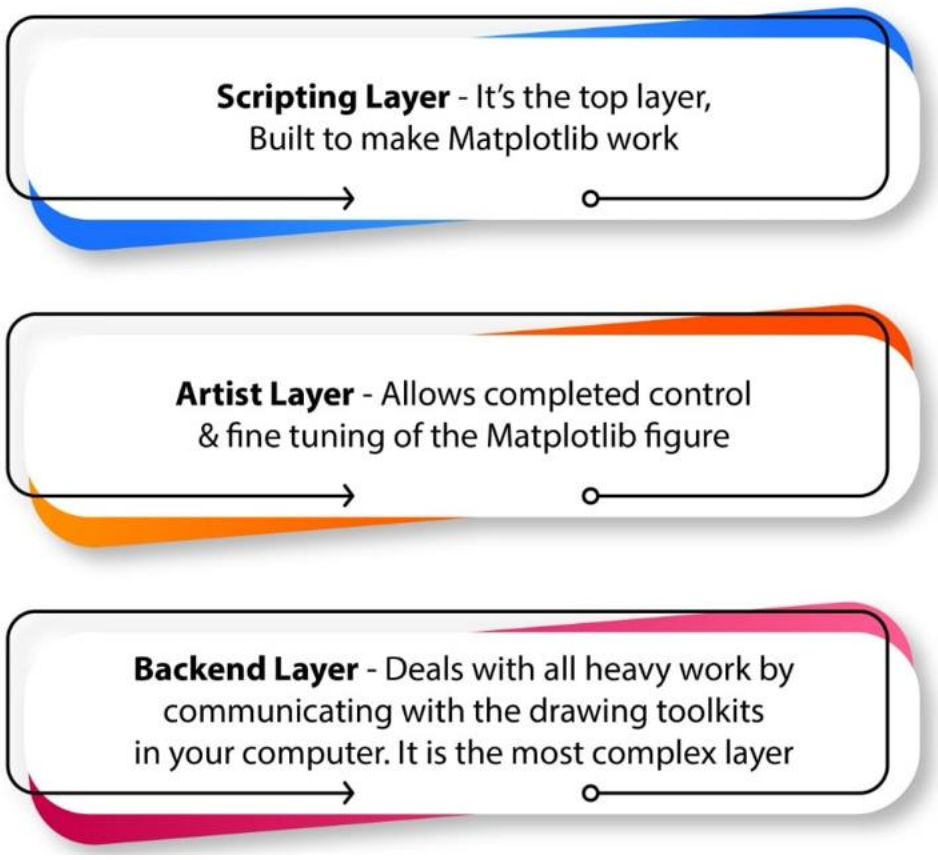


Figure 3-6: Layers of Matplotlib Architecture

3.6 Machine Learning Algorithms:

3.6.1 K-Nearest Neighbour (KNN) Algorithm :

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset.

How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbours
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.

- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

3.4.2 Random Forest Algorithm:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.

3.7 Overview of the backend:

NodeJS

Node.js is an open-source and cross-platform JavaScript runtime environment. A Node.js app runs in a single process, without creating a new thread for every request.

Node.js is actually not a framework or a library, but a runtime environment, based on Chrome's V8 JavaScript engine.

When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

Node.js has a unique advantage because millions of frontend developers that write JavaScript for the browser are now able to write the server-side code in addition to the client-side code without the need to learn a completely different language.

The Node.js Framework refers to a workspace platform that is supportive of the use of node.js. Today, a lot of developers use JavaScript for the development of frontend and backend applications. Node.js framework works in the form of the collection of the framework built on Node. Besides, it has the properties and functionalities that can give further assistance. The open-source, cross-platform JavaScript Runtime environment is good enough to execute the JavaScript code outside the web browser.

Node.js proves to be the best for simple and real-time applications. Moreover, the incorporation of real-time chat applications is an easier approach.

Node.js works in the form of a lightweight framework and is a major reason most developers choose it for the development of apps. Besides, the type of techniques that find implementation with Node.js and for JavaScript gives the fast and extremely efficient opportunity to create the apps. When the use case is not containing CPU-intensive operations, you can get the benefits of Node.js while enjoying the fast and scalable network applications.

- Easy to use and Learn
- Expressive Language
- Extensible
- Learn Standard Library
- GUI Programming Support
- Integrated
- Embeddable
- Dynamic Memory Allocation
- Wide Range of Libraries and Frameworks
- Better efficiency and overall developer productivity
- Code sharing and reuse
- Speed and performance
- Easy knowledge sharing within a team

3.8 Testing:

TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing is the exposure of the system to trail input to see whether it produces correct output.

Testing Phases: -

Software testing includes the following:

- Test activities are determined and test data selected.
- The test is conducted and test results are compared with the expected results.

The testing phase is an important part of software development. It is the computerized system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

Software testing is carried out in three steps:

UNIT TESTING:

The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether the objectives have been met. Errors are noted down and corrected immediately.

Unit testing is the important and major part of the project. So, errors are rectified easily in particular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So, unit testing is conducted to individual modules.

INTEGRATION TESTING:

The second step includes Integration testing. it need not be the ease, the software whose modules when run individually and showing perfect results, will

also show perfect results when run as a whole. The individual modules are clipped under this major module and tested again and verified the results. This is due to poor interfacing, which may result in data being lost across an interface. A module can have inadvertent, adverse effect on any other or on the global data structures, causing serious problems.

VALIDATION TESTING:

The final step involves validation and testing which determines which the software functions as the user elected. Here also some modifications were. In the completion of the protect it is satisfied fully by the end user.

ACCEPTANCE TESTING:

It is performed with realistic data of the client to demonstrate that the software is working, satisfactorily.

SYSTEM TESTING:

It is mainly used if the software meets its requirements. The reference document for this process is the requirement document.

TESTING FUNDAMENTALS

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully, it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

TEST CASES

A Test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not.

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

A test case is a detailed procedure that fully tests a feature or an aspect of a feature. Whereas the test plan describes what to test, a test case describes how to perform a particular test.

Unit Testing methods, the software engineer can drive test cases that

- Guarantee that logical decisions on their true and false sides.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds. • Exercise internal data structure to assure their validity

Chapter 4. Proposed System

Proposed System

4.1 The Proposal:

The past decade has witnessed that rice is one of the major grain crops worldwide, over half of the world's population consumes rice as the staple food, and over 85% of consumption accounts for Asia. To precisely estimate the grain yield and quality of rice, exploring the hill number of rice seedlings is a key component for cultivation density and uniform maturity of precision agriculture.

This project aims to detect and analyse the quality of rice seeds. For demonstration, the rice seed dataset was adopted to identify the number and position of rice seed using a lightweight CNN classification architecture. The proposed CNN model is trained with a cross-validation dataset, which reduces the effect of bias data on the model. In addition, the performance is evaluated by classification accuracy.

The aim of this paper is to provide a platform of for rice seed detection by making labelled and unlabelled data findable and accessible through domain-specific repositories. For this scope, this project focuses on the description of the dataset, including what methods used for collecting and producing the data, where the dataset may be found, and how to use the data with useful information and detect the quality of rice grain.

We start with the data gathering phase – we will gather data from multiple sources such as Kaggle, GitHub, etc. The next step in the process is to use pre-processing techniques to scraping, cleaning, augmentation and erosion for model creation.

Once we have acquired data, we train the ML model using the pre-processed data. We have used ML algorithms to analyse and compare the quality of seeds.

We will compare the rice seed and draw the results; results will indicate that our proposed framework provides a feasible solution to detect to quality of seeds and its severity in online social networks.

Finally, we will compare the results of proposed and baseline features. Findings of the comparison indicate the significance of the proposed features in Rice Seed Quality Detection.

Figure 4-1:WorkFlow

4.2 Benefits of Proposed System:

The current system had a lot of challenges that are overcome by this system :

• **Economic:** The proposed system is economic as it is effectively useful in detecting quality of rice grain.

• **Real-Time Usage:** Application/platform which will be able to detect the seed quality and classify them accordingly in real time.

• **Man Power:** It does not require any person or their efforts to analyse the quality of seed.

• **24 x 7 Availability :** This tool will also collect the seed quality data and show the graphical representation for the data according to the seed, quality and various other aspects and will analyse it.

• **Statistical analysis:** This tool can be used by the industries to classify seeds and their quality, this can also be used by the local/small businesses and can also be used by the ecommerce website to show the quality for the pulses they are selling online.

4.3 Feasibility Study:

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

4.3.1 Technical:

For any real-time detection system, there is a need to process images from the video. For this, the kind of framework used must be the one that is capable of extracting those objects from the images easily and accurately in real-time. The framework used in this is TensorFlow, which is a framework designed by Google for efficiently dealing with deep learning and concepts like neural networks , making the system technically feasible. The system once set up completely, works automatically without needing any person to operate it. The result (count and other information), gets automatically saved in the database, without requiring any manual effort for saving it. For making the system technically feasible, there is a requirement of GPU built system with high processor for better performance.

4.3.2 Economical:

For any real-time object detection system, there is a need of a High definition Camera for better and accurate results. Since the system is completely automated, there is a need of continuous electricity supply for it to operate 24X7. The TensorFlow framework used in the system works great with GPU built systems, which are a little on the expensive side. Since the system uses high performance processors continuously, so to save any disaster from occurring due to very high temperatures, there is a requirement of a cooling system in the environment where it is implemented.

4.3.3 Operational:

The main motto of our system is to reduce the manual efforts of counting the students and vehicles by automating it. The system is able to do that accurately and efficiently making the system operationally feasible. Proposed system is beneficial only if it will meet the organizations operating requirements. The current business operations are considered. This test of feasibility asks if the system will work when it is developed and installed. Here are questions that will help test the operational feasibility of a project.

4.4 Design Representation:

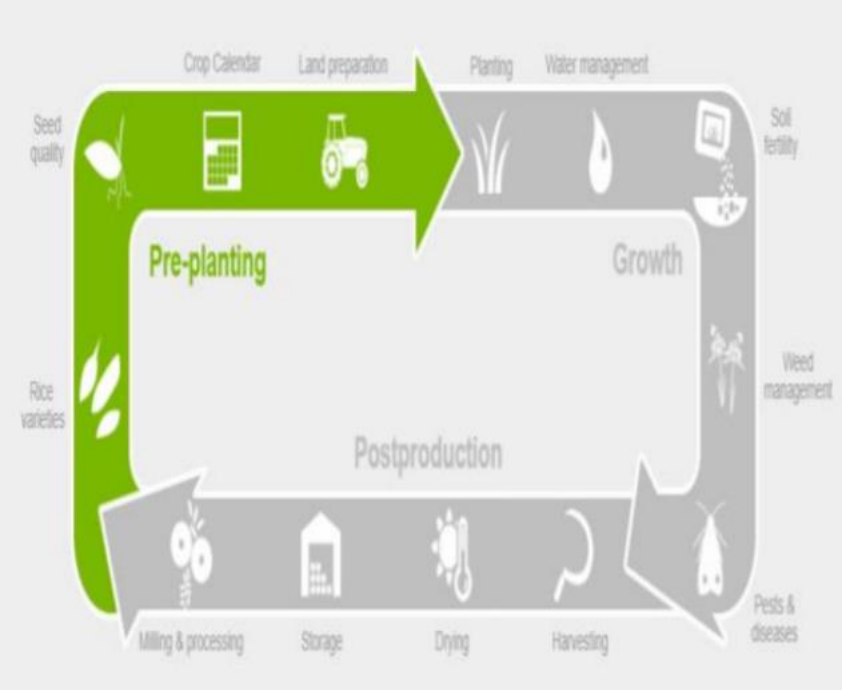


Figure4-2: Life Cycle of Rice

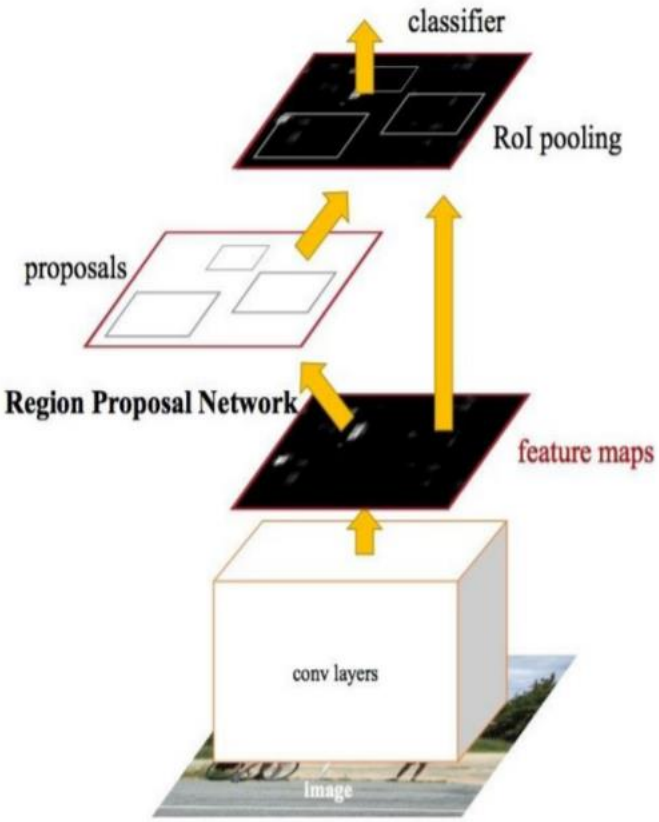


Figure 4-3: CNN architecture

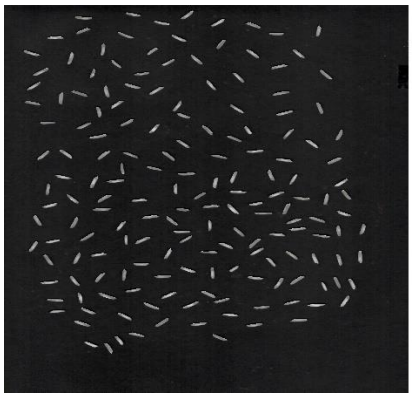


Figure 4-4: Test Image1-Flatining Image

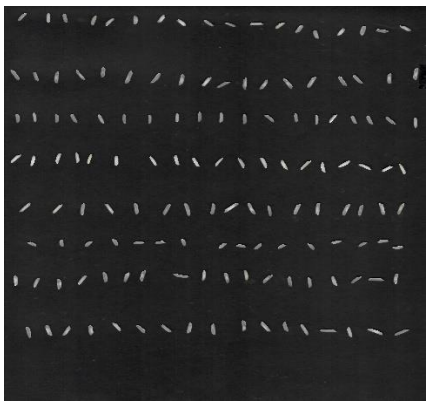


Figure 4-5: Test Image2-Rebuild Edged Image

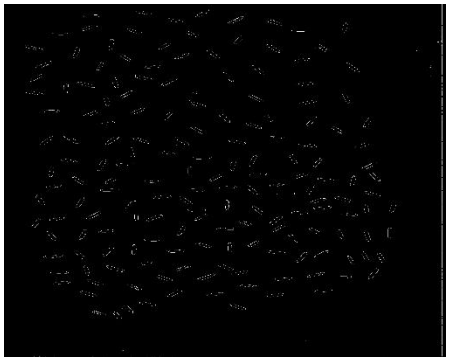


Figure 4-6 : Test Image 3- Canny Image

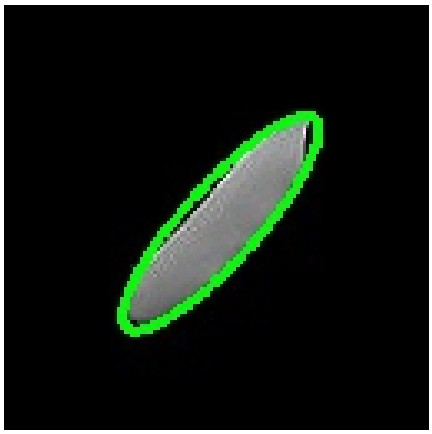


Figure 4-7: Test Image 4-Ellipse Image

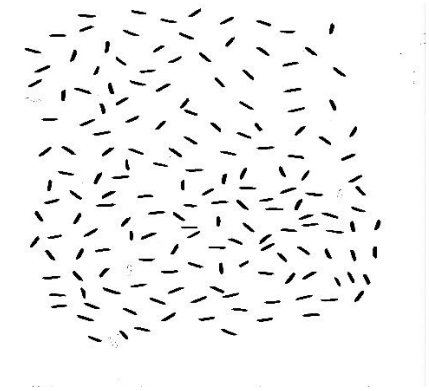


Figure 4-8: Test Image 5-Masked Image

Figure 4-9: Test Image 6-Predicted Image

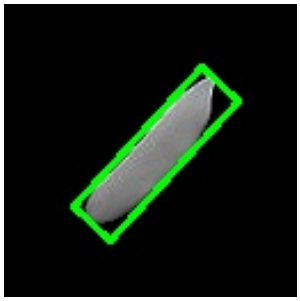


Figure 4-10: Test Image 7-Rectangle Image

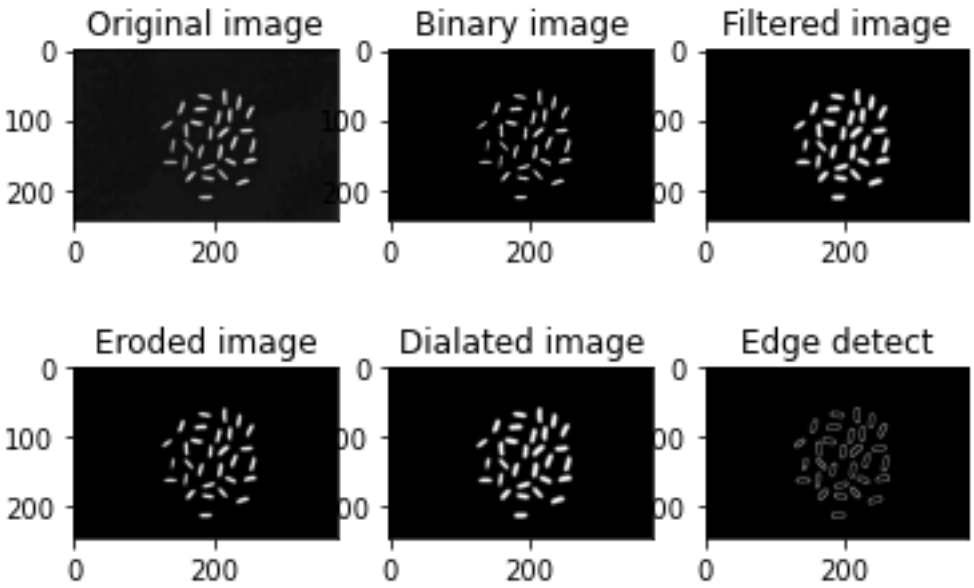


Figure 4-11: Seed Recognition Output

4.4.1 Data Flow Diagram:

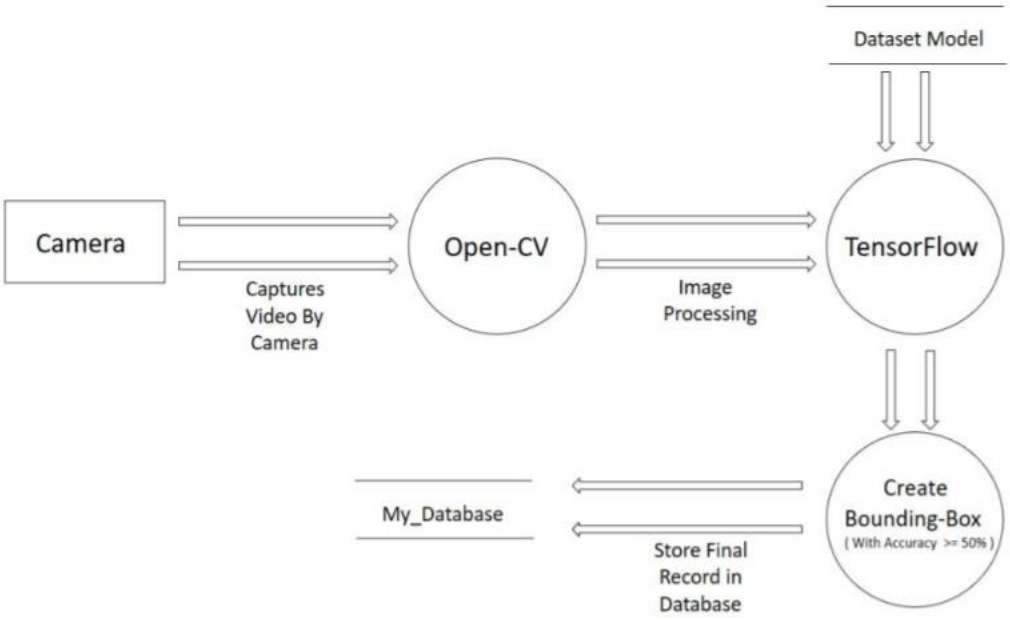


Figure 4-12: Data Flow Diagram Level 1

4.4.2 *Flow Diagram:*

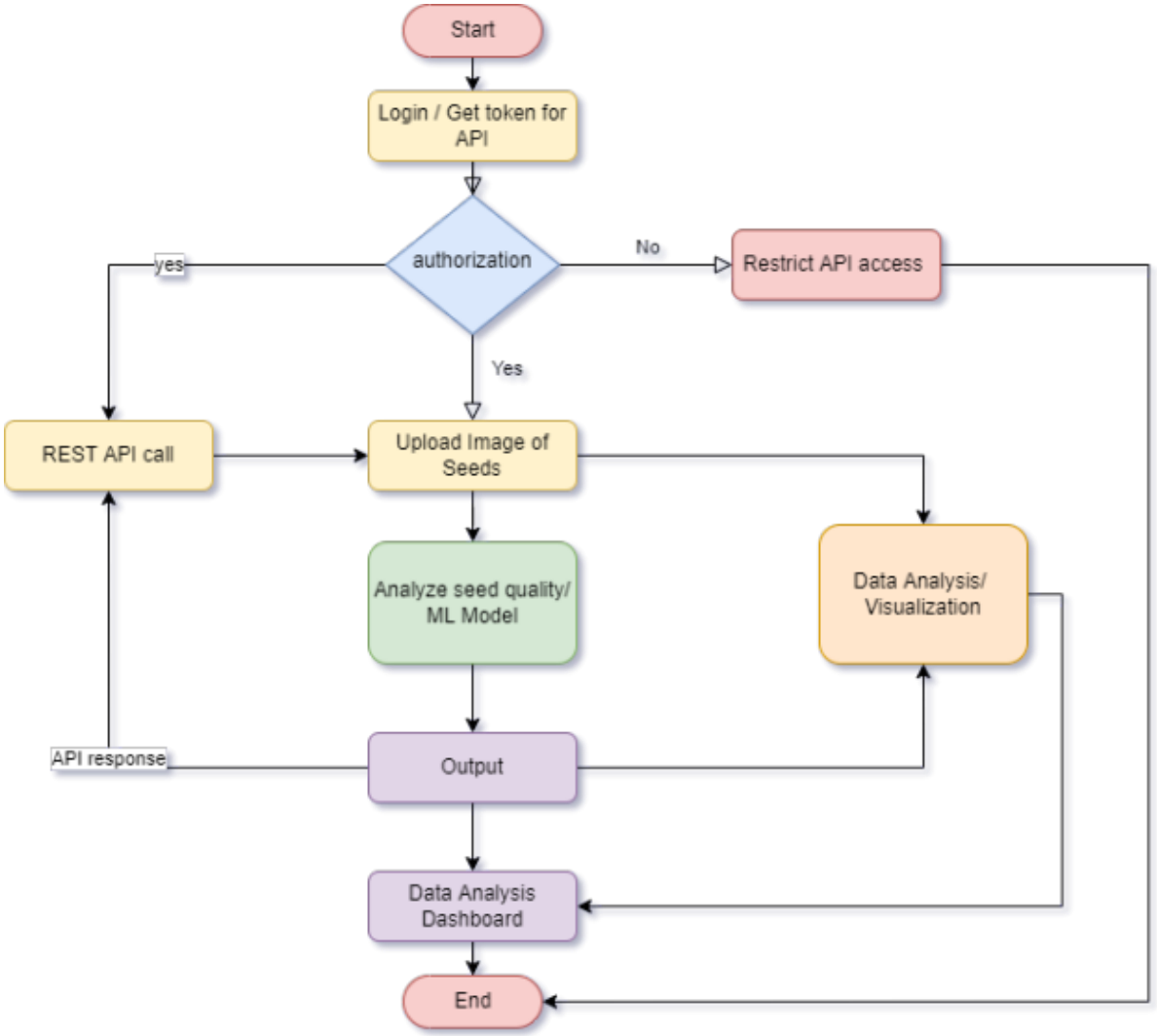


Figure 4-13: Flow Diagram

4.4.3 *Sequential Diagram*

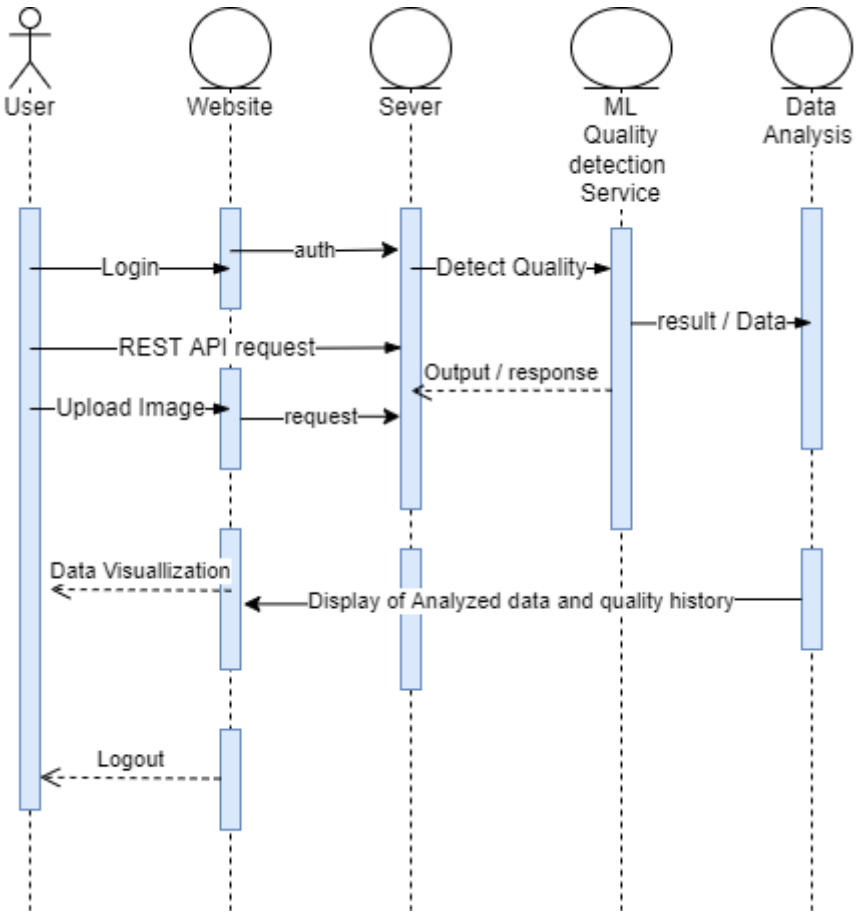


Figure 4-14-Sequential Diagram

4.4.4 Use case Diagram

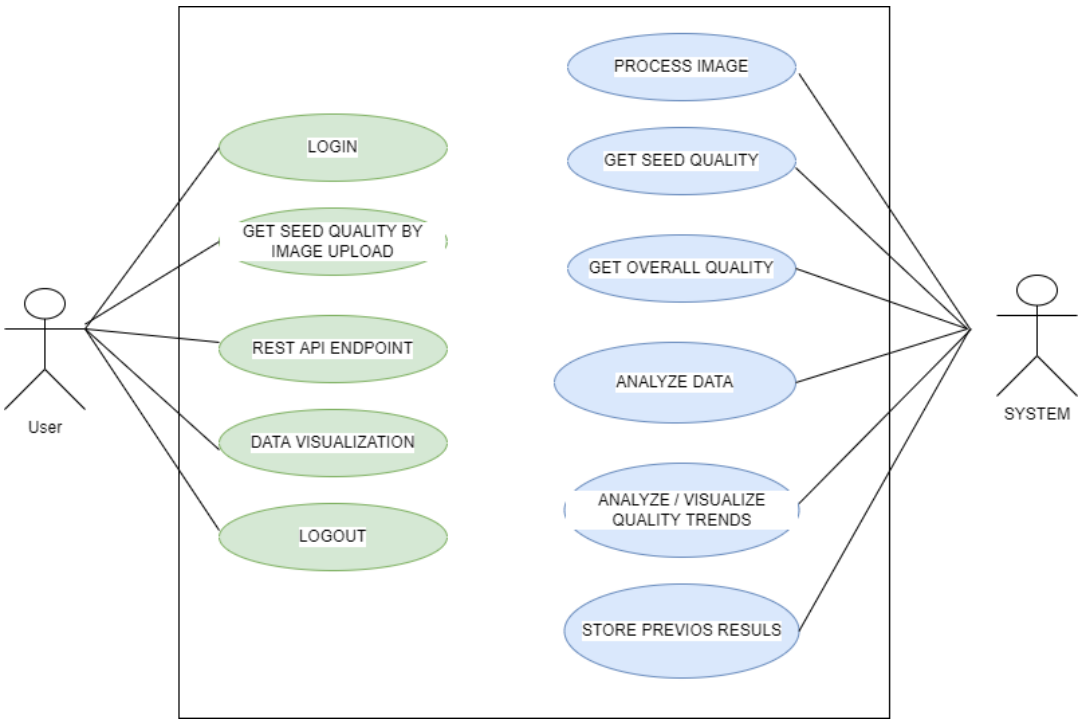


Figure 4-15: Use Case Diagram

4.5 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below :

4.5.1 Hardware

- 32-bit, x86 Processing system
- Windows 7 or later operating system
- High processing computer system without GPU or with GPU(high performance)

4.5.2 Software

- OpenCV
- Python and its supported libraries
- Tensor Flow
- If Installing TensorFlow in GPU systems : 1. CUDA® Toolkit 9.0. 2. The NVIDIA drivers associated with CUDA Toolkit 9.0. cuDNN v7.0. 3. GPU card with CUDA Compute Capability 3.0 or higher

Chapter 5. Conclusion

Conclusion

5.1 Conclusion:

The aim of the project that was to detect, identify and analyse the quality of seeds and save their details in the database is successfully and accurately done by this project with the use of concepts like Deep learning, Convolutional Neural Networks, OpenCV and TensorFlow. The work done manually can now be completely replaced by this automated system and it can reduce all the extra efforts of maintain the records.

5.2 Limitations:

Considering the complexity of the social dynamics underlying the target of classification, and the costly collection and annotation of training data, data scarcity is one of limitations because significant part of actual bullying takes place 'behind closed doors.

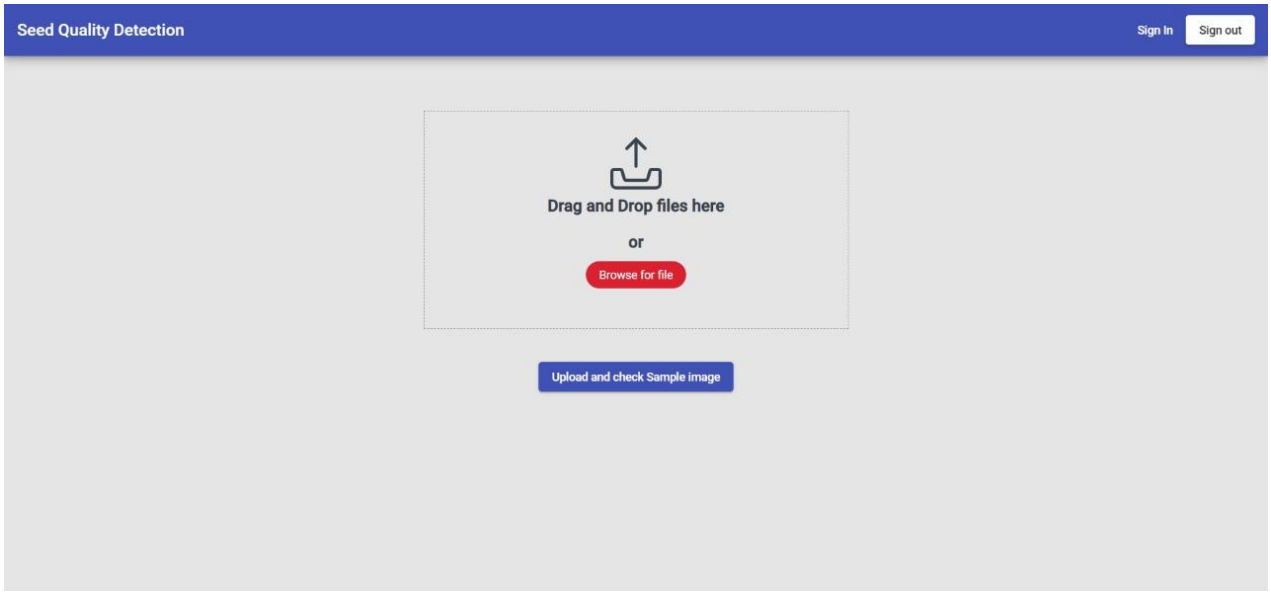
The working of this project would be a little slow because framework like TensorFlow, & deep learning need high-processing hardware and GPU(graphical processing unit) systems but we are using CPU only.

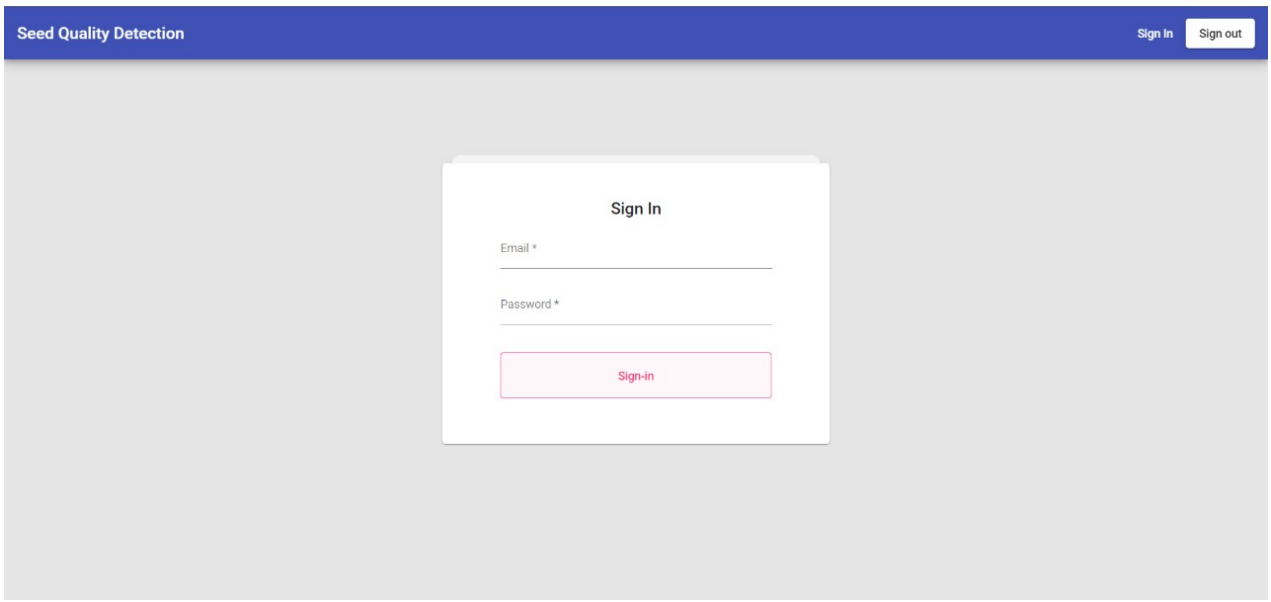
The models that we are using for identifying the objects are pretrained models. So, if we want to train our own model, it takes a lot of time and processing.

In the system, scanning of each frame is one per second but still it needs improvement. If the objects move too fast, it may not detect them.

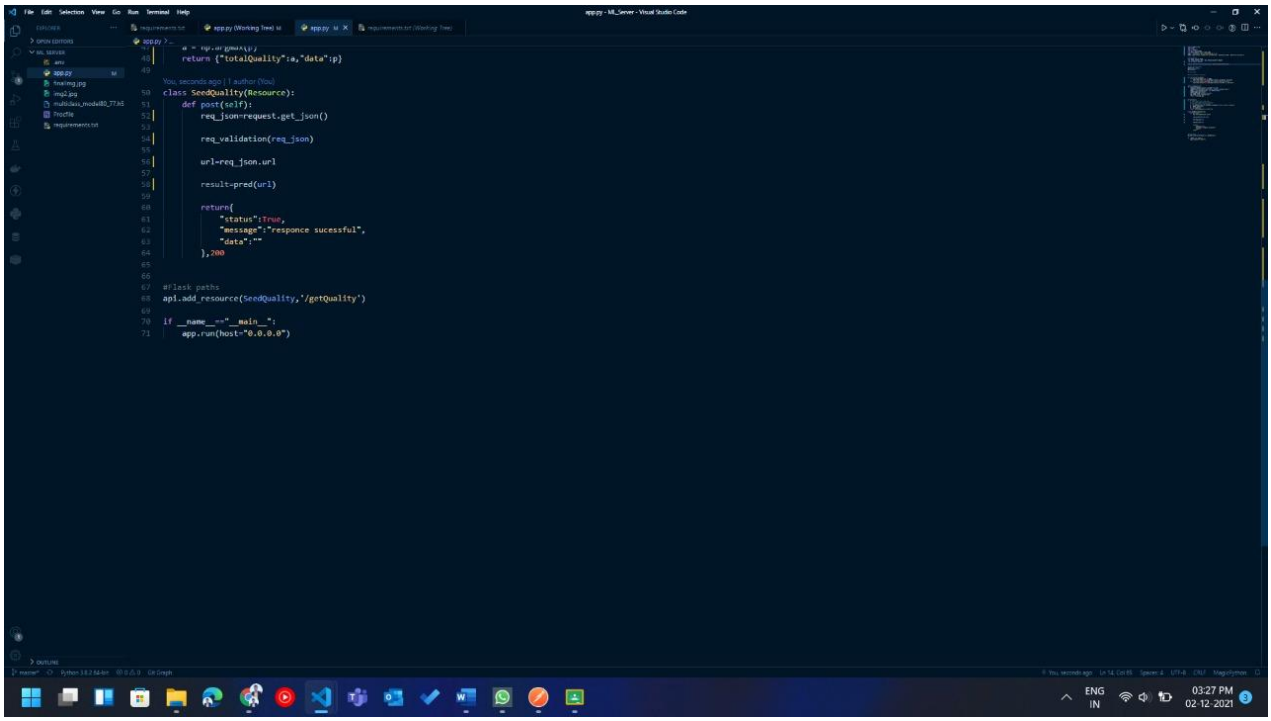
`

Sample Screenshots

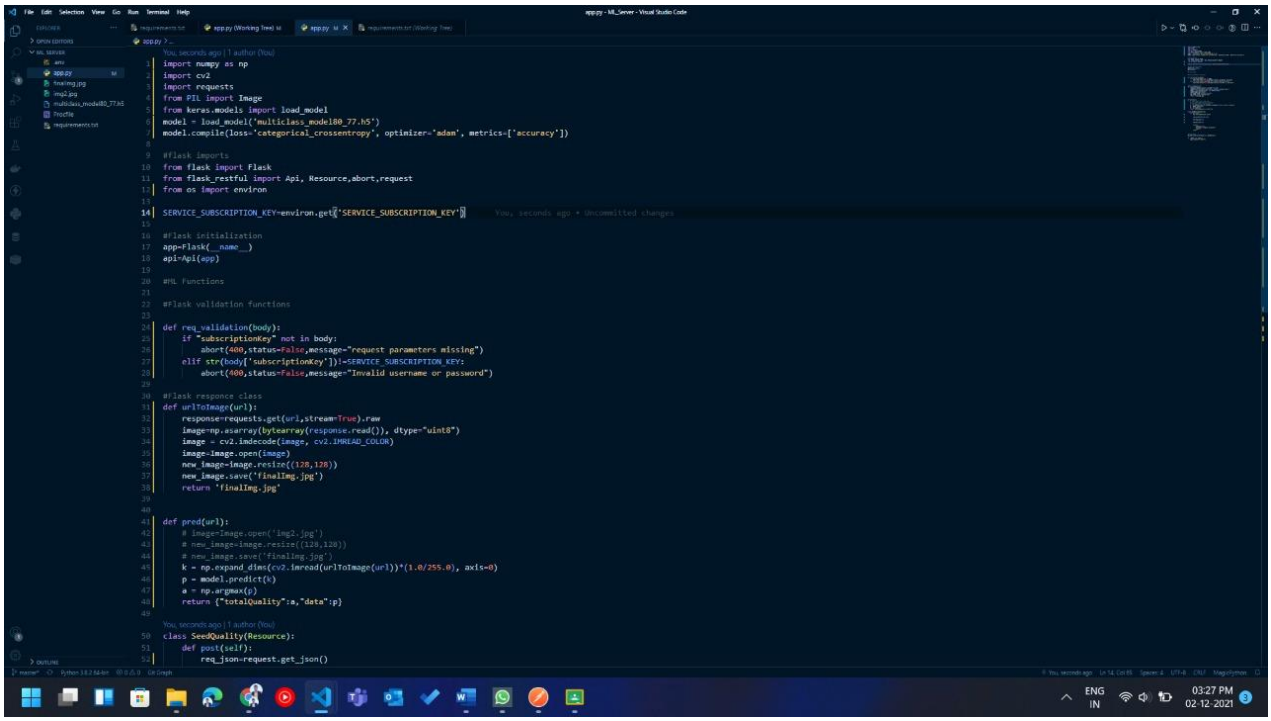




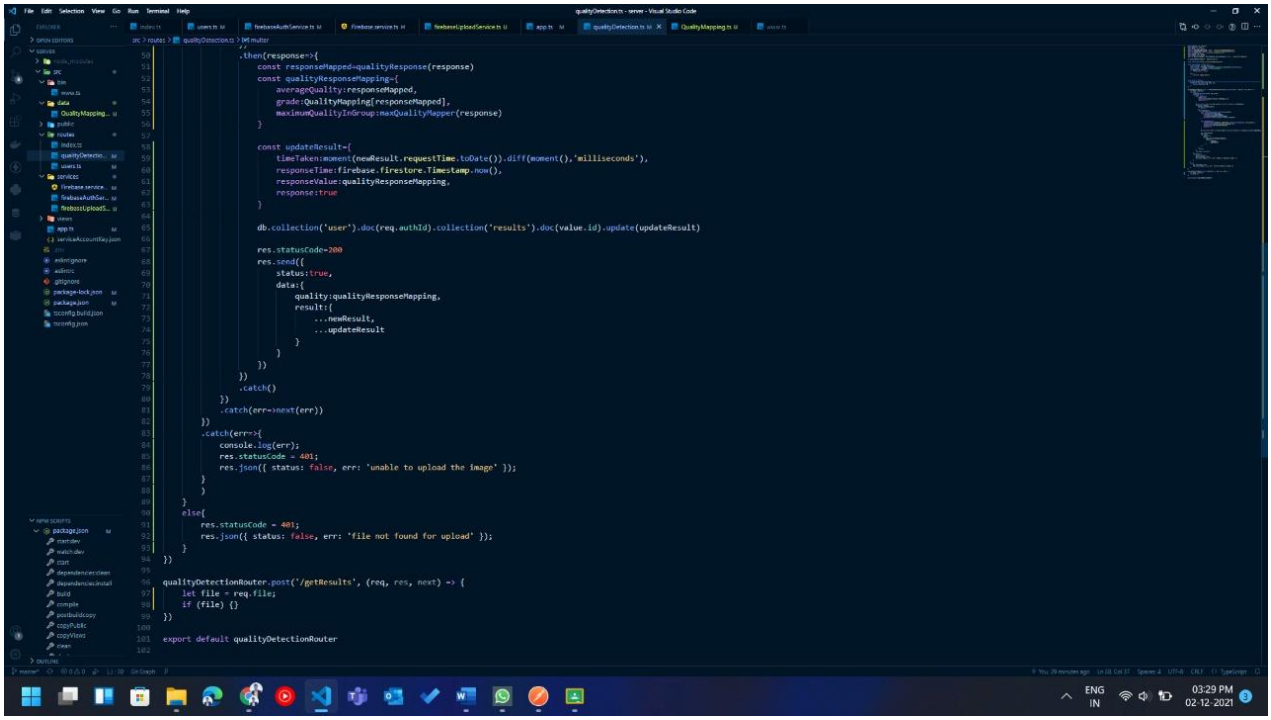
Source Code



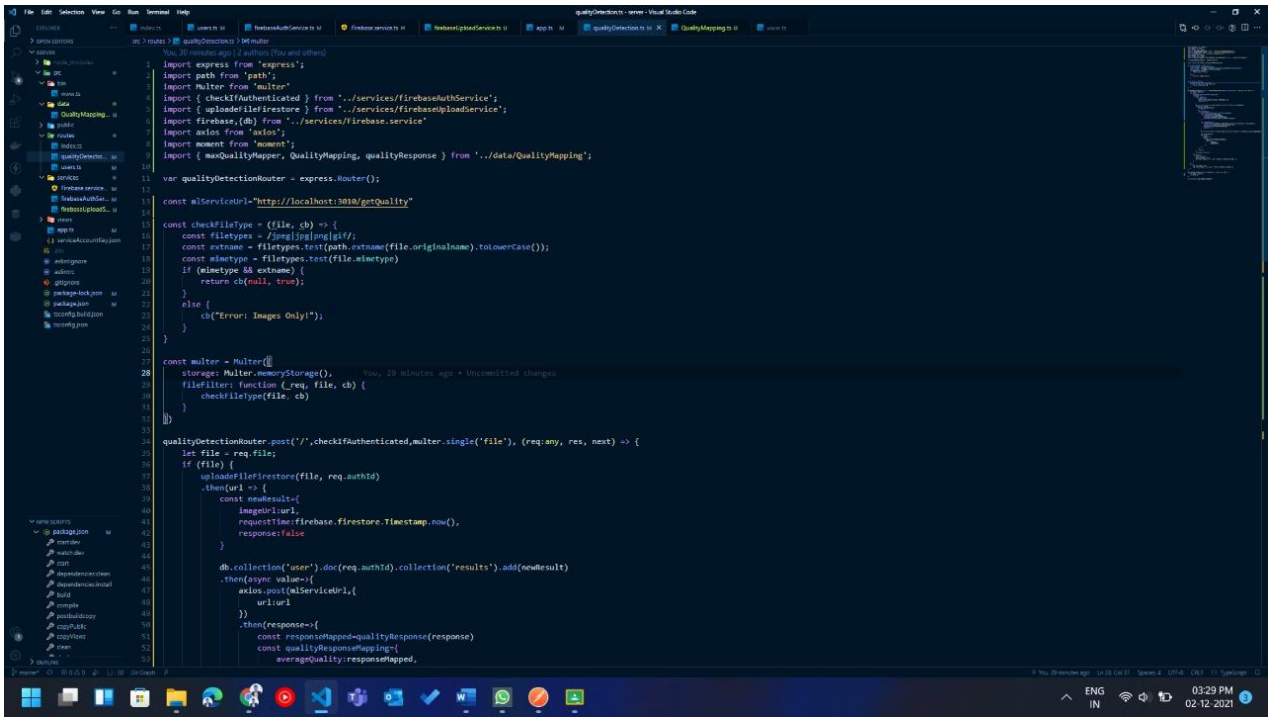
```
    return {"totalQuality":a,"data":p}

class SeedQuality(Resource):
    def post(self):
        req_json=request.get_json()

        req_validation(req_json)

        url=req_json.url

        result=pred(url)

        return{
            "status":True,
            "message":"responce sucessful",
            "data":""
        },200


#flask paths
api.add_resource(SeedQuality,'/getQuality')

if __name__=="__main__":
    app.run(host="0.0.0.0")
```



```
import numpy as np
import cv2
import requests
from PIL import Image
from keras.models import load_model
model = load_model('multiclass_model80_77.h5')
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

#flask imports
from flask import Flask
from flask_restful import Api, Resource,abort,request
from os import environ

SERVICE_SUBSCRIPTION_KEY=environ.get('SERVICE_SUBSCRIPTION_KEY')

#flask initialization
app=Flask(__name__)
api=Api(app)

#ML functions

#flask validation functions

def req_validation(body):
    if "subscriptionKey" not in body:
        abort(400,status=False,message="request parameters missing")
    elif str(body['subscriptionKey'])!=SERVICE_SUBSCRIPTION_KEY:
        abort(400,status=False,message="Invalid username or password")

#flask responce class
def urlToImage(url):
    response=requests.get(url,stream=True).raw
    image=np.asarray(bytearray(response.read()), dtype="uint8")
    image = cv2.imdecode(image, cv2.IMREAD_COLOR)
    image=Image.open(image)
    new_image=image.resize((128,128))
    new_image.save('finalImg.jpg')
    return 'finalImg.jpg'


def pred(url):
    # image=Image.open('img2.jpg')
    # new_image=image.resize((128,128))
    # new_image.save('finalImg.jpg')
    k = np.expand_dims(cv2.imread(urlToImage(url))*(1.0/255.0), axis=0)
    p = model.predict(k)
    a = np.argmax(p)
    return {"totalQuality":a,"data":p}


class SeedQuality(Resource):
    def post(self):
        req_json=request.get_json()
```



```
                    */
                    .then(response=>{
                        const responseMapped=qualityResponse(response)
                        const qualityResponseMapping={
                            averageQuality:responseMapped,
                            grade:QualityMapping[responseMapped],
                            maximumQualityInGroup:maxQualityMapper(response)
                        }

                        const updateResult={
                            timeTaken:moment(newResult.requestTime.toDate()).diff(moment(),'milliseconds'),
                            responseTime:firebase.firestore.Timestamp.now(),
                            responseValue:qualityResponseMapping,
                            response:true
                        }

                        db.collection('user').doc(req.authId).collection('results').doc(value.id).update(updateResult)

                        res.statusCode=200
                        res.send({
                            status:true,
                            data:{
                                quality:qualityResponseMapping,
                                result:{
                                    ...newResult,
                                    ...updateResult
                                }
                            }
                        })
                    })
                    .catch()
                })
                .catch(err=>next(err))
            })
            .catch(err=>{
                console.log(err);
                res.statusCode = 401;
                res.json({ status: false, err: 'unable to upload the image' });
            }
            )
        }
        else{
            res.statusCode = 401;
            res.json({ status: false, err: 'file not found for upload' });
        }
    })

qualityDetectionRouter.post('/getResults', (req, res, next) => {
    let file = req.file;
    if (file) {}
})

export default qualityDetectionRouter
```



```
import express from 'express';
import path from 'path';
import Multer from 'multer'
import { checkIfAuthenticated } from '../services/firebaseAuthService';
import { uploadFileFirestore } from '../services/firebaseUploadService';
import firebase,{db} from '../services/firebase.service'
import axios from 'axios';
import moment from 'moment';
import { maxQualityMapper, QualityMapping, qualityResponse } from '../data/QualityMapping';

var qualityDetectionRouter = express.Router();

const mlServiceUrl="http://localhost:3010/getQuality"

const checkFileType = (file, cb) => {
    const filetypes = /jpeg|jpg|png|gif/;
    const extname = filetypes.test(path.extname(file.originalname).toLowerCase());
    const mimetype = filetypes.test(file.mimetype)
    if (mimetype && extname) {
        return cb(null, true);
    }
    else {
        cb("Error: Images Only!");
    }
}

const multer = Multer({
    storage: Multer.memoryStorage(),
    fileFilter: function (_req, file, cb) {
        checkFileType(file, cb)
    }
})

qualityDetectionRouter.post('/',checkIfAuthenticated,multer.single('file'), (req:any, res, next) => {
    let file = req.file;
    if (file) {
        uploadFileFirestore(file, req.authId)
        .then(url => {
            const newResult={
                imgUrl:url,
                requestTime:firebase.firestore.Timestamp.now(),
                response:false
            }

            db.collection('user').doc(req.authId).collection('results').add(newResult)
            .then(async value=>{
                axios.post(mlServiceUrl,{
                    url:url
                })
                .then(response=>{
                    const responseMapped=qualityResponse(response)
                    const qualityResponseMapping={
                        averageQuality:responseMapped,
```

Bibliography

[1] Date set for training and testing the model https://github.com/dhishku/Machine-Learning-forGrain-Assaying/blob/master/Dataset_Wheat_Grain.md

[2] YOLO: Real-Time Object Detection (pjreddie.com)

[3] https://github.com/ModelDepot/tfjs-yolo-tiny

[4] https://github.com/shaqian/tfjs-yolo

[5] https://cloudxlab.com/blog/object-detection-yolo-and-python-pydarknet/

[6]https://towardsdatascience.com/how-to-train-a-custom-object-detection-model-with-yolo-v5- 917e9ce13208

[7] (PDF) Image Analysis: A Modern Approach to Seed Quality Testing (researchgate.net)