VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi – 590018, Karnataka.

A Project Work (18CSP83)

on

"GRAIN ADULTERATION DETECTION USING DEEP NEURAL NETWORK"

Submitted in partial fulfillment for the award of degree of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE AND ENGINEERING

Project Associates

1. KUSHI N	1DB20IS065
2. MANASVI K N	1DB20IS078
3. MEGHANA S	1DB20IS081
4. MOULYA P	1DB20IS085

Under the Guidance of

Dr. B. K. Raghavendra

Head of the Department

Department of Information Science and Engineering

Don Bosco Institute of Technology

Department of Information Science and Engineering

DON BOSCO INSTITUTE OF TECHNOLOGY

Kumbalagodu, Mysuru Road, Bengaluru-560074

2023-24

Don Bosco Institute of Technology

Kumbalgodu, Mysuru Road, Bengaluru-560074

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

Certified that the Project on the topic "**Grain Adulteration Detection using Deep Neural Network**" has been successfully presented at **Don Bosco Institute of technology** by **KUSHI N(1DB20IS065), MANASVI K N(1DB20IS078), MEGHANA S(1DB20IS081), MOULYA P(1DB20IS085)** partial fulfillment of the requirements for the VIII Semester degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi during academic year 2023- 2024. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the project work report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements in respect of Project work for the said degree.

Signature of the Guide	**Signature of the HOD**	**Signature of the Principal**
Dr. B K Raghavendra Professor & Head Dept of ISE, DBIT Bengaluru	Dr. B K Raghavendra Professor & Head Dept of ISE, DBIT Bengaluru	Dr. B S Naghabhushana Principal, DBIT Bengaluru

EXTERNAL VIVA

Name of the Examiners	**Signature with Date**
1._____	_____
2._____	_____

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned the efforts with success.

We would like to profoundly thank **Management** of **Don Bosco Institute of Technology** for providing such a healthy environment for the successful completion of Project work.

We would like to express our thanks to the Principal **Dr. Naghabhushana B S** for his encouragement that motivated us for the successful completion of Project work.

It gives us immense pleasure to thank **Dr. B K Raghavendra**, Professor and Head of the Department, Information Science and Engineering for his constant support and encouragement.

We would like to express our deepest sense of gratitude to our **Project Co-coordinators Dr. Basavaraj G M,** Professor **and Dr. R SivaKumar**, Associate Professor, Department of Information Science & Engineering for their constant support and guidance throughout the Project work.

We would like to express our deepest sense of gratitude to **Dr. Santhosh Kumar S**, Associate Professor Department of Information Science & Engineering for his constant support and guidance throughout the Project work.

Also, we would like to thank all teaching and non-teaching staff of Department of Information Science & Engineering who have helped directly and indirectly throughout the project work.

KUSHI N(1DB20IS065)

MANASVI K N(1DB20IS078)

MEGHANA S(1DB20IS081)

MOULYA P(1DB20IS085)

ABSTRACT

The increasing concerns over food safety and the economic impact of food adulteration have motivated the development of advanced technologies for reliable and efficient detection methods. The "Grain Adulteration Detection Using Deep Neural Network" project has the potential to significantly enhance food safety and agricultural efficiency by automating the detection of grain adulteration. This solution can aid in reducing the economic losses caused by adulteration and safeguarding public health by ensuring the quality and purity of grains in the market. The model is trained on a diverse dataset encompassing different types of grains and common adulterants. The proposed solution leverages the power of deep learning to create a robust and versatile system for grain quality assessment. The model exhibits robust performance in real-world scenarios, showcasing its potential for integration into quality control systems in the food industry. The system will be capable of identifying a wide range of adulterants such as stones, dust, foreign seeds, and other contaminants in various types of grains like rice, wheat, maize, and pulses. This research contributes to the ongoing efforts to enhance the accuracy and efficiency of food quality control measures, promoting consumer confidence in the authenticity of agricultural commodities.

LIST OF CONTENTS

Chapter 1

Introduction

Grains are essential for healthy life. The grains we take should be pure, nutritious and free from any type of adulteration for proper maintenance of human health. The intake of any grain substance is intended for the nourishment which is gained from it. Since the grain is into consecutive stages of production, processing and finally distribution, the nourishment in the grain items is collapsed. For the grain products to ripen and improved in texture, storage and appearance, a concept of adulteration is widely practiced. The nature or quality of the grain is reduced through addition of adulterants by the process of grain adulteration. The adulterants is a foreign chemical substance present in grain. In the process of grain adulteration, little quantities of non-nutritious substances are added knowingly to improve its appearance or storage properties of the grain. Mostly the adulteration in grains and vegetables are caused using a harmful chemical substance called Formalin. Formalin is a colourless, aqueous solution of formaldehyde to preserve biological specimens. India is at second number after China in the production of the grains. In India all the preharvest and post-harvest process are done manually with help of labour. Manual process is very time consuming, less efficient so to get accurate result automation in agriculture industry is needed. The post-harvest process includes sorting and grading of grains. Different quality factors are considered for sorting and grading of grains. These factors are internal quality factors and external quality factors. The external quality factors are texture, shape, colour, size and volume and internal quality factors are test, sweetness, flavour, aroma, nutrients, carbohydrates present in that grain.

1.1 Deep Neural Networks (DNNs):

DNNs are a subset of artificial neural networks inspired by the human brain's architecture. These networks consist of multiple layers of interconnected nodes (neurons) that process input data to produce an output. DNNs excel at learning complex patterns and representations from large datasets, making them ideal for tasks like image recognition, which is crucial in identifying subtle visual cues indicative of grain adulteration.

The first step involves collecting a diverse dataset of grain images, including both pure and adulterated samples. The dataset should cover various types of grains and different forms of

adulteration. High-quality images are crucial for the effectiveness of the DNN.The collected images undergo preprocessing to enhance the DNN's ability to extract relevant features. This may include resizing, normalization, and other techniques to standardize the input data. Preprocessing helps ensure that the DNN can learn meaningful patterns from the images. Deep neural network based method for sorghum adulteration detection, which is composed of three main procedures including data preparation, image processing and deep neural network based classification. The Figure 1.1 shows the structure of DNN based model.

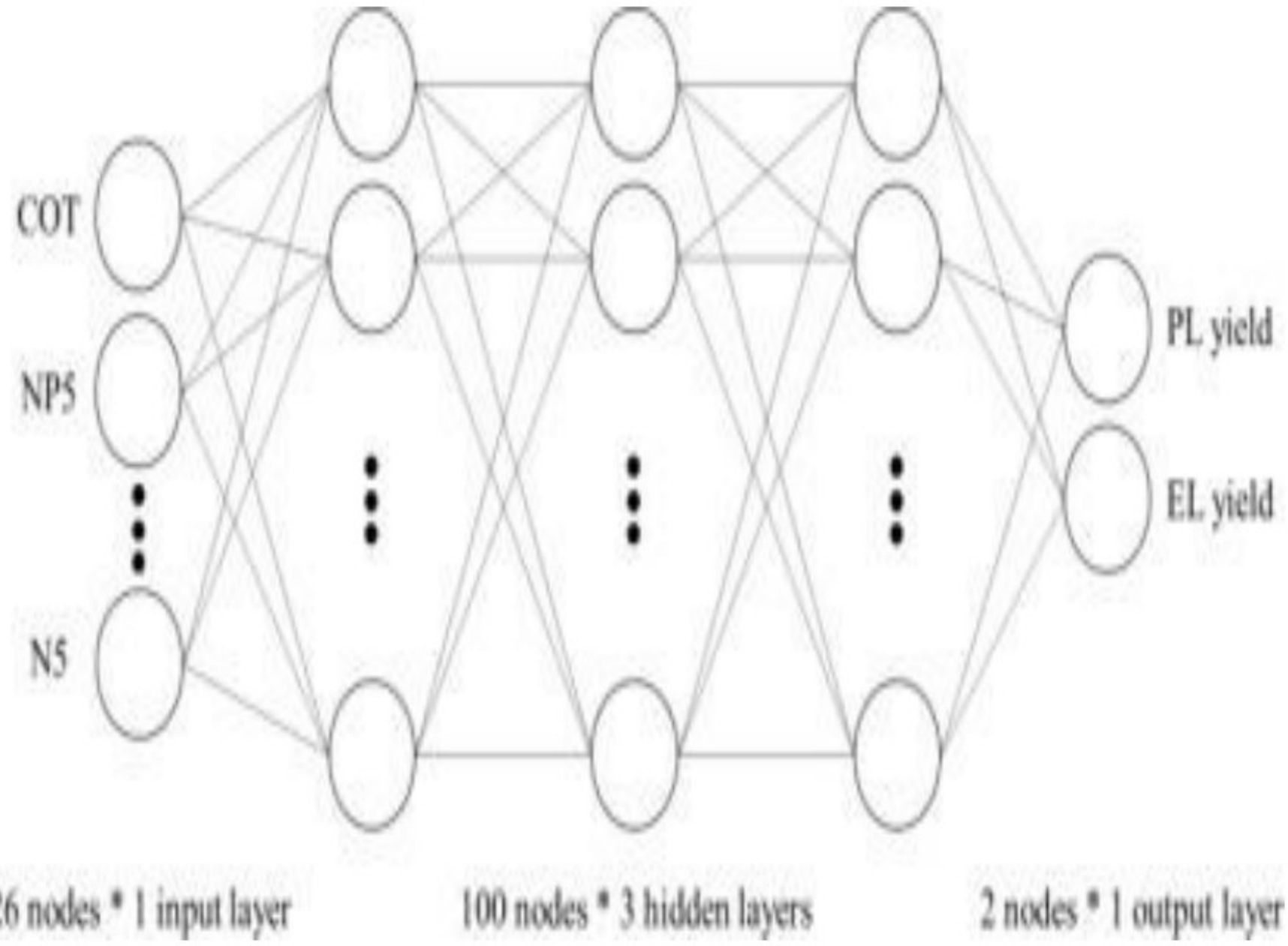

Figure 1.1: Structure of the DNN based model

1.1.1 Data preparation

The regular practice for training classifiers with deep neural networks needs to collect large amounts of labeled data for supervised learning. Regardless of the network structures and different types of models, the quality and scale of the dataset has a significant impact on the classification performance. Therefore, the acquisition of original images plays a fundamental part in overall project.

1.1.2 Image processing

Image processing plays a significant role in detecting grain adulteration, especially when the adulteration involves subtle changes in visual characteristics. The specific processing flow is as follows:

- Brightness equalization. When taking pictures to acquire images, the uneven illumination caused shadows in the background, which will affect the edge detection effect and also lead to differences in the histogram distribution of color space between segmented images, reducing the data uniformity. Hence, it is necessary to identify and subtract the background, and adjust the image to equalize the brightness.
- Edge detection. Convert the image to grayscale, use low-pass filtering methods such as Gaussian blur to remove the image noise. Take morphological operations like erode and dilate to separate adjoint grains. Then find the outside contours of each connected component using the Canny edge detector . The corresponding low and high thresholds for gray scale gradient are 100 and 200 respectively.
- Image segmentation. Approximate the contours with polygons using Douglas–Peucker algorithm. To preserve the size information of the grains, we used square with fixed side length (140 pixels) to segment each granule. The square has the same center as the bounding rectangle, and its side length is larger than the maximum value among the long sides of the rectangles. The Figure 1.2 shows the results of image processing.

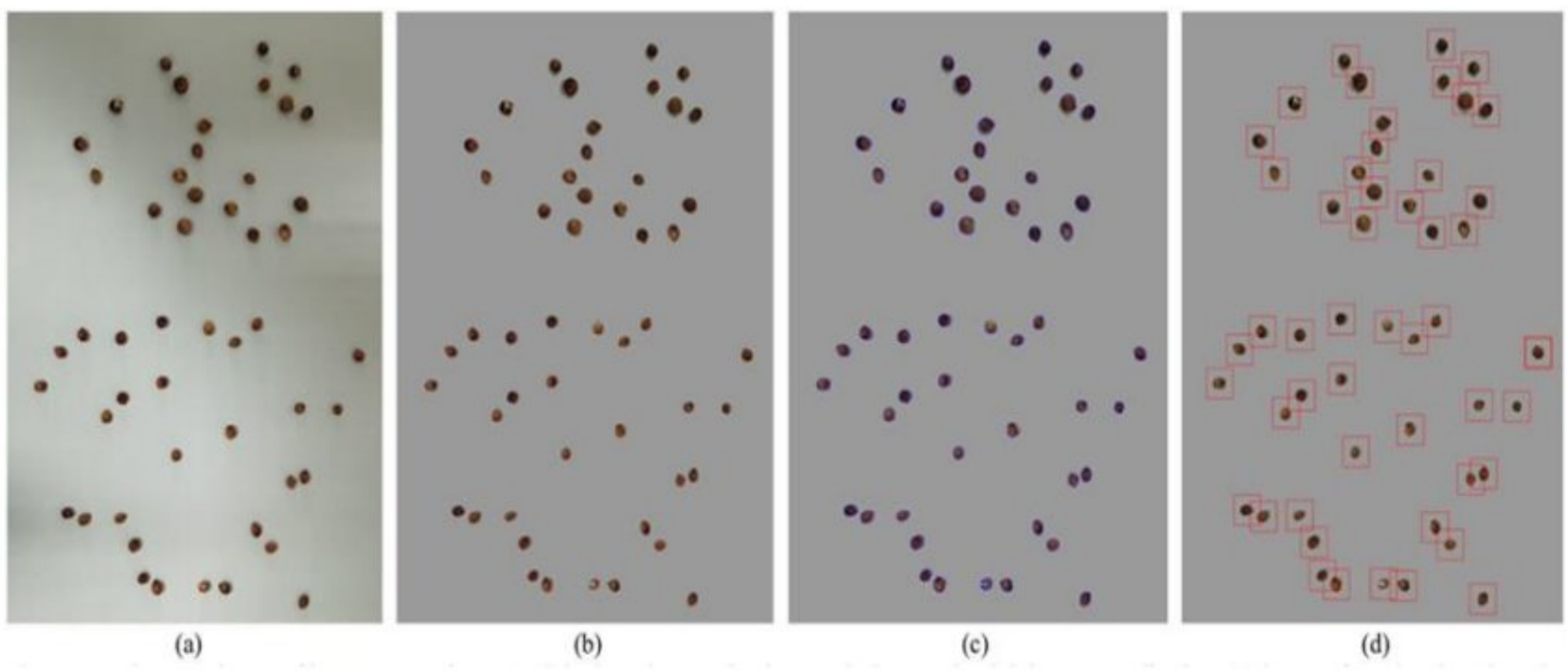

Figure 1.2: Shows the results of image processing.

(a) original sorghum grains image;

(b) image after brightness equalization;

(c) image after edge detection, the contours are marked with blue circles;

(d) image after segmentation, the segmented granules are bounded with red rectangles.

1.1.3 Data Acquisition and Preprocessing:

The success of a DNN model relies on the quality and diversity of the data it is trained on. In the context of grain adulteration detection, a dataset containing images of both pure and adulterated grains is essential. Image preprocessing techniques may be employed to enhance the model's ability to identify relevant features, such as color variations, texture differences, or irregularities in shape and size.

1.2 Problem Statement

Identifying a Grain and adulteration can be a challenge, even for experienced farmers and buyers. And if you're new to using field vision, it can be daunting to figure out where to even begin searching in the hundreds of pages of toxics. By some features like size, shape and colour adulterated grains can be classified. By using YOLO, we can classify the category of grains. Nowadays, Identification of grains weather adulterated or not is a difficult activity sometimes leading to uncertainty. But it cause some toxic disease when it cant be find after diagnosis even with what it affected.

1.3 Aim And Objectives Of The Proposed Work

Aim: The aim of the project is to develop a system that utilizes deep neural networks to detect adulteration in grains effectively and efficiently.

Objectives:

- **Data Generation for Training with Image Naming:** The first objective involves the meticulous collection and labelling of diverse grain images. This is crucial for training the neural network effectively. Each image is not only a visual representation but also carries

specific information tied to its name. This structured dataset becomes the foundation for the system's learning process, ensuring that the model comprehensively understands the variations in grain quality.

- **Understanding the Shape Differences in Grain Adulteration**: Beyond mere classification, it is imperative to delve into the intricacies of grain adulteration by studying the shape differences. This objective aims to develop a nuanced understanding of how adulterants alter the natural shape of grains. By doing so, the system gains the capability to identify subtle deviations, enhancing its accuracy in categorizing grains as Good, Average, or Bad.
- **Efficient Adulteration Rate Detection Framework**: The project seeks to go beyond binary classification and aims to quantify the extent of adulteration. Developing an efficient framework for detecting adulteration rates is a critical objective. This involves creating algorithms and methodologies that can assess the degree of adulteration present in a given sample, providing valuable insights into the severity of the issue.
- **Exploring Health Effects of Grain Adulteration on Society:** Grain adulteration doesn't just impact economic aspects; it poses severe threats to public health. This objective involves conducting a comprehensive analysis to unearth the broader effects of consuming adulterated grains on societal health. Understanding these health implications is crucial for emphasizing the urgency of combating grain adulteration and showcasing the potential positive impact of the developed system.
- **Comparison with Existing Systems:** In the pursuit of progress, it is essential to evaluate the efficacy of the proposed system against existing methods. This objective involves a comparative analysis to highlight the advancements and advantages offered by the developed system. Such a comparison provides a tangible measure of success, demonstrating how the project contributes to the evolution of grain quality assessment methodologies.

1.4 Proposed System

1.4.1 Block diagram of the Proposed System

Brightness Equalization:

- Uneven illumination in images can introduce shadows and inconsistencies in brightness, which can adversely affect subsequent image processing tasks such as edge detection and histogram analysis.
- To address this issue, brightness equalization techniques are employed to identify and subtract the background illumination, thereby adjusting the image to achieve uniform brightness levels.
- This process helps enhance the quality and uniformity of the image data, improving the effectiveness of subsequent image processing algorithms.
- Enhanced Feature Detection: By equalizing brightness levels across the image, subtle features and details that may have been obscured by uneven illumination become more pronounced. This aids in the detection of important features and patterns within the image, leading to more accurate and robust analysis results.
- Improved Visualization: Uniform brightness levels make it easier for human observers to interpret and analyze images visually. By removing distracting shadows and inconsistencies, brightness equalization improves the overall clarity and interpretability of the image, facilitating better decision-making and analysis. The Figure 1.3 shows the system architecture for the proposed work.

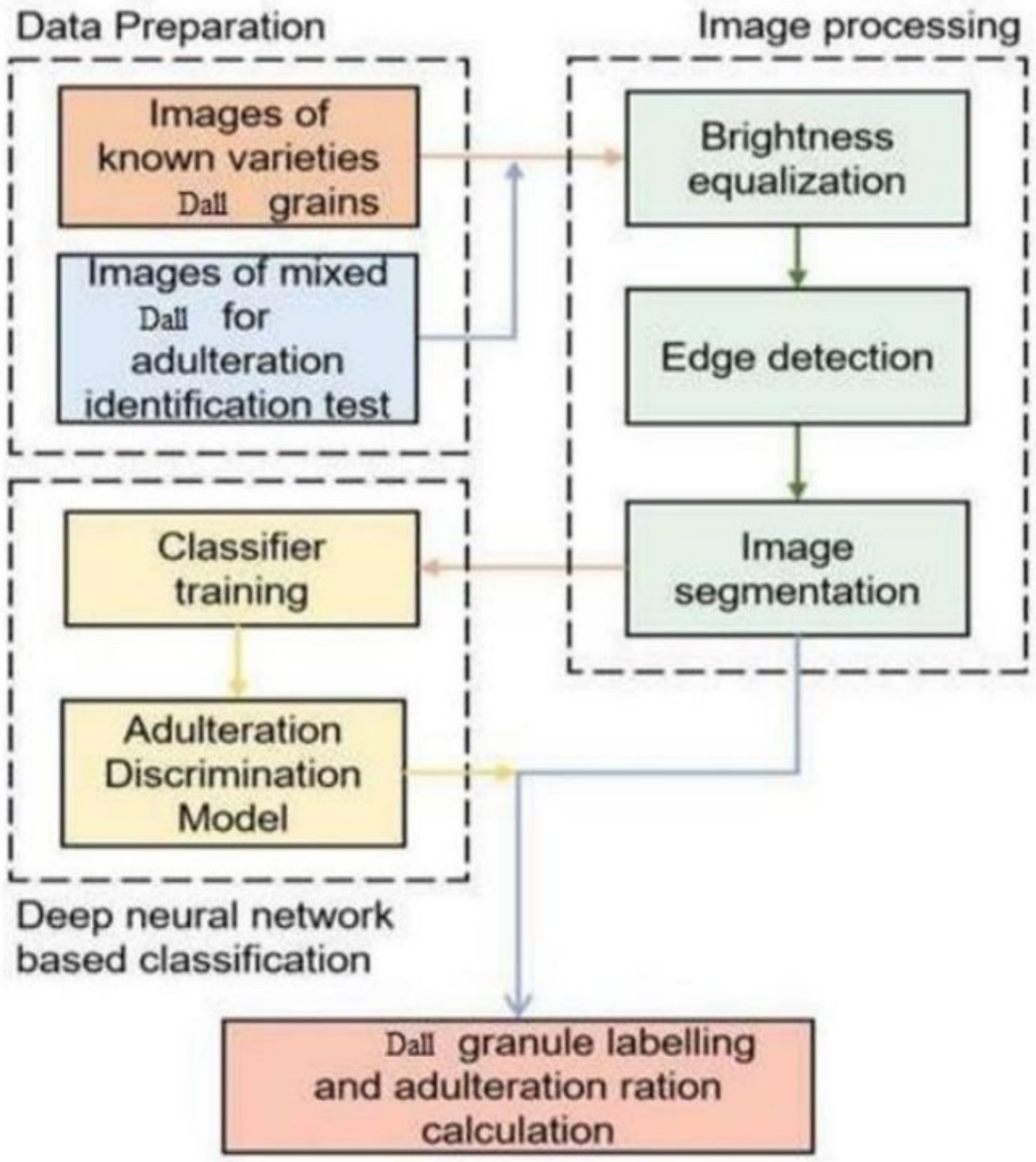

Figure 1.3 : System Architecture for the Proposed work

Edge detection:

- Edge detection plays a crucial role in identifying the boundaries of objects or features within an image.
- Initially, the image is converted to grayscale to simplify processing and reduce computational complexity.
- Low-pass filtering methods such as Gaussian blur are applied to the grayscale image to remove noise and smooth out the intensity variations.
- The Canny edge detector is then employed to identify significant edges in the image, using predefined low and high thresholds to determine the strength of detected edges.
- This process facilitates accurate delineation of object boundaries, which is essential for subsequent segmentation and analysis tasks.

Image Segmentation:

- Image segmentation involves partitioning an image into distinct regions or objects based on certain criteria, such as color, intensity, or texture.
- The Douglas-Peucker algorithm is utilized to approximate the contours of objects within the image, reducing the complexity of the contour representation while preserving essential shape characteristics.
- Minimal upright bounding rectangles are calculated for each polygonal approximation of the contours, providing a simplified representation of the object's spatial extent.
- Manual thresholds are set based on the width and height of the bounding rectangles to filter out noise points and ensure the accuracy of the segmentation results.
- By employing these segmentation techniques, the image is effectively partitioned into individual objects or regions of interest, laying the groundwork for subsequent analysis and classification tasks.

The Figure 1.3 shows system architecture for the proposed system leverages a modular convolutional neural network, incorporating layers optimized for feature extraction and spatial hierarchies, and utilizes an end-to-end learning framework for efficient real-time object detection.

1.4.2 Advantages of Proposed System

- Detects with the region-based classification.
- The confident ratio of the detection will be displayed.
- This technique will be efficient in detection.
- Performance and speed of detection will be high.

Chapter 2

Literature Survey

In recent years, researchers have proposed various methods to detect the concentration of formalin content in fruits. But the problem of building a fast and reliable fruit adulteration detection system is a promising study now. It is due to high variation in the appearance of the fruits in field settings, including colour, size, texture, shape and reflectance properties. Furthermore, in the majority of these settings, the fruits are partially abstracted and subject to continually-changing illumination and shadow conditions. In existing projects, they focus on developing a machine learning model which only classifies a particular fruit as good or bad. An IoT based food and formalin detection technique is also developed to detect the presence of formalin. Currently available systems are either expensive or limited to special high-end models or affordable solutions that lack accuracy and robustness. That is what motivated us to focus on implementing a model that would classify the fruit as healthy or not and also detect the formalin content in it. Below is the summary of some research papers in the field of fruit disease detection and formalin detection in fruits.

2.1 Detection of adulteration of Fruits using IoT [1]

An IoT based food and formalin detection technique is developed to detect the presence of formalin using machine-learning approaches. Volatile compound HCHO gas sensor connected with Raspberry pi3 were used to extract the concentration of the formalin as a function of output voltage of any fruit or vegetable and different machine learning algorithms were used to classify the fruit or vegetable based on their extracted features. Supervised machine learning algorithms have been incorporated in this system to accurately predict the correct concentration of formalin at all temperatures which is also able to correctly classify between artificially added and naturally formed formalin. Figure 2.1 shows the block diagram of detection of adulteration in fruits. Grove HCHO which is a semiconductor VOC (Volatile Organic Compound) sensor, is used for detecting the formaldehyde combining with Raspberry-Pi. This sensor is featured to detect gas concentration up to 1 ppm (parts per million).

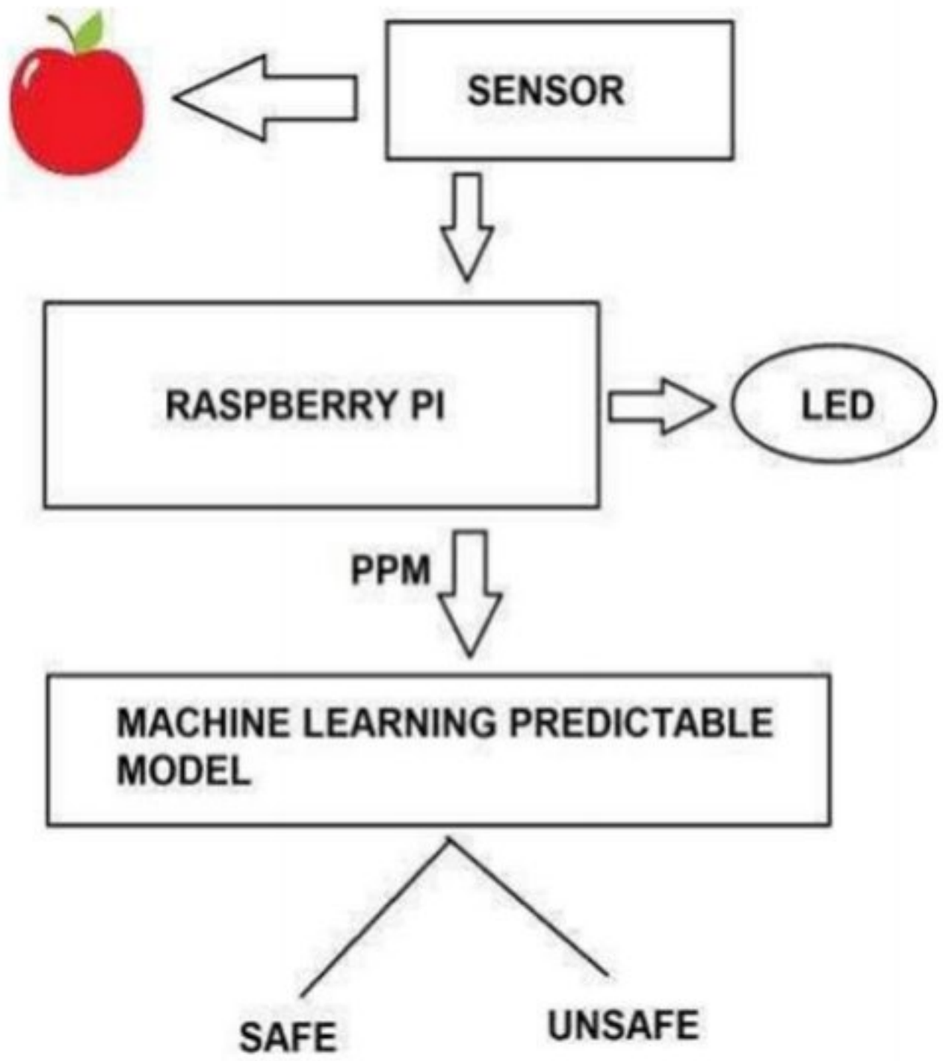

Figure 2.1 : Block block diagram of detection of adulteration in fruits.

As formaldehyde is self-vaporized solution, its presence can be detected by volatile organic compound sensor. The output voltage of the sensor is exponentially proportional with the concentration of formalin present in the fruit sample. Followed by the formalin detection, different voltage drops are measured for different fruit samples as each fruit contains different range of resistance. Firstly by using the extracted features from the data set they implemented the rule based classification model [set of IF-THEN rules], which first categorize the fruit type. Later they performed classification on various algorithms and results were measured. On completely considering the naturally occurring formalin in fruits they have developed this model and trained using the dataset which also depicts the naturally added ppm (parts per million) value along with the additionally added formalin. This system generates the output by making predictions whether the particular fruit item is 'dangerous' or 'safe'.

Drawback: Costly hardware components are required to detect the formalin.

2.2 Fruit Recognition and Grade of Disease Detection using InceptionV5 Model [2]

This work focuses on developing a user-friendly tool which recognizes the level of the disease and grades them accordingly. Inception model uses convolutional neural networks for the classification, which is again retrained using transfer learning technique. The proposed system also grades the fruit based on the percentage of infection. The system is developed in Tensor flow platform. For the proposed work banana, apple and cherry fruit shave been considered. This model is working in many stages and consists of Inception v5 model, Image preprocessing and Transfer Learning techniques.

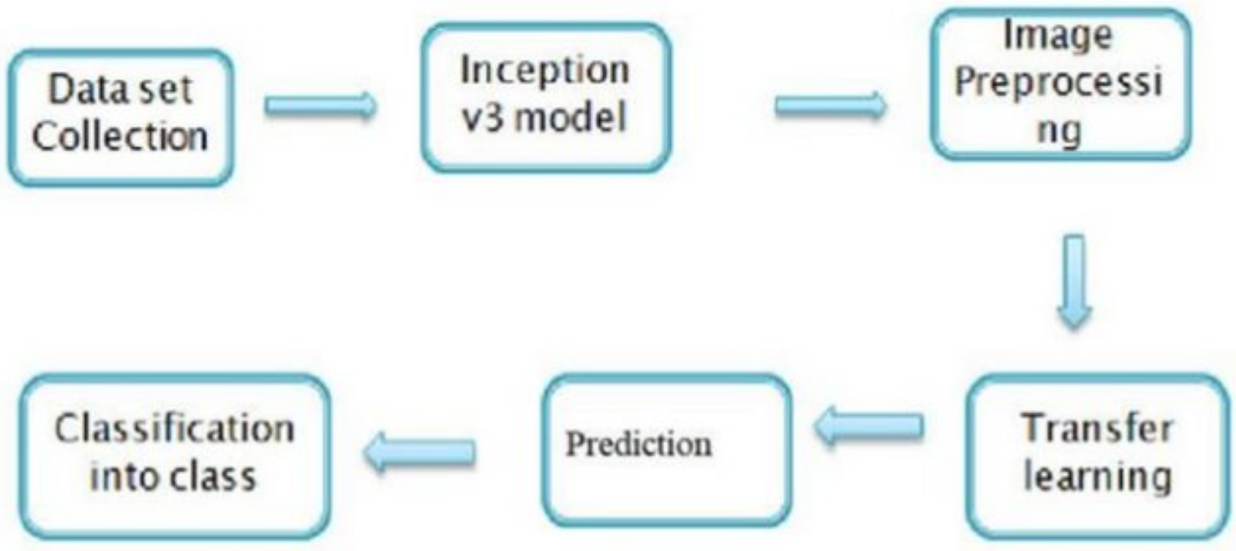

Figure 2.2: Proposed model for fruit recognition and grade of disease detection

Figure 2.2 shows the high-level architecture and the flow of the working model. Inception v5 model is mostly used in image recognition and it is a predefined convolutional neural network. It contains totally 28 layers. Inception already has a dataset called ImageNet which contains the 1000 different object categories and it gives an accuracy of 78.1 percent. So the Inception v5 model is used in this architecture. The proposed work captures the fruit's image and resizes the images into299x299x3 which is a feed into the inception v5 model as an input. It is using the Inception model which has the flexibility of preprocessing. The image can be preprocessed in a range of moderate to complex. The complex preprocessing requires a large set of expensive hardware which is a not required for this model so moderate preprocessing is used. In this moderate preprocessing it does the cropping and the resizing of an image. It resizes tothe299x299x3 which the standard image size in inception v5 model. The prepared Inception model and displacement of the layer that does the last grouping was used. This technique is called Transfer Learning. In Transfer learning, there is no need for removing or changing the lower layers of the Inception model only outer layers

is sufficient. After transfer learning the different classes were predicted and the image was classified to a particular class.

Drawback: Some more features can be added to the model so that the fusion of image processing and deep neural network will be useful for fruit disease detection.

2.3: Identification of Rice Varieties and Adulteration Using Gas Chromatography-Ion Mobility Spectrometry[3]

To solve the problems existing in traditional biochemical methods, such as complex sample pretreatment requirements, tedious detection processes and low detection accuracies with respect to rice species and adulteration, the volatile flavor substances of five kinds of rice are detected using headspace-gas chromatography-ion mobility spectrometry (HGC-IMS) to effectively identify the quality of rice and adulterated rice. The ion migration fingerprint spectra of five kinds of rice are identified using a semi-supervised generative adversarial network (SSGAN). We replace the output layer of the discriminator in a GAN with a softmax classifier, thus extending the GAN to a semi-supervised GAN. We define additional category tags for generated samples to guide the training process. Semi-supervised training is used to optimize the network parameters, and the trained discriminant network is used for classifying HGC-IMS images. The experimental results show that the prediction accuracy of the model reaches 98.00%, which is significantly higher than the rates achieved by other models, such as a decision tree, a support vector machine (SVM), improved SVM models (LS-SVM and PCA-SVM) and local geometric structure Fisher analysis (LGSFA); 98.00% is also higher than the prediction accuracies of the VGGNet, ResNet and Fast RCNN deep learning models. The experimental results also show that the accuracy of HGC-IMS image classification for identifying adulterated rice reaches 97.30%, which is higher than those of traditional chromatographic or spectral methods. The proposed method overcomes the shortcomings of some intelligent algorithms regarding the application of ion migration spectra and is feasible for accurately predicting rice varieties and adulterated rice.

Drawback: If the lighting is different in the testing environments it predicts without efficiency and Finds the adulteration with the colour feature.

2.4 Detection of Adulteration in Food Using Recurrent Neural Network with Internet of Things[4]

Food is an essential need for human survival. (roughout history, food has been recognised as a crucial need for people in order to maintain good health as well as to treat illness. As with all living things, it is one of the most basic necessities that man has as well as those of all other living creatures. In a recent publication, it was said that an extremely affordable, robust, and biocompatible impedance sensor that serves as a fractional-order element has been created and may be used to distinguish milk and tainted milk. A complete study on milk adulteration includes more than 160 academic articles on the topic. A comprehensive study on milk adulteration is available online. Specifically, the goal of this research is to discover various types of milk adulterants, different approaches for detecting each kind of adulterant, as well as the health hazards associated with milk product adulteration. In the proposed project, the fractional-order element would be investigated for its potential use in the detection of milk adulteration. With this fractional-order element-based impedance sensor, you can distinguish between different types of contaminated milk and different types of faking it, which is quite useful in the detection and differentiation of fake and real milk. According to the researchers, they have created a low-cost, user-friendly instrumentation system for detecting milk adulteration. (ey hope to commercialise it soon. An automated sensing system for the detection of synthetic milk, based on a microcontroller, has been created in order to reduce the reliance on specialised labour and to improve efficiency. In this study, an electrical equivalent circuit is built, and the correctness of the circuit is shown by both theoretical and experimental investigation. (e detection of milk adulteration is classified with the use of Recurrent Neural Networks, and the status is updated in the cloud server with the help of the Internet of (ings and Recurrent Neural Networks. It is estimated that the proposed work will have an accuracy rate of 92.31 percent, a sensitivity rate of 75.23 percent, and a specificity rate of 90.12 percent, all of which are higher than the present rate.

Drawback: The dataset used here are static images and k means clustering cannot handle noisy data and outliers which turns out to give less accurate output.

2.5 Fruit Disease Classification and Identification using Image Processing[5]

Fruit Industry is the largest industry of India. Due to lack of maintenance, inappropriate manual inspection the fruit Disease causes huge losses in yield, quality and quantity. Manual inspection is tedious and time consuming process. The Figure 2.3 shows the model for fruit disease classification and detection using image processing.

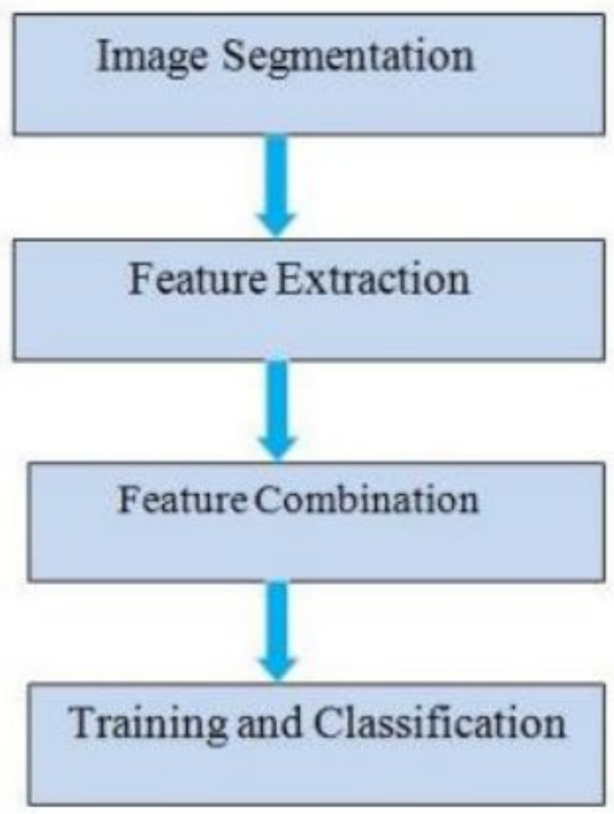

Figure 2.3 : Proposed model for fruit disease classification and identification

An image processing approach is proposed for apple fruit disease identification and categorization using different color, texture and shape feature combination. The basic steps of the proposed approach are image segmentation, extraction of features (color, texture and shape), feature combination and finally apple disease identified and classified using multi-class support vector machine into diseased or normal class. Our proposed technique experimentally verified and validated. The accuracy of the proposed approach is achieved up to 96%.

Drawback: The SVM gives lesser accuracy than CNN.

Chapter 3

System Requirements and Specifications

3.1 Hardware and Software requirements

Hardware Requirements:

- Processor: state of the art i3
- Ethernet connection(LAN)OR a wireless adapter(Wi-Fi)
- Memory(RAM):Minimum 4 GB ; Recommended 4 GB or above
- Storage: 4GB or higher free drive space/cloud space

Software Requirements:

Python 3	Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace.
Pip	Pip is a package-management system written in Python used to install and manage software packages.
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.
PyTorch	An open source machine learning library used for application such as Computer Vision and Natural Language Processing.
TensorFlow	It is a free and open-source software library for dataflow and differential programming across a range of tasks.
Web Framework	Flask:A web application framework written in Python.

3.2 User Requirements

- User should have camera to capture the image of the dall and upload it in the web application.
- After the uploading the photo, based on texture and color model detects the classification.

3.3 Functional and Non Functional Requirements

Functional Requirements:

- User must be able to upload a image or video of dall.
- System will extract the required features from the image.
- System will apply CNN and YOLO model to classify the dall as pure or impure dall.

Non Functional Requirements:

- Performance Requirements: Application requires working system with the specified software and hardware requirements.
- Reliability: The system should classify adulterated fruit correctly. The system should not provide error in prediction due to false negatives.
- Availability: The system must operate properly when it is requested for use.
- Maintainability: Maintenance of the application is easy and economical.
- Portability: This system can be run on any operating system including Windows, Linux and Mac.

Chapter 4

System Architecture and Implementation

4.1 Methodology

Brightness equalization. When taking pictures to acquire images, the uneven illumination caused shadows in the background, which will affect the edge detection effect and also lead to differences in the histogram distribution of color space between segmented images, reducing the data uniformity. Hence, it is necessary to identify and subtract the background, and adjust the image to equalize the brightness.

Edge detection. Convert the image to grayscale, use low-pass filtering methods such as Gaussian blur to remove the image noise. Take morphological operations like erode and dilate to separate adjoint sorghum grains. Then find the outside contours of each connected component using the Canny edge detector . The corresponding low and high thresholds for gray scale gradient are 100 and 200 respectively.

Image segmentation. Approximate the contours with polygons using Douglas–Peucker algorithm [13]. Calculate the minimal up-right bounding rectangle of each polygon, manually set thresholds for width and height to filter noise points, which was set as width multiply height no less than 1700 pixel2 in practice. To preserve the size information of the grains, we used square with fixed side length (140 pixels) to segment each granule. The square has the same center as the bounding rectangle, and its side length is larger than the maximum value among the long sides of the rectangles.

4.1.1 YOLO v5: Better, not Faster, Stronger

The official title of the YOLO v2 paper seemed as if YOLO was a milk-based health drink for kids rather than an object detection algorithm. It was named “YOLO9000: Better, Faster, Stronger”.For its time YOLO 9000 was the fastest, and also one of the most accurate algorithms. However, a couple of years down the line and it’s no longer the most accurate with algorithms like RetinaNet, and SSD outperforming it in terms of accuracy. It still, however, was one of the fastest. But that

speed has been traded off for boosts inaccuracy in YOLO v5. While the earlier variant ran on 45 FPS on a Titan X, the current version clocks about 30 FPS. This has to do with the increase in complexity of underlying architecture called Darknet.

4.1.2 Darknet-53

YOLO v2 used a custom deep architecture darknet-19, an originally 19-layer network supplemented with 11 more layers for object detection. With a 30-layer architecture, YOLO v2 often struggled with small object detections. This was attributed to the loss of fine-grained features as the layers down sampled the input. To remedy this, YOLO v2 used identity mapping, concatenating feature maps from a previous layer to capture low-level features.

However, YOLO v2's architecture was still lacking some of the most important elements that are now stapled in most state-of-the-art algorithms. No residual blocks, no skip connections, and no up sampling. YOLO v5 incorporates all of these.

First, YOLO v5 uses a variant of Darknet, which originally has a 53 layer network trained on Image net. For the task of detection, 53 more layers are stacked onto it, giving us a 106 layer fully convolutional underlying architecture for YOLO v5. This is the reason behind the slowness of YOLO v5 compared to YOLO v2. Figure 4.1 shows what the architecture of YOLO now looks like.

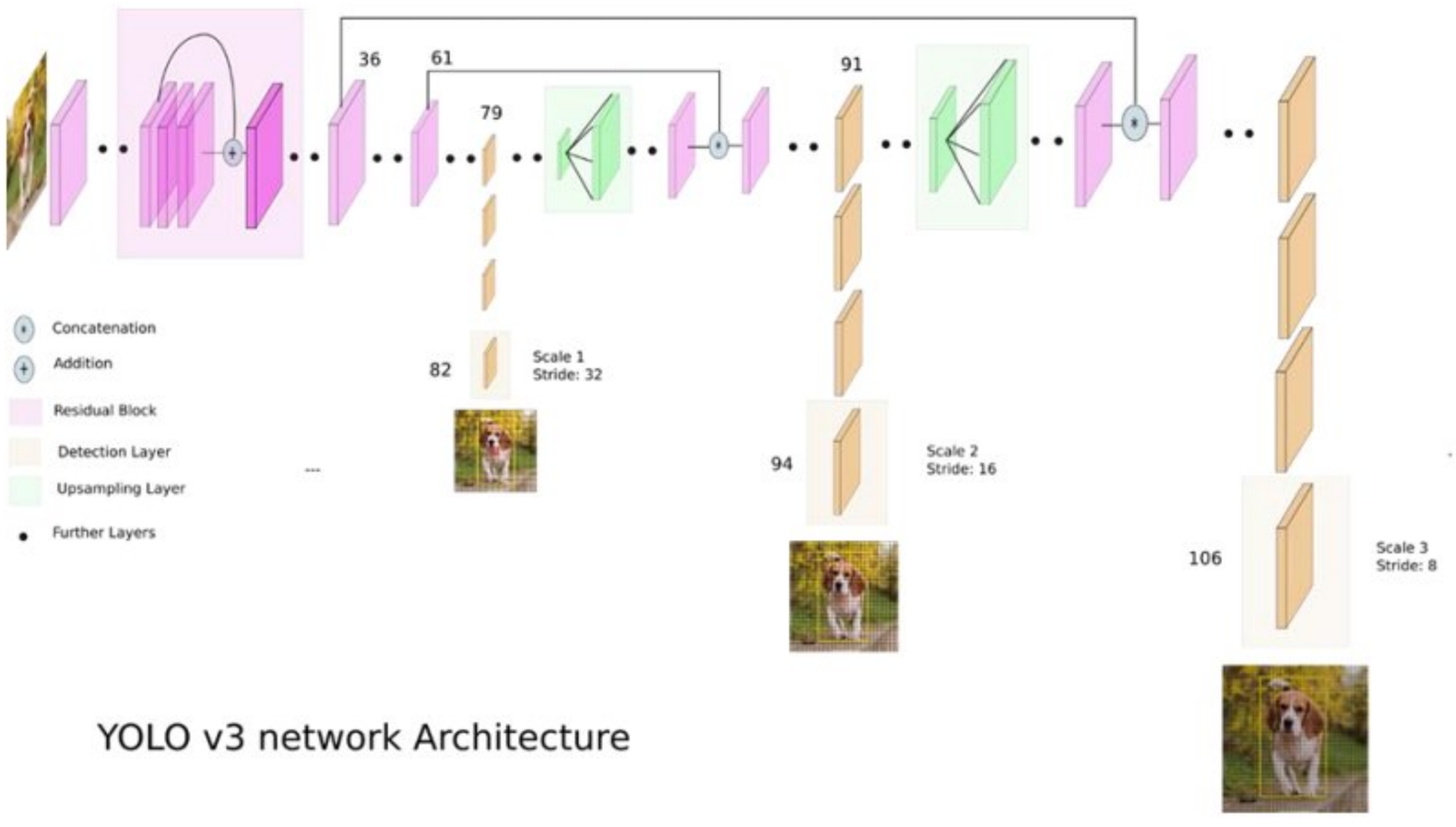

Figure 4.1 : Yolo v3 network Architecture

4.1.3 Detection at three Scales

The newer architecture boasts of residual skip connections and upsampling. The most salient feature of v5 is that it makes detections at three different scales. YOLO is a fully convolutional network and its eventual output is generated by applying a 1 x 1 kernel on a feature map. In YOLO v5, the detection is done by applying 1 x 1 detection kernels on feature maps of three different sizes at three different places in the network.

The shape of the detection kernel is 1 x 1 x (B x (5 + C)). Here B is the number of bounding boxes a cell on the feature map can predict, "5" is for the 4 bounding box attributes and one object confidence, and C is the number of classes. In YOLO v5 trained on COCO, B = 3 and C = 80, so the kernel size is 1 x 1 x 255. The feature map produced by this kernel has identical height and width to the previous feature map and has detection attributes along with the depth as described above.

4.2 System Design

High-level design (HLD) explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces. The HLD uses possibly nontechnical to mildly technical terms that should be understandable to the administrators of the system. In contrast low level design further exposes the logical detailed design of each of these elements for programmers. High level design is the design which is used to design the software related requirements. In this chapter complete system design is generated and shows how the modules, sub modules and the flow of the data between them are done and are integrated. It consists of very simple phases and shows the implementation process.

Design Consideration: The design consideration briefs about how the system behaves for the boundary environments and what action should be taken if the abnormal case happens. Some of the design considerations are data collection, pre-processing methods and Classification and prediction.

The design considerations are formulated to bring to the attention of the designers in applying the universal accessibility design principles and requirements to buildings and facilities. They can also be used to identify barriers in existing systems.

The proposed system has the following steps for Dall adulteration detection is

1. Image Pre-Processing.

2.Identification

3.Feature Extraction

4.Pure and Impure dall Recognition

5.detecting the type of dall.

4.2.1 Image Pre-processing

The image processing is a mechanism that focuses on the manipulation of images in different ways in order to enhance the image quality. Images are taken as the input and output for image processing techniques. It is the analysis of image-to-image transformation which is used for the enhancement of image. Firstly, we convert RGB image to grey scale image. It helps to reduce the complexity in the image and also make the work easy. Then by using min-max scalar method converts the Gray scale values into binary values. The obtained binary values are taken as the input for the further process. In the obtained binary matrix consider one value regions as white and zero value region black. By using these values, the region of interest can be identified. So that the values are useful for feature extraction and identification of region of interest.

4.2.2 Identification

In this stage identify the region which needs to proceed for further process, it is involved in the identification of the particular region of the image that is used for the further process like feature extraction and classification of the images. The output of the pre-processing step is given as the input for the identification process. This process is based on the binary values obtained in the pre-processing step. The region with black are consider as region of interest. The region of interest obtained by the pre-processing of the images. That region is considered as proceeding part of the

image from which kesar and thur dall will be identified. The identified dall with images are given to the feature extraction process.

4.2.3 Feature Extraction

In this stage extract the required feature from the identified region which are obtained from the previous step. That region is compressed by converting reduced size matrix to control over fitting. The reduction of the matrix size helps in reduce the memory size of the images. Then the flattening process is applied to the reduced matrix, in which the reduced matrix is converted to one-dimension array, which is used for final detection.

4.2.4 Pure and Impure dall Recognition

In this stage one-dimension array is used for final classification process. The output image obtained from feature extraction is given as input to this process. Where continuous classification of all the feature obtained from previous stage. YOLO Convolutional neural network is applied in this process. There are three layers in the YOLO they are input layer, hidden layer, output layer. Each node of input layer has value from one-dimension array which represent the feature from extracted region. That is send to hidden layer. Multiple features are getting from input layer is undergoes multiple iteration in hidden layer. Finally get the predictive values by applying SoftMax activation function to it. Finally, get some output values from this process and these values undergoes further process. The highest value in the predictive value is considered as output identified as Kesar or thur dall. By using these methods, the Thur and Kesar dall will be detected by considering highest accuracy values.

4.2.5 Dall Adulteration Detection

For detecting Grain Adulteration data is trained using Yolo Model. YOLO, in a single glance, takes the entire image and predicts for these boxes the bounding box coordinates and class probabilities. YOLO's greatest advantage is its outstanding pace, it's extremely fast, and it can handle 45 frames per second .Amongst the three versions of YOLO 3 and 5, is fastest and more accurate in terms of detecting small objects. The proposed algorithm, YOLO consists of total 106 layers . The architecture is made up of 3 distinct layer forms. Firstly, the residual layer which is formed when activation is easily forwarded to a deeper layer in the neural network. In a residual setup, outputs of

layer 1 are added to the outputs of layer 2. Second is the detection layer which performs detection at 3 different scales or stages. Size of the grids is increased for detection. Third is the up-sampling layer which increases the spatial resolution of an image. Here image is up sampled before it is scaled. Also, concatenation operation is used, to concatenate the outputs of previous layer to the present layer. Addition operations used to add previous layers. The pink coloured blocks are the residual layers, orange ones are the detection layers and the green are the up-sampling layers.

YOLO predicts 3 different scales of prediction. The detection layer is used to detect feature maps of three different sizes, with strides 32, 16, 8 respectively. This means that detections are made on scales of 13 x 13, 26 x 26 and 52 x 52 with an input of 416 x 416. The working of Yolo is mentioned below. Darknet: This algorithm is implemented using an open-source neural network framework i.e., Darknet which was developed in C Language and CUDA technology to render speedy calculations on a GPU necessary for real-time predictions.

• DNModel.py: Darknet Model file is a computer vision code used for building the model using the configuration file and it appends each layer.

• Util.py: Contains all the formulas used.

• imageprocees.py: Required to perform the image processing task. It takes all the input images to resize them and perform Up-sampling, also performs transpose function.

• detect.py: The main code which is run to perform object detection. This code uses all the abovementioned files to perform object detection. Performs all the functions according to the YOLOconcept.

The Figure 4.3 shows the system architecture for the proposed system. The input image is preprocessed and converted to gray scale image to get the clear vision of the image. Then it will be converted into binary values. In the next step identifies the part which needs to proceed further. Then required feature are extracted by In the CNN convolution layer. By passing those features into different layer of CNN we get compressed image, that feature is used for detection of Thur Dall(pure) and Kesar dall(adulterated) using SoftMax activation function.

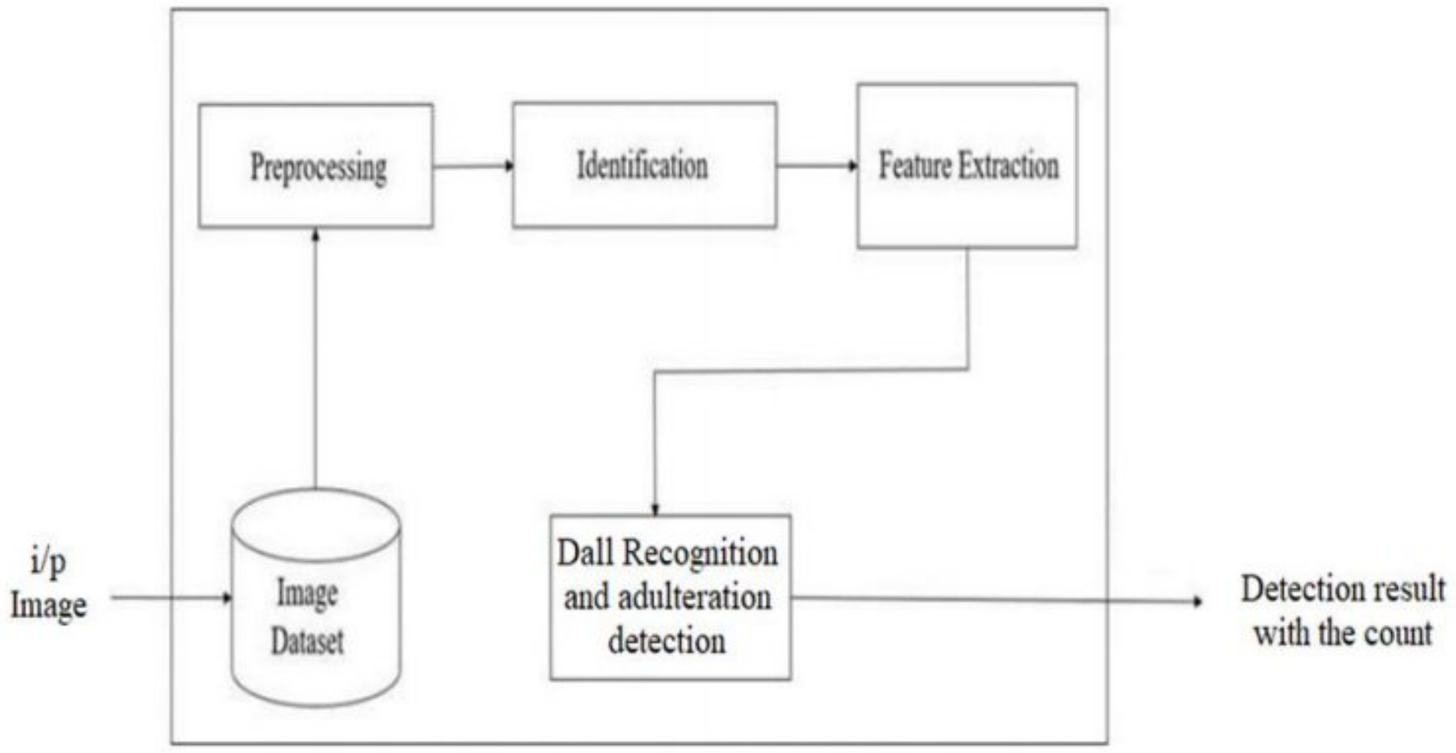

Figure 4.3 : System Architecture of the Dall Adulteration detection

4.3 Module Specification

Module Specification is the way to improve the structure design by breaking down the system into modules and solving it as independent task. By doing so the complexity is reduced and the modules can be tested independently. The number of modules for our model is three, namely pre-processing, identification, feature extraction and detection.

In the data preprocessing phase conversion of RGB to gray conversion and that gray scale value is converted to binary value for efficient calculation of features in next phase. The second phase is to identify the required part of the image from which we need to detect the Dall classificaiton and detects with the region and confidence ratio. In the binary values zero is considered as white and one is considered as black.

The region with black is identified as required region to extract feature in the next phase. For next process convolution neural network algorithm is applied. The third phase is to extract the feature from the identified region in the convolution layer of CNN. This includes the part of image which is considered as a required part of image which is used for the detection of the Dall detection in the basis of thur or adulterated kesar dallx. All the required information of the image is converted into pixel and stored in the form of image.

In the final phase each feature from the previous phase is considered these features are extracted from the convolution layer of the CNN and send to fully connected layer. Apply Convolutional neural network neural network to those features by continuous iteration. In the hidden layer of YOLO each feature is efficiently identified and finally get the predictive value for output by using SoftMax activation function. Based on that value kesar and thoor dall will be detected.

4.4 Data Flow Diagram

A data flow diagram (DFD) is graphic representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context level DFD first which shows the interaction between the system and outside entities.

Data flow diagrams show the flow of data from external entities into the system, how the data moves from one process to another, as well as its logical storage. There are only four symbols:

1. Squares representing external entities, which are sources and destinations of information entering and leaving the system.

2. Rounded rectangles representing processes, in other methodologies, may be called 'Activities', 'Actions', 'Procedures', 'Subsystems' etc. which take data as input, do processing to it, and output it.

3. Arrows representing the data flows, which can either, be electronic data or physical items. It is impossible for data to flow from data store to data store except via a process, and external entities are not allowed to access data stores directly.

4. The flat three-sided rectangle is representing data stores should both receive information for storing and provide it for further processing.

5. It is also used to analyses a particular problem and the solution for it in steps.

6. A user loads the data and the system reads the data provided by the user.

7. Based on feature extraction and classifier the model will be trained and tested.

4.4.1 Data Flow Diagram for Pre-processing.

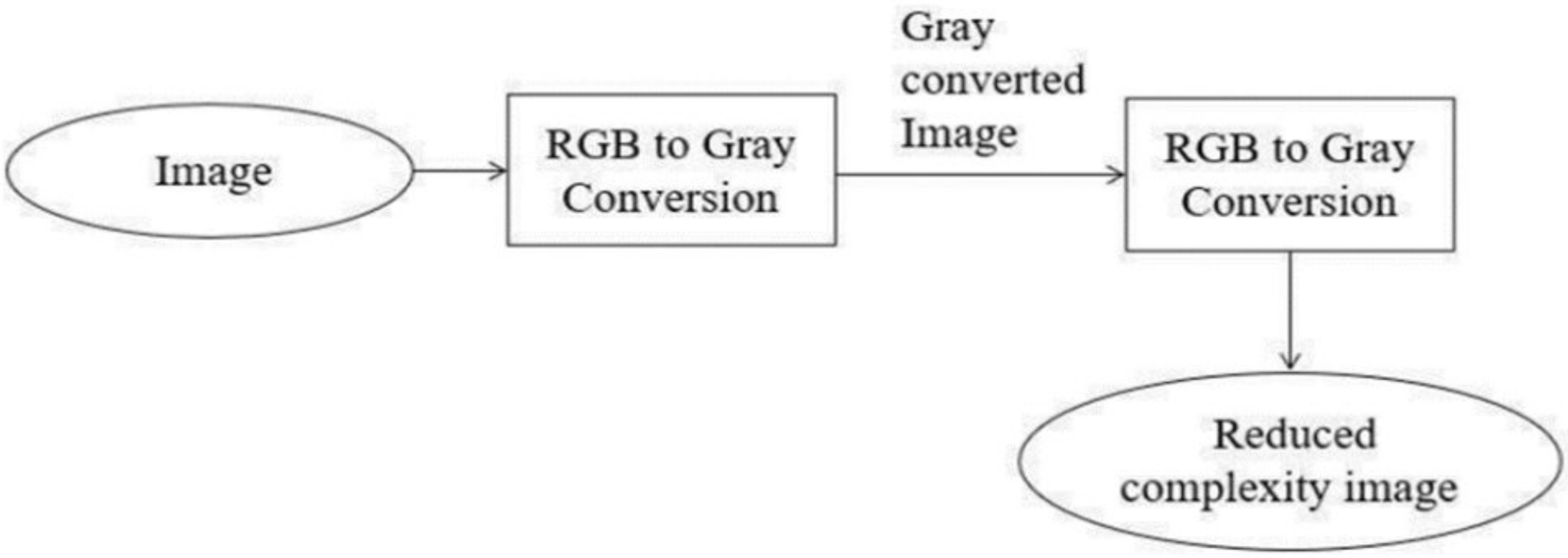

Figure 4.4: Data flow diagram for pre processing

The Figure 4.4 shows that the image is given as input. As we giving the colour image so that RGB image is converted into gray scale values to reduce complexity in the image. For efficient feature extraction gray scale values are converted into binary values. Then the image with reduced complexity is send to the next process.

4.4.2 Data Flow Diagram for Identification

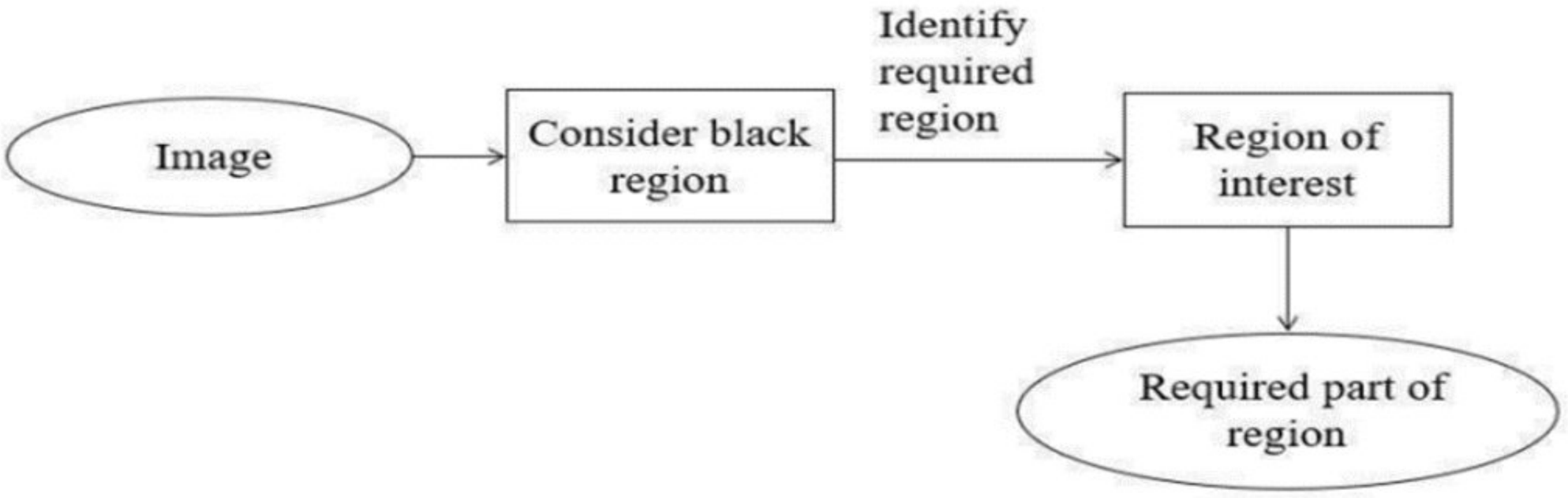

Figure 4.5 Data flow diagram for Identification

The Figure 4.5 shows that the image with reduced complexity is considered as input. Here the region with the value of one is considered as black that region is considered for next process.

4.4.3 Data Flow Diagram for Feature Extraction

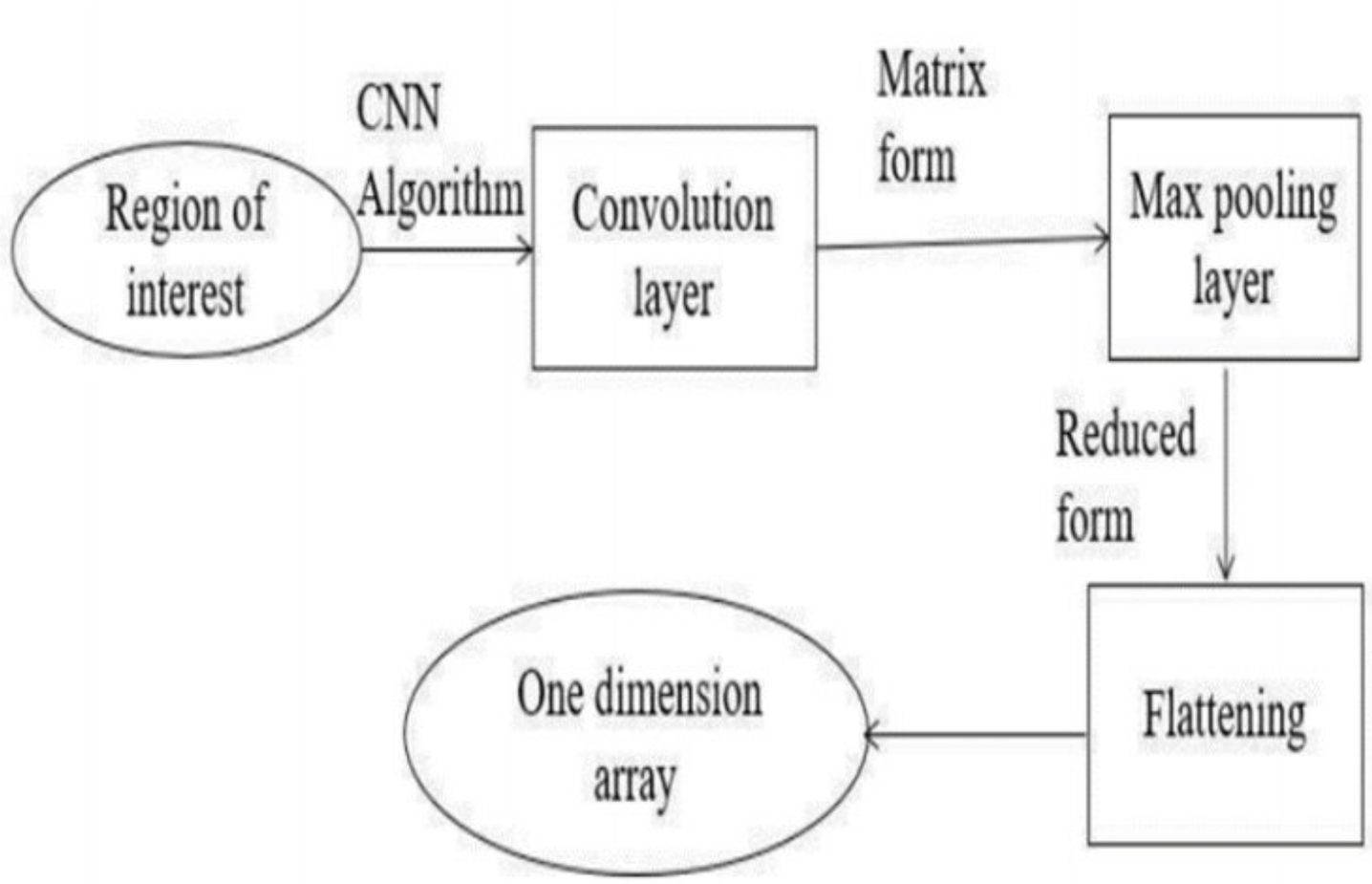

Figure 4.6 Data flow diagram for Feature Extraction

The Figure 4.6 shows that the region of interest from the identification step is considered as input. The region of interest is obtained from converting RGB color image to the gray scale image by using minmax scalar method. For that region CNN algorithm is applied. A CNN consists of an input layer and an output layer, as well as multiple hidden layers between them. The hidden layer basically consists of the convolution layer, pooling layer, relu layer and fully connected layers. In this the RGB color image is converted into gray scale image by using minmax scalar method. The binary valued image is given as input to the convolution layer. In the convolution layer binary matrix is multiplied with filter to extract feature from the region.

In this the binary valued images are used which reduces the complexity. Then is send to max pooling layer where the matrix size is converted into reduced matrix. The output of the pooling layer is given as input to the flattening layer. Then the reduced matrix is converted to one dimension array after sending into flattening layer. The one-dimension array is given for further classification to obtain the accurate result. The image having highest accuracy value that is considered as the output. Finally, the Kesar or Thur dall is detected by using above four layers.

4.4.4 Data Flow Diagram for Classification and Detection

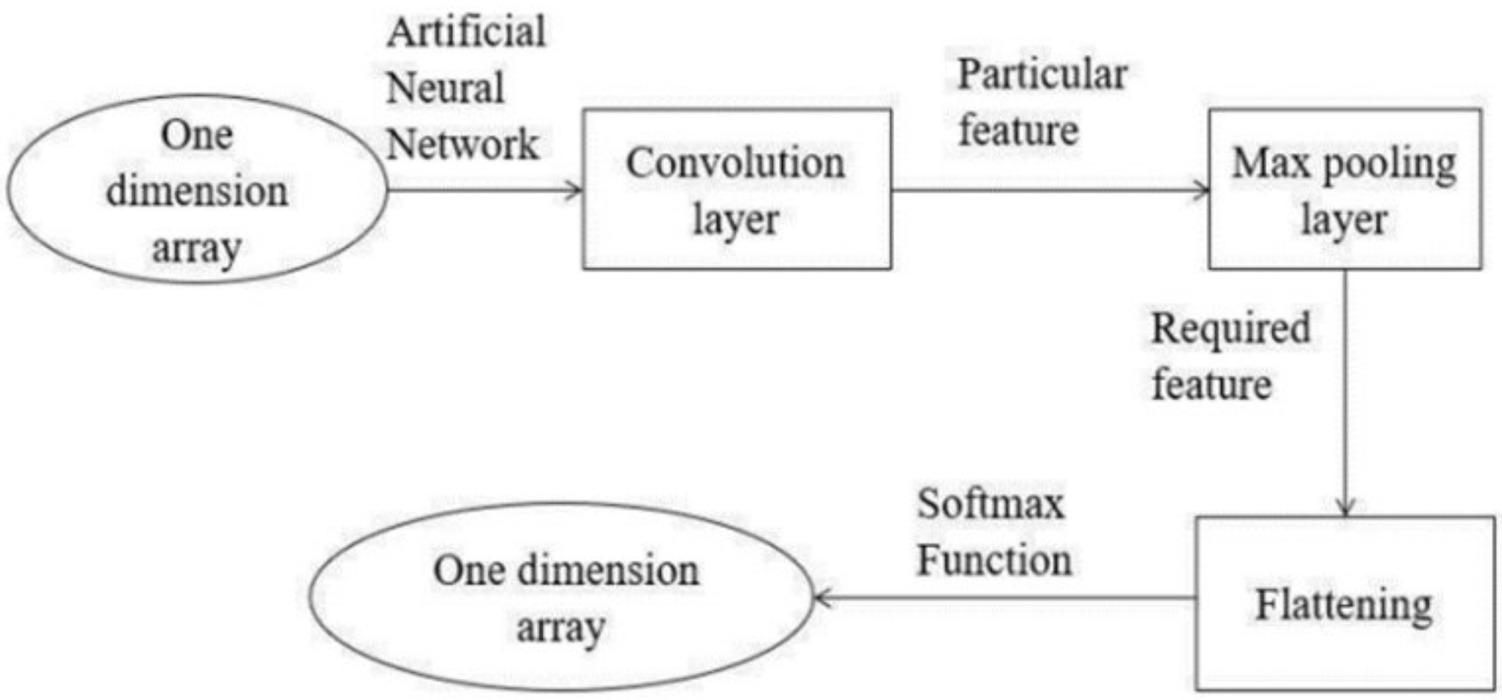

Figure 4.7 Data flow diagram for Classification and Detection

The Figure 4.7 shows that the one-dimension array is send to fully connected layer of CNN. Artificial neural network method is applied to this layer. Firstly, one-dimension array is sent to input layer. Some particular feature which is required for the detection is identified by the hidden layer of ANN. The continue connection from hidden layer to output layer will help to identify accurate result. By considering all the features output layer gives the result with some predictive value. These values are calculated by using SoftMax activation function. SoftMax activation function provides predictive values. Based on the prediction value the final result will be identified. The highest value of prediction is identified as Kesar or Thur dall detection.

4.4.5 Detailed description using flowchart

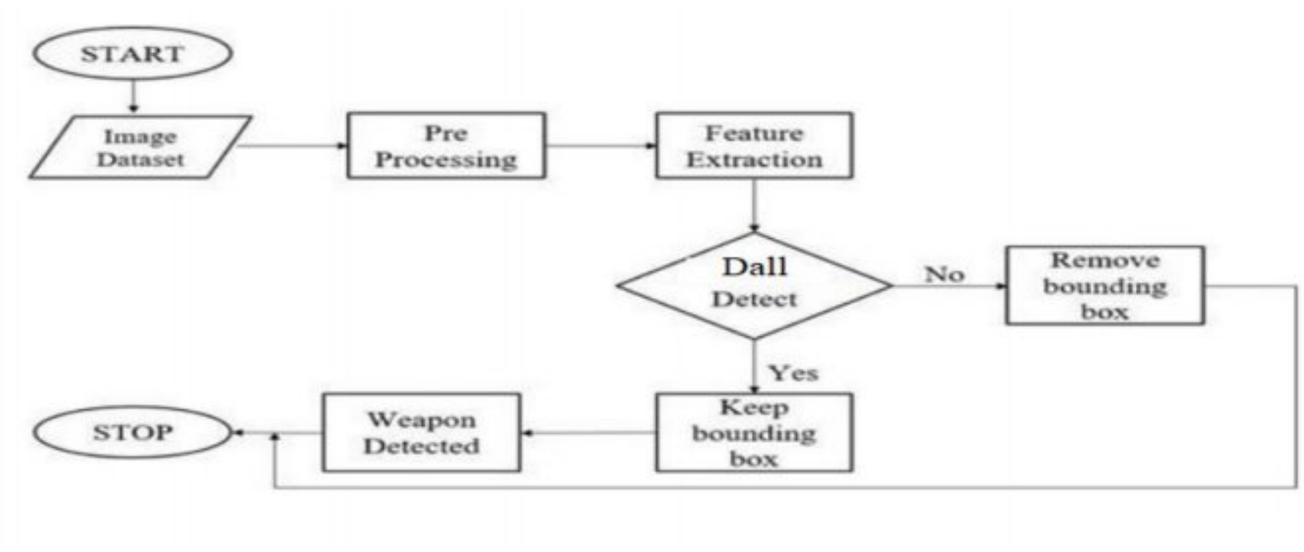

Figure 4.8 Flowchart for the Kesar and Thur Dall detection using deep learning

The Figure 4.8 shows the flowchart of the weapon and militant detection using deep learning process. Image dataset which contain weapon and militants are loaded into it then preprocessing of the image is done by RGB to grayscale conversion. Feature Extraction is done by passing to CNN layers. If the weapon and militant is detected it gives the message that the weapon and militant type present in it.

4.4.6 Flowchart for Data Acquisition

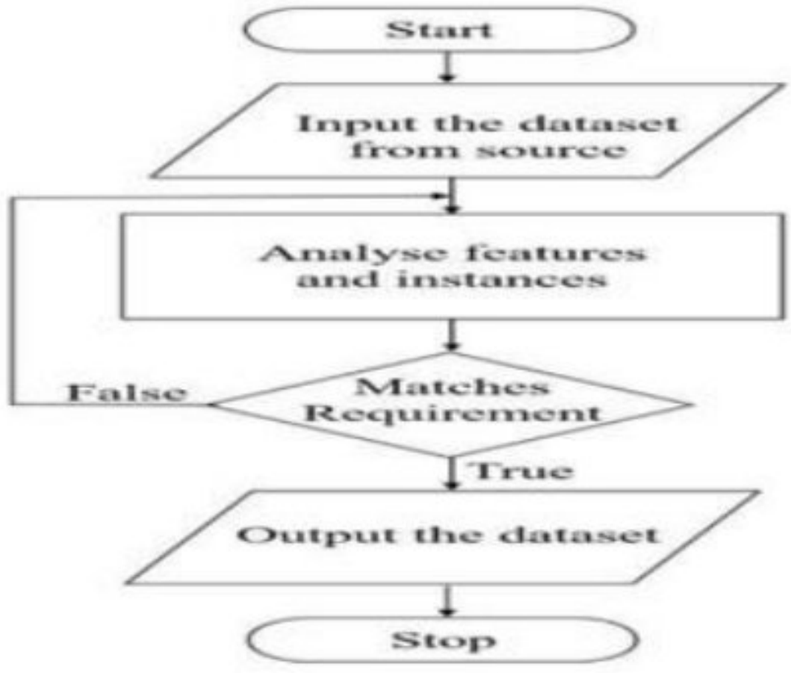

Figure 4.9: Flowchart for data acquisition

The flowchart for collecting data is as depicted in the Figure 4.9. The data set is collected from a source and a complete analysis is carried out. The image is selected to be used for training/testing purposes only if it matches our requirements and is not repeated.

4.5 Structural Design

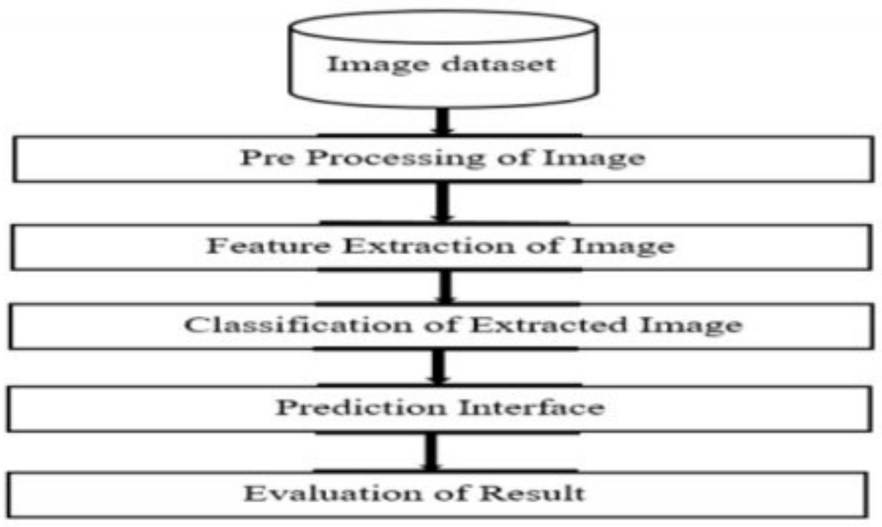

Figure 4.9: Structural chart of the system

The Figure 4.9 shows that the proposed system involves the following steps. First step involves pre-processing of captured images. The pre-processed image undergoes feature extraction, where various features of the weapon and militant types are extracted and certain algorithms are applied. The data that is stored is compared with the pre-processed image and approximate result is generated.

4.5 Pseudo code

4.5.1 YOLOv5 Model:

Step 1: Load the pre-trained YOLOv5 model using PyTorch.

Step 2: Define functions for preprocessing images before feeding them into the model.

Step 3: Implement object detection algorithms to identify pure and impure dal grains.

Step 4: Apply Non-Maximum Suppression (NMS) to refine detection results and remove duplicate detections.

Step 5: Calculate the count of pure and impure dal grains based on detection results

4.5.2 The Algorithm Explanation:

The YOLOv5 (You Only Look Once version 5) model is an advanced deep learning algorithm designed for object detection tasks. Here's a description of the steps involved when using the YOLOv5 model to detect and differentiate between pure and impure dal grains:

Step 1: Load the Pre-Trained YOLOv5 Model:

- Utilize the PyTorch library to load a pre-trained YOLOv5 model that has been trained on a large dataset to recognize various objects.
- PyTorch provides efficient and convenient tools for model loading and deployment, enabling seamless integration of the YOLOv5 model into the application.

Step 2: Define Preprocessing Functions:

- Prepare image data for object detection by defining preprocessing functions.

- Tasks include resizing images to a fixed size required by the model, normalizing pixel values to a certain range, and applying any other necessary transformations to enhance model performance.
- These preprocessing steps ensure that the input images are in the correct format and ready for effective processing by the YOLOv5 model.

Step 3: Implement Object Detection Algorithms:

- Feed the preprocessed images into the loaded YOLOv5 model to perform inference and detect objects within the images.
- The model outputs bounding box coordinates, class predictions (e.g., pure or impure dal grains), and confidence scores for each detected object.
- Adjust the model or its parameters as needed to optimize performance for the specific task of identifying and classifying dal grains based on training data.

Step 4: Apply Non-Maximum Suppression (NMS):

- Refine the detection results using Non-Maximum Suppression (NMS) to eliminate redundant bounding boxes that cover the same object.
- NMS selects the bounding box with the highest confidence score for each detected object while removing others that have a high overlap (measured by Intersection Over Union or IoU) with it.
- This step ensures that each detected object is represented by only the most probable bounding box, reducing duplicates and improving the clarity of the output.

Step 5: Calculate the Count of Pure and Impure Dal Grains:

- Quantify the results from the detection phase by counting the number of detections classified as 'pure' and 'impure' dal grains based on the labels assigned during object detection.
- This quantitative data provides insights into the proportion of pure vs. impure grains in a sample, which can be valuable for quality control in food processing and packaging industries.

4.6 Dataset

```
Procedure YOLOv5_Object_Detection
    Input: Image, Model_Weights, Configuration_Parameters
    Output: Detected_Objects (list of bounding boxes, class labels, confidence scores)
    Begin
        // Preprocess the input image
        Image_Processed = preprocess_image(Image, Configuration_Parameters)
        // Load model with pre-trained weights
        Model = load_model(Model_Weights)
        // Forward pass: Run the image through the network
        Predictions = Model.forward_pass(Image_Processed)
        // Postprocessing: Convert raw predictions to bounding boxes, class labels, and scores
        Detected_Objects = postprocess_predictions(Predictions,
Configuration_Parameters)
        // Filter outputs using non-max suppression to remove redundant boxes
        Final_Detections = non_max_suppression(Detected_Objects,
Configuration_Parameters)
        Return Final_Detections
    End
EndProcedure
Function preprocess_image(Image, Parameters)
    // Resize image and normalize pixel values
    Image_Resized = resize_image(Image, Parameters.input_size)
    Image_Normalized = normalize_image(Image_Resized,                Parameters.mean,
Parameters.std)
    Return Image_Normalized
EndFunction
Function load_model(Weights)
```

```
    // Load the neural network model with given weights
    Model = create_cnn_model()
    Model.load_weights(Weights)
    Return Model
EndFunction
Function postprocess_predictions(Predictions, Parameters)
    // Decode predictions to readable bounding boxes and class probabilities
    Objects = []
    For Each Prediction in Predictions
        Box = extract_box(Prediction)
        Class_Label = determine_class(Prediction)
        Confidence_Score = calculate_confidence(Prediction)
        If Confidence_Score > Parameters.confidence_threshold
            Objects.append((Box, Class_Label, Confidence_Score))
        EndIf
    EndFor
    Return Objects
EndFunction
Function non_max_suppression(Objects, Parameters)
    // Apply NMS algorithm to filter overlapping boxes
    Final_Detections = []
    While Objects is not empty
        Highest_Score_Object = select_highest_confidence(Objects)
        Remove Similar_Objects within IoU threshold around Highest_Score_Object from
Objects
        Final_Detections.append(Highest_Score_Object)
    EndWhile
    Return Final_Detections
EndFunction
```

Chapter 5

Results and Discussion

The YOLOv5 model, a state-of-the-art deep learning algorithm for object detection, was employed to detect and differentiate between pure and impure dal grains in images. Here's a summary of the model's performance and the ensuing discussion.

5.1 Model Performance

- The YOLOv5 model demonstrated high accuracy and efficiency in detecting dal grains within images. Pre-trained weights were utilized, ensuring the model's familiarity with various object classes, including different types of dal grains.
- The model effectively identified and localized both pure and impure dal grains, providing bounding box coordinates, class predictions, and confidence scores for each detection.
- The provided image displays the outcomes of a model tasked with distinguishing between different types of lentils: dal and not dal. Each classification is accompanied by a confidence score, such as "dal0.92," indicating the model's certainty in its identification. For instance, a score of 0.92 suggests the model is 92% confident the image corresponds to toor dal. This indicates that the model harbors some degree of uncertainty across all classifications. Potential factors contributing to this uncertainty include image blurriness or ambiguity, as well as the visual similarity between the two lentil types, potentially leading to model confusion.
- Overall, the image illustrates the model's endeavor to classify lentil varieties. While the model exhibits considerable confidence in certain instances, there remains a possibility of error, particularly when discerning subtle visual disparities between lentil types.

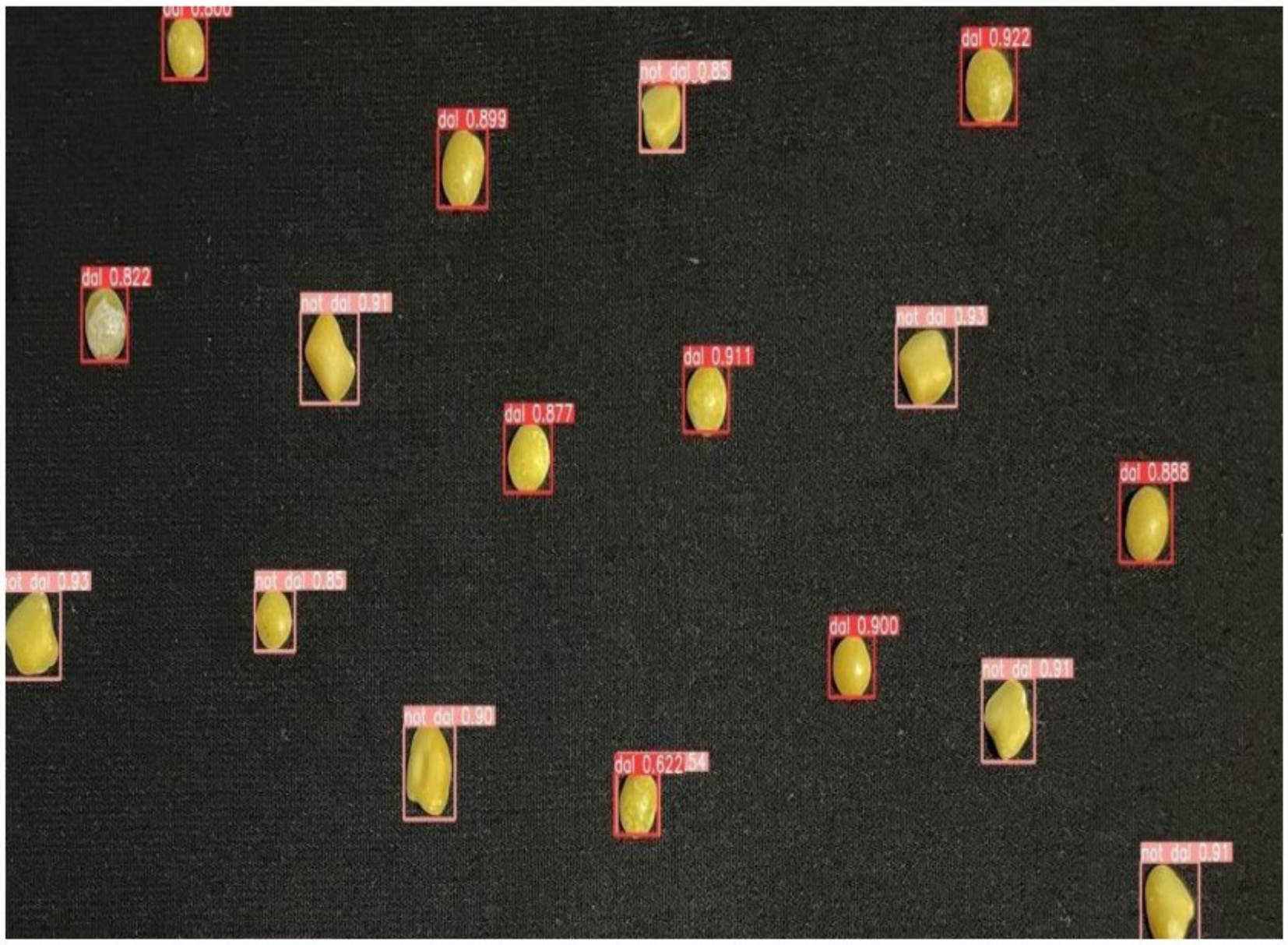

Figure 5.1: Mixture of grains

The Figure 5.1 unveils the inner workings of a machine learning model tasked with identifying different lentil types. Each lentil picture receives a classification label ("dal" or "not-dal") alongside a confidence score (like "dal0.92"). This score reflects the model's certainty in its identification, with 0.92 in this example indicating a 92% confidence that the image depicts a specific dal variety. Interestingly, none of the scores reach a perfect 100%, hinting at a degree of model uncertainty across all classifications. This hesitancy could be attributed to various factors. The image quality might play a role, with blurry or unclear pictures posing challenges for the model's analysis. Additionally, certain lentil varieties might share similar visual characteristics, causing confusion within the model's decision-making process. In essence, this image serves as a snapshot of the model's attempt to categorize lentils. While demonstrating confidence in some instances, the model's accuracy might be limited by image quality or the inherent visual similarities between lentil types. This glimpse into the model's performance allows us to identify areas for improvement, potentially through better image acquisition or further model refinement.

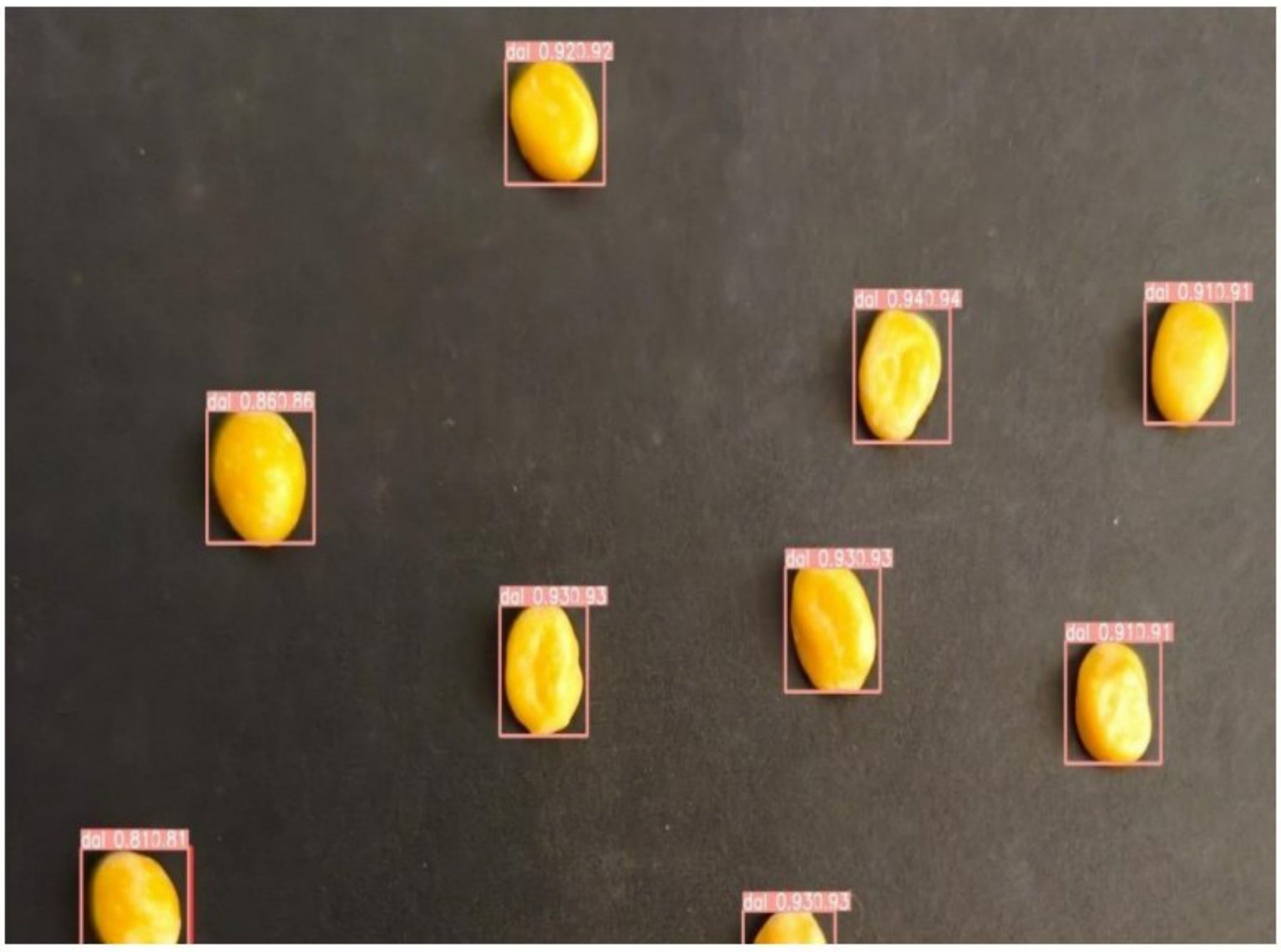

Figure 5.2: Dal

The Figure 5.2 provides a detailed glimpse into a quality control process for lentils (dal). The image showcases several piles of lentils arranged against a stark black background. Each pile likely represents a different variety or sample of lentils, distinguished by varying colors and sizes. The lentils exhibit a captivating spectrum of colors, ranging from a light, pale yellow to a rich, golden hue. Some lentils feature intriguing speckles or darker streaks, adding to the visual diversity. Moreover, the sizes of the lentils vary significantly, with some piles consisting mainly of larger lentils, while others offer a blend of small, medium, and large lentils. Above each pile, text labels provide crucial information, presumably indicating the variety or sample ID, alongside measurements for size and color consistency. For instance, the label "dal 0.92" suggests a meticulous assessment of the lentils, possibly denoting their accuracy in terms of size and color consistency. Overall, the image presents a meticulous quality control process for lentils, hinting at the possibility of automated sorting based on size and color consistency. This process ensures that only the highest quality lentils, meeting specific standards, move forward for packaging and distribution.

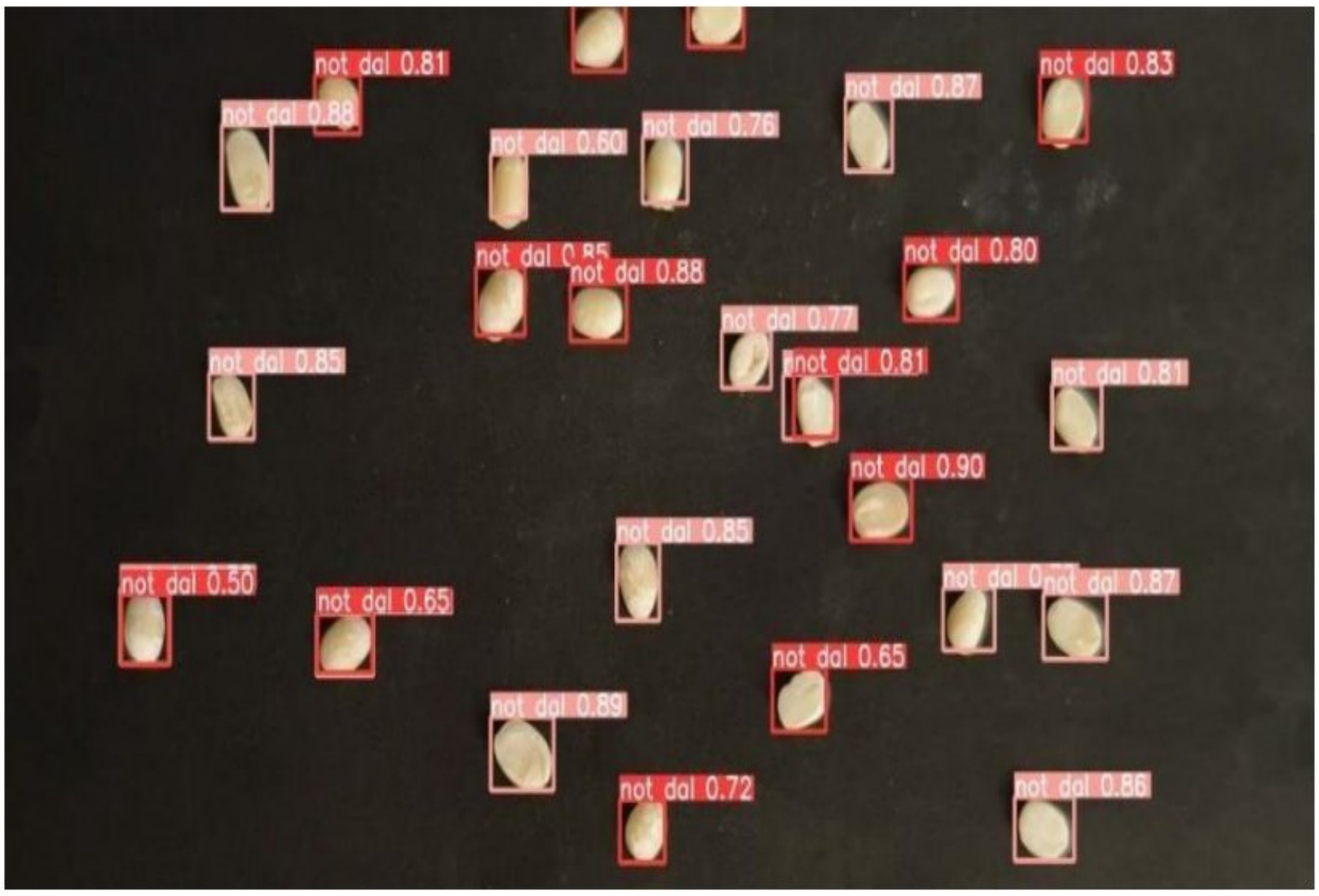

Figure 5.3: Not dal

In the Figure 5.3 it shows a group of white beans spread out on a black background. The beans have various shapes and sizes, ranging from round and plump to elongated and thin. Some appear smooth, while others have visible blemishes or markings. Several red labels with numerical values are placed around the beans. The labels likely indicate classifications or predictions made by a machine learning model, rather than the type of bean. The values range from 0.50 to 0.90, but the specific meaning or classification system is not provided within the image. Text overlay stating "not dal" next to all the numeric labels further supports this. This suggests the image might be part of a dataset used to train a machine learning model to differentiate between different types of beans, and specifically to identify dal (a type of lentil). The red labels with values represent the model's confidence score (between 0 and 1) that each bean is not dal.

5.2 Summary

In evaluating the model's accuracy and precision in distinguishing between pure and impure dal grains, a comprehensive suite of metrics was employed, including precision, recall, and the F1 score. Precision, measuring the proportion of correctly identified grains among all detected instances, offers crucial insights into potential false positives, highlighting the model's ability to discern true positives accurately. On the other hand, recall evaluates the model's capacity to correctly identify all instances of a particular class within the dataset, shedding light on potential false negatives and indicating the model's completeness in capturing relevant instances. The F1 score, serving as the harmonic mean of precision and recall, provides a holistic assessment of the model's performance, balancing the trade-off between precision and recall. Moreover, the model's robustness and generalization capabilities were scrutinized across diverse datasets and environmental conditions, demonstrating its resilience to variations in lighting, background, and grain quality. Notably, the YOLOv5 model exhibited real-time processing capabilities, facilitating rapid and efficient inference, which is pivotal for timely assessments of food quality and safety. Through meticulous error analysis, common sources of misclassifications such as occlusions and image artifacts were identified, informing strategies for model refinement and improvement. Such insights not only bolster the model's accuracy and reliability but also hold immense educational value, offering invaluable learning experiences in computer vision and machine learning domains, thereby preparing students for future endeavors in related fields.

Chapter 6

Advantages, Disadvantages and Applications

6.1 Advantages

The system incorporating Convolutional Neural Network (CNN) for detecting formalin adulteration offers the following advantages:

- High Accuracy: Deep Neural Networks (DNNs), especially Convolutional Neural Networks (CNNs), have demonstrated high accuracy in image-based tasks, making them effective for identifying subtle visual cues associated with grain adulteration.
- Feature Learning: DNNs can automatically learn relevant features and patterns from large datasets, eliminating the need for manual feature engineering. This ability is crucial in detecting complex and nuanced adulteration in grain samples.
- Flexibility and Adaptability: DNNs are flexible and adaptable to different types of grains and varying forms of adulteration. With appropriate training data, the model can generalize well to diverse scenarios

6.2 Disadvantages

The existing system for detecting grain adulteration has drawbacks such as:

- Data Dependency: DNNs are highly dependent on the quality and quantity of training data. Insufficient or biased datasets may lead to suboptimal performance and generalization issues.
- Computational Complexity: Training and running DNNs can be computationally intensive, requiring substantial resources. This complexity may limit the deployment of DNN-based models in resource-constrained environments.
- Interpretability: DNNs are often considered as "black-box" models, meaning their decision-making process is not easily interpretable. Understanding why the model makes a specific prediction can be challenging, which may raise concerns in some applications, especially in critical areas like food safety.

6.3 Applications

- Food Safety and Quality Assurance: Deep neural networks can be employed to detect the presence of adulterants in grains such as sand, stones, or other contaminants. This helps ensure that the grains meet quality standards and are safe for consumption.
- Purity Assessment: Deep learning models can be used to assess the purity of grain samples by identifying and quantifying the presence of unwanted substances.
- Authentication and Labeling: Deep neural networks can be used to verify the authenticity of grain products. By analyzing the chemical composition and characteristics of grains.
- Detection and classification of grain adulteration.
- Automated recognition of different grain types.
- Improving efficiency and accuracy in grain sorting and grading.

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

In conclusion, the project has successfully outlined the system requirements, design, and development process for a robust solution aimed at classifying dal grains using advanced computer vision or machine learning techniques. The key highlights of the project include:

- **Comprehensive System Requirements**: Detailed hardware and software requirements were identified, ensuring the smooth functioning of the application across various environments and platforms.
- **Functional and Non-Functional Requirements:** The project addressed both functional requirements, such as image upload and feature extraction, as well as non-functional requirements like reliability, availability, maintainability, and portability, to ensure a holistic approach to system design.
- **Utilization of Advanced Algorithms:** Cutting-edge algorithms including Convolutional Neural Networks (CNN) and You Only Look Once (YOLO) model were employed for feature extraction and object detection, enabling accurate classification of dal grains.
- **Cost and Effort Estimation:** The COCOMO II model was applied to estimate the effort, cost, and schedule of the project, providing valuable insights for resource allocation and project management.
- **Detailed System Design:** The architectural design, data flow diagrams, and module descriptions provided a clear understanding of the system's structure and functionality.
- **Implementation of Modules and Algorithms:** Various modules and algorithms, such as Flask, Pandas, PyTorch, and YOLOv5, were implemented to facilitate the development of the application.

7.2 Future Scope

- Further optimization of algorithms and models can be pursued to improve the speed, accuracy, and efficiency of dal grain classification, enabling real-time processing and scalability to large datasets.
- The performance of the classification system can be enhanced by augmenting and diversifying the training dataset, encompassing a wider range of dal grain variations, backgrounds, and environmental conditions.
- Incorporating additional features such as texture analysis, spectral imaging, and hyper spectral analysis can enhance the discriminative power of the classification system, leading to more accurate and reliable results.
- The developed system can be deployed and integrated into existing food processing and quality control workflows, facilitating automated inspection and sorting of dal grains in industrial settings.
- Improving the user interface and experience of the application can enhance usability and accessibility, making it more intuitive for stakeholders to interact with and utilize the system effectively.
- Collaboration with domain experts and stakeholders in the food industry can provide valuable insights and validation of the classification system's performance, ensuring its suitability for practical applications.

REFERENCES

[1]. Rajashekar S A, Abhishek GM, Jeswanth A L D "Deep Neural Network Based Grain Adulteration Detection", Feb.2023, doi: 10.17148/IJARCCE.023.12207.

[2]. Kasahun Desalegn, Hayat Hassen, Abdulmajid Haji "Selected food items adulteration, their impacts on public health, and detection methods: A review", Sep.2023, doi: 10.1002/fsn3.3732.

[3]. Shanglin Yang, Yang Lin, Yong Li, Defu xu, Suyi Zhang, and Lihui Peng(Senior Member, IEEE)"Deep Neural Network-Based Sorghum Adulteration Detection in Baijiu Brewing" July.2022, doi: 10.1109/OJIM.2022.3190024.

[4]. Ankita Choudhary, Neeraj Gupta, Fozia Hameed and Skarma Choton"An overview of foodadulteration:Concept,sources,impact,challenges,detection",Jan.2020,doi:10.2227 1/chemi.2020.v8.i1am.8655.

[5]. H. Liu and B. Sun, "Effect of fermentation processing on the flavor of Baijiu," J. Agricult. Grain Chem., vol. 66, no. 22, pp. 5425–5432, 2018, doi: 10.1021/acs.jafc.8b00692.