# RICE SORTER

Author
Muhammad Hasnain
Kashif Ali
Muhammad Qasim Sardar Tarar
Tajammal Riaz

Supervisor
Dr. Saddaf Rubab

Submitted to the faculty of Department of Computer Science,
Military College of Signals, National University of Sciences and Technology,
in partial fulfillment for the requirements of B.E Degree in Software Engineering

July, 2020

## CERTIFICATE OF CORRECTIONS & APPROVAL

Certified that work contained in this thesis titled "RICE SORTER”, carried out by Muhammad Hasnain, Kashif Ali, Muhammad Qasim Sardar Tarar and Tajammal Riaz under the supervision of Asst. Prof. Dr. Saddaf Rubab and Asst. Prof. Ahmed Muqeem Sheri for partial fulfillment of Degree of Bachelors of Software Engineering, in Military College of Signals, National University of Sciences and Technology, Islamabad during the academic year 2019-2020 is correct and approved. The material that has been used from other sources it has been properly acknowledged / referred.

**Approved by**

______________________

**(Main Supervisor)**
**Asst. Prof. Dr. Saddaf Rubab**
**Department of CSE, MCS**

______________________

**(Co-Supervisor)**
**Asst. Prof. Dr. Ahmed Muqeem Sheri**
**Department of CSE, MCS**

**Date:______________________**

## DECLARATION

**We hereby declare that the work contained in this report and the intellectual content of this report are the product of our work. This thesis report has not been formerly published in any stracture nor does it include any verbatim of the published resources which could be treated as violation of the international copyright decree. We also affirm that we do recognize the terms 'plagiarism' and 'copyright' and that in case of any copyright infringement or plagiarism established in this thesis, we will be held fully accountable of the consequences of any such violation.**

**Plagiarism Certificate (Turnitin Report)**

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student
Muhammad Hasnain
Kashif Ali
Muhammad Qasim
Tajammal Riaz

Registration Number

Signature of Supervisor
Asst. Prof. Dr. Saddaf Rubab
Department of CSE, MCS

**Acknowledgements**

I am thankful to my *Khaliq* Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which You brought up in my mind to refine it. Indeed, I could have done utterly nothing but with Your magnificent help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your absolute will, so indeed none be worthy of glory and praise but You Subhana-Watala.

I am lavishly appreciative to my beloved parents who raised me up when I was not equipped for strolling and kept on supporting me along the way in each branch of my life.

I would also like to express special thanks to my supervisor Dr. Saddaf Rubab for her help throughout my thesis I can safely say that I haven't learned any other engineering subject in such depth than the ones which she taught.

I would also like to pay special thanks to Dr. Ahmed Muqeen Sheri for his rich and much needed support and cooperation. Each time I got stuck in literally anything, he came up with the most needed solutions. Without his help I wouldn’t have been able to complete my thesis.  I appreciate his longing patience and fruitful guidance throughout the whole thesis.

I would also like to thank Dr. Malik Muhammad Zaki Murtaza Khan and Mam Mobeena Shahzad for being on my thesis guidance and evaluation committee and express my special thanks again to Dr. Ahmed Muqeem Sheri for his help.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

*Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.*

**Abstract**

In twenty-first century, life has become too fast and common men hardly find any time to resolve their domestic problems which are small but necessary for the smooth running of their family. One of such problem is rice sorting. Rice purchased from the market usually is not one hundred percent rice; some impurities are always there. So in turn, men have to take them to the market again to get them sorted. He might have to wait in queues for a while or might sometimes submit and come back again after a day or two.

The project "RICE SORTER" is a prototype to an image-processing based, economical and time efficient solution to the problem of rice sorting for domestic purposes. A machine which is easy to maintain, flexible in scalability, and most importantly, it fits into the kitchen.

The working principle is simple; a thin stream of rice is falling in front of a fixed camera. This camera is embedded to a digital system which applies Machine-Learning (ML) algorithms on the video feed, identifies the impurities and then sends signal to the pneumatic valve which will in turn, blow away the impurity from the stream but let the rice pass. The ideal goal is to obtain pure rice after single or multiple cycle(s) of sorting.

**Keywords:** *Economical, Domestic Use, Time Efficient, Image-Processing, Machine Learning*

## Table of Contents

## TABLE OF FIGURES

# CHAPTER 1

# INTRODUCTION

The introductory section will illustrate the background, nature and need-of-solution of the problem of Rice Sorting. To describe the content in detail, this section is subdivided into five parts. The Overview; a general review or summery of the problem and its proposed solution, The Problem; describing the problem in detail, The Statement; summarizing the problem, The Approach; of the proposed solution, The Objectives; formally listing out the milestones to cover to proposed solution to the problem of sorting.

## 1.1 Overview

It's the 21st Century and daily life has become a ride of the roller coaster. Today, man is more stuck in the shackles of earning a livelihood than ever before. There is absolutely no time in working days to look after even our basic needs. So, it is tried to make things as simple, reliable and available as possible.

One of such a problem which consumes much of the precious time is cleaning and sorting out rice from the rice normally purchased from the markets. Rice purchased from the market usually is not one hundred percent rice; some impurities are always there. They need to be taken to the market again to get them sorted. Might have to wait in queues for a while or might sometimes submit and come back again after a day or two.

It is obvious to sort out impurities from the rice to cook them, as it is to save precious work hours. That sums up the problem and the required goal or objective of any proposed solution to this problem.

## 1.2 Problems

The problem is to sort the rice

- in minimal time
- at optimal cost;

Ideally; should be done in the kitchen any time before cooking, with normal domestic voltage supply.

The solution needs to maintainable, reliable and scalable.

### 1.3 Solution

The overall working of the system is simple and basic. A video is captured through IP camera, which is communicated through an RJ45 connector to personal computer. Personal computer on the basis of image processing decides whether to activate valve or not. The activation of valve is done with help of Arduino Uno and a Transistor. Whenever the valve activates, it eject the rice in front of it.

A simple 2D model of the solution is shown in figure 1.1:

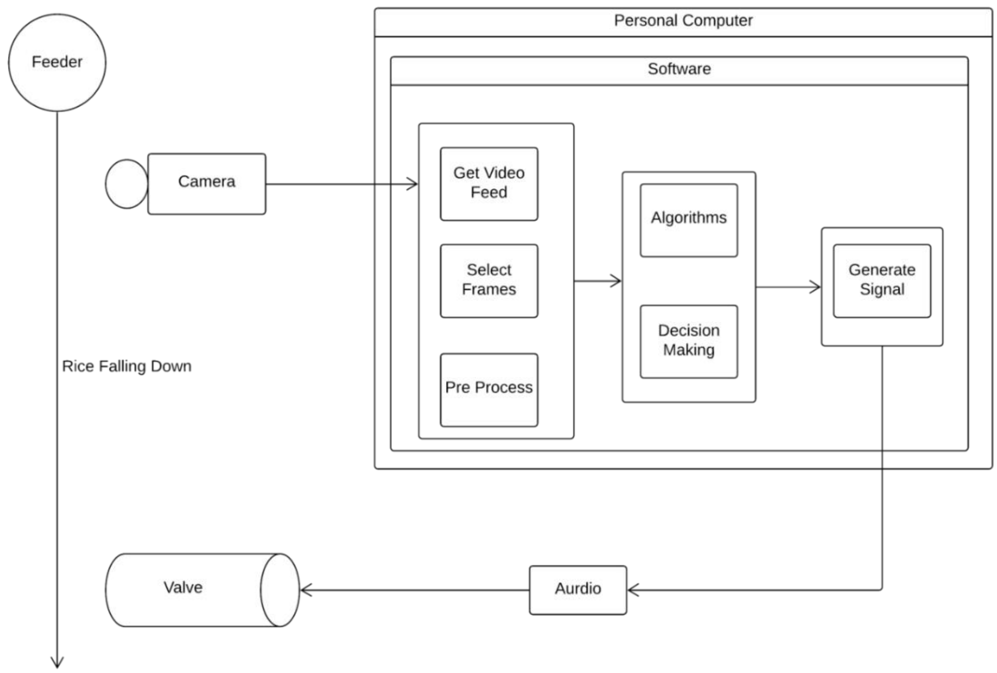


**Figure 1.1:** 2D-Model of the Solution

The software system will have mainly 3 subsystems.

- The first subsystem will acquire the video feed, select frames, pre-process them and pass them on to the next sub system.
- The second subsystem will apply algorithms developed using OpenCV, a framework of python, to decide whether the particle is defected or not. The result is passed on to next and last subsystem.
- Based on the result, a signal will be generated to the Arduino which is connected to the valve.

## 1.4 Approach

The problem is divided into smaller ones and then each is resolved individually. At the end, all the solutions are chained up to make the system work. The project has five basic modules. First module is to drop the Rice from moving chute, Second module is to acquire the image using a camera of falling rice, Third module is based upon image preprocessing, Fourth module is to detect the impurities rice from image processing and the fifth module will send the signal through Arduino to ejector to blow the unwanted impurities to improve the quality of product.

We will see the minor details in the sections to come.

## 1.5 Objectives

The rice needs to be sorted in a minimal time and cost. It is important to mention here that this is a prototype of an idea. It can always be further developed or enhanced at a larger scale in the years to come.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Multi Object Detection in OpenCV

Object Detection has been pulling in much enthusiasm because of the wide range of uses that utilization it. Article identification innovation has been driven by an expanding handling power accessible in programming and equipment. In this work we present a created application for numerous articles location dependent on OpenCV libraries.

The multifaceted nature related viewpoints that were considered in the item discovery utilizing course classifier are depicted. Besides, w Article recognition has been pulling in much enthusiasm because of the wide range of uses that utilization it. Article identification innovation has been driven by an expanding handling power accessible in programming and equipment. In this work we present a created application for numerous articles location dependent on OpenCV libraries. The multifaceted nature related viewpoints that were considered in the item discovery utilizing course classifier are depicted. Besides, we talk about the profiling and porting of the application into an implanted stage and contrasting the outcomes and the normal stage.

The proposed application manages ongoing frameworks execution and the outcomes give a sign of where the instances of item recognition applications might be progressively unpredictable and where it might be less complex then talking about the profiling and porting of the application into an implanted stage and contrasting the outcomes and the normal stage.

## 2.2 Mapping Paddy Rice Using CNN

Rice is one of the world's significant staple nourishments, particularly in China. Exceptionally exact checking on rice-delivering land is, in this way, pivotal for evaluating food supplies and efficiency. As of late, the profound learning convolutional neural system (CNN) has made impressive progress in remote-detecting information investigation. A CNN-based paddy-rice planning strategy utilizing the multitemporal Landsat 8, phenology information, and land-surface temperature (LST) was created during this examination. To start with, the spatial–worldly versatile reflectance combination model (STARFM) was utilized to mix the moderate-goal imaging spectroradiometer (MODIS) and Landsat information for acquiring multitemporal Landsat-like information. In this manner, the limit strategy is applied to get the phenological factors from the Landsat-like (Normalized distinction vegetation list) NDVI time arrangement. At that point, a summed up single-channel calculation was utilized to get LST from the Landsat 8. At long last, multitemporal Landsat 8 ghostly pictures, joined with phenology and LST information, were utilized to extricate paddy-rice data utilizing a fix based profound learning CNN calculation. The outcomes show that the proposed strategy accomplished a general precision of 97.06% and a Kappa coefficient of 0.91, which are 6.43% and 0.07 higher than that of the help vector machine technique, and 7.68% and 0.09 higher than that of the irregular timberland technique, separately. In addition, the Landsat-inferred rice region is emphatically related (R2 = 0.9945) with government measurable information, showing that the proposed technique has potential in enormous scope paddy-rice planning utilizing moderate spatial goal pictures.

## CHAPTER 3: DESIGN AND DEVELOPMENT

### 3.1 System Overview

The project has five basic modules (Figure 3.0). First module is to drop the Rice from moving chute, Second module is to acquire the image using a camera of falling rice, Third module is based upon image preprocessing, Fourth module is to detect the impurities rice from image processing and the fifth module will send the signal through Arduino to ejector to blow the unwanted impurities to improve the quality of product.

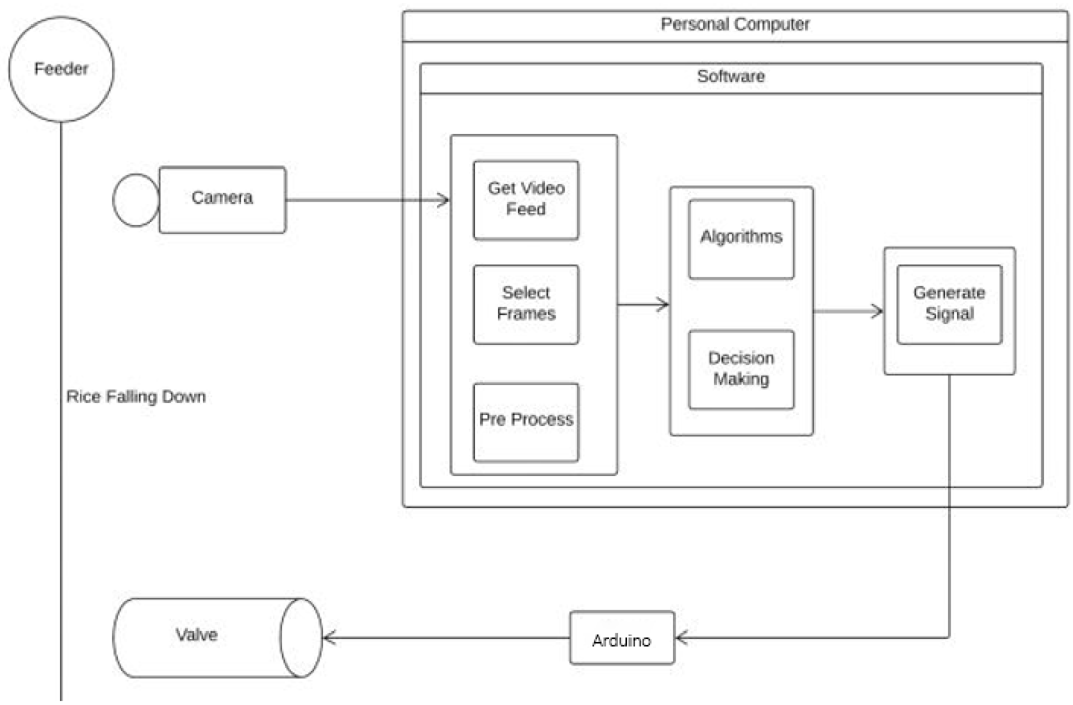


**Figure 3.1:** System Overview

#### 3.1.1 Product Perspective

Main idea behind the project Rice Sorter is to reduce the hectic work force on rice grading with high accuracy. It is also impossible for humans to accurately inspect impurities on a large scale. Software Design Document 5 This project will minimize the time loss and also will reduce amount of work required to sort the rice.

### 3.1.2 Product Functions

The features of this product, as entrusted to be, are as below:

1. Rice will be dropped from a moving feeder.
2. Camera attached to the PC or Laptop examines this falling rice.
3. The video feed is than processed and impurities are identified
4. A signal is generated to the Arduino
5. Arduino sends signal to pneumatic valve which in turn will blow out impurity.

### 3.1.3 End Stakeholders

The following section describes the types of users of the RICE SORTER. There are explanations of the user followed by the interactions the user(s) shall be able to make with the software.

1) The product rice sorter is helpful in everyday use.
2) For commercial Use RICE SORTER idea is highly effective in commercial domain where on large scale humans are ineffective in terms of money, time and quality than of RICE SORTER

We will see how the system will be implemented using various UML diagrams in the following sections. Some main UML diagrams are shown and discussed with every detail in the following. For the sake of assignment-scope, it is assumed that basic knowledge of each diagram is pre-known to the reader. Let us finally jump to the core of the document.

## 3.2 System Architecture

The overall working of the system is simple to understand. A video is captured through ip camera, which is used for communication through an RJ45 connector to personal computer. Personal computer, on the basis of image processing decides whether to activate valve or not. The activation of valve is done with help of Arduino Uno and a Transistor. Whenever the valve activates, it blows the rice in front of it.

### 3.2.1 Architectural Design

The system will work as the feeder will provide a thin video feed to the cameras which is directly connected to a personal computer where the video will be processed.

First certain frames are chosen. After preprocessing, algorithms are applied to identify the defected particles in the stream of rice and a electronic signal is given to the Arduino which is connected to a valve.

This valve turns on and off based on the signal generated by the PC and Arduino.

The general structure of product is showed in figure 3.1.

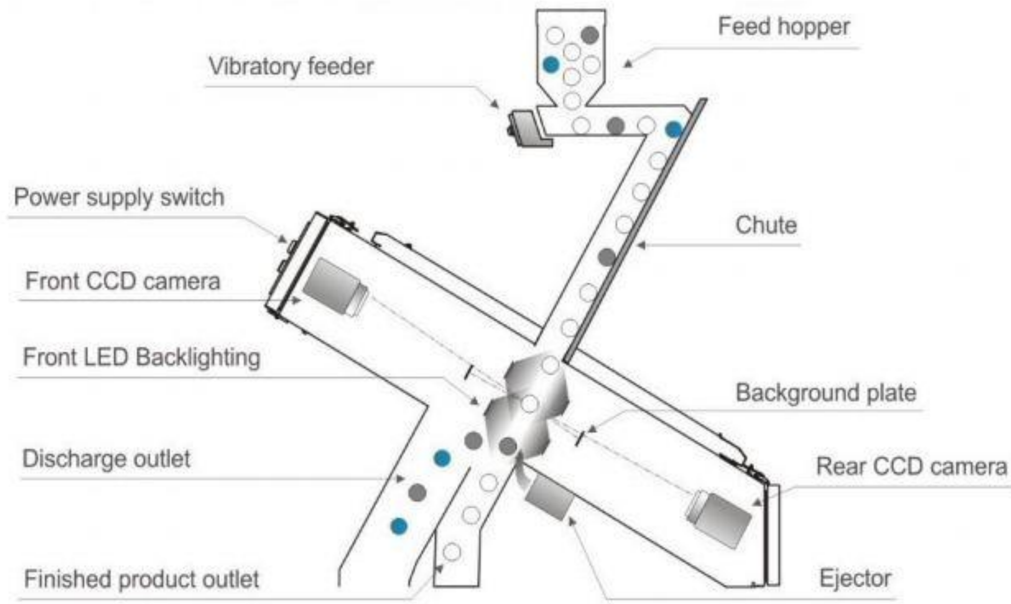


**Figure 3.2:** Structure of The Product

Hardware component connections can be seen in fig 3.2. The camera providing the video feed is connected to the system which processes it makes decisions based on machine learning algorithms and then generates a signal to the Arduino which in turn switches the pneumatic valve on/off based on the signal.

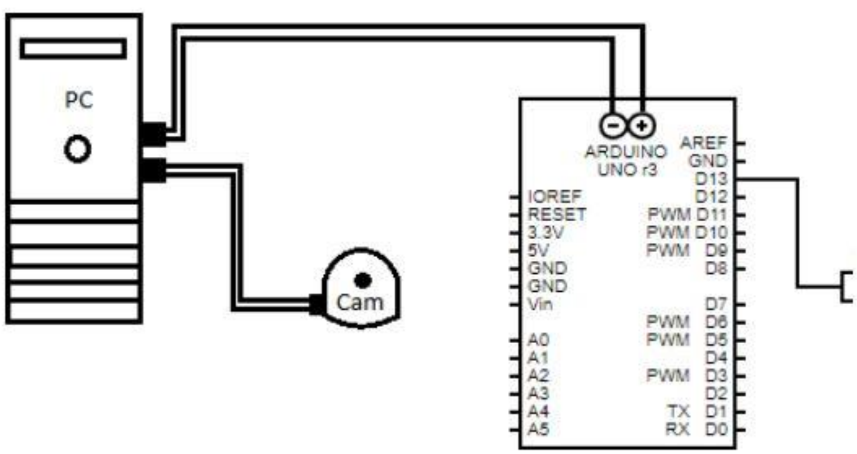


**Figure 3.3:** Hardware Component Connections

Talking about the architectural design of the software system, it is divided into three steps to make the system easier to debug and understandable. The system model is shown in fig. 3.4

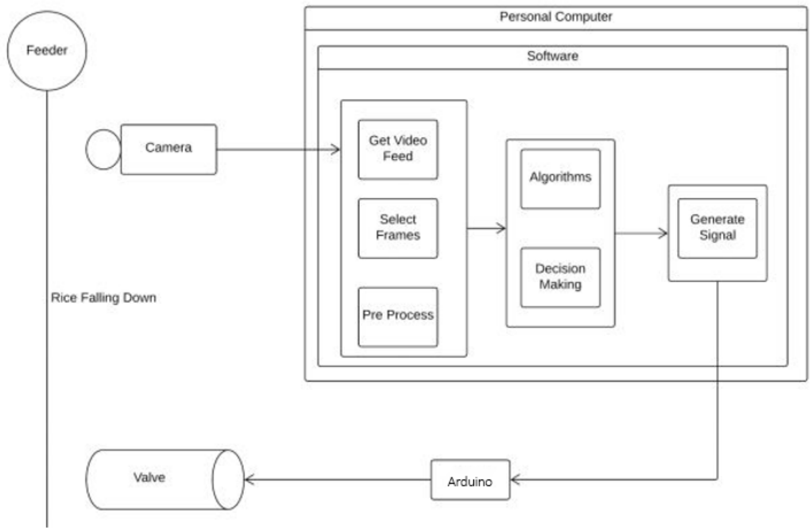


**Figure 3.4:** Software-Hardware Integration

The three modules are as:

1) Image-Processing-Module: This module gets the video feed and pre-processes it.
2) Decision-Making-Module: This module makes use of machine learning algorithms to return a decision either it’s a rice particle or an impurity.
3) Signal-Generator: This module signals the Arduino when the pneumatic valve is to switched on or off.

The software system will have mainly 3 subsystems.

1) The first subsystem will acquire the video feed, select frames, preprocess them and pass them on to the next sub system.

2) The second subsystem will apply algorithms developed using OpenCV, a framework of python, to decide whether the particle is impurity or not. The result is passed on to next and last subsystem.
3) Based on the result, a signal will be generated to the Arduino which is connected to the valve.

When Arduino receive the signal to turn on the ejector valve, ejector valve opens and ejects the product falling in front of it.

### 3.2.2 Decomposition Description

This section will discuss the description of the decomposition of the system using different UML diagrams. UML diagrams are very effective in understanding the system just by previewing a set of diagrams, each having its own purpose, definitions and constraints. Let's see the UML diagrams for our system.

#### 3.2.2.1 Use Case Diagram

Use cases specify the expected behavior (what), and not the exact methods of making it happen (how). Above diagram shows a generalized view of the system regardless of the subsystem structure.

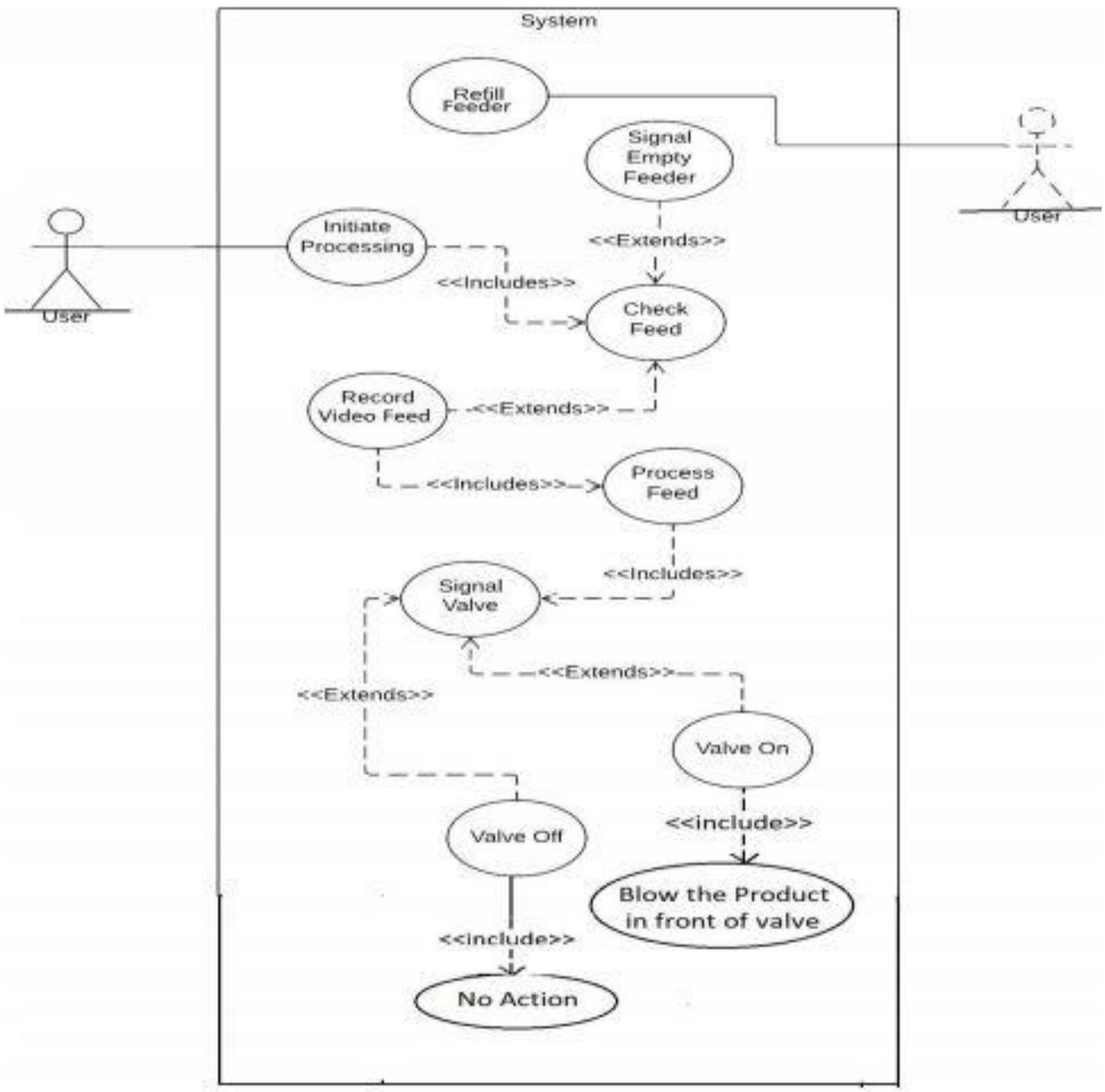


**Figure 3.5:** Use Case Diagram

The user will have to initiate the system by adding feed in the chute. Cameras provide video feed of the feed to the software system. This video feed is analyzed and defected particles are identified and removed by the valve which operates based on the signal generated by Arduino connected to the computer. The secondary actor, the user itself, needs to refill the feeder over time (if any of the rice is left unsorted).

#### 3.2.2.2 Class Diagram

There are three subsystems and five classes to simplify the development and understanding of the entire working to this system.

They are video feed, preprocess and the processor. They are discussed in detail on the next page.

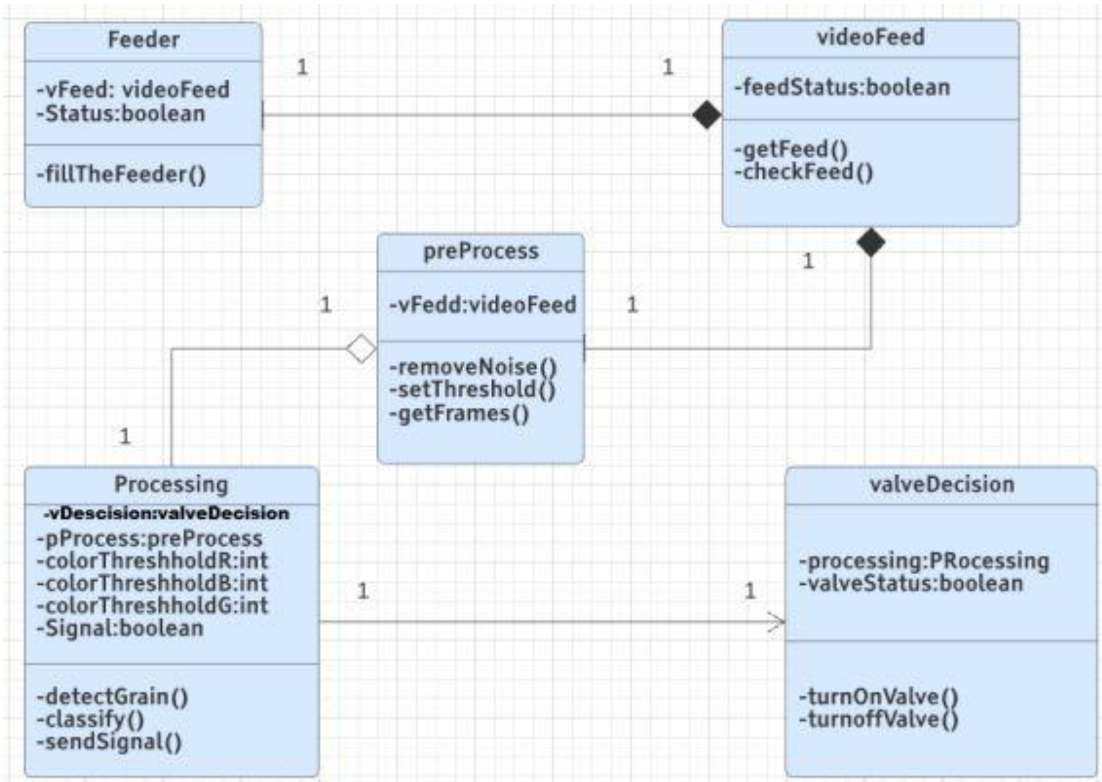


**Figure 3.6:** Class Diagram

**1) Video Feed**

The video feed provided by a camera; of the rice feed falling down from the moving feeder. The method getVFeed() will do these works; capture video, selecting frames.

2) **Pre-Processing**

Now following techniques are applied to the selected frames from the video feed to obtain a high-res image:

- loading the image
- Resizing this image
- Removing noise from it (Denoising)
- Segmentation: cropping to interested area
- Smoothening the edges of this image (Morphology)

3) **The Processor**

There will be an instance of camera, pneumatic valve and Decision class. After getting the selected frames it applies algorithms and takes a decision i.e. either it is an impurity or a correct rice particle. It will have another function i.e. SignalToValve(). Which will generate a signal to the pneumatic valve via Arduino based on the feed.

4) **Pneumatic Valve**

This valve component will have its timeout time. This time will tell when to close the valve once it is let open. Status to be the present state of the valve i.e. open or close. setStatus() changes the status of the valve between two possible values (on/off , true/false). Restore() will reset the status variable to default settings and checkDecision() will fetch the value of the signal variable in the processor object.

**5) The Feeder**

Rice is filled into this feeder from where it will be let to fall vertically downwards. Feeders will have two functions to make it sure, the rice particle-stream is continuously falling from it whenever the system is on.

### 3.2.2.3 Other UML Diagrams

- **Sequence Diagram**

A sequence diagram is shown in figure 3.7

The user adds the feed into the feed which lets a thin stream of rice to fall down. This stream is recorded by the camera and passed on to the processor module where it is preprocessed and then classified. Based on the classification, a signal is generated to the valve which in turn blows away the impurity.

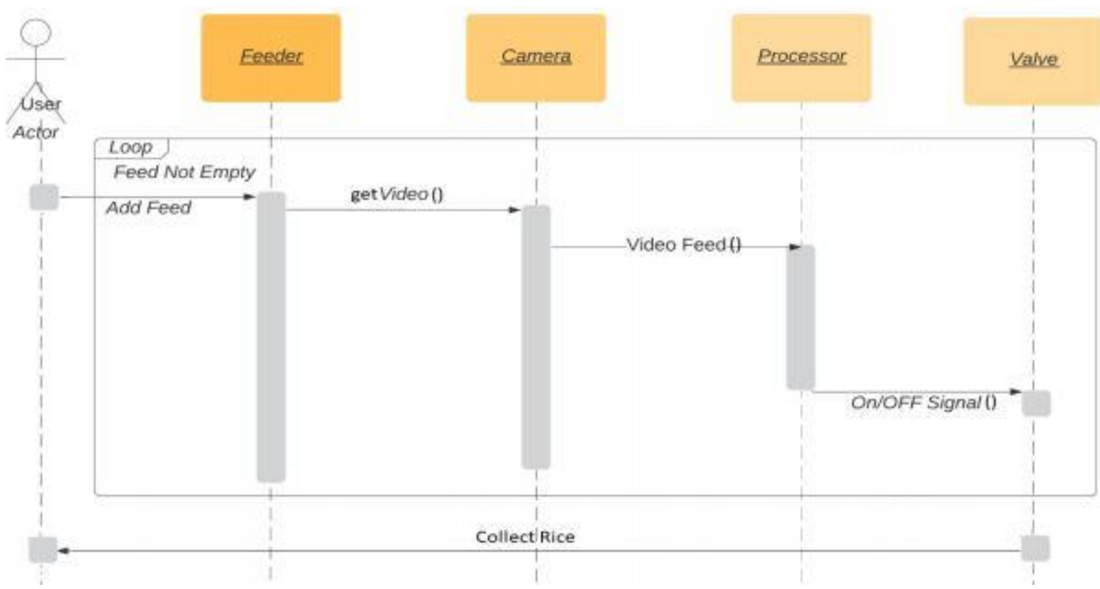


**Figure 3.7:** Sequence Diagram

- **Component Diagram**

A component diagram is shown as figure 3.8

The system can be divided into three components. The camera, the processor and the valve. Same as we have seen previously; the user adds the feed into the feed which lets a thin stream of rice to fall down. This stream is recorded by the camera and passed on to the processor module where it is preprocessed and then classified. Based on the classification, a signal is generated to the valve which in turn blows away the impurity.

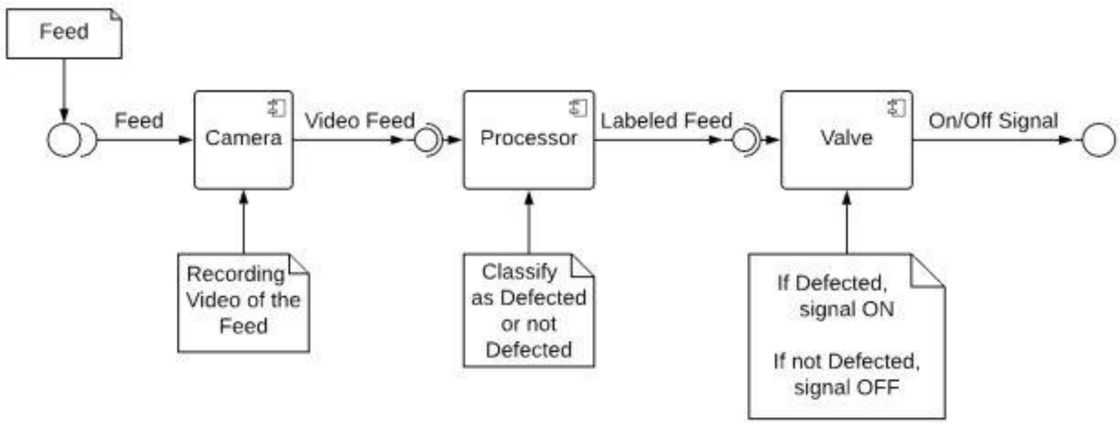


**Figure 3.8:** Component Diagram

- **Activity diagram**

Activity diagram is shown in figure 3.9 which shows the normal working of the system. As soon as the system is started after the user fills the feeder, the feeder drops the thin stream of feed. This stream is recorded by the camera and passed on to the processor module where it is preprocessed and then classified. Based on the classification, a signal is generated to the valve which in turn blows away the impurity.

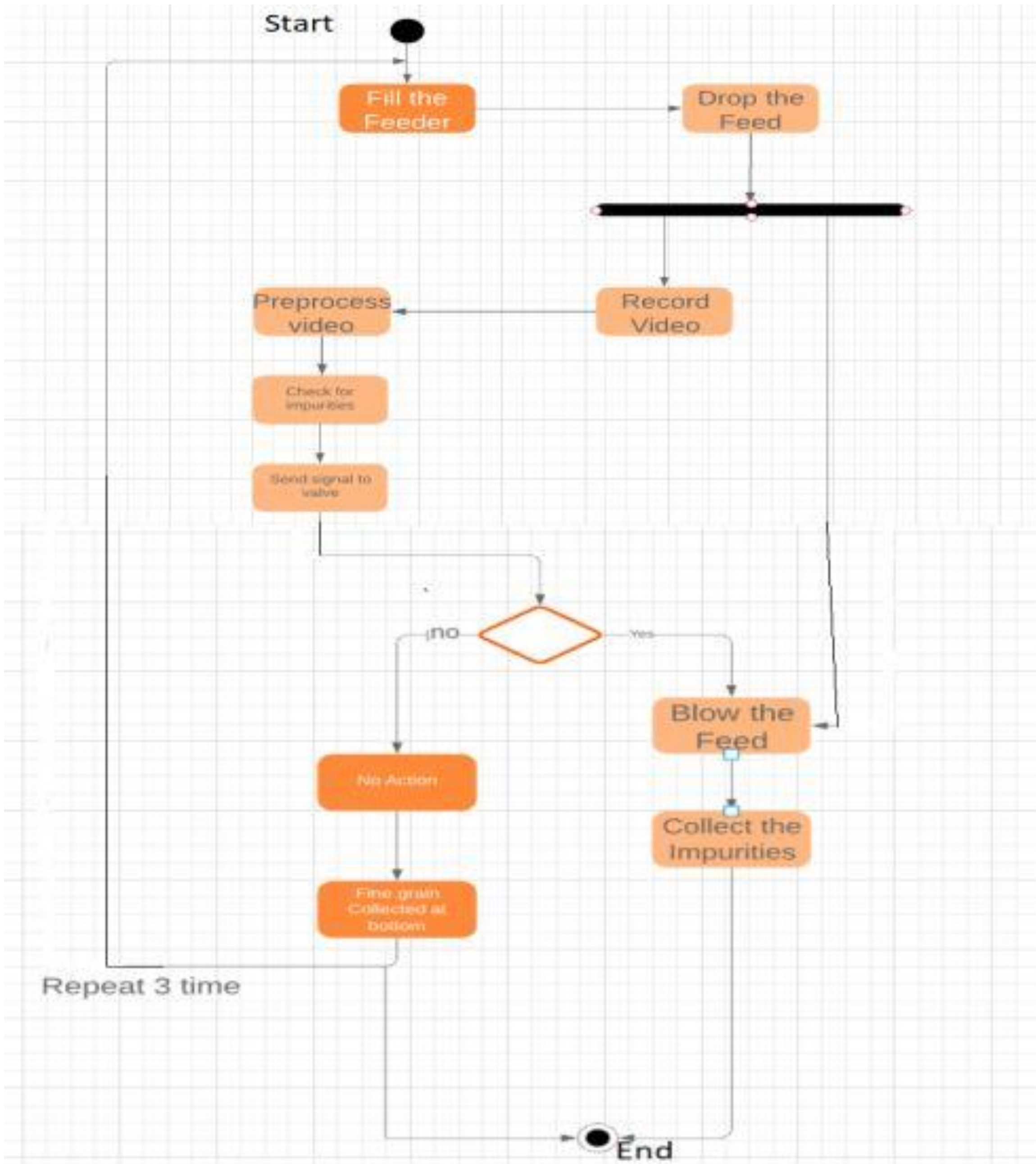


**Figure 3.9:** Activity Diagram

### 3.2.3 Design Rationale

An Object-Oriented-Programming (OOP) approach is adopted to produce an optimal solution to this problem of sorting. If the entire system is divided into small subsystems, it becomes far easier understand, design, implement and debug the solution of this problem. As these subsystems are generated and finalized, they are integrated into one other as required to get the parent system. Some key catching advantages are list below:

- This approach has a transparent modular structure which makes it quite easy for declaring abstract data-types with hidden details.
- Instances of classes can be reused anywhere, this system or another one. This highly reduces the cost of developing the system. The more one puts effort in analysis and design phase, lower is the overall cost.
- Maintainability is also very effectively possible of an OOP system. Since the design is component based, certain parts of the system can be changed, updated of deleted when needed without make significant changes in the overall system.
- Reusability also results in speeding-up the development process. OOP languages have prebuilt opensource big libraries of classes, and this is also reusable in future projects.
- It gives a productive framework for prebuilt libraries where the supplied software modules can be easily integrated and updated by the developer. This is specifically helpful for developing GUI based applications.
- High Productivity as OOP helps to get more work done in less time; These applications when finished will run better, possibly have more features and are quite easier to maintain and update. OOP Developers take new and existing software instances and bind them together to release a new and genuine application.

- Because prebuilt opensource libraries have almost all needed useful methods, developers often don't have to reproduce the running wheel; greater value of their precious time goes in developing the new software.

## 3.3 Data Design of The System

### 3.3.1 Data Description of the System

The entire information related work of this system will use file-storage approach. For simplicity, it will be correct to say "File Handling". The system will save and reuse the data in the program when needed. For ease, all data operations and management will be done on files. Mostly, the data will be of numerical matrix related because we have to process the pictures or frames from the video feed in this project to sort the rice. All the images will be stored in files and will be used in the program when needed, fetching them out of the files to the application. Python 3 code for fetching the image is as under:

```
try:
        image = Image.open(path)
        except IOError:
                pass
```

The output will be stored in another file. A database is not used or say not needed for now as large amount of work is beyond the scope of the project, file handling is useful alternative in this scenario.

### 3.3.2 Data Dictionary of The System

#### 3.3.2.1 Functional Description (Data Dictionary):

- **Feed:** This step involves the user filling the feed-container with the rice feedwill by the function FillTheFeeder().
- **Video Feed:** checkFeed() function will refill the feed container whenever it gets empty, or if the container is filled up it will signal to stop adding feed to the feeder.
- **Process Feed:** Now the video from the camera will be processed and impurities will be identified. This will be done in two steps. Firstly preprocessing and then the actual decision making processing.

#### 3.3.2.2 Pre-Process functions:

This contains VideoFeed() function that sill produce a video for actual processing.

- **RemoveNoise():** This function will denoise the image. A set of variables are adjusted to gain a noise free photo of the source image. We may resize or downscale the image if needed
- **SetThreshold():** This function will adjust the margins of the image. It may resize it to obtain the accuracy.
- **GetFrames():** This function will fetch out selected and preprocessed frames from the video feed.
- **Processing() Method:** The preprocessing techniques will be applied inside this function.

- **DetectGrain():** This function will detect a particle from the frame. And call the classify() function.
- **Classify():** This will check if the detected particle is a rice particle of an impurity and then call the SendSignal() function.
- **SendSignal():**This function will generate a signal to the Arduino based on the classification of the detected particle.

#### 3.3.2.3 Valve Switching:

There will be two functions in this step:

- **ValveOn():** This function will turn the valve on. It will be called if an impurity is detected.
- **ValveOff():** This will be called either when the timer is out or the particle detected is a rice particle

## 3.4 Component Design

Now we will see each component is a more systematic manner.

- **Video Feed and Pre-Processing**

The following code snippet will allow us to do the job:

```
If (connected):
    getVideoFeed()
else
    prompt error in console.
```

This will format the selected frames of the video, denoise it and will let us select the reduced area for processing to get video feed.

cv2.fastNlMeansDenoisingColored

This will eliminate noise from the background.

Then we must declare the dimensions using .crop_ROI() function from OpenCV it will ecrop the focus area of the video frame.

- **Processing**

This code will classify the photo and signal either to eject the particle or let it pass. Steps:

import ObjectDetection

detector.detectObjectsFromImage

detect the RGB-color of the photo

send the relevant signal to valve via Arduino

- **Arduino**

This code snippet will make the Arduino to turn on the valve for a mare millisecond then it will turn it off imminently.

```
serial = serial.Serial('COM4', 9800, timeout=1)

serial.write(b'T')

serial.write(b'F')
```

# CHAPTER 4

# ANALYSIS AND EVALUATION

First of all, image preprocessing was done after getting the dataset. As each picture in dataset had a lot of grains, each grain was separated by finding contours.

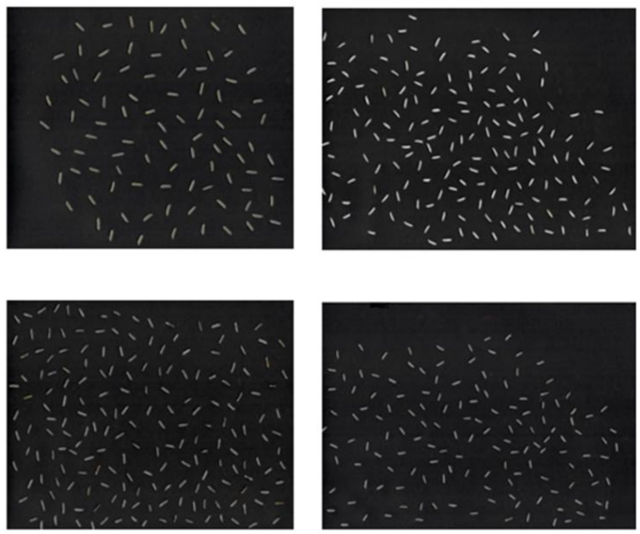

**Figure 4.1:** Sample Images

Before finding contours, threshold or canny edge detection was applied and then grains were cropped them. Then each grain was padded to make the size of all grains same and was saved.

**Figure 4.2:** After PreProcessing

Dataset was increased by data augmentation and was then put in the model for training. The model used for the training process was Convolutional Neural Network (CNN). A convolution is a technique that changes a function/method to something else. Convolutions is done so that one can transform the original function into a form to get more information.

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_5 (Conv2D)            (None, 128, 128, 6)       456
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 64, 64, 6)         0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 64, 64, 16)        2416
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 32, 32, 16)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 30, 30, 64)        9280
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 15, 15, 64)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 13, 13, 128)       73856
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 6, 6, 128)         0
_________________________________________________________________
flatten_2 (Flatten)          (None, 4608)              0
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              4719616
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_4 (Dense)              (None, 8)                 8200
=================================================================
Total params: 4,813,824
```

**Figure 4.3:** Training

Training data was 60 percent of the whole data set and validation was 40 percent. The accuracy of the model was (65%).

| | |
|---|---|
| Training Data | 60% |
| Validation Data | 40% |
| Image Size | (128,128,3) |
| Batch Size | 64 |

After training the class of rice grains is predicted using the model. Image preprocessing was done as before when dataset was prepared and around each grain a bounded box was created its class was written on right side of grain. Size was predicted by getting contours of each grains and the minimum area rectangle each formed by rice grain was found for each contour and its length and width was calculated by Euclidean distance.

The Euclidean distance or Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space. With this distance, Euclidean space becomes a metric space. The associated norm is called the Euclidean norm. Size of the original rice grain was taken as reference to measure the size so that the size can be predicted accurately and showed it around the bounded box of rice grain.

# CHAPTER 5

# CONCLUSION

The rice needs to be sorted in a minimal time and cost. It is important to mention here that this is a prototype of an idea. It can always be further developed or enhanced at a larger scale in the years to come.

The working principle is simple; a thin stream of rice is falling in front of a fixed camera. This camera is embedded to a digital system which applies Machine-Learning (ML) algorithms on the video feed, identifies the impurities and then sends signal to the pneumatic valve which will in turn, blow away the impurity from the stream but let the rice pass. The ideal goal is to obtain pure rice after single or multiple cycle(s) of sorting.

The model used here in this training is CNN (Convolutional Neural Network).

# CHAPTER 6

# FUTURE WORK

If the system gets developed successfully, it can be deployed into some hardware mechanism and a number of applications can be achieved. For instance, a rice sorter which can separate rice based on color, size and quality.

The initial motivation of the system was to solve the problem of sorting efficiently within the premises of a house kitchen. Though it can be scaled up to perform commercial-level use and earn a living out of it.

Moreover, the system enlightens the path to follow in order to develop solutions similar to rice sorter, for example, one may use the concept to sort pulses or any particular substance.

# BIBLIOGRAPHY

All references in this document are provided where necessary, however where not present, the meaning is self-explanatory. All ambiguous terms have been clarified. Some basic references are as under:

a) IEEE standard 10161998 recommended practice for software design descriptions.

b) Freeman, P. and A. I. Wasserman, Tutorial on Software Design Techniques. 4th Edition, IEEE Computer Society Press.

c) Gamma, E. etal ., Design Patterns, Addison Weslet, 1995.

d) UML: The Unified Modeling Language User Guide by G. Brooch, J. Rumbaugh and I. Jacobson, Addion Weslay Pub Co; ISBN:0201571684.

e) Roger S. Pressman, Software engineering: a practitioner's approach. 6th edition, McGrawHill international edition.

f) Doğru,A.,Component oriented software engineering , The Atlas Publishing,2006.

FYP thesis v4.0

ORIGINALITY REPORT

| 18% | 11% | 4% | 16% |
| --- | --- | --- | --- |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | | |
| --- | --- | --- | --- |
| 1 | Submitted to Higher Education Commission Pakistan | Student Paper | 10% |
| 2 | www.mdpi.com | Internet Source | 3% |
| 3 | www.swedishryk.edu.pk | Internet Source | 1% |
| 4 | Submitted to Universiti Teknologi MARA | Student Paper | 1% |
| 5 | eprints.usq.edu.au | Internet Source | <1% |
| 6 | Submitted to Oxford Brookes University | Student Paper | <1% |
| 7 | Submitted to King Mongkut's Institute of Technology Ladkrabang | Student Paper | <1% |
| 8 | Submitted to La Trobe University | Student Paper | <1% |

| | | |
|---|---|---|
| 9 | www.scribd.com<br>Internet Source | <1% |
| 10 | hdl.handle.net<br>Internet Source | <1% |
| 11 | Submitted to Informatics Education Limited<br>Student Paper | <1% |
| 12 | senior.ceng.metu.edu.tr<br>Internet Source | <1% |
| 13 | umpir.ump.edu.my<br>Internet Source | <1% |
| 14 | Submitted to Tshwane University of Technology<br>Student Paper | <1% |
| 15 | fedetd.mis.nsysu.edu.tw<br>Internet Source | <1% |
| 16 | Meng Zhang, Hui Lin, Guangxing Wang, Hua Sun, Jing Fu. "Mapping Paddy Rice Using a Convolutional Neural Network (CNN) with Landsat 8 Datasets in the Dongting Lake Area, China", Remote Sensing, 2018<br>Publication | <1% |

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | On | | |

| | | | |
|---|---|---|---|
| 9 | www.scribd.com<br>Internet Source | <1% |
| 10 | hdl.handle.net<br>Internet Source | <1% |
| 11 | Submitted to Informatics Education Limited<br>Student Paper | <1% |
| 12 | senior.ceng.metu.edu.tr<br>Internet Source | <1% |
| 13 | umpir.ump.edu.my<br>Internet Source | <1% |
| 14 | Submitted to Tshwane University of Technology<br>Student Paper | <1% |
| 15 | fedetd.mis.nsysu.edu.tw<br>Internet Source | <1% |
| 16 | Meng Zhang, Hui Lin, Guangxing Wang, Hua Sun, Jing Fu. "Mapping Paddy Rice Using a Convolutional Neural Network (CNN) with Landsat 8 Datasets in the Dongting Lake Area, China", Remote Sensing, 2018<br>Publication | <1% |

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | On | | |