# Two-Stage Automated Coffee Bean Sorter: A Precise System for Green Coffee Beans Using Machine Vision and Density-Based Analysis

---

A Thesis
Presented to the Faculty of the
Department of Electronics and Computer Engineering
Gokongwei College of Engineering
De La Salle University

---

In Partial Fulfillment of the
Requirements for the Degree of
Bachelor of Science in Computer Engineering

---

by

DELA CRUZ John Carlo Theo S.
PAREL Pierre Justine P.
TABIOLO Jiro Renzo D.
VALENCERINA Ercid Bon B.

March, 2025

De La Salle University

# ORAL DEFENSE RECOMMENDATION SHEET

This thesis, entitled **Two-Stage Automated Coffee Bean Sorter: A Precise System for Green Coffee Beans Using Machine Vision and Density-Based Analysis**, prepared and submitted by thesis group, AISL-1-2425-C3, composed of:

DELA CRUZ, John Carlo Theo S.
PAREL, Pierre Justine P.
TABIOLO, Jiro Renzo D.
VALENCERINA, Ercid Bon B.

in partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Engineering** (**BS-CPE**) has been examined and is recommended for acceptance and approval for **ORAL DEFENSE**.

**Dr. Melvin K. Cabatuan**
*Adviser*

March 31, 2025

# ABSTRACT

The study proposes to develop a two-stage automated coffee bean sorter that identifies the good beans, less-dense beans and at the same time segregating the defective coffee bean using machine vision and density-based analysis. In the first stage, the defective beans will be detected through the use of machine vision, parameters such as size and defects are taken into account. The second stage is used to categorize each bean by its density, which is calculated by its mass and volume. Thus, beans with relatively low density and not within the size threshold, are sorted out. The system aims to incorporate machine vision and density analysis to reduce human labor and provide an alternative to manual sorting methods for the farmers and coffee bean producers.

*Index Terms*—computer vision, deep learning, density-based analysis, Arabica, green coffee beans, sorting.

De La Salle University

# TABLE OF CONTENTS

De La Salle University

De La Salle University

De La Salle University

De La Salle University

# LIST OF FIGURES

De La Salle University

# LIST OF TABLES

# ABBREVIATIONS

| | | |
|---|---|---|
| AC | Alternating Current | 138 |
| HTML | Hyper-text Markup Language | 138 |
| CSS | Cascading Style Sheet | 138 |
| XML | eXtensible Markup Language | 138 |

# NOTATION

| | | |
|---|---|---|
| $\mathcal{S}$ | a collection of distinct objects | 140 |
| $\mathcal{U}$ | the set containing everything | 140 |
| $\emptyset$ | the set with no elements | 140 |
| $\lvert\mathcal{S}\rvert$ | the number of elements in the set $\mathcal{S}$ | 140 |
| $h(t)$ | impulse response | 130 |
| $x(t)$ | input signal represented in the time domain | 130 |
| $y(t)$ | output signal represented in the time domain | 130 |

Throughout this thesis, mathematical notations conform to ISO 80000-2 standard, e.g., variable names are printed in italics, the only exception being acronyms like, e.g., $\mathrm{SNR}$, which are printed in regular font. Constants are also set in regular font like $\mathrm{j}$. Standard functions and operators are also set in regular font, e.g., in $\sin(\cdot)$, $\max\{\cdot\}$. Commonly used notations are $t$, $f$, $\mathrm{j} = \sqrt{-1}$, $n$ and $\exp(\cdot)$, which refer to the time variable, frequency variable, imaginary unit, $n$th variable, and exponential function, respectively.

# GLOSSARY

| | | |
|---|---|---|
| matrix | a concise and useful way of uniquely representing and working with linear transformations; a rectangular table of elements |
| Functional Analysis | the branch of mathematics concerned with the study of spaces of functions |

# LISTINGS

# Chapter 1

# INTRODUCTION

De La Salle University

## 1.1 Background of the Study

Coffee is one of the most globally consumed beverages. It is a vital product in the global market, with production reaching 168.2 million bags in 2022-2023. The coffee industry is expected to grow even more in the coming years, with output projected to rise by 5.8% in 2023-2024 [International Coffee Association, 2023]. In the Philippines, coffee holds a strong cultural significance, with the local industry continuously expanding. The country is the 14th largest coffee producer in the world. Locally, the industry is expected to grow at a compound annual growth rate (CAGR) of 3.5% from 2021 to 2025, driven by small-scale farm households [Santos and Baltazar, 2022]. With a growing popularity among coffee enthusiasts, the demand for specialty coffee is increasing as well. Consumers are becoming more selective about the quality of their coffee beans [Tampon, 2023].

To stay competitive in the rapidly evolving coffee industry, farmers carefully select high-quality coffee beans for production. Grading green coffee beans is a crucial part of coffee production, as it is directly associated with the quality of the cup quality of coffee brews [Barbosa et al., 2019]. Coffee grading is a process in the industry that determines the quality of coffee beans, using various parameters such as size, density, color, and defects, ensuring that only high quality beans are selected for consumption [Córdoba et al., 2021]. The size of coffee beans is determined using a screen size and sorting procedure, where the coffee beans are categorized into different screen sizes, with larger beans considered higher quality [González et al., 2019]. The density of a bean can be calculated by the ratio of its mass and volume, which greatly influences the roasting process and overall quality of the coffee [Datov and Lin, 2019]. Color is also another indicator for quality, with darker beans being preferred for their richer flavor profile. On the other hand, defects are classified

among 3 categories: Category 1 includes the most severe issues such as foreign matter and black beans, Category 2 includes less severe defects like broken beans, and Category 3 includes minor defects like slight discoloration. Determining the quality of the coffee beans in relation to their defect values is based on quality standards and grading systems such as SCAA protocols guidance or the Philippine National Standard on Green Coffee Bean [Bureau of Agriculture and Fisheries Standards, 2012].

Traditionally, this stage of assessing and categorizing coffee beans relies on visual evaluation, which is time-consuming and labor-intensive, making it prone to human error. One of the biggest challenges in coffee bean production is ensuring consistency in quality. As the demand for specialty coffee continues to grow, there has also been an increase for the need of more efficient and accurate sorting methods. The application of modern technology can help reduce the labor costs and minimize human errors in these tasks. In recent years, computer vision was used alongside various machine learning models and techniques, such as convolutional neural networks (CNNs), support vector machines (SVMs), or K-nearest neighors (KNN) models, where the models were trained on labeled data to classify images of coffee beans into different quality categories. The proposed aims to utilize this technology to develop a two-stage automated coffee bean sorting system using machine vision and density-based analysis to categorize and identify and segregate specialty-grade green coffee beans from non-specialty and defective coffee beans.

## 1.2 Prior Studies

Identifying and sorting specialty-grade coffee beans can be strenuous since the traditional way of classifying a specialty-grade coffee is by manually sorting the coffee bean batch and

classifying them according to the set of standards of the SCAA. The existing work aims to solve these problems through image processing and implementing deep learning-based models to automatically sort the coffee beans while achieving high accuracy. However, these solutions only automate detecting either one of the parameters such as defects, color, and size, while the proposed system considers density, size, color and defects all in one system. Hence, eliminating human intervention or labor. The table below shows the comparison of existing solutions to the researcher's proposal aligning with the traditional way of sorting coffee beans.

TABLE 1.1 SUMMARY OF THE LITERATURE REVIEW

| Existing Literature | Description |
|---|---|
| Defect Detection | The existing literature focuses on using various machine learning models such as YOLO, KNN, and CNN to detect defects in green coffee beans, through identifying visible defects like black spots, broken beans, discoloration, and more. These existing approaches heavily rely on visual characteristics and do not consider other key factors that affect green coffee bean quality like density, which can enhance classification accuracy. The proposed system integrates density and size analysis alongside the defecting various levels of defects on the coffee bean for a more holistic detection and classification. |

| | |
|---|---|
| Coffee Bean Grading and Quality Assessment | The existing literature utilize algorithms such as artificial neural networks, support vector machine, and random forest to grade and classify coffee beans according to the specified grading system. These methods primarily focus on visual features of the beans, which do not account the bean's density and size, which are both essential factors for classifying specialty-grade coffee beans. Additionally, there is a lack of practical implementation of automated sorting systems, as these focus on simply classifying the beans. Through a two-stage process, the proposed system will take into consideration both the visual inspection and the density measurement, which leads to a more complete classification of coffee beans. |

| | |
|---|---|
| Automated Sorting and Classification System | Research has been conducted on developing that automate the process of sorting coffee beans according to various parameters. Some studies focus on sorting defectives against non-defective, while others focus on other visual parameters like defects and roast profiles. These systems focus only on visual characteristics, without considering the actual size of the bean and its density as parameters for better classification accuracy. The proposed system will integrate the use of visual, density, and size parameters to enable a comprehensive automated sorting solution for classifying specialty-grade coffee beans. |

354

TABLE 1.2 COMPARISON TABLE ON EXISTING STUDIES

| Proposed System | [Balay et al., 2024] | [Lualhati et al., 2022] |
|---|---|---|

| | | |
|---|---|---|
| • Defect sorting using EfficientNetV2.<br>• Considers classification of 10 defect types.<br>• The system considers density parameters to sort out less-dense beans.<br>• The system includes a graphical user interface for farmers to visualize the cumulative data of the defects present in the batch.<br>• The system also includes AI-generated recommendations on the possible interventions for the farmers based on the data gathered from the sorting system. | • Defect sorting using YOLOv8<br>• The study considered only 6 types of defects. | • Defect sorting using YOLOv2 and InceptionV3.<br>• The study considered only 2 types of defects. |

## 1.3 Problem Statement

The Philippine coffee industry is a growing market, however it is stuck with using traditional methods in sorting green coffee beans. Often relying on manually sorting the beans, it exposes a number of problems that are apparent in the industry. Relying on manual sorting increases production cost which results in higher prices for quality coffee beans. To make the Philippine coffee beans more competitive to the exported beans, reducing the price is crucial. Another problem that is encountered in manual sorting heavily focuses only on the physical attributes of the bean like size and appearance. There are standards that need to be met, which forces the farmers to resort to manual sorting to comply with the standards

of the SCAA. The SCAA standards require a 300g batch of green coffee beans must not contain any defects and the size consistency of the beans must not exceed 5% variance. Another reason why coffee processors still opt to do manual sorting is because there are no commercially available and reliable GCB sorting machines [Lualhati et al., 2022]. There is a need for a coffee sorter that is able to efficiently and accurately sort GCB. Coffee bean selection is carried out either manually, which is a costly and unreliable process [Santos et al., 2020]. The manual sorting process limits scalability and quality control, putting the strain on farmers as coffee shop owners' demands for high-quality coffee continue to rise [Lualhati et al., 2022].

## 1.4 Objectives and Deliverables

### 1.4.1 General Objective (GO)

GO: To develop an automated (Arabica) green coffee bean sorter that identifies good, less-dense and defective beans from an unsorted batch of coffee beans. The system will utilize machine vision and density-based analysis for defect detection and classification of the coffee beans, ensuring efficient coffee bean sorting.;

### 1.4.2 Specific Objectives (SOs)

- SO1: To gather and create a dataset consisting of 500 high-resolution images of good Arabica green coffee beans and 200 high-resolution images per classification of defective beans (Category 1 & Category 2).;
- SO2: To improve the synchronization between the machine vision system and the

embedded sorting mechanism, ensuring defect sorting of at least 20 beans per minute for stage one, solving issues such as non-synchronization of the system.;

- SO3: To achieve an accuracy of at least 85% in classifying defective green coffee beans using computer vision;
- SO4: To achieve an accuracy of at least 85% in filtering out less-dense green coffee beans;

### 1.4.3 Expected Deliverables

Table 1.3 shows the outputs, products, results, achievements, gains, realizations, and/or yields of the Thesis.

TABLE 1.3 EXPECTED DELIVERABLES PER OBJECTIVE

| Objectives | Expected Deliverables |
|---|---|
| GO: To develop an automated (Arabica) green coffee bean sorter that identifies good, less-dense and defective beans from an unsorted batch of coffee beans. The system will utilize machine vision and density-based analysis for defect detection and classification of the coffee beans, ensuring efficient coffee bean sorting. | A Two-Stage Automated Coffee Bean Sorter System that identifies defective, good beans, and less-dense green coffee bean using machine vision and density-based analysis. |
| SO1: To gather and create a dataset consisting of 500 high-resolution images of good Arabica green coffee beans and 200 high-resolution images per classification of defective beans (Category 1 & Category 2). | • Data Gathering<br>• Image Collection through High Quality Camera |
| SO2: To improve the synchronization between the machine vision system and the embedded sorting mechanism, ensuring defect sorting of at least 20 beans per minute for stage one, solving issues such as non-synchronization of the system. | • Improving the synchronization of machine vision and embedded sorting mechanism of the system. |
| SO3: To achieve an accuracy of at least 85% in classifying defective green coffee beans using computer vision | • Computer Vision Program<br>• Sorting Mechanism |
| SO4: To achieve an accuracy of at least 85% in filtering out less-dense green coffee beans | • Density-based Analysis<br>• Sorting Mechanism |

## 1.5 Significance of the Study

The study explores the implementation of machine Vision and density analysis of an automated coffee been sorter that can identify and sort out the defective, less-dense and good green coffee beans. This said system would aid coffee sorters to mitigate manual labor and to ensure that the sorting process of the GCB are accurate. In order to test the effectiveness of the system, the study would gather data and compare the time efficiency and accuracy of the manual sorting by a an expert sorter to be compared with the proposed system. The system proposes significance to specific parts of society as follows:

### 1.5.1 Technical Benefit

This study would benefit the academe as this introduces a significant advancement in coffee bean sorting technology by implementing both machine vision and density-based analysis to detect and sort good coffee beans, less-dense and separating defective ones. The proposed system would mitigate manual sorting that leads into insufficency like human error and fatigue. The system would improve the overall efficiency by operating at a faster rate compared to manual labor. As a result, it would serve as a proof of concept for the implementation of machine vision and density-based analysis in agricultural industries specifically in the Philippine coffee industry.

### 1.5.2 Impact to the Coffee Industry

The study would aid coffee farmers and producers, by providing an automated system that ensures accurate sorting of Arabica green coffee beans, the system aims to have an accurate output to help maintain to yield higher quality coffee beans and allows coffee bussinesses

to scale up their operations, increase the competitiveness of exporting those beans, and meet demand more efficiently. The productivity given from the system would potentially strengthen the foundation of local coffee producers.

## 1.6 Assumptions, Scope, and Delimitations

### 1.6.1 Assumptions

1. There would be a defective coffee bean from the green coffee bean test batch;
2. Identifying the defective coffee beans using the machine vision and density-based analysis would be much more efficient and accurate than manually sorting them;
3. During testing, test batches will contain 50% good beans and 50% defective beans, 60% good beans and 40% defective beans, 70% good beans and 30% defective beans, 80% good beans and 20% defective beans, 90% good beans and 10% defective beans, 100% good beans;

### 1.6.2 Scope

1. The study only focuses on Arabica green coffee beans;
2. The study has two stages, the first stage would segregate the defective green coffee beans from the batch, then the second stage would identify the specialty-grade green coffee beans depending on its density;

### 1.6.3 Delimitations

1. The batch of coffee beans to be used for testing and dataset collection will consist solely of Arabica beans from the same origin, farmer, and processed in the same way;
2. The system is only limited to unroasted green coffee beans;
3. The batch of coffee beans to be used should only be dehulled and not sorted visually and by density;
4. Since the system is considering several types of defects and density parameter, sorting time is compromised;
5. The system is designed to perform individual scanning of each coffee bean;

# Chapter 2

# LITERATURE REVIEW

## 2.1 Existing Work

TABLE 2.1 REVIEW OF RELATED LITERATURE

| Literature | Description of the Literature |
|---|---|
| [Balay et al., 2024] | This study focused on the development of an automatic green coffee bean sorter. The algorithm used is the YOLOv8 to train the model, while a Raspberry Pi was used in order to test the model along with the sorting mechanism. There are a total of 6 defects that the system can detect these are full black, partial black, chipped, dried cherry, shell and dried cherries. A total of 10 trial were done to effectively test the system. Out of the 10 trials, 9 trials were found to have an average target sensitivity of 97.8%, with an average time of 2 minutes and 32 seconds for a total of 100 beans. |
| [Amadea et al., 2024] | In this study, a system was developed to detect defects in Arabica green coffee beans. The study used two different models such as Detection Transform (DETR) and You Only Look Once version 8 (YOLOv8). Upon comparison, YOLOv8 showed strengths in defect detection. On the other hand, DETR model showed significant strengths than the YOLOv8 model when it comes to defect detection. |

| | |
|---|---|
| [de Oliveira et al., 2016] | This study constructed a computer vision system that outputs measurements of green coffee beans, classifying them based on their color. In the system, Artificial Neural Network (ANN) was used as the transformation model. On the other hand, the Bayes classifier was used in classifying the coffee beans into four (whitish, cane green, green, and bluish-green). The model was able to achieve a small error of 1.15%, while the Bayes classifier achieved a 100% accuracy. To concluded, the developed system was able to effectively classify the coffee beans based on their color. |
| [Balbin et al., 2020] | In this study, the objective is to provide better technology for local coffee producers to increase export-quality beans production. Thus, the study proposed a device that can evaluate the size, quality, and roast level of a batch of beans fed into the machine. The model used in the system was the Black Propagation Neural Network (BPNN), together with other image processing techniques such as K-mean shift, Blob, and Canny Edge. These techniques were used to extract the features of the beans and analyzed using RGB analysis. |

| | |
|---|---|
| [Pragathi and Jacob, 2024] | The paper discusses the use of machine learning algorithms such as KNN and CNN to classify the specialty type coffee bean for Arabica. The coffee bean quality of an Arabica can be classified by the number of defective coffee bean presents in a sample. The defects are classified into two categories named primary and secondary. |
| [Lualhati et al., 2022] | With the lack of a locally made green coffee bean sorter in the Philippines, the researchers aimed to design and implement a device that will handle the sorting. The paper discusses the development of a Green Coffee Bean (GCB) quality sorter. The system used a PID based algorithm and image processing algorithm for sorting. It utilized two cameras to capture images of both sides of the GCB, this was done to check for the quality of the GCB through a prediction test. The paper conducted a total of 5 tests, each with varying conditions. The designed system on average got an accuracy score of 89.17% and sorting speed of 2 h and 45 mins per 1 Kg of GCB. |

| | |
|---|---|
| [García et al., 2019] | The paper discusses the use of computer vision for quality and defective inspection for GCBs. The paper makes use of parameters such as color, morphology, shape, and size to determine the quality of the GCB. It makes use of the algorithm k-nearest neighbors (KNN) to differentiate the quality and to identify the defective beans. The designed prototype makes use of an Arduino MEGA board to gather the data and a DSLR camera to capture the GCB. The type of bean used was an Arabica, and a total of 444 grains were used to test the prototype. The accuracy score for both the quality evaluation and defective beans resulted in an average of 94.79% and 95.78% respectively. |

| | |
|---|---|
| [N.S. Akbar et al., 2021] | The researchers proposed a system that sorts the Arabica coffee into 2 classes, defective and non-defective. After the classification into two classes, the coffee beans are then graded based on the quality consisting of: specialty grade, premium grade , exchange grade, below grade, and off grade. Utilizing computer vision for classifying the defective and non-defective beans, the researchers used the color histogram and the Local Binary Pattern (LBP) to get the color and the texture of the beans. The data gathered from both the color histogram and LBP are used to train two models, the random forest algorithm and the KNN algorithm. The results from both algorithms are both promising, with an average accuracy score of 86.56% using the random forest algorithm and 80.8% for the KNN algorithm, However, this result shows that utilizing the random forest algorithm provided better accuracy scores for the model. |

| | |
|---|---|
| [Huang et al., 2019] | The paper discusses the development of a GCB sorter in real-time by using Convolutional Neural Network (CNN) . The researchers used a total of 72,000 images of good and bad beans, 36,000 per category respectively. A total of 7,000 images for the beans were picked at random to test the model, while the remaining was used to train the model. To test the model, a webcam was used to record the coffee beans, however this resulted in capturing only the topside of the bean,to solve this the beans were flipped to provide accurate results. This resulted in an average accuracy score of 93.34% with a false positive rate of 0.1007. |

| | |
|---|---|
| [Luis et al., 2022] | The paper focuses on using You Only Look Once (YOLOv5) as the algorithm for detecting the defective GCB. The researchers used a Raspberry Pi camera to capture the images of the coffee beans. To test the effectiveness of the developed system a total of 45 trials were conducted with varying classification that the model was trained on. The model tested a total of 15 trials for each classification, these classifications are black, normal and broken. Each classification provided different accuracy results, for the blackened coffee bean, a total of 106 coffee beans were tested which resulted with an 100% accuracy by correctly identifying 106 blackened coffee beans. For the normal coffee bean, a total of 117 beans were used which resulted in an accuracy score of 91.45% since only 107 out of 117, were accurately classified. Lastly, a total of 104 broken beans were used, which resulted with an accuracy score of 94.23% since only 98 beans were correctly classified. The average accuracy score of the system developed resulted in an average of 95.11%. |

| | |
|---|---|
| [Santos et al., 2020] | In this study, the development of quality assessment of coffee beans through computer vision and machine learning algorithms. The main parameters that this study considers are the shape and color features of the coffee bean and they used machine learning techniques such as Support Vector Machine (SVM), Deep Neural Network (DNN) and Random Forest (RF), to identify the coffee beans' defect. The script written in Python Language was used to extract shape and color features of the coffee beans based on the datasets. Overall, the system had a very high accuracy (¿88%) on classifying coffee beans through the models that have been developed. |
| [Arboleda et al., 2020] | The study proposed a novel solution that deals with the low signal-to-noise ratio. The study shows a way of extracting features of an image in context with green coffee beans. The researchers concluded a new edge detection approach for green coffee beans. It was achieved by using the heuristic approach in calculating the right values for the discriminant and finding the best pixel formation. |

| | |
|---|---|
| [Susanibar et al., 2024] | The proposed system aims to implement a GCB automated classification based on size and defects. The paper classified each bean into three different sizes. The system used two stages to identify the sizes of each bean. Firstly the entrance of the system was measured to ensure that the bigger beans are not able to pass through. The second stage involves the use of a cylindrical sieve with holes. This resulted with an average accuracy score of 96% for classifying the beans in size. However, the system does not provide a good accuracy score in classifying beans in terms of its defect since it only averaged 80% when classifying the defects of the beans. |
| [Srisang et al., 2019] | The study proposed an oscillating sieve as the main way for sorting Robusta coffee beans. Sizes are differentiated into 4 classes: extra large (XL), large (L), medium (M), small (S). The sieve resulted in an accuracy score of around 79% in classifying the sizes of the coffee beans. |

## 2.2 Lacking in the Approaches

TABLE 2.2 COMPARING PROPOSED STUDY AND EXISTING STUDIES

| **Existing Studies** | **Proposed Study** |
|---|---|

| | |
|---|---|
| • Uses computer vision to classify green coffee bean grade based on its visual characteristics such as size, color, and shape.<br>• Most related studies classify defective and non-defective beans only.<br>• The density parameter of the green coffee beans is not considered.<br>• Similar study [Lualhati et al., 2022] only considered three classifications of GCBs: Good, Black, and Irregular-Shaped beans.<br>• Similar automated GCB sorter [Balay et al., 2024] only considered one side of the bean.<br>• Existing classification of GCBs with automated sorters do not have an integrated graphical user interface (GUI) for data analytics. | • Computer vision will be used to analyze the physical characteristics of the bean, including its volume.<br>• Density parameters will be considered by implementing a weighing scale to the system.<br>• The system will implement two stages of sorting:<br>– The first stage sorts out the defective beans.<br>– The second stage sorts out the potential specialty-grade beans based on their density and size.<br>• The system is designed to inspect both sides, utilizing two cameras.<br>• The system is designed with a GUI for farmers to visualize the cumulative data of the defects present in the batch. |

## 2.3 Summary

The various related literature discusses the numerous technological advancements related to coffee bean sorting to aid coffee farmers and producers on efficient sorting and classification of beans. These studies provide insights regarding the various methods used in the field of coffee sorting that utilize machine vision, density-based analysis, and deep learning to identify and classify coffee beans based on their physical parameters. Numerous studies discussed parameters like size, defects, and color. However, existing studies tend to focus primarily on visual characteristics and lack integration density analysis for accurate classification of green coffee beans. The review literature identifies and acknowledges the gaps in current sorting practices, such as the lack of comprehensive systems that implement machine vision and density-based analysis. The study aims to address these gaps by proposing a two-stage sorting system that automates both detection of defective beans and the classification of less-dense beans. Density and size will play a significant role, as it is linked to identifying the quality of the coffee bean. However, related literature mentioned overlooks this parameter for classifying the coffee bean. Higher density beans are often associated with higher quality coffee beans, into being potential specialty-grade coffee after roasting and cupping.

# Chapter 3

# THEORETICAL CONSIDERATIONS

## 3.1 Theoretical Framework

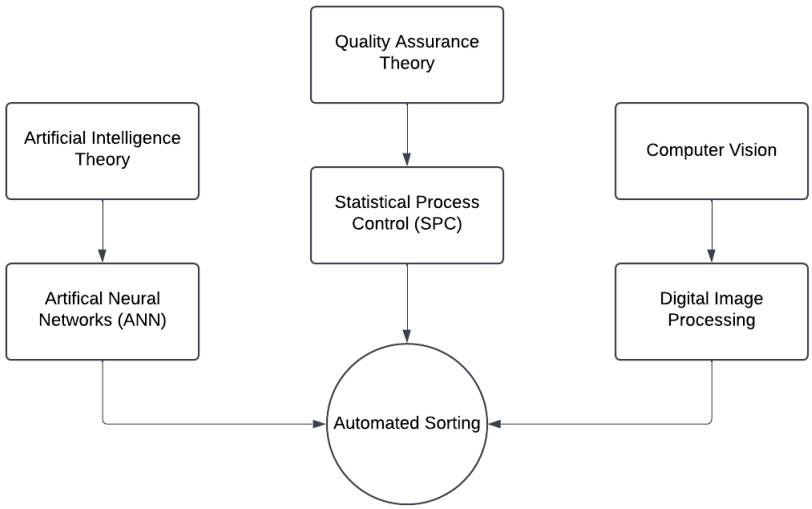



Fig. 3.1 Theoretical Framework

The theoretical framework discusses the multiple concepts that are involved in this study. These key concepts are crucial to ensuring the success of the thesis. There are three main concepts that are key to this study, the Artificial Intelligence Theory, the Quality Assurance Theory and lastly, Computer Vision.

## 3.2 Conceptual Framework

The conceptual framework shows the implementation of two systems which consists of machine vision and embedded systems. The framework describes the thought process of both systems with the end goal of integrating both systems. The machine vision handles the defect classification of the system, whereas the embedded system handles the sorting of

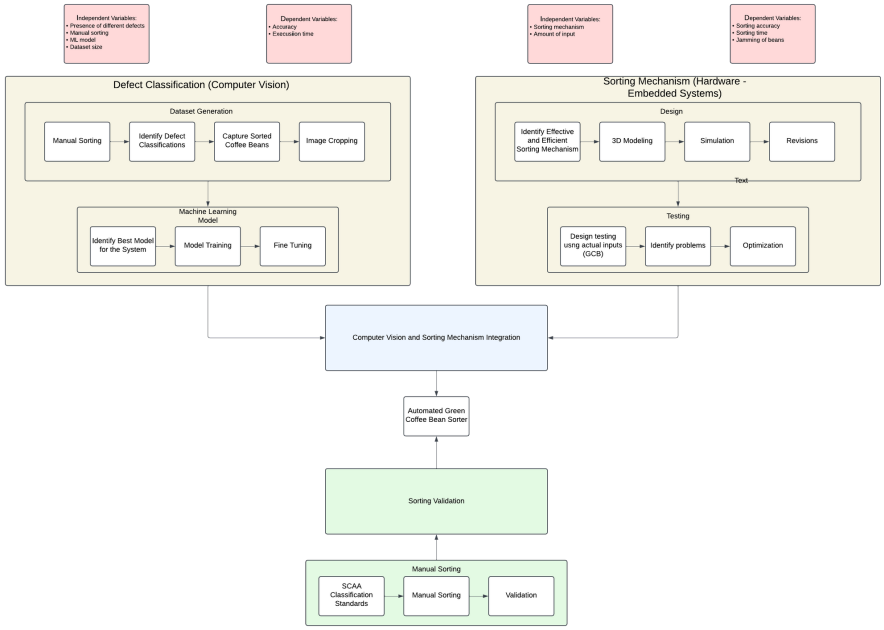



Fig. 3.2 Conceptual Framework

the beans. By integrating both systems together, creates an automated green coffee bean sorter. The data validation is done by sorting through the tested coffee beans by the system following the standards of the SCAA.

## 3.3 Quality Assurance Theory

Quality assurance theory refers to the set of principles and practices that focuses on establishing a systematic process to ensure that a product or service conforms to a predetermined standard. In the aspect of food and agriculture, there are a number of practices and principles that ensure the safety and quality of food products. According to [da Cruz et al., 2006], there are a number of practices in place that must be followed, one of which is Good Agricultural Practices, where these procedures are aimed to reduce hazards related to

product safety at the farm level. Another one of said practices is the Good manufacturing practice, which were formerly called support programs that provide foundations to the overall food safety management programme. This includes cleaning, maintenance, personnel training, calibration equipment, quality control, and pest control. Industries that adopt such practices produce the following results, better quality products, greener initiatives and better productivity within a department. Lastly, hazard analysis and critical control points (HACCP), is a science-based system that was created to identify potential hazards and actions to control said hazards. This practice is used to ensure food safety.

In the context of coffee beans, there are a number of systems in place to ensure that quality beans are being provided to the consumer market. The governing body known as the Specialty Coffee Association (SCAA) has implemented grades to green coffee beans to provide a better way to classify said beans. These grades can be differentiated into 5 grades namely, Specialty Grade, Premium Coffee Grade, Exchange Coffee Grade, Below Standard Coffee Grade, and Off grade Coffee. They are classified according to the number of defects found in a sample batch of 300 grams and according to their size. Specialty grade coffee beans are supposed to contain less than 5 defects in a sample batch while also not allowing any primary defects to be present; it should only have less than 5% difference between its sizes. Coffee beans in this grade should also contain a special attribute whether in its body, flavor, aroma, or acidity, and its moisture content should only be in the range of 9-13%. Premium Coffee grade beans should only contain 8 full defects in a sample batch but primary defects are allowed in the sample batch. Similarly to specialty grade coffee beans, its sizes should only contain a 5% difference to one another; it should also contain a special attribute and moisture content should also be similar to its specialty grade counterpart. Exchange coffee grade should contain defects ranging from 9-23 beans in a

sample batch, with sizes that can vary up to 50% difference in weight but also only 5% in its sizes. Below standard and off grade coffee beans are classified according to the number of defects present in a sample batch; 24-86 beans for below standard while more than 86 beans for off grade. These gradings are used to ensure that quality green coffee beans are produced and ensure that consumers are provided with the best quality available.

## 3.4 Artificial Intelligence Theory

Artificial Intelligence in defect classification are widely used in this industry which are commonly used in manufacturing and industrial applications. Several deep learning techniques are used in order to achieve an effective defect classification. Models such as convolutional neural networks (CNNs) and You Only Look Once (YOLO) are widely used for classification. CNN utilizes an image based analysis and feature extraction approach to identify different classifications. CNN is more effective in analyzing grid-like data like images, making it suitable for defect classification [Das et al., 2019]. One of its major advantages is its ability to automatically detect important features such as shape, patterns, and edges. Although it may have its own advantages, there are also disadvantages that need to be taken into account, mainly in scenarios that involve class imbalance and complex backgrounds (Moon, 2021) . YOLO is another model that is suitable for defect classification, its ability to provide real-time defect classification while also providing high accuracy is essential in some industries. In YOLO, there are several versions that are developed over the years, which are supposed to bring several improvements in terms of speed, accuracy, and computational efficiency. Combining different models is also effective, in the case of [Deepti and Prabadevi, 2024], they combined transformer architecture with YOLOv7

to enhance its feature extraction, this resulted in an increase of 5.4% in mean average precision and F1 score.

## 3.5 Computer Vision Theory

There are fundamental concepts that need to be done for image processing in detection. There are pre-processing techniques like preprocessing and segmentation. Pre-processing is a general term for preparing an image to be analyzed by the system, this includes techniques such as denoising an image, applying filters, and enhancing the image to further improve the visibility of defects [Lee and Tai, 2020] . Segmentation is dividing the images into segments to make the analysis simpler, methods such as histogram segmentation and active contour models helps in isolating the regions of interest.

For defect classification, feature extraction is important to identify the relevant features then extracting said features to to help indicate specific defects, this utilizes the edges, textures, and shapes to help in defect classification [Wu et al., 2024]. BY utilizing OpenCV and deep learning models is advisable for automatic feature extraction. Models like CNN, can automatically extract features from images, which greatly reduces the need for manual extraction, this helps in a more robust and scalable solution [Bali and Tyagi, 2020]. The versatility of OpenCV library which allows support for multiple image pre-processing tasks, when combined with deep learning models can be applied to different fields.

## 3.6 Performance Evaluation

Accuracy, precision, recall, and F1 score are common measures to assess how well classification models predict. Accuracy measures how good a model is by computing the ratio of correct predictions to all predictions. While appropriate for balanced datasets, accuracy can be deceptive when dealing with imbalanced classes, since a model can be very accurate by predicting the majority class. Precision measures how well positive predictions are obtained by calculating the number of correct predicted positive instances. This is particularly important when false positives are costly, such as in the case of spam. Recall, or sensitivity, measures how well a model identifies true positive instances, which is very important in cases where failing to detect a positive instance is costly, such as in medical diagnosis. Since precision and recall trade off each other, the F1 score reconciles the two by computing their harmonic mean. This measure is particularly appropriate when a trade-off between precision and recall is desired, so that neither false positives nor false negatives dominate the assessment. In general, these measures provide a general impression of how good a model is and help decide how well-suited the model is for different applications.

## 3.7 Existing Technologies and Approaches

The paper done by [Lualhati et al., 2022], is a green coffee bean sorter that utilizes MATLAB as its image processing. The system created uses a PID based algorithm and image processing algorithm for sorting. The system utilized two cameras to capture both sides of the bean. The system of Lualhati et al. comprises only 3 green coffee bean classifications, which are good, black and deformed coffee beans. The developed system uses multiple stepper motors for the defect sorting, while 2 cameras were used to handle

the green coffee bean detection.

The paper of [Balay et al., 2024], is an automatic sorting for green coffee beans utilizing computer vision and machine learning for defect classification. The system developed uses the YOLOv8 model alongside a Raspberry Pi based image processing to identify and classify the green coffee beans. The defects that the group classified are full black, partial black, chipped, dried cherry, shell, and insect damage. The system developed uses a conveyor belt and sorting motor for an automated defect separation. They used one camera module, the raspberry pi camera module 3 NoIR for the defect detection of the system.

## 3.8 Density Measurement

In measuring the density of the coffee bean there are a number ways this can be done, one way is by measuring the bulk density of the batch. This is done by measuring the mass of a batch then dividing it to a fixed volume. The more appropriate method for measuring the density of the coffee bean is called “free settle” density or free-flow density. This is defined as the ratio of the mass of the coffee beans to the volume they occupy after being allowed to flow freely into a container. It is expressed in grams per liter or kilograms per cubic meter.

## 3.9 Summary

This chapter gives the theoretical and conceptual backgrounds of an automated green coffee bean sorter using Artificial Intelligence (AI), Quality Assurance, and Computer Vision. The theoretical background focuses on key concepts like deep learning models (CNNs, YOLO) used for defect classification, quality assurance principles (GAP, GMP, HACCP) ensuring

food safety, and computer vision algorithms (preprocessing, segmentation, and feature extraction) used for image analysis. The conceptual background explains the integration of machine vision for defect detection with embedded systems for sorting, thus conforming to the SCAA coffee grading standards. Performance metrics like accuracy, precision, recall, and F1 score are used for evaluating the performance of the model. Current technologies, for instance, those of [Lualhati et al., 2022] and [Balay et al., 2024], provide insights relevant to image processing and machine learning-based sorting techniques, thus contributing to automated coffee bean classification development.

# Chapter 4

# DESIGN CONSIDERATIONS

## 4.1 Mechanical Design

### 4.1.1 Screw Feeder

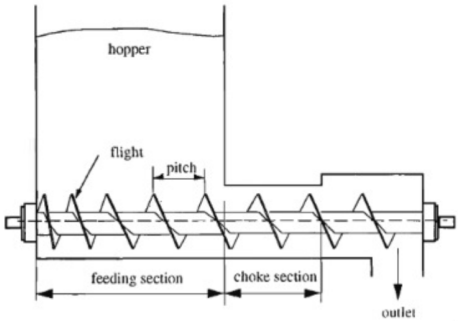



Fig. 4.1 Screw Feeder Diagram

Figure 4.1 shows the diagram of a screw feeder. Screw feeders are usually used in industrial fields like agriculture, chemicals, plastics, cements, poultry and food processing. According to [Minglani et al., 2020], screw feeders are specifically used to transport or move granular materials at a controlled rate like corn and wheat. It consists of a rotating screw and small feeding section or the hopper. Despite having big batches of a certain material, screw feeders can control the rate of which these materials are dispensed. With this concept, the group decided to utilize a screw feeder as the input mechanism for the system. This mechanism allows a controlled rate of coffee bean dispensing, which is a significant factor to avoid overcrowding in the rotating conveyor table causing the beans to jam. In addition, batches of coffee beans can be put at once instead of just adding a certain amount of beans at a time.

Fig. 4.2 Rotating Conveyor Table 3D Design, 32-inch Rotary Table Accumulator (RTA)

### 4.1.2 Rotating Conveyor Table

After the inputted beans comes out from the screw feeder, the coffee beans would then be placed in the rotating conveyor table. According to the study of [**?**]. The conveyor table is used as a transportation systems for all forms of bulk materials to a certain machine or destination. The system utilizes the rotating conveyor table to have a controlled movement of coffee beans towards the first stage of the system. The improvised linearization system, consisting of metal guide rails and dividers ensures that beans align in a single path, reducing random movement, and improving the flow of the input beans. An infrared sensor would detect each bean as it passes, to control the movement of the bean preventing clogging and ensuring efficient operation.

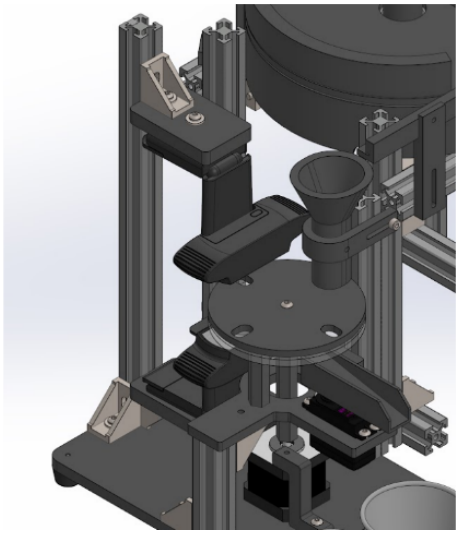


Fig. 4.3 Inspector Tray 3D Design

### 4.1.3 Inspection Tray (1st Stage)

The inspection tray serves as the platform for the machine vision based analysis of coffee beans. It is designed with 8 holes, allowing uniform placements and optimal camera positioning for the system. The system utilizes a two-layer structure: a stationary acrylic platform and a rotating 3D-printed platform with holes. The rotating mechanism sequentially positions each bean between two webcams, which captures and analyzes its physical characteristics from top and bottom perspective. This design captures both sides of the bean, ensuring a better classification of the bean. After inspection, the bean moves onto a slide, where it is either directed to the second stage for density analysis (Good) or sorted out as a defect.

### 4.1.4 Density Sorter (2nd Stage)

In measuring the density of the coffee bean there are a number ways this can be done, one way is by measuring the bulk density of the batch. This is done by measuring the mass of a batch then dividing it to a fixed volume. The more appropriate method for measuring the density of the coffee bean is called “free settle” density or free-flow density. This is defined as the ratio of the mass of the coffee beans to the volume they occupy after being allowed to flow freely into a container. It is expressed in grams per liter or kilograms per cubic meter.

## 4.2 Embedded Systems

### 4.2.1 Microcontroller

Fig. 4.4 Arduino Nano Microcontroller

Since the system is composed of two stages of sorting: defect sorting through computer vision and density-based analysis–the group decided to utilize two Arduino Nano microcontrollers to modularize the control process. The first Arduino Nano microcontroller is tasked to handle the computer vision-based defect sorting through serial communication with OpenCv operating in Python. In addition, it handles the operation of defect sorting

consisting of a stepper motor for the rotation of the inspection tray and a servo motor for the slider, which directs the beans to the designated bin (defect or good bin). On the other hand, the second Arduino Nano microcontroller manages the density-based analysis and sorting, which consists of another stepper motor to direct the beans to its respective bin (dense and less-dense bin), the precision scale which is interfaced through RS232, and the top feeder where the input beans are poured. The use of separate Arduino microcontrollers is advantageous when it comes to the computer vision-based sorting of beans. This is because serial communication is much faster when code complexity is significantly reduced. With this, a designated microcontroller handles the computer vision part and two-way serial communication between the microcontroller and the computer vision algorithm running in Python. Most importantly, the use of two microcontrollers allowed the system to not rely solely on a sequential approach. This means that the two stages of sorting are not relying on the timing of each other, allowing the inspection tray and the top feeder to operate independently. Thus, resulting in a much faster and efficient sorting process.

### 4.2.2 Sensors

Fig. 4.5 Infrared Sensor

To ensure that the beans are falling in a one-by-one manner onto the inspection tray, the group placed an IR sensor at the edge of the top feeder. This IR sensor triggers the DC motor that runs the feeder to stop, and runs small steps until the bean is dropped. The addition of the IR sensor at the edge of the feeder allows the motor to run continuously until another bean is detected. With this, the waiting time for the next bean at the inspection tray is significantly lessened.


Fig. 4.6 TOF10120

TOF10120 or Time of Flight sensor is utilized in the system due to its high precision, non-contact measurement capability. This sensor is used to estimate the volume of each bean, which is essential for computing the density. In the second stage of sorting, where beans are classified based on density, the sensor plays a crucial role in determining the approximate volume of each bean by measuring its height or dimensions as it passes through the system.

### 4.2.3 Motor control

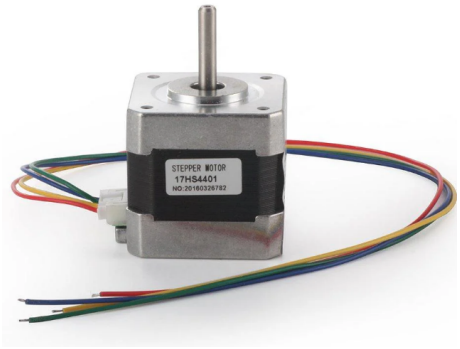



Fig. 4.7 12V NEMA 17 Stepper Motor

Two NEMA 17 12V stepper motors, paired with L298N motor drivers were used to control the movement of the inspection tray in the first stage and the density-based sorting mechanisms in the second stage. In these mechanisms, the group decided to use stepper motors to ensure precise and accurate movements. Precise and accurate movements are needed for the inspection tray to make sure every movement of the hole is perfectly aligned to the camera. Thus, allowing a more uniform and consistent angle for each bean to be inspected through the computer vision. In addition, NEMA 17 stepper motors were the best choice for these mechanisms due to its high torque, which is essential because it will be moving weighted objects.

Fig. 4.8 6V DC Motor

For the rotating conveyor table (top feeder), where the beans are initially poured, a 6V DC motor is used. The group decided to use this motor due to its high RPM, which is needed for a fast rotation of the rotating conveyor table. The speed of the feeder is regulated to prevent clogging and ensure that the beans are evenly spaced before they enter the inspection tray. The motor speed is fine-tuned through pulse-width modulation (PWM) to synchronize with the stepper motor-driven inspection tray, ensuring a steady input without overwhelming the system.

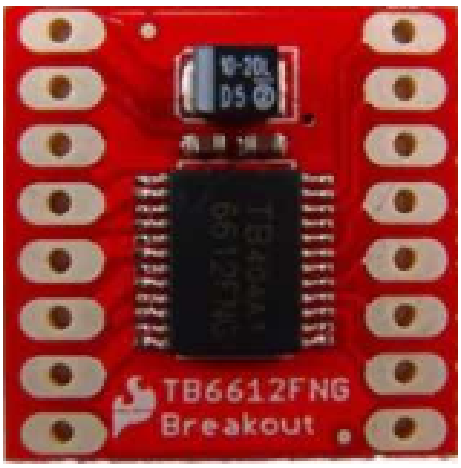



Fig. 4.9 TB6612FNG Motor Driver

To drive the 6V DC motor, the group utilized TB6612FNG, a motor driver module. This module also allowed PWM control for the motor, which is essential for reducing the speed of the motor when needed.

### 4.2.4 Operating Voltage


Fig. 4.10 12V Power Supply

The main power supply comes from a 12V external power supply, which provides enough voltage for all the components and keeps the voltage from dropping and interfering with system performance. The Arduino microcontroller is powered via its VIN pin, so it can function without the need for a USB connection and maintains a stable 5V logic output for sensor and actuator control. The NEMA 17 stepper motors that operate the inspection tray and density sorter are directly powered from the 12V supply and fed into L298N motor drivers to adjust voltage and monitor current flow. Operating these motors at 12V provides best torque output, which is vital in ensuring consistent movement during the sorting process.


Fig. 4.11 MT3608 Step-Up Module

For the top feeder mechanism, a step-up module is needed to supply the sufficient voltage needed for the motor–6V. From the 5V output of the Arduino, the step-up module will be utilized to convert it into 6V.

## 4.3 Computer Vision System

### 4.3.1 Image Processing

Fig. 4.12 C920 Camera

The system requires clear images of the coffee beans for accurate processing by the detection and classification models. Two C920 cameras will be used to capture images from opposite sides of each bean—one positioned on top and the other at the bottom. The captured images will then be processed within the laptop using the detection and classification models to identify and categorize the beans.

### 4.3.2 Object Detection and Classification Models

The object detection model identifies and isolates the coffee beans from the background. For this task, different models were explored:

1. **RF-DETR**

   A transformer-based object detection model that eliminates the need for anchor boxes, improving small object detection.

2. **YOLOv11**

   A CNN-based YOLO variant that incorporates the C3k2 block, SPPF, and C2PSA components to enhance feature extraction and detection accuracy.

3. **YOLOv12**

   The latest YOLO version and attention-centric model that integrates transformer-based components to enhance performance while maintaining real-time efficiency.

### 4.3.3 Object Classification Models

Following detection, each identified coffee bean was cropped and classified based on its defect type. The classification models used included:

1. **EfficientNetV2**

   A convolutional neural network (CNN) designed for high efficiency and accuracy, balancing computational cost and performance.

2. **YOLOv8**

   A lightweight yet highly accurate model that supports both object detection and classification, making it suitable for real-time applications.

3. **YOLOv11**

   A classification-specific adaptation of YOLOv11, leveraging enhanced feature extraction techniques for defect recognition.

4. **YOLOv12**

   A classification variant of YOLOv12, incorporating advanced attention mechanisms to improve accuracy.

## 4.4 Serial Communication

Serial communication is used for sensors and motors for arduino due to the simplicity, reliability and efficient transfer of data between different devices. The precision scale uses a RS232 and a MAX TTL converter to send the data from the precision to the arduino to get the weight values of each green coffee bean. To sort out the good from defective beans the system utilizes a servo motor. The data from python is received by the arduino through serial communication. The python side is responsible for the decision and defect classification while the arduino is responsible for controlling the servo motor.

## 4.5 Graphical User Interface (GUI)


Fig. 4.13 Graphical User Interface

The proposed system would be integrating a graphical user interface developed using PyGui and ChatGPT API. The GUI would serve as the control center platform for the system. This would provide real-time feedback and insights for users. As shown in Figure 8, a concept of how the GUI would interact with the system would be a start button, once the button is executed the system would then be expecting inputs and start sorting. There would be real-time feedback during the sorting process, then some visual markers to indicate their classification, and an elapsed time so the user would be aware of the time of the sorting process. Once the system is done, the user can click the end button and the summary report would generate in an orderly manner, providing tables of classification that was detected through the process. In the bottom part of the GUI, ChatGPT API would be integrated and would offer recommendations based on the detected quality and classification of the coffee beans.

## 4.6 Density Analysis

The density analysis works by using a precision scale to measure the mass of the bean. To get the data from the precision scale, serial communication is used from the scale to an arduino nano. This is done by using a RS232 with a Max TTL converter for the arduino to read the data from the precision scale. To sort out the good from defective beans the system utilizes a servo motor for the density sorting mechanism. The servo motor is used to sort the dense from the less dense beans. The sorting mechanism developed consists of gears and cross-shaped modules to properly capture the beans and properly sort them out.

## 4.7 Technical Standards

### 4.7.1 Hardware

In the design and development of the system, the group incorporated and followed a series of technical standards. One of which is ISO 12100:2010 – Safety of Machinery, where general principles for risk assessment and reduction are discussed. Thus, the system is designed, while keeping in mind the hazards associated with moving parts, making sure that all moving parts in the system do not need to be touched for operations. An emergency stop is also integrated into the system to stop all the moving parts in case of undesirable incidents [International Organization for Standardization, 2010].

On top of this, ISO 14121-1 – Risk Assessment for Machinery was also followed to further assess the potential risks throughout the system. The standard includes identifying and quantifying hazards such as electrical short circuits, faulty wirings, and motor overheating [International Organization for Standardization, 2007]. With this, the system

included protective enclosures for the electrical wirings, proper grounding of the circuits, and controlled motor actuation. More specifically, for motors, it was made sure that the design has sufficient voltage and ampere to power the different kinds of motors used with the use of L298N, and MT3608 modules. These are the main components for adjusting motor speeds dynamically during the sorting process.

Lastly, ISO 30071-1 was standard used to provide sufficient lighting during data collection, and real time bean inspection during sorting process. This standard helps ensure consistent and non-glare lighting conditions, which are essential for the machine vision cameras to accurately capture bean features [International Organization for Standardization, 2019]. Uniform illumination improves the reliability of image classification by reducing shadow artifacts and reflections, thereby enhancing overall detection performance.

### 4.7.2 Software

For the software side of the system, the first applicable standard is ISO/IEC 25024 – Systems and Software Engineering – Measurement of Data Quality, which offers a systematic method for measuring the quality of datasets utilized in information systems [International Organization for Standardization, 2015]. This standard was used during the dataset gathering and training for the different coffee bean defects like black, sour, insect damage, fungus damage, broken, floaters, and dried cherry. Practically, this included pre-processing the image data to eliminate noise, balance class distribution, and verify ground truth labels.

Lastly, ISO/IEC 23053 – Framework for Artificial Intelligence (AI) offers a reference architecture to build and integrate machine learning building blocks [International Organization for Standardization, 2022]. This standard was highly applicable in determining the design of the machine vision module, where a pre-trained deep learning model is utilized

for the classification of bean defects. This standard provides guidelines on best practice for the overall machine learning cycle, ranging from data acquisition, feature extraction, and model training through to model evaluation, deployment, and monitoring.

### 4.7.3 Green Coffee Bean Sorting

For sorting green coffee beans, Specialty Coffee Association of America (SCAA) Standards for Green Coffee Bean Sorting was incorporated to maintain conformity. The standards set the definition for the classification of primary and secondary defects (i.e., black, sour, insect-damaged, broken, and floater beans) and sets the maximum allowable defect counts for specialty-grade coffee. The SCAA standards were applied to mark the training set of the machine vision model and also to set up the thresholds of defect classification, so visually defective beans can be correctly classified and rejected. Also, the sorting mechanism based on density points towards SCAA bean weight and volume guidelines using a precision scale and ToF sensor to sort beans based on within-acceptability density limits.

On the other hand, the system also adheres to PNS/BAFS 341:2022, the Philippine National Standard for Agricultural Machinery – Coffee Green Bean Grader – Specifications and Methods of Test [of Agriculture and Standards, 2022]. It sets local criteria for testing coffee grading equipment on performance, safety, construction aspects, and methods of test. For the purposes of this research, PNS/BAFS 341:2022 is used as a reference for the design of the sorting mechanism, specifically in terms of the materials used in construction, handling of beans, and the efficiency with which the mechanical and electronic subsystems segregate. It also guides the testing procedure employed to verify sorting precision, capacity, and rates of misclassification under test conditions.

# Chapter 5

# METHODOLOGY

TABLE 5.1 SUMMARY OF METHODS FOR REACHING THE OBJECTIVES

| Objectives | Methods | Locations |
|---|---|---|
| GO: To develop an automated (Arabica) green coffee bean sorter that identifies good, less-dense and defective beans from an unsorted batch of coffee beans. The system will utilize machine vision and density-based analysis for defect detection and classification of the coffee beans, ensuring efficient coffee bean sorting. | • DDR Methodology<br>• Description of the System | Sec. 5.1 on p. 56<br>Sec. 5.2 on p. 59 |
| SO1: To gather and create a dataset consisting of 500 high-resolution images of good Arabica green coffee beans and 200 high-resolution images per classification of defective beans (Category 1 & Category 2). | • Dataset Collection<br>• Manual Sorting | Sec. 5.3 on p. 60 |

*Continued on next page*

*Continued from previous page*

| Objectives | Methods | Locations |
|---|---|---|
| SO2: To improve the synchronization between the machine vision system and the embedded sorting mechanism, ensuring defect sorting of at least 20 beans per minute for stage one, solving issues such as non-synchronization of the system. | • Data Collection<br>• Dataset preprocessing<br>• Model Training<br>• Serial Communication | Sec. 5.3 on p. 60<br>Sec. 5.5 on p. 66 |
| Sec. 5.7.1 on p. 76<br>SO3: To achieve an accuracy of at least 85% in classifying defective green coffee beans using computer vision | • Dataset preprocessing<br>• Model Training | Sec. 5.5 on p. 66 |
| SO4: To achieve an accuracy of at least 85% in filtering out less-dense green coffee beans | • Density Threshold Calibration Using Water Displacement Method<br>• Density Sorter | Sec. 5.4 on p. 65<br>Sec. 5.6.4 on p. 75 |

## 5.1 Description of the System

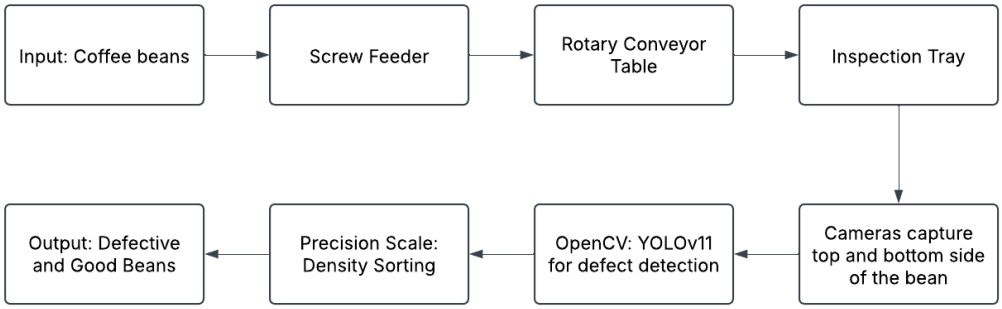



Fig. 5.1 System Block Diagram

The proposed system is a two-staged automated green coffeee bean sorting machine, integrating both machine vision and density analysis. Firstly, the coffee beans are introduced into the system through a funnel, which directs them to a conveyor belt mechanism. In the first stage, the green coffee beans will be sorted depending on their visual characteristics. In this stage, the physical qualities of the bean is analyzed such as size, color, and defect. If the bean is defective, the system will automatically sort it out. Then, all the non-defective beans will go through the second stage of the system. In the second stage, there will be an IR sensor and a weighing scale. The IR sensor will help the system to calculate for the estimated volume of the bean. The volume and mass of the bean in hand, the density of the bean can be calculated. Depending on the density threshold and size threshold set by the user, the bean will be classified whether it is good or not.

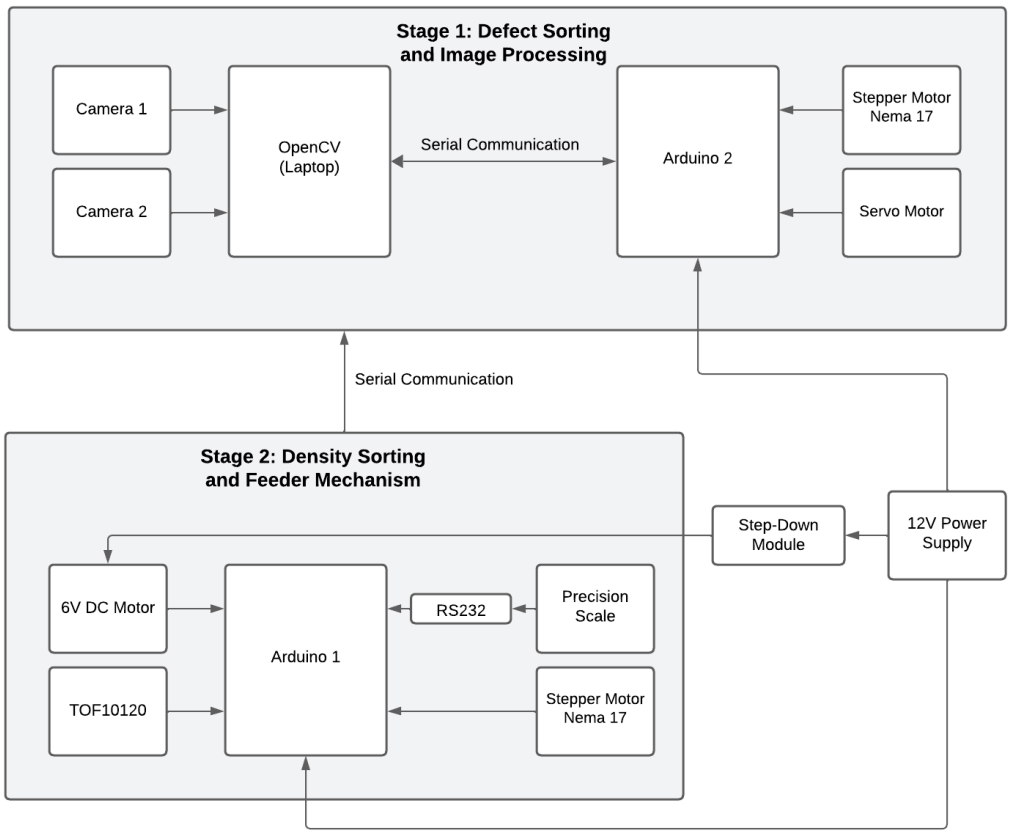



Fig. 5.2 Schematic Diagram of the System

Figure 5.2 shows the schematic diagram of the proposed system. Arduino Uno microcontroller makes all the mechanical components such as the servo motor, stepper motors, and the conveyor belt. The servo motor controls the rotating mechanism for bean sorting. On the other hand, the stepper motors operate a slide mechanism to direct the beans. Two cameras, integrated with OpenCV via Python, handle machine vision algorithms, and image processing for defect detection of the beans. A ToF10120 sensor provides precise distance measurement. A precision weighing scale measures the density of each bean for classification. The Arduino communicates with the OpenCV system through serial

communication, ensuring smooth coordination.

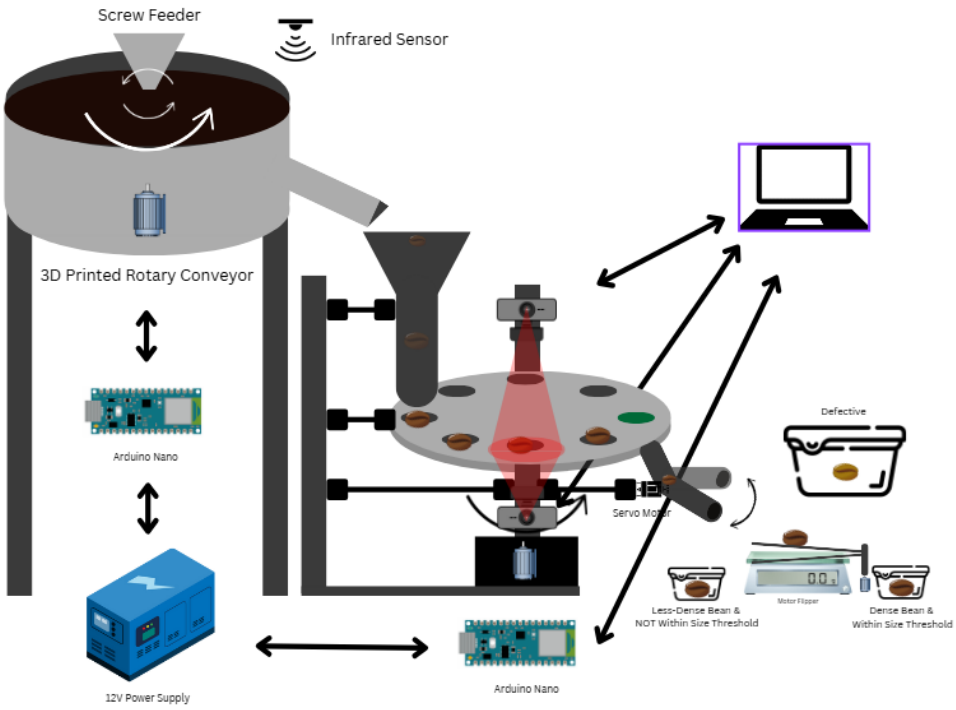



Fig. 5.3 Design Overview of the System

Figure 5.3 shows the design overview of the system. Beans are first arranged through a hopper and a conveyor belt. On top of the conveyor belt, a 3D-printed guide is attached for the beans to maintain a linear formation. Then, the beans are expected to fall into another funnel attached to a tube. The tube is directly attached to a rotating mechanism that allows the beans to be inspected and sorted one-by-one. In this stage, defective beans are sorted out. Then, the non-defective beans are transferred onto the precision scale to analyze the density. The less-dense beans are sorted out of the batch.

De La Salle University

## 5.2 Research Design

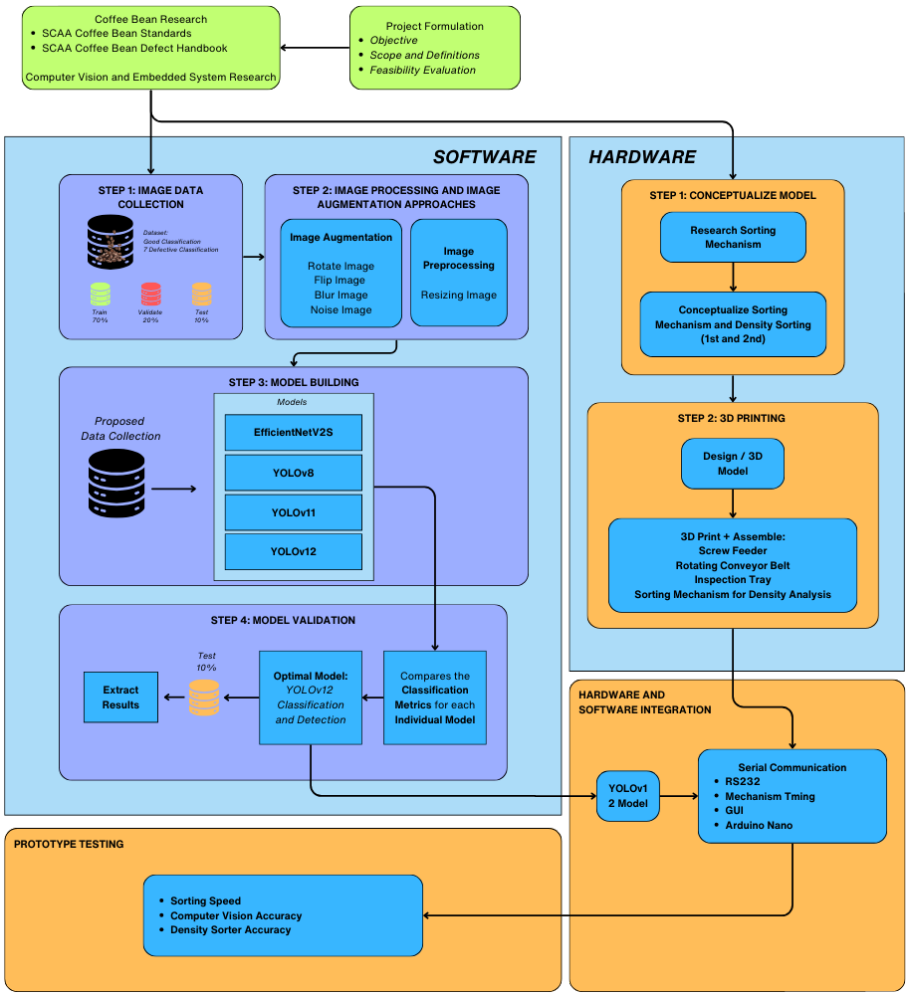



Fig. 5.4 Design and Development Research (DDR) Methodology

The researchers opted for a Design and Development Research model for the research. As shown in Figure 5.4, there are multiple levels that were needed in order to develop a working prototype for the system.

## 5.3 Dataset Collection

For dataset collection, Arabica green beans from a farm will be used. Each bean will be captured by a high-resolution camera under sufficient and consistent lighting. Proper lighting is crucial, as it directly affects the visibility of the bean's physical features, minimizing shadows, grain, and other noise that could result from inconsistent illumination. The top and bottom side pictures of the beans are to be collected. In addition, defective beans of the same type and origin will be gathered to identify the different classification of defects (primary and secondary). This study focuses on defects such as Broken, Dried Cherry, Floater, Full Black, Full Sour, Fungus Damage, and Insect Damage. The dataset will include at least 500 images of good beans and a minimum of 200 images for each defect category. To expand the dataset and enhance model training, augmentation techniques such as scaling, rotation, and mirroring will be applied.

### 5.3.1 Dataset Collection and Model Training

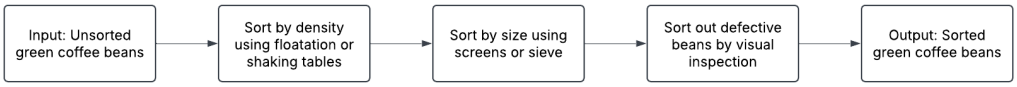



Fig. 5.5 Manual Sorting Process

The diagram in Figure 5.5 depicts the representation of the process of manual sorting of unsorted green coffee beans through a series of steps. First, the beans are sorted by density using methods such as floatation or shaking tables. This helps in separating the denser beans, usually pertaining to a more developed and higher quality bean. Then, the beans are sorted by size using screens and sieves with specific dimensions depending on the variety

of the beans. After this, a thorough visual inspection is performed by the sorters to identify and remove the defective beans from the batch. To ensure consistency and accuracy, the group follows the Specialty Coffee Association of America (SCAA) Standards Defect Handbook, which provide documentation and guidelines for identifying and classifying defective beans. Finally, the process results in the output of sorted green coffee beans, ready for further processing or sale. To ensure the dataset reflects real-world conditions, the group acquired Arabica green coffee beans from Davao. These beans were manually sorted to properly classify defective characteristics before capturing images for dataset creation. This step was crucial for improving the efficiency of batch image capture and ensuring accurate model training, making the system more applicable to Philippine coffee producers.

### 5.3.2 Utilization of Open-Source Database

To establish a foundation for the system's model, the group initially referenced an open-source dataset from Kaggle. This dataset provides an original 500x500px images of Arabica green coffee beans categorize as defective or good. This dataset also provided insights into how individual beans were captured, including factors such as lighting, camera positioning, focus, and resolution. By analyzing the dataset, the group gained a better understanding of how to achieve a high-quality data collection, ensuring that the collected dataset would contribute to high model accuracy when it is fed into the system.

### 5.3.3 First Iteration of Dataset Collection

Fig. 5.6 First Iteration of Data Collection Setup

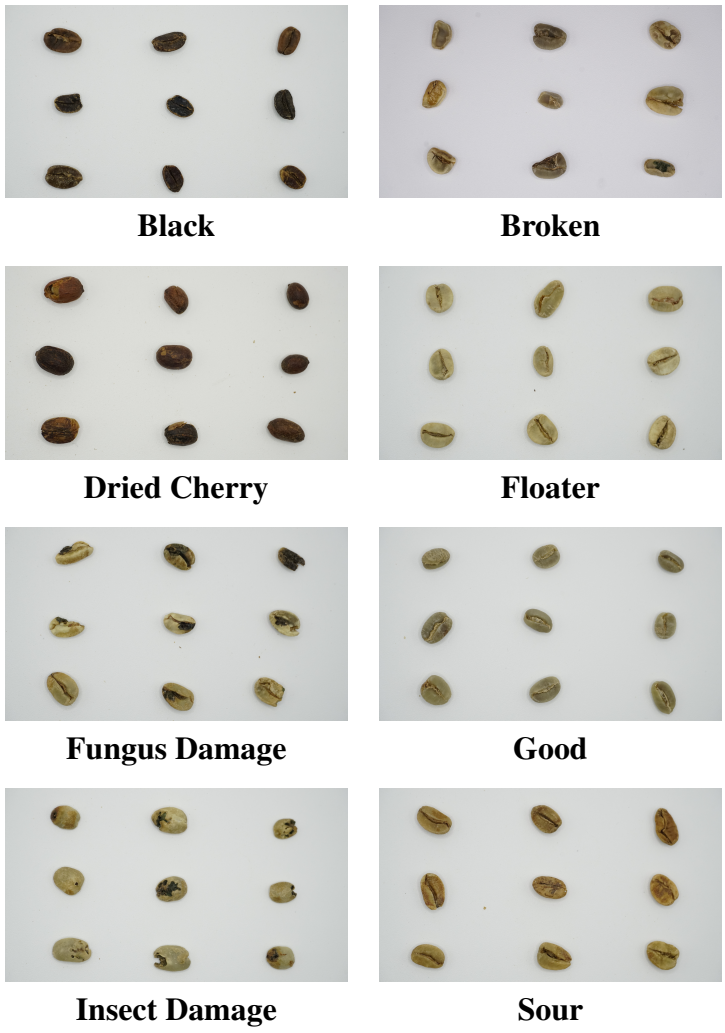



Fig. 5.7 Sample Images from the First Iteration of Dataset Collection

The first iteration of data collection utilized a Sony A6300 camera with its Kit Lens, set at 1/200 Shutter Speed, 1000 ISO, and a Distance of 50mm. The beans were captured in batches of nine, carefully arranged within the camera's field of view following the rule of thirds. The rule of thirds is a photographic composition principle where an image is divided into a 3x3 grid, creating nine equal grid lines to create balance to the photo. By aligning

the coffee beans with the rule of thirds, the group ensured a structured and even distribution of the beans within the frame. This setup also made it easier to automate the cropping process, as the predefined positions of the beans allowed a Python script to accurately extract individual images.

### 5.3.4 Second Iteration of Dataset Collection

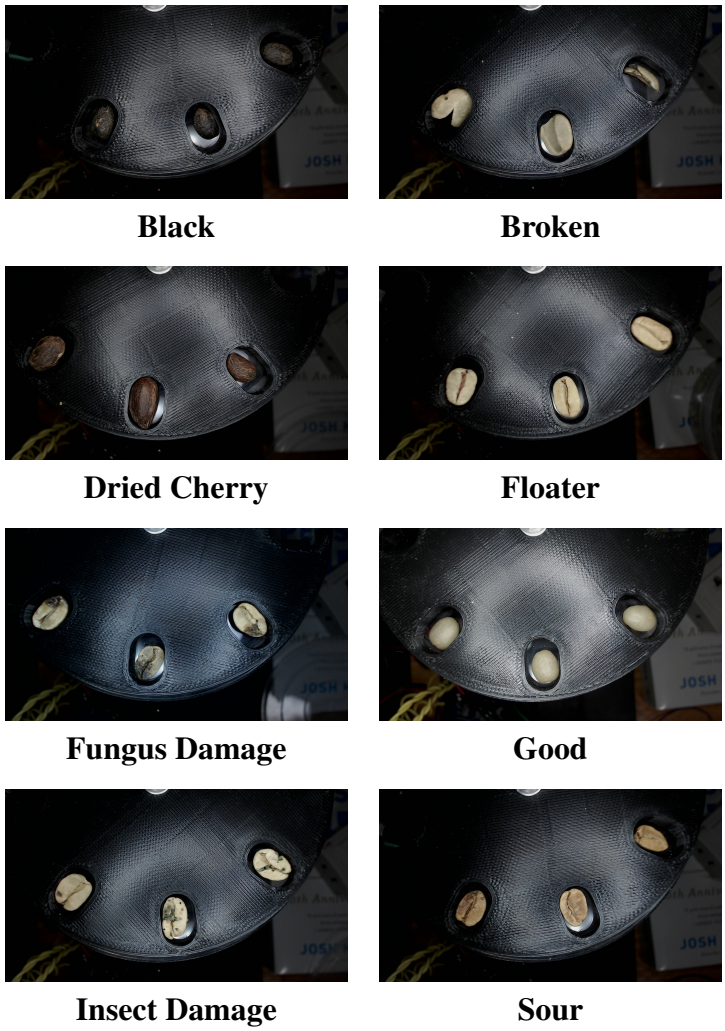



Fig. 5.8 Sample Images from the Second Iteration of Dataset Collection

The second iteration focused on real-world implementation, using the system's built-in webcam to capture images directly from the inspection tray. This setup represents the ideal condition, as it replicates the actual environment where the model will operate. The images captured in this iteration directly reflect what the system will process in a practical application, allowing for better generalization and real-time adaptability.

## 5.4 Density Threshold Calibration Using Water Displacement Method

Setting the threshold for bean density is crucial for the stage 2 sorting of the system, which involves measuring the density of each bean. In order to set a threshold for density-based classification, a calibration batch of Good quality coffee beans was chosen. The beans were confirmed to be free of defects and representative of typical specialty-grade coffee by the farmer. The threshold density was calculated by determining the average density of this batch through direct measurements of mass and volume.

The total volume of the batch of beans was measured by the water displacement technique, a commonly used method to measure the volume of solids that are irregularly shaped. The beans were fully immersed in a water-filled graduated cylinder, and the rise in water level was measured. The volume of water displaced is equivalent to the combined volume of the batch of beans, measured in cubic centimeters ($cm^3$).

The overall weight of the beans was determined by a high-precision digital scale (at least to 0.001 g resolution). Both the mass and volume are known, and the batch density may be calculated through the use of the standard formula for density:

$$\text{Batch Density} = \frac{\text{Total Mass of Beans } (g)}{\text{Total Volume Displaced } (cm^3)}$$

This computed average density served as the threshold value in the system. During automated classification, individual bean density is calculated using estimated volume (from image analysis) and actual weight (from the precision scale via RS232 communication). Beans with a density lower than the threshold are classified as less dense, while those meeting or exceeding the threshold are considered dense, indicating higher quality.

## 5.5 Dataset Preparation and Model Training

### 5.5.1 Dataset Splitting

The dataset is divided into train, validation, and test sets in a 70-20-10 ratio. The training dataset will be used for model learning, which allows it to identify patterns in the image. The validation set is used to assess the model's performance and fine-tune the parameters of the model during training. This is an iterative process wherein the model learns from the training data and is then evaluated on and fine-tuned on the validation dataset. Finally, the test set is used for evaluating the model's final performance, assessing its ability to generalize to new data.

### 5.5.2 Image Annotation

Roboflow Annotate was used to label images of coffee beans. The platform was used for two separate datasets: one for the detection model, the other for the classification model. In the detection dataset, bounding boxes were drawn around individual coffee beans and

labeled accordingly. For the classification dataset, the trained detection model was used to crop individual coffee beans from the raw dataset, which were the categorized into the eight different classifications. Roboflow was chosen for its ability to store datasets in the cloud and its support for different annotation formats, such as COCO and YOLO, ensuring compatibility with different deep learning models during experimentation.

### 5.5.3 Dataset Augmentation Techniques

Data augmentation techniques were applied using Roboflow's tools to improve the model generalization. Different augmentations such as rotation, flipping, blur, brightness and contrast adjustment, and noise were used to simulate variations, which helps prevent overfitting and improve the model's ability to identify defects in different lighting conditions and orientations.

### 5.5.4 Model Evaluation

Each trained model will be tested on the system, with a predetermined set of beans. The results from this test are analyzed by using a confusion matrix, providing a detailed breakdown of the model's performance for each category. The confusion matrix provides a way to interpret classification results by defining the following parameters:

- **True Positives (TP)** - The number of correctly classified instances for a specific defect type.
- **False Positives (FP)** - The number of times a different category was incorrectly classified as this defect type.

- **True Negatives (TN)** - All correctly classified instances excluding the defect category in question.
- **False Negatives (FN)** - The number of times this defect type was classified as something else.

Through these parameters, key performance metrics such as accuracy, precision, recall, and F1-score were computed to evaluate the system's performance in different classifications as shown below. This test will assist in determining what types of defects the system correctly classifies and which types might need improvements in image preprocessing, dataset expansion, or optimization of the machine learning model. The outcome will be applied to optimize the sorting algorithm for minimal misclassifications to ensure greater reliability in real-world defect detection.

1. **Accuracy** measures overall correctness of the classification model

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

2. **Precision** measures how many of the predicted positive classifications were actually correct

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

3. **Recall** evaluates how well the model identifies actual positive cases

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

4. **F1-score** represents the harmonic mean of precision and recall

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.4}$$

### 5.5.5 Model Benchmarking and Selection

Several models were trained and tested within the actual system to determine the most effective one. These models trained and evaluated include EfficientNetV2, YOLOv8, YOLOv11, and YOLOv12. Each model was assessed using the defined performance metrics and compared accordingly. The model with the highest overall performance will be selected for deployment in the system.

## 5.6 Hardware Development

The hardware elements of the system, two-stage automated coffee bean sorter, are developed to provide effective and precise sorting using a mix of mechanical and electronic components. Each element is designed and tested to maximize the sorting process while providing system reliability.

### 5.6.1 Screw Feeder

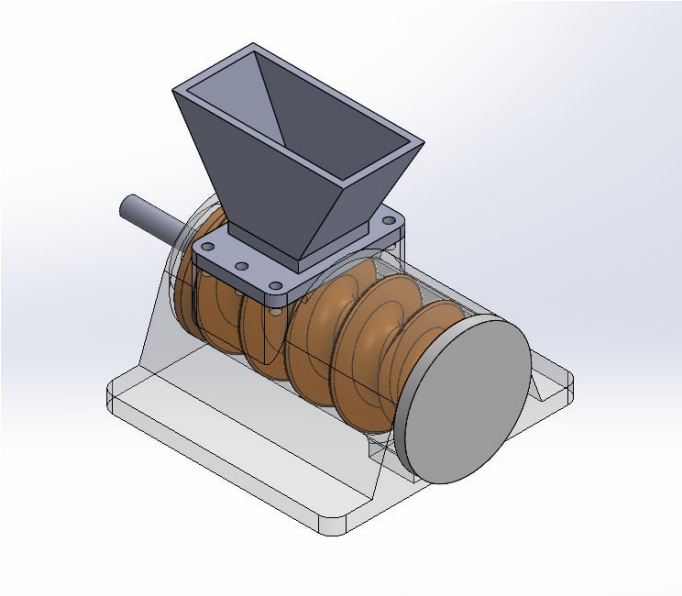


Fig. 5.9 Screw Feeder 3D Design

Screw feeder is the most essential of the devices as it governs the beans of coffee moving into the system. It operates mostly to deliver the beans consistently in terms of volume and ensures they do not bundle up and fall into the system in heavy masses, causing beans build up on the rotating conveyor table. The feeder is driven by a 12V DC motor, and the rotation speed is regulated using PWM. Through a constant and controlled flow, the screw feeder avoids clogging and provides a consistent input into the inspection tray, enhancing overall system performance. Figure 5.9 shows the actual 3D model design of the screw feeder used in the system.

### 5.6.2 Rotating Conveyor Table

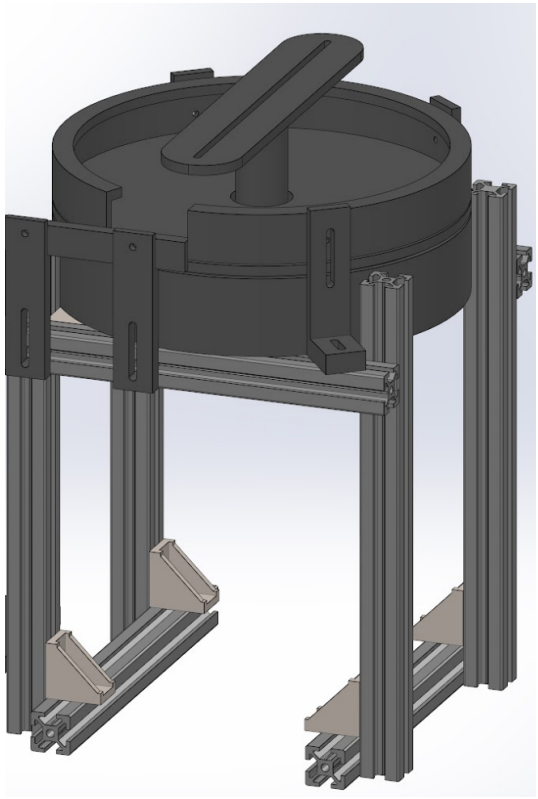


Fig. 5.10 Rotating Conveyor Table 3D Design

The conveyor table, as shown in Figure 5.10, rotates to move the coffee beans from the feeding mechanism to the inspection tray. The table contains aluminum guides to linearly arrange the beans prior to dropping on the inspection tray. The conveyor is powered by a 12V DC motor, which offers consistent movement and regulated speed to avoid misalignment. By incorporating a turning mechanism, the conveyor guarantees beans are well oriented prior to inspection tray entry, minimizing classification errors due to faulty positioning.

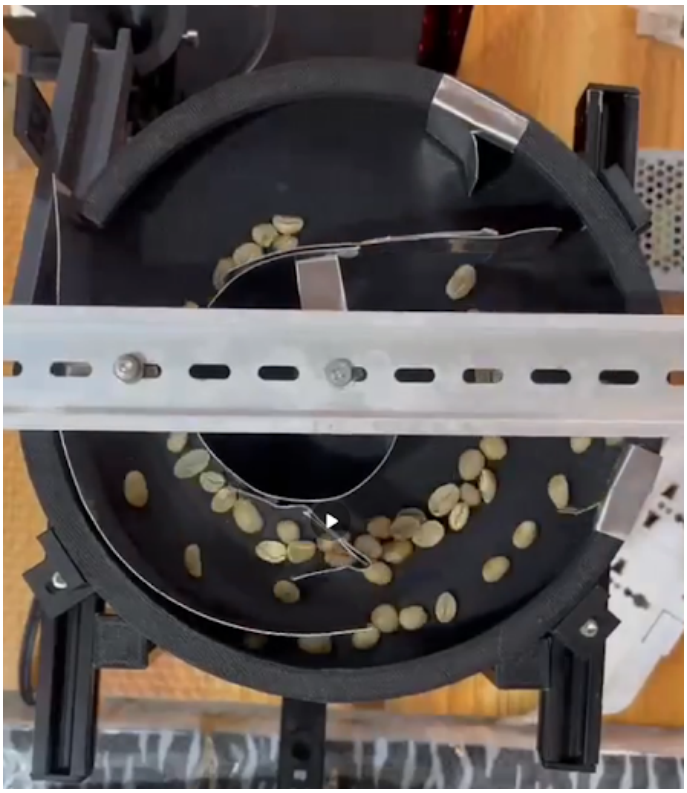


Fig. 5.11 Rotating Conveyor Table with Aluminum Guides

As shown in Figure 5.11, the installed aluminum guides on the rotating conveyor table ensures coffee beans to be linearly arranged. This linear arrangement of beans significantly helped the system to ensure that coffee beans are dropped onto the slide, which connects the conveyor table to the inspection try, in a one-by-one manner. In addition, the aluminum guides are also installed to keep the beans from accumulating in one area, which can cause the jamming of beans. The researchers tested the different motor speeds to observe the optimal settings that will not cause bean jamming and meet the minimum sorting speed of the system.

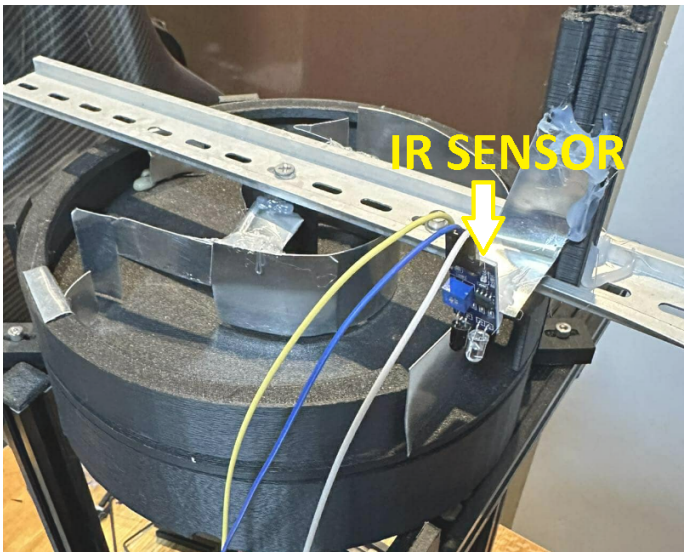



Fig. 5.12 Rotating Conveyor Table with IR Sensor

Initially, the rotating conveyor table is set at a fixed and slow speed to ensure that coffee beans are dropped into the inspection tray one-by-one. However, at this rate, the time travel time of the first bean dropped from the center of the table is very long. Thus, the group decided to add an IR sensor at the edge of the rotating table as seen in Figure 5.12. The sensor's responsibility is to detect if there is a bean at the edge. If there is no bean detected, the rotating table is set to a higher speed to expedite the process. On the other hand, if a bean is detected by the sensor, the rotation of the table is adjusted in such a way that it is able to drop the beans one-by-one onto the inspection tray. With this sensor integrated into the system, a higher speed can be set for the rotating table, minimizing the time travel of the beans from the center to the inspection tray, resulting to a faster sorting time for the first stage.

### 5.6.3 Inspection Tray

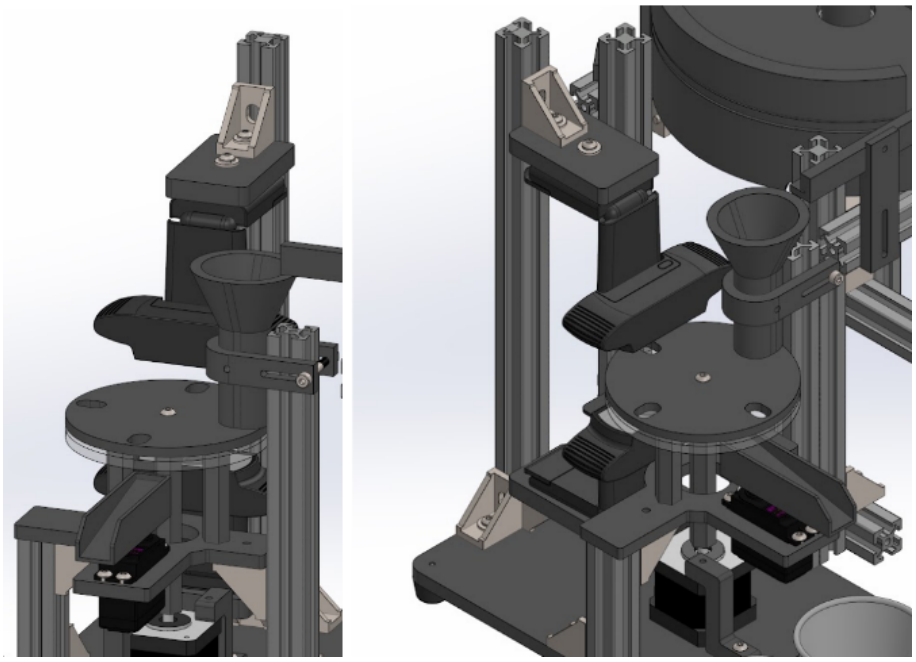


Fig. 5.13 Inspection Tray 3D Design

The inspection tray is the main component for the first-stage sorting mechanism. The inspection tray is used to support beans in a stable and constrained position for a short time, enabling the camera to take high-resolution images without motion blur. The NEMA 17 stepper motor drives the movement of the inspection tray, enabling accurate alignment with the vision system's image processing pipeline. The tray surface is created to reduce reflections and enhance contrast so that the camera can precisely detect defects like cracks, discoloration, or insect infestation. In addition, the surface is made of clear acrylic to allow a clear image for the camera positioned at the bottom of the tray. Lastly, a rotatable slider controlled by a 5V servo motor serves as the main segregator of the good beans from the defective beans.

### 5.6.4 Density Sorter

The density sorter is the second-stage sorting system, tasked with sorting coffee beans according to their measured density. This is achieved by initially measuring each bean's mass using a precision weighing scale and volume using the ToF10120 infrared sensor. After calculating the density, the system triggers a sorting system powered by a NEMA 17 stepper motor, which sorts beans into various collection bins according to their classification. This sorting operation is such that high-density, specialty-grade beans are kept separate from low-density, commercial-grade or defective beans. The density sorter's accuracy is verified by comparing the results of its classification to manual weighing measurements (ground truth data).

Fig. 5.14 Precision Scale

The U.S. Solid Electronic Precision Balance (0.01g, 1200g capacity, RS232 port, AC/DC power) was selected for the density sorting mechanism because it is highly accurate, transmits data in real-time, and is well-calibrated. Its 0.01g precision guarantees accurate mass readings, which are critical to precise density calculations in sorting coffee beans. The RS232 port facilitates smooth integration with the microcontroller for automatic data

processing and sorting decisions, minimizing manual errors. Its dual power source (AC and battery) also guarantees uninterrupted operation in different environments, making it a dependable and efficient part of the coffee bean sorting system.

## 5.7 Hardware and Software Integration

### 5.7.1 Serial Communication

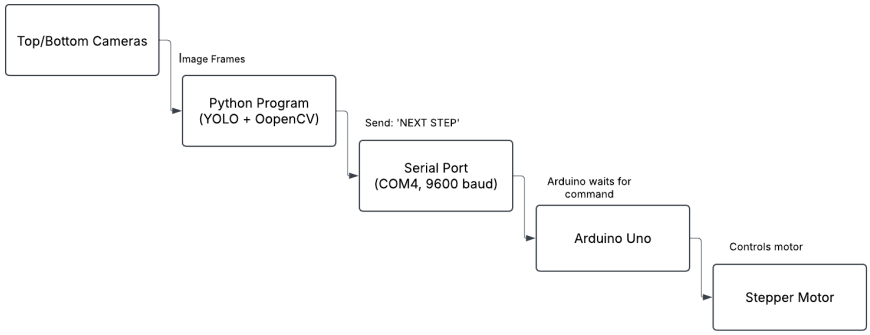



Fig. 5.15 Serial Communication Flow for Stage 1 Classification

The system is generally composed of hardware and software components. Hardware components are mainly responsible for collecting data from the coffee beans such as the camera and IR sensor, and the sorting mechanisms such as servo motors and stepper motors. On the other hand, the software components are the brain of the system which is mainly responsible for data processing such as image detection, defect classification of the beans, volume and density computation, and control of the mechanisms. Since the system has two major components, software and hardware, they should be integrated together for the system to be as effective. Thus, serial communication was utilized to integrate the hardware and software components of the system. Serial communication is a significant

component in the system as it serves as the communication medium of the hardware and software. It enables real-time coordination between the software (YOLO-based image detection, classification, and density computation) and the hardware (running in Arduino microcontrollers). The said communication is established with the use of a USB serial interface using the pyserial library in Python. In addition, this is configured at a baud rate of 9600.

### 5.7.2 Recommended Standard 232 (RS-232)

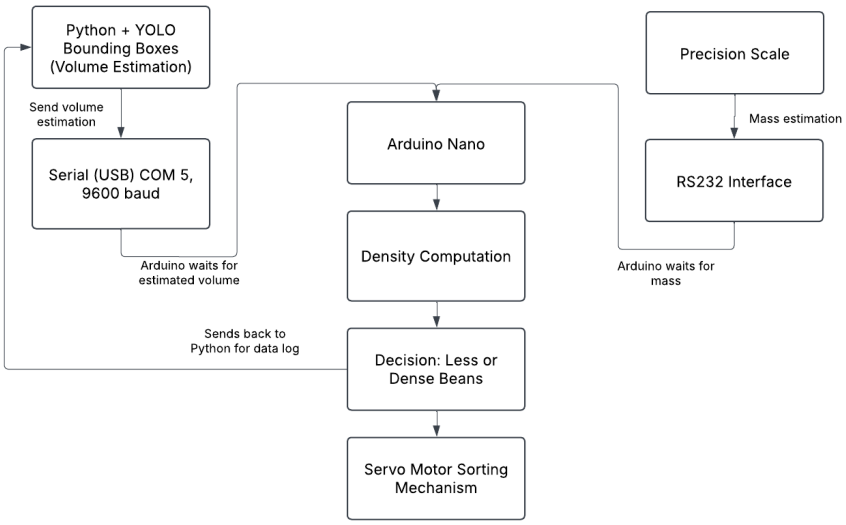



Fig. 5.16 Precision Scale Integration with RS232 for Stage 2 Classification

The stage 2 classification is mainly composed of the sorting mechanism itself, and the precision scale to measure the mass of each bean. The bounding boxes from the stage 1 classification are used to estimate each bean's volume. Additionally, the beans depth is also estimated through the IR sensor placed in the rotating conveyor table. With these measurements, the volume of each bean, the volume can be calculated using the Tri-axial

Ellipsoid's volume formula. The system, specifically at the inspection tray mechanism where the YOLO detection and classification is implemented, has a function move_stepper() responsible for sending the command from the Python code to the Arduino microcontroller. When the Arduino receive this command, it executes motor movement that allows the stepper motor to move at a certain angle that allows the camera to capture the bean. This function is crucial for the system as this is how each bean in the inspection tray is fed to the image processing side of the system. This movement rotates the mechanism holding the coffee beans, positioning the next bean beneath the top and bottom cameras for inspection. After the motor completes the movement, the Arduino will send back a message to the program running Python, signalling that the bean is ready for image capture and further processing. In addition, the Python script is continuously or constantly waiting for the Arduino's message through the arduino.readline() function, ensuring seamless communication and faster processing.

## 5.8 Prototype Setup

### 5.8.1 Actual Setup

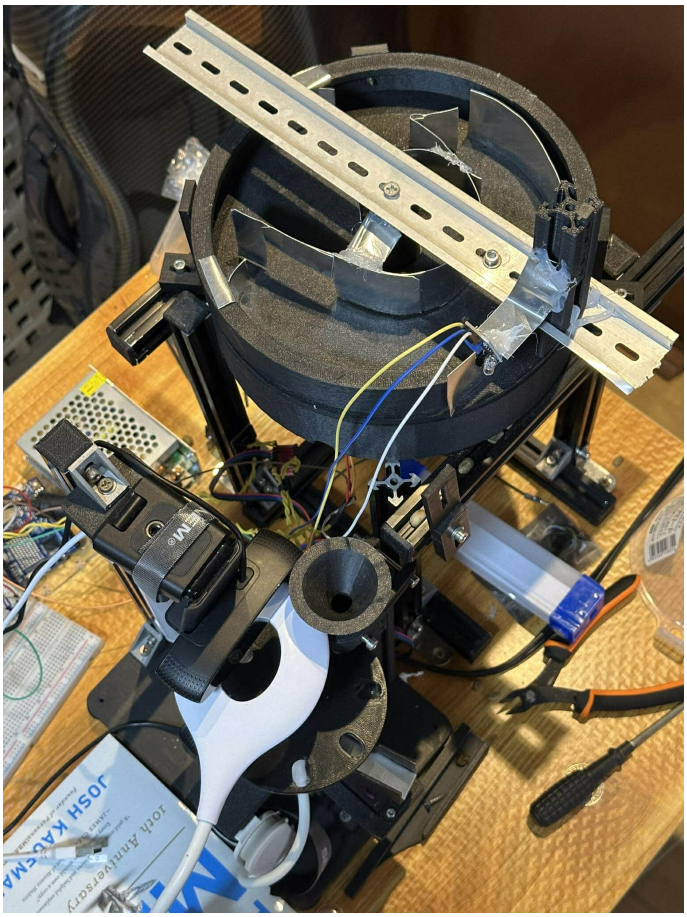


Fig. 5.17 Actual System Setup

Physical integration of the automatic coffee bean sorter system comprises various integrated parts with the purpose of enabling effective, accurate, and methodical sorting in terms of visual defects as well as density categorization. The system involves integration of mechanical, electronic, and computer vision technologies for optimizing sorting. To

begin the process, coffee beans are added to a revolving conveyor table, which is the main mechanism of transport used for feeding the beans into the inspection system. The conveyor features aluminum guides positioned strategically along it to ensure linear alignment of the beans as they travel. Linear alignment is required to avoid overlap and misclassification, since individual processing by the machine vision system is necessary for each bean.Once the beans travel further along the conveyor, they are conveyed onto the inspection tray. There, they are viewed in multiple perspectives by two high-definition cameras. A two-camera imaging process ensures improved defect detection by providing a full, thorough evaluation of the surface, shape, and texture of the bean. The images are then processed with a deep learning-based classification algorithm that classifies each bean as either defective or good according to predefined defect types like black beans, dried cherries, fungus damage, insect damage, sour beans, floaters, and broken beans.

After classification, the system triggers the defect sorting mechanism, which physically takes out defective beans from the processing line. The mechanism includes a servo motor-powered sorting slide, which diverts defective beans into a distinct collection bin. Good beans that are classified are taken to the second level of sorting, which is density-based classification. At the density-based sorting level, good beans are weighed individually with a high-precision electronic balance. The U.S. Solid Electronic Precision Balance (0.01g, RS232) is embedded within the system to accurately weigh the mass of each bean. A Time-of-Flight (ToF) sensor also estimates the volume of each bean, permitting the calculation of the density of beans. According to the calculation of density, beans are automatically sorted into corresponding collection bins using a second sorting mechanism regulated by a NEMA 17 stepper motor.

### 5.8.2 Lighting Setup for Inspection Tray

Lighting has a key importance in the image-based detection and classification system, specifically for the inspection tray. For the model to be more accurate and precise in classifying good and defective beans, correct lighting is important such that details like surface texture, color difference, and defects are properly rendered by the imaging system. Asymmetrical, unsteady, or low-quality lighting can create shadows, reflections, or over-exposure, all of which lower the quality of input images and thus decrease the accuracy of object detection and classification models like YOLO. To improve the consistency and definition of images taken during inspection, the lighting arrangement above the inspection tray was refined incrementally throughout development. The refinements were intended to maximize the illumination conditions for both the top and bottom camera modules.

Fig. 5.18 First Iteration of Lighting Setup

Figure 5.18 shows the initial lighting setup that the researchers implemented on the system. The initial lighting arrangement was based on a single top-mounted LED lighting. Although the arrangement was more than bright enough for the top camera, it introduced random shadows and highlights onto the bottom camera. As a result, only one side of the bean is accurately inspected. These random elements impacted the model's performance in detecting bean contours and separating surface flaws, particularly for dark beans or reflective-surface beans.

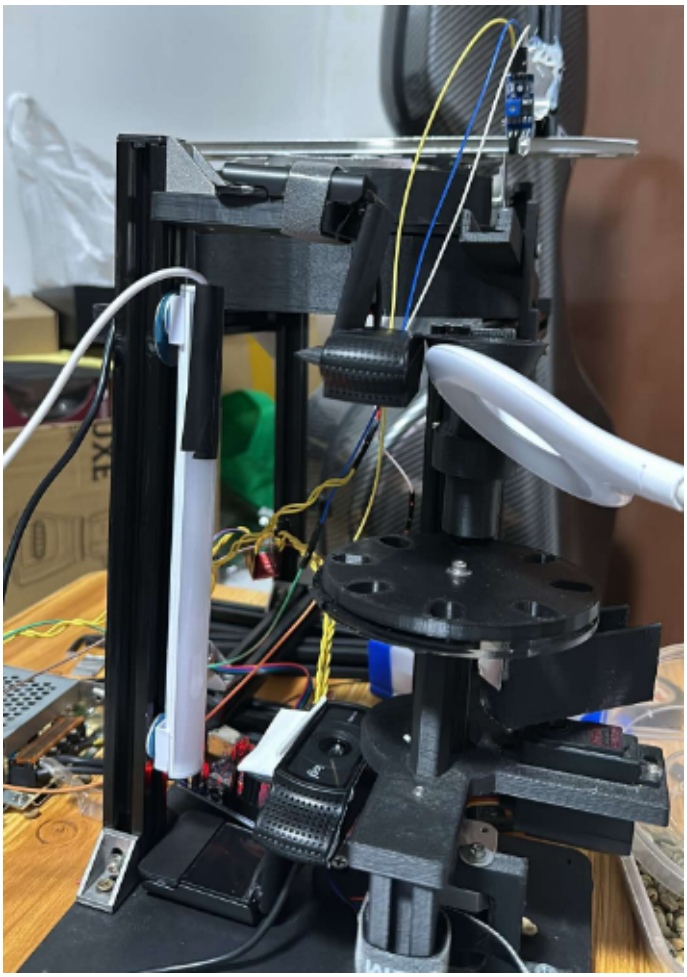


Fig. 5.19 Second Iteration of Lighting Setup

For the second iteration of the lighting setup, the researchers decided to add another LED strip lighting at the side of the inspection tray, while keeping the LED lighting mounted at the top. This provided good lighting for both top and bottom cameras. However, the view of the bottom camera is still a bit dark.

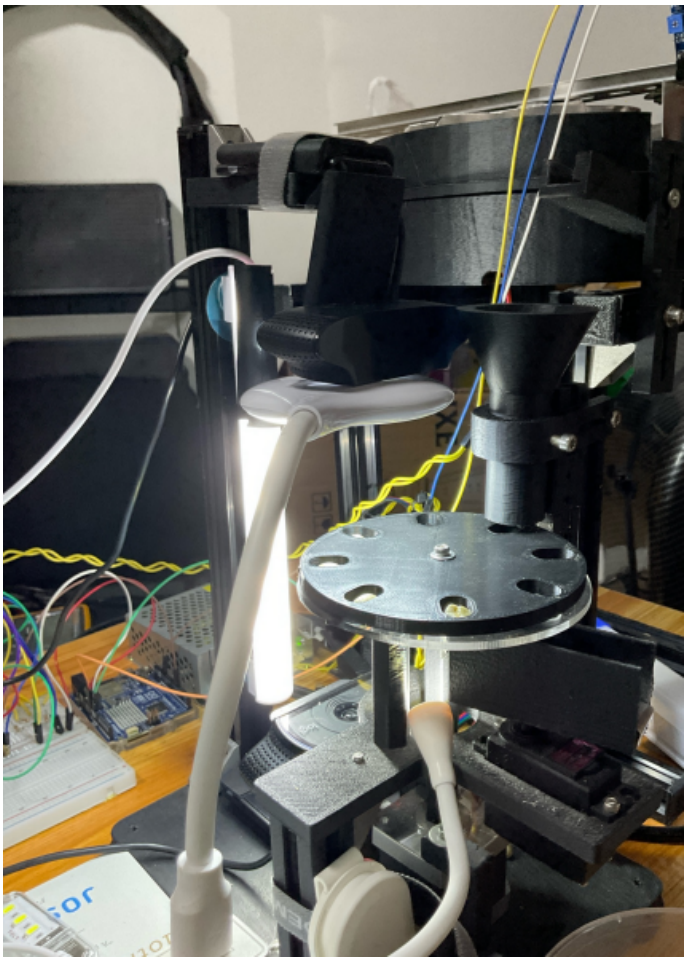


Fig. 5.20 Final Iteration of Lighting Setup

To ensure that both camera views have sufficient lighting and avoid shadows, the researchers decided to use a total of three LED lights. One is a small ring light placed exactly above the inspection tray. Another LED light is a stip light placed at the side of the inspection tray to improve lighting at the side of each bean. Lastly, another small LED light is placed under the inspection tray to ensure that the bottom camera has enough lighting.

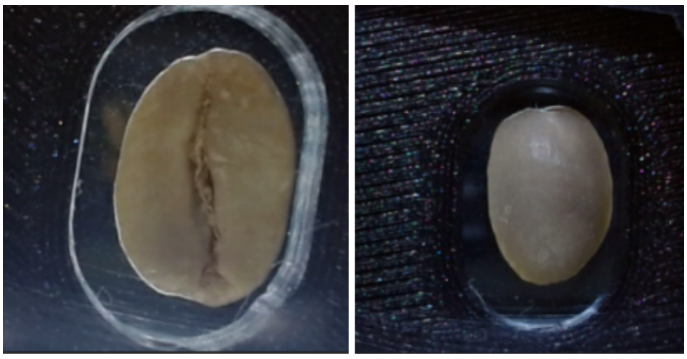


Fig. 5.21 Top and Bottom View of the Cameras

### 5.8.3 System Operation

The system operation follows a sequential process to ensure the effective sorting of green coffee beans (GCBs) based on its classification and density. The automated system consists of two primary stages: 1st Stage which is the machine vision-based classification and 2nd stage which is the density-based sorting.

The process beings in the inputting of unsorted GCBs (Contains good and defective beans) into the screw feeder, which regulates the controlled and consistent delivery of the beans into the rotary conveyor table. The conveyor table is designed with aluminum guides to ensure a linearized formation of the beans to mitigate jamming. This also ensures a controlled movement of beans, ensuring that they drop onto the inspection tray one at a time. As the bean goes towards the edge of the conveyor table, the IR sensors detects the beans and stops the rotation to ensure the one-by-one inspection of the beans, this also prevents clogging, and jamming once the beans are dropped into the inspection tray.

The first phase involves machine-vision classification. Once the GCBs reach the inspection tray, each bean is analyzed one-by-one using a machine vision system consisting

of top and bottom cameras. The system captures high-resolution images of the bean and processes the data to determine which classification it belongs. If the bean is identified as defective, a signal is sent to the servo motor, which redirects the bean into the defective bin for disposal, if the bean is classified as good, it then proceeds to the second phase of the system

The second stage involves density-based sorting, where each GCB's weight is measured using a precision scale, while its volume is determined by the ToF10120 infrared sensor. The system then calculates the density and classifies the bean accordingly.

The sorting mechanism activates, directing beans into designated collection bins based on their density. High-density beans, often associated with specialty-grade quality, are separated from low-density, commercial-grade, or defective beans.

## 5.9 Prototype Testing

### 5.9.1 Sorting Speed

TABLE 5.2 SORTING SPEED TESTING TABLE

| Test Condition | Conveyor Table Speed (RPM) | Inspection Tray Speed (RPM) | Sorting Speed (Beans per Minute) |
|---|---|---|---|
| 100% Good Beans | | | |
| 80% Good, 20% Defective Beans | | | |
| 70% Good, 30% Defective Beans | | | |
| 50% Good, 50% Defective Beans | | | |
| 100% Defective Beans | | | |

The sorting speed of the system will be determined by conducting at least five trials. Each trial will be exactly conducted for one minute. The number of beans sorted out within the time frame are considered as the sorting speed in beans per minute. Then, the average sorting speed from the five trials is computed. In each trial session, controlled variables such as motor speed of the inspection tray and rotating conveyor table are varied to observe the optimal setting for the system, ensuring that there are no beans jamming in the tray and fast enough to meet the minimum sorting speed. Table 6.6 shows the different conditions for each trial to ensure that the sorting speed across different type of beans are considered.

### 5.9.2 Defect Sorting Accuracy

TABLE 5.3 GOOD BEAN CLASSIFICATION ACCURACY TESTING TABLE

| Test Condition | Correctly Classified Beans | Misclassified Beans | Total Number of Beans |
|---|---|---|---|
| 100% Good Beans | | | 100 |
| 80% Good, 20% Defective Beans | | | 100 |
| 70% Good, 30% Defective Beans | | | 100 |
| 50% Good, 50% Defective Beans | | | 100 |
| 100% Defective Beans | | | 100 |

The defect sorting accuracy by feeding 100 beans on each trial. For testing its accuracy for detecting good beans and defective beans, five trials are conducted containing 100 beans of good beans for the first trial, 80 good and 20 defects for the second trial, 50 good and 50 defects for the third trial, 20 good and 80 defects for the fourth trial, and 100 defects for the last trial. With these, the number of correctly classified and misclassified beans are logged into the system to compute for accuracy using the formula:

$$\text{Accuracy}(\%) = \left(\frac{\text{Correctly Classified Beans}}{\text{Total Beans Tested}}\right) \times 100 \tag{5.5}$$

TABLE 5.4 SPECIFIC DEFECT CLASSIFICATION ACCURACY TESTING TABLE

| **Test Condition** | **Correctly Classified Beans** | **Misclassified Beans** | **Total Number of Beans** |
|---|---|---|---|
| 100% Good Beans | | | 100 |
| 80% Good, 20% Defective Beans | | | 100 |
| 70% Good, 30% Defective Beans | | | 100 |
| 50% Good, 50% Defective Beans | | | 100 |
| 100% Defective Beans | | | 100 |

For further accuracy testing of the computer vision model in actual implementation, the researchers also included testing trials for each defect type. Table 5.4 shows how each trial is conducted. For example, the defect type chosen for the test is the Sour defect type. The first trial contains 100 sour beans. For the second trial, 80 sour beans and 20 randomly selected beans, excluding the chosen defect type which is sour. Thus, the random beans are always the other classes except the chosen defect type to be tested. In this test, correctly classified beans and misclassified beans are also considered to compute for the accuracy of the system. By testing the system under different defect distributions, the robustness of the machine vision model can be assessed.

TABLE 5.5 DATASET DISTRIBUTION FOR OVERALL TESTING

| **Bean Classification** | **Bean Count** |
|---|---|
| Black | 20 |
| Broken | 20 |
| Dried Cherry | 20 |
| Floater | 20 |
| Fungus Damage | 20 |
| Good | 20 |
| Insect Damage | 20 |
| Sour | 20 |
| **Total Beans** | **160** |

Lastly, to assess the overall accuracy and reliability of the first stage, machine vision-based defect classification, a trial consisting of a predefined dataset of 160 coffee beans was conducted. Each category consists of 20 beans as shown in Table 5.5, including good beans and the other defect types such as black, dried cherry, fungus, insect damage, sour, floater, and broken beans.

### 5.9.3 Density Sorting Accuracy

To assess the accuracy of the mechanism, it will rely on measuring the accuracy and the reliability of the density sorting mechanism in sorting out the dense beans to the less dense beans. To successfully determine the accuracy of the system, the basis will be the scale, where the system should be able to sort the dense beans to the less dense bean in relation to the detected weight in the scale. A successful system should be able to sort with an accuracy of 85

# Chapter 6

# RESULTS AND DISCUSSIONS

TABLE 6.1 SUMMARY OF RESULTS FOR ACHIEVING THE OBJECTIVES

| Objectives | Results | Locations |
|---|---|---|
| GO: To develop an automated (Arabica) green coffee bean sorter that identifies good, less-dense and defective beans from an unsorted batch of coffee beans. The system will utilize machine vision and density-based analysis for defect detection and classification of the coffee beans, ensuring efficient coffee bean sorting. | • Achieved to gather and create a unique dataset consisting of 500 good and 200 defective beans<br>• Achieved improvisation of the synchronization between the machine vision and embedded system. | Sec. 6.1 on p. 95 |
| SO1: To gather and create a dataset consisting of 500 high-resolution images of good Arabica green coffee beans and 200 high-resolution images per classification of defective beans (Category 1 & Category 2). | • Acquired 257 images of Black coffee beans<br>• Gathered 301 images of Broken coffee beans<br>• Gathered 305 images of Dried Cherry coffee beans<br>• Acquired 288 images of Floater coffee beans<br>• Acquired 301 images of Fungus Damage coffee beans<br>• Gathered 1565 images of Good coffee beans<br>• Acquired 345 images of Insect Damage coffee beans<br>• Gathered 320 images of Sour coffee beans | Sec. 6.1 on p. 95 |

*Continued on next page*

*Continued from previous page*

| Objectives | Results | Locations |
|---|---|---|
| SO2: To improve the synchronization between the machine vision system and the embedded sorting mechanism, ensuring defect sorting of at least 20 beans per minute for stage one, solving issues such as non-synchronization of the system. | • Achieved 22 beans per minute for stage one of the system | Sec. 6.4 on p. 107 |

*Continued on next page*

*Continued from previous page*

| Objectives | Results | Locations |
|---|---|---|
| SO3: To achieve an accuracy of at least 85% in classifying defective green coffee beans using computer vision | • Achieved 90.07% testing accuracy in classifying Black coffee beans.<br>• Achieved 90.07% testing accuracy in identifying Broken coffee beans.<br>• Attained 90.65% testing accuracy in recognizing Dried Cherry coffee beans.<br>• Recorded 87.78% testing accuracy in detecting Floater coffee beans.<br>• Achieved 90.65% testing accuracy in classifying Fungus Damage coffee beans.<br>• Reached 90.07% testing accuracy in identifying Good coffee beans.<br>• Attained 90.07% testing accuracy in detecting Insect Damage coffee beans.<br>• Achieved 90.65% testing accuracy in classifying Sour coffee beans.<br>• Achieved 90.00% overall testing accuracy of the system. | Sec. 6.3 on p. 104 |
| SO4: To achieve an accuracy of at least 85% in filtering out less-dense green coffee beans | • To achieve 90% in filtering out less-dense coffee beans | |

## 6.1 Description of the New Custom Dataset

TABLE 6.2 CLASS DISTRIBUTION SUMMARY

| Class Name | Image Count |
|---|---|
| Black | 205 |
| Broken | 203 |
| Dried Cherry | 206 |
| Floater | 202 |
| Fungus Damage | 207 |
| Good | 604 |
| Insect Damage | 201 |
| Sour | 202 |
| **Total** | **2030** |

Table 6.2 presents the dataset's class distribution after adjustments. The image counts for each category were increased such that the minimum is above 200, with "Good" exceeding 543; for instance, Black has 205 images and Good has 604 images. The table confirms a total of 2,030 images distributed across the eight classes, ensuring a balanced dataset that maintains diversity while meeting the minimum requirements.

TABLE 6.3 DATASET SPLIT SUMMARY

| Split | Percentage | Image Count | Augmentation |
|---|---|---|---|
| Train | 88% | 1786 | Original training images are augmented three times |
| Validation | 8% | 162 | Non-augmented |
| Test | 4% | 82 | Non-augmented |

Table 6.3 outlines the dataset split into training, validation, and test sets. The training set comprises 88% (1,786 images), while the validation and test sets account for 8% (162

images) and 4% (82 images) respectively, with the training images later augmented 3× per image.

## 6.2 Performance of Classification Models on Custom Dataset

Four different classification models, such as EfficientNet, YOLOv8, YOLOv11 and YOLOv12, were benchmarked to determine the most optimal model to be used for the system. Each model was trained using a custom dataset manually gathered by the researchers. In addition, augmentations such as rotation, flip, blur and noise, were applied.

### 6.2.1 EfficientNetV2S

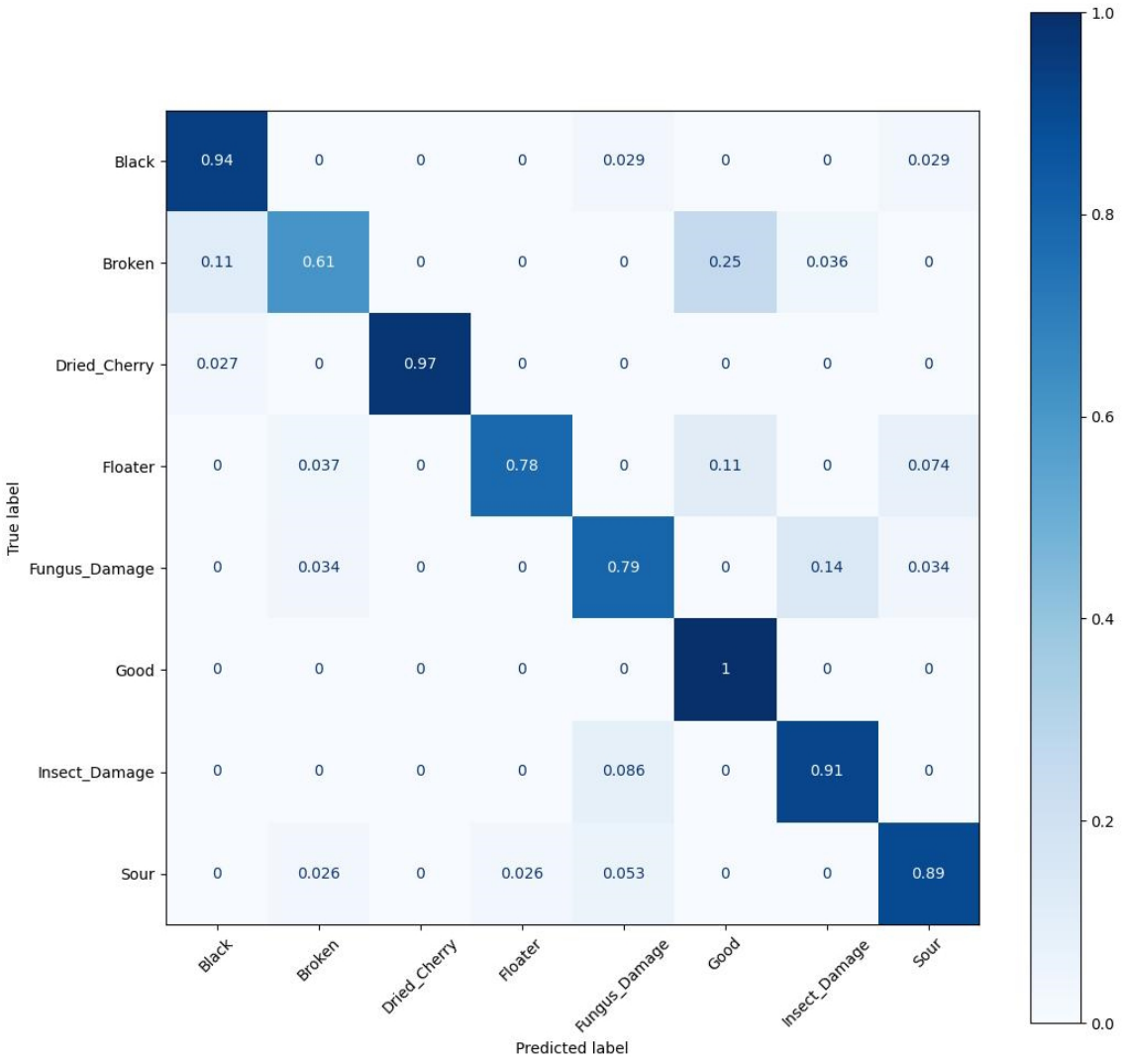



Fig. 6.1 Normalized Confusion Matrix for EfficientNetV2S on Test Dataset

The confusion matrix depicted in Figure 6.1 shows how the EfficientNetV2 classification model performed against the validation dataset, where normalized values are used to represent percentage predictions by each class. The matrix is seen to indicate that even

though EfficientNet was able to classify the Good bean class perfectly (1.00) and accurately for classes like Dried Cherry (0.97) and Black (0.94), its classification was poor for many defect classes. In particular, the model exhibited significant misclassification in the Broken bean class, with just 61% correctly classified, while a significant 25% were misclassified as Good. Likewise, for Floater and Fungus Damage, EfficientNetV2 had true positive rates of only 0.78 and 0.79, respectively, with some floaters being mistaken as Fungus Damage (11%) and Sour (7.4%). This trend indicates that EfficientNet found it difficult to identify subtle visual variations between defect types, particularly when texture or color change overlapped among classes.

### 6.2.2 YOLOv8

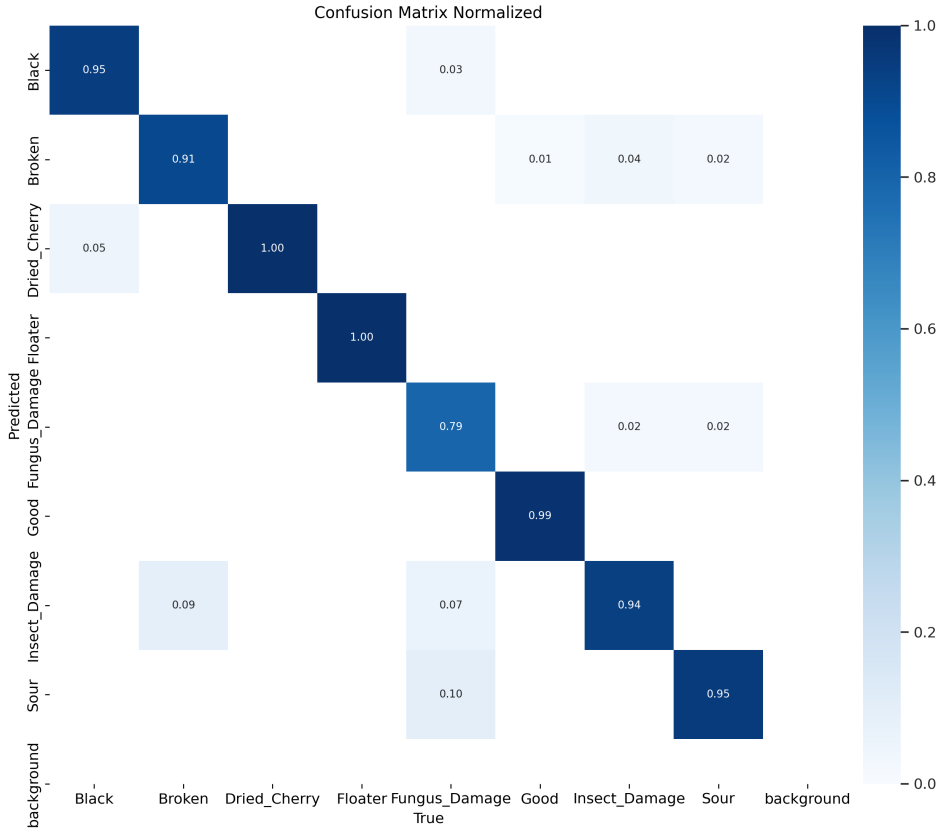



Fig. 6.2 Normalized Confusion Matrix for YOLOv8 on Test Dataset

The YOLOv8 confusion matrix shows excellent classification accuracy in the majority of defect classes, with exceptionally good performance in separating Dried Cherry, Floater, and Good beans, each of which had a perfect or near-perfect true positive rate (TPr) of 1.00, 1.00, and 0.99, respectively. The model also correctly classified Black beans at 0.95, reflecting excellent robustness in detecting strongly distinguishable visual features.However, there was some confusion between visually cimilar classes, like Fungus Damage, which had a true positive rate of 0.79. Misclassifications for the category were distributed between

Insect Damage and Sour beans, at 2% each, which would suggest some overlap in texture or color patterns that the model found difficulty in distinguishing. However, there was a lesser, but still significant confusion between Sour and Fungus Damage, where Sour beans were misclassified at 0.10 within other classes. The Insect Damage class performed well at 0.94, though there was some confusion (6%) with Fungus Damage. Broken beans reached 0.91, with small misclassifications into Dried Cherry and others. Most importantly, there was no confusion with the Background class, indicating YOLOv8's excellent capability of isolating and detecting bean contours well. In general, YOLOv8 provides balanced performance, with satisfactory overall accuracy across different classes.

### 6.2.3 YOLOv11-cls

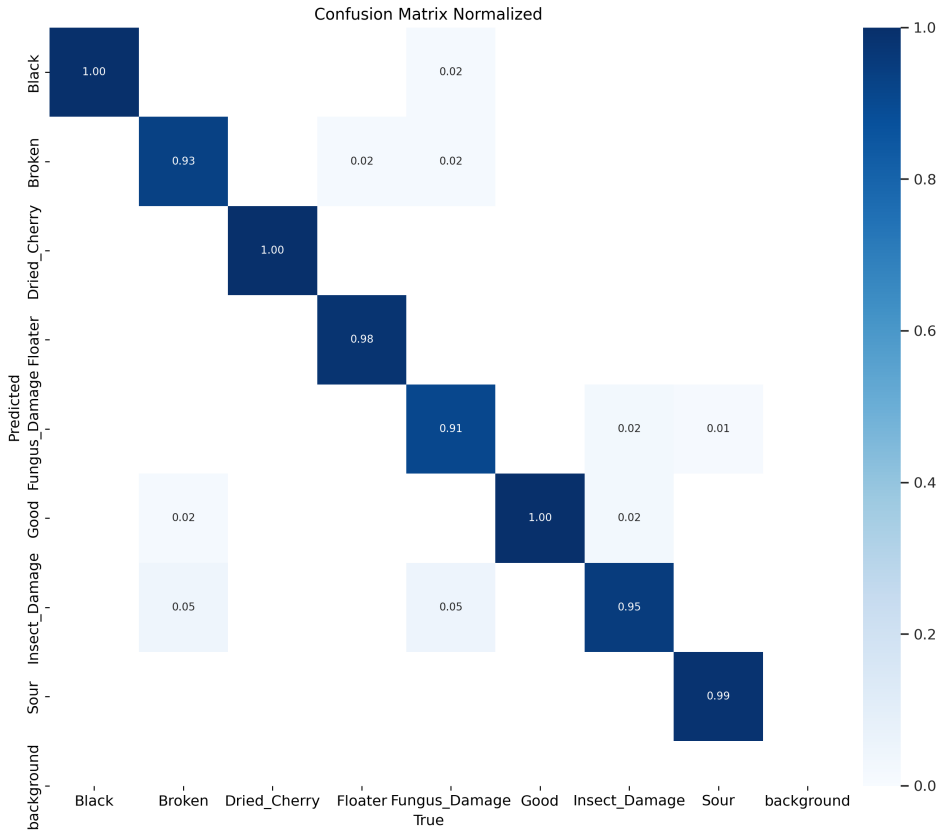



Fig. 6.3 Normalized Confusion Matrix for YOLOv11 on Test Dataset

The YOLOv11 confusion matrix shows significant gains in classification consistency, especially in visually different categories. The model obtained ideal classification (1.00) for both Good beans and Floater, which means a high capability to identify well-defined, good beans and floating defects. Likewise, excellent true positive rates were achieved for Black (0.97), Dried Cherry (0.97), and Broken (0.94) beans with limited confusion (at most 3%) with adjacent defect classes, showing the robustness of YOLOv11 in detecting salient visual features. More complex defects, YOLOv11 achieved a true positive of 0.90 for Fungus

Damage, though misclassification did occur into Sour beans (7%) and Insect Damage (2%), which points to some confusion between defects that have comparable texture degradation. The Insect Damage class achieved a strong 0.92, but was at times confused with Black and Fungus Damage, both by 3%. The performance of the model slightly declined in the Sour bean class, which exhibited the lowest true positive rate of 0.89, with significant misclassifications to Fungus Damage (7%), indicative of visual discoloration or wrinkling overlap. In general, YOLOv11 shows a good balance in performance, being excellent in clean categories and keeping stable results for complicated defect types. Its high precision with low false positives on most classes indicate its potential in real-time defect detection applications, with room for improvement through additional dataset augmentation for biologically deteriorated beans.

### 6.2.4 YOLOv12-cls


Fig. 6.4 Normalized Confusion Matrix for YOLOv12 on Test Dataset

The YOLOv12 performance, as reflected in the normalized confusion matrix, presents good classification performance for most defect classes. Most importantly, Sour beans and Good beans were classified with a true positive rate of 0.99, and Dried Cherry and Black beans followed closely with 0.99 and 0.96, respectively. This implies excellent sensitivity of the model to clearly distinguishable visual features, particularly those with color homogeneity and texture contrast. However, some defect types caused classification difficulties. Broken beans had the worst classification accuracy of 0.80, with high misclassifications spread

over other classes like Dried Cherry, Floater, and Insect Damage, each contributing 1–2% to the confusion. Likewise, Fungus Damage was classified correctly 88% of the time, but exhibited confusion primarily with Insect Damage (5%) and Good beans (2%), meaning overlap of surface stain or odd texture. The Floater class was highly accurate at 0.97 and had little confusion. Insect Damage, despite maintaining a consistency of 0.92, had some misclassifications as Fungus Damage (10%). Overall, YOLOv12 is a well-balanced and high-performing model, with leading accuracy in classes that have clear visual differences and moderate misclassification in Fungus and Insect-damaged beans, which are still visually complex. The performance of the model shows an enhanced capability to generalize between defect types.

## 6.3 Actual Performance of Trained Models in the System

TABLE 6.4 SPECIFIC PERFORMANCE OF THE MODELS FOR EACH DEFECT

| Model | Defect | TP | TN | FP | FN | Prec. | Rec. | F1 | Acc. |
|---|---|---|---|---|---|---|---|---|---|
| EffNetV2 | Black | 15 | 134 | 6 | 5 | 71.4 | 75.0 | 73.2 | 81.27 |
| YOLOv8 | Black | 16 | 135 | 5 | 4 | 76.2 | 80.0 | 78.0 | 85.67 |
| YOLOv11 | Black | 17 | 137 | 3 | 3 | 85.0 | 85.0 | 85.0 | 88.87 |
| YOLOv12 | Black | 18 | 139 | 1 | 2 | 94.7 | 90.0 | 92.3 | 90.07 |
| EffNetV2 | Broken | 15 | 134 | 6 | 5 | 71.4 | 75.0 | 73.2 | 81.27 |
| YOLOv8 | Broken | 16 | 135 | 5 | 4 | 76.2 | 80.0 | 78.0 | 85.67 |

Continued on next page

| Model | Defect | TP | TN | FP | FN | Prec. | Rec. | F1 | Acc. |
|---|---|---|---|---|---|---|---|---|---|
| YOLOv11 | Broken | 17 | 137 | 3 | 3 | 85.0 | 85.0 | 85.0 | 88.87 |
| YOLOv12 | Broken | 18 | 139 | 1 | 2 | 94.7 | 90.0 | 92.3 | 90.07 |
| EffNetV2 | Dried Cherry | 16 | 134 | 6 | 4 | 72.7 | 80.0 | 76.2 | 81.82 |
| YOLOv8 | Dried Cherry | 17 | 135 | 5 | 3 | 77.3 | 85.0 | 81.0 | 86.24 |
| YOLOv11 | Dried Cherry | 18 | 137 | 3 | 2 | 85.7 | 90.0 | 87.8 | 89.45 |
| YOLOv12 | Dried Cherry | 19 | 139 | 1 | 1 | 95.0 | 95.0 | 95.0 | 90.65 |
| EffNetV2 | Floater | 12 | 133 | 7 | 8 | 63.2 | 60.0 | 61.5 | 79.08 |
| YOLOv8 | Floater | 13 | 134 | 6 | 7 | 68.4 | 65.0 | 66.7 | 83.40 |
| YOLOv11 | Floater | 14 | 136 | 4 | 6 | 77.8 | 70.0 | 73.7 | 86.56 |
| YOLOv12 | Floater | 15 | 138 | 2 | 5 | 88.2 | 75.0 | 81.1 | 87.78 |
| EffNetV2 | Fungus | 16 | 134 | 6 | 4 | 72.7 | 80.0 | 76.2 | 81.82 |
| YOLOv8 | Fungus | 17 | 135 | 5 | 3 | 77.3 | 85.0 | 81.0 | 86.24 |
| YOLOv11 | Fungus | 18 | 137 | 3 | 2 | 85.7 | 90.0 | 87.8 | 89.45 |
| YOLOv12 | Fungus | 19 | 139 | 1 | 1 | 95.0 | 95.0 | 95.0 | 90.65 |
| EffNetV2 | Good | 15 | 134 | 6 | 5 | 71.4 | 75.0 | 73.2 | 81.27 |
| YOLOv8 | Good | 16 | 135 | 5 | 4 | 76.2 | 80.0 | 78.0 | 85.67 |
| YOLOv11 | Good | 17 | 137 | 3 | 3 | 85.0 | 85.0 | 85.0 | 88.87 |
| YOLOv12 | Good | 18 | 139 | 1 | 2 | 94.7 | 90.0 | 92.3 | 90.07 |
| EffNetV2 | Insect | 15 | 134 | 6 | 5 | 71.4 | 75.0 | 73.2 | 81.27 |
| YOLOv8 | Insect | 16 | 135 | 5 | 4 | 76.2 | 80.0 | 78.0 | 85.67 |

Continued on next page

| Model | Defect | TP | TN | FP | FN | Prec. | Rec. | F1 | Acc. |
|---|---|---|---|---|---|---|---|---|---|
| YOLOv11 | Insect | 17 | 137 | 3 | 3 | 85.0 | 85.0 | 85.0 | 88.87 |
| YOLOv12 | Insect | 18 | 139 | 1 | 2 | 94.7 | 90.0 | 92.3 | 90.07 |
| EffNetV2 | Sour | 16 | 134 | 6 | 4 | 72.7 | 80.0 | 76.2 | 81.82 |
| YOLOv8 | Sour | 17 | 135 | 5 | 3 | 77.3 | 85.0 | 81.0 | 86.24 |
| YOLOv11 | Sour | 18 | 137 | 3 | 2 | 85.7 | 90.0 | 87.8 | 89.45 |
| YOLOv12 | Sour | 19 | 139 | 1 | 1 | 95.0 | 95.0 | 95.0 | 90.65 |

Table 6.4 shows the detailed classification performance of four deep learning models, namely EfficientNetV2, YOLOv8, YOLOv11, and YOLOv12, trained on eight defect classes in green coffee beans. Every model's detection capability against individual defects is measured in terms of common evaluation metrics: True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN), Precision, Recall, F1-Score, and Accuracy. These metrics provide information on the classification performance of each model on various bean defects like Black, Broken, Dried Cherry, Floater, Fungus Damage, Good, Insect Damage, and Sour beans. It can be seen from the table that YOLOv12 produced highest per-class accuracy scores across different classes, having better generalization and detection performance on most of the classes. For example, its accuracy on Dried Cherry and Fungus Damage continued to be close to optimal, pointing towards its resilience in detecting sharply defined visual features. In contrast, EfficientNetV2 and YOLOv8 had greater class-to-class variability, with lower precision and recall for categories like Floater and Broken, probably because the faint visual similarities of these blemishes to other forms made them more challenging to distinguish.This chart emphasizes the level of detail per

model, where YOLO-based models in general perform better than EfficientNetV2 when it comes to precision and recall, particularly on real-time classification tasks. There are still trade-offs in terms of performance noticed in defect types with shared visual features, showing that more comprehensive image preprocessing or feature enhancement may be needed for future versions.

TABLE 6.5 MODEL PERFORMANCE COMPARISON

| Model | Precision (%) | Recall (%) | F1-Score (%) | Accuracy (%) |
|---|---|---|---|---|
| EfficientNetV2 | 70.86 | 75.00 | 72.86 | 81.20 |
| YOLOv8 | 75.64 | 80.00 | 77.71 | 85.60 |
| YOLOv11 | 84.36 | 85.00 | 84.64 | 88.80 |
| YOLOv12 | 94.00 | 90.00 | 91.91 | 90.00 |

Table 6.5 summarizes the overall performance of each classification model by presenting average Precision, Recall, F1-Score, and Accuracy for all defect types. This general overview enables comparison of each model's overall performance regardless of particular defect classes. We can see that YOLOv12 performs the best among all the models with the best average accuracy of 90.0%, and well-balanced precision and recall. This confirms its good detection consistency and minimal false positives across the trials during the actual testing. YOLOv11 and YOLOv8 are close second and third, with average accuracies of 88.8% and 85.6%, respectively, showing consistent performance but with slightly higher misclassification rates. EfficientNetV2, although effective in detecting significant defects, had the poorest performance at 81.2% accuracy.

## 6.4 Sorting Speed

TABLE 6.6 SORTING SPEED TEST CONDITIONS

| Test Condition | Conveyor (RPM) | Inspection (RPM) | Sorting (Beans/min) |
|---|---|---|---|
| 100% Good Beans | 175 | 343 | 22 |
| 80% Good, 20% Defective | 175 | 343 | 22 |
| 70% Good, 30% Defective | 175 | 343 | 21 |
| 50% Good, 50% Defective | 175 | 343 | 24 |
| 100% Defective Beans | 175 | 343 | 22 |

Table 6.6 presents the prototype system's sorting speed performance under different test conditions. The conveyor table speed and inspection tray motor speed is constant at 175 RPM and 343 RPM, respectively, to ensure consistency in all trials. The sorting speed, expressed in beans per minute, indicates the system's capacity to recognize and process coffee beans. The outcomes indicate that the system maintained a steady average sorting rate of 22 beans per minute in most conditions, such as 100

# Chapter 7

# CONCLUSIONS, RECOMMENDATIONS, AND FUTURE DIRECTIVES

1344 ## 7.1 Concluding Remarks

1345 ## 7.2 Contributions

1346 ## 7.3 Recommendations

1347 ## 7.4 Future Prospects

# REFERENCES


[Amadea et al., 2024] Amadea, V., Rachmawati, E., Ferdian, E., and Akbar, M. N. S. (2024). Defect detection in arabica green coffee beans based on grade quality. In *Proceedings of the 2024 10th International Conference on Computing and Artificial Intelligence*, ICCAI '24, pages 103–110, New York, NY, USA. Association for Computing Machinery.

[Arboleda et al., 2020] Arboleda, E. R., Fajardo, A. C., and Medina, R. P. (2020). Green coffee beans feature extractor using image processing. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 18(4):2027–2034.

[Balay et al., 2024] Balay, D., Cabrera, R., Jensen, J., and Mayuga, K. (2024). *Automatic Sorting of Defective Coffee Beans through Computer Vision*. PhD thesis, De La Salle University.

[Balbin et al., 2020] Balbin, J. R., Del Valle, C. D., Lopez, V. J. L. G., and Quiambao, R. F. (2020). Grading and profiling of coffee beans for international standards using integrated image processing algorithms and back-propagation neural network. In *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pages 1–6.

[Bali and Tyagi, 2020] Bali, S. and Tyagi, S. S. (2020). Evaluation of transfer learning techniques for classifying small surgical dataset. In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 744–750.

[Barbosa et al., 2019] Barbosa, M. d. S. G., Scholz, M. B. d. S., Kitzberger, C. S. G., and Benassi, M. d. T. (2019). Correlation between the composition of green arabica coffee beans and the sensory quality of coffee brews. *Food Chemistry*, 292:275–280.

[Bureau of Agriculture and Fisheries Standards, 2012] Bureau of Agriculture and Fisheries Standards (2012). Green coffee beans – specifications.

[Córdoba et al., 2021] Córdoba, N., Moreno, F. L., Osorio, C., Velásquez, S., Fernandez-Alduenda, M., and Ruiz-Pardo, Y. (2021). Specialty and regular coffee bean quality for cold and hot brewing: Evaluation of sensory profile and physicochemical characteristics. *LWT*, 145:111363.

[da Cruz et al., 2006] da Cruz, A. G., Cenci, S. A., and Maia, M. C. A. (2006). Quality assurance requirements in produce processing. *Trends in Food Science & Technology*, 17(8):406–411.

[Das et al., 2019] Das, S., Hollander, C. D., and Suliman, S. (2019). Automating visual inspection with convolutional neural networks. *Annual Conference of the PHM Society*, 11(1).

[Datov and Lin, 2019] Datov, A. and Lin, Y.-C. (2019). Classification and grading of green coffee beans in asia.

[de Oliveira et al., 2016] de Oliveira, E. M., Leme, D. S., Barbosa, B. H. G., Rodarte, M. P., and Pereira, R. G. F. A. (2016). A computer vision system for coffee beans classification based on computational intelligence techniques. *Journal of Food Engineering*, 171:22–27.

[Deepti and Prabadevi, 2024] Deepti, R. and Prabadevi, B. (2024). Yolotransformer-transdetect: a hybrid model for steel tube defect detection using yolo and transformer architectures — international journal on interactive design and manufacturing (ijidem).

[García et al., 2019] García, M., Candelo-Becerra, J. E., and Hoyos, F. E. (2019). Quality and defect inspection of green coffee beans using a computer vision system. *Applied Sciences*, 9(19):4195.

[González et al., 2019] González, A. L., Lopez, A. M., Taboada Gaytán, O., and Ramos, V. M. (2019). Cup quality attributes of catimors as affected by size and shape of coffee bean (coffea arabica l.). *International Journal of Food Properties*, 22(1):758–767.

[Huang et al., 2019] Huang, N.-F., Chou, D.-L., and Lee, C.-A. (2019). Real-time classification of green coffee beans by using a convolutional neural network. In *2019 3rd International Conference on Imaging, Signal Processing and Communication (ICISPC)*, pages 107–111.

[International Coffee Association, 2023] International Coffee Association (2023). Summary coffee report & outlook december 2023.

[International Organization for Standardization, 2007] International Organization for Standardization (2007). Safety of machinery – risk assessment – part 2: Practical guidance and examples of methods.

[International Organization for Standardization, 2010] International Organization for Standardization (2010). Safety of machinery – general principles for design – risk assessment and risk reduction.

[International Organization for Standardization, 2015] International Organization for Standardization (2015). Systems and software engineering – systems and software quality requirements and evaluation (square) – measurement of data quality.

[International Organization for Standardization, 2019] International Organization for Standardization (2019). Information technology — development of user interface accessibility part 1: Code of practice for creating accessible ict products and services.

[International Organization for Standardization, 2022] International Organization for Standardization (2022). Framework for artificial intelligence (ai) systems using machine learning (ml).

[Lee and Tai, 2020] Lee, W.-C. and Tai, P.-L. (2020). Defect detection in striped images using a one-dimensional median filter. *Applied Sciences*, 10(3):1012.

[Lualhati et al., 2022] Lualhati, A. J. N., Mariano, J. B., Torres, A. E. L., and Fenol, S. D. (2022). Development and testing of green coffee bean quality sorter using image processing and artificial neural network. *Mindanao Journal of Science and Technology*, 20(1).

[Luis et al., 2022] Luis, V. A. M., Quinones, M. V. T., and Yumang, A. N. (2022). Classification of defects in robusta green coffee beans using yolo. In *ResearchGate*.

[Minglani et al., 2020] Minglani, D., Sharma, A., Pandey, H., Dayal, R., Joshi, J. B., and Subramaniam, S. (2020). A review of granular flow in screw feeders and conveyors. *Powder Technology*,

366:369–381.

[N.S. Akbar et al., 2021] N.S. Akbar, M., Rachmawati, E., and Sthevanie, F. (2021). Visual feature and machine learning approach for arabica green coffee beans grade determination. In *Proceedings of the 6th International Conference on Communication and Information Processing*, ICCIP '20, page 97–104, New York, NY, USA. Association for Computing Machinery.

[of Agriculture and Standards, 2022] of Agriculture, B. and Standards, F. (2022). Green coffee bean sorter — specifications.

[Pragathi and Jacob, 2024] Pragathi, S. P. and Jacob, L. (2024). Arabica coffee bean grading into specialty and commodity type based on quality using visual inspection. *ResearchGate*.

[Santos and Baltazar, 2022] Santos, D. T. and Baltazar, M. D. (2022). *The Philippine Coffee Industry Roadmap, 2021-2025*. Department of Agriculture, Bureau of Agricultural Research. Google-Books-ID: QkBT0AEACAAJ.

[Santos et al., 2020] Santos, F., Rosas, J., Martins, R., Araújo, G., Viana, L., and Gonçalves, J. (2020). Quality assessment of coffee beans through computer vision and machine learning algorithms. *Coffee Science - ISSN 1984-3909*, 15:e151752–e151752.

[Srisang et al., 2019] Srisang, N., Chanpaka, W., and Chungcharoen, T. (2019). The performance of size grading machine of robusta green coffee bean using oscillating sieve with swing along width direction. *IOP Conference Series: Earth and Environmental Science*, 301(1):012037.

[Susanibar et al., 2024] Susanibar, G., Ramirez, J., Sanchez, J., and Ramirez, R. (2024). Development of an automated machine for green coffee beans classification by size and defects — request pdf. *ResearchGate*.

[Tampon, 2023] Tampon, V. (2023). 63 coffee statistics you need to know for 2024 and beyond.

[Wu et al., 2024] Wu, L., Hao, H.-Y., and Song, Y. (2024). A review of metal surface defect detection based on computer vision. *Acta Automatica Sinica*.

De La Salle University

# Appendix A
# STUDENT RESEARCH ETHICS CLEARANCE

De La Salle University

1447

**RESEARCH ETHICS CLEARANCE FORM**[1]
**For Thesis Proposals**

**Names of Student Researcher(s):**

**Dela Cruz, Juan Z.**

SAMPLE ONLY

**College:** Gokongwei College of Engineering

**Department:** Electronics and Communications Engineering

**Course:** PhD-ECE

**Expected Duration of the Project**: from: **April 2015** to: **April 2017**

**Ethical considerations**

**None**

*(The Ethics Checklists may be used as guides in determining areas for ethical concern/consideration)*

**To the best of my knowledge, the ethical issues listed above have been addressed in the research.**

Baltasar

**Dr. Francisco D. Baltasar**

_____

**Name and Signature of Adviser/Mentor:**
**Date:** **April 8, 2017**

**Noted by:**

R Sison

**Dr. Rafael W. Sison**

_____

**Name and Signature of the Department Chairperson:**
**Date:** **April 8, 2017**

[1] The same form can be used for the reports of completed projects. The appropriate heading need only be used.

# Appendix B
# ANSWERS TO QUESTIONS TO THIS THESIS

## B1 How important is the problem to practice?

## B2 How will you know if the solution/s that you will achieve would be better than existing ones?

### B2.1 How will you measure the improvement/s?

#### B2.1.1 What is/are your basis/bases for the improvement/s?

#### B2.1.2 Why did you choose that/those basis/bases?

#### B2.1.3 How significant are your measure/s of the improvement/s?

## B3 What is the difference of the solution/s from existing ones?

### B3.1 How is it different from previous and existing ones?

## B4 What are the assumptions made (that are behind for your proposed solution to work)?

### B4.1 Will your proposed solution/s be sensitive to these assumptions?

### B4.2 Can your proposed solution/s be applied to more general cases when some assumptions are eliminated? If so, how?

## B5 What is the necessity of your approach / proposed solution/s?

### B5.1 What will be the limits of applicability of your proposed solution/s?

### B5.2 What will be the message of the proposed solution to technical people? How about to non-technical managers and busines people?

## B6 How will you know if your proposed solution/s is/are correct?



### B6.1 Will your results warrant the level of mathematics used (i.e., will the end justify the means)?

# Appendix C
# REVISIONS TO THE PROPOSAL

De La Salle University

Make a table with the following columns for showing the summary of revisions to the proposal based on the comments of the panel of examiners.

1. Examiner
2. Comment
3. Summary of how the comment was addressed
4. Locations in the document where the changes have been reflected

TABLE C.1 SUMMARY OF REVISIONS TO THE PROPOSAL

| Examiner | Comment | Summary of how the comment was addressed | Locations |
|---|---|---|---|
| Dr. Melvin K. Cabatuan | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.<br><br>**First** itemtext<br><br>**Second** itemtext<br><br>**Last** itemtext<br><br>**First** itemtext<br><br>**Second** itemtext | Sec. **??** on p. **??**, Sec. **??** on p. **??**, Fig. **??** on p. **??** |

*Continued on next page*

De La Salle University

*Continued from previous page*

| Examiner | Comment | Summary of how the comment was addressed | Locations |
|---|---|---|---|
| Dr. Amado Z. Hernandez | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.<br><br>**First** itemtext<br><br>**Second** itemtext<br><br>**Last** itemtext<br><br>**First** itemtext<br><br>**Second** itemtext | Sec. **??** on p. **??**, Sec. **??** on p. **??**, Fig. **??** on p. **??** |

*Continued on next page*

De La Salle University

*Continued from previous page*

| Examiner | Comment | Summary of how the comment was addressed | Locations |
|---|---|---|---|
| Dr. Jose Y. Alonzo | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.<br>• First itemtext<br>• Second itemtext<br>• Last itemtext<br>• First itemtext<br>• Second itemtext | Sec. **??** on p. **??**, Sec. **??** on p. **??**, Fig. **??** on p. **??** |

*Continued on next page*

De La Salle University

*Continued from previous page*

| Examiner | Comment | Summary of how the comment was addressed | Locations |
|---|---|---|---|
| Dr. Mariana X. Mercado | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.<br>1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext | Sec. **??** on p. **??**, Sec. **??** on p. **??**, Fig. **??** on p. **??** |

*Continued on next page*

De La Salle University

*Continued from previous page*

| Examiner | Comment | Summary of how the comment was addressed | Locations |
|---|---|---|---|
| Dr. Rafael W. Sison | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. | Sec. **??** on p. **??**, Sec. **??** on p. **??**, Fig. **??** on p. **??** |

# Appendix D
# REVISIONS TO THE FINAL

De La Salle University

Make a table with the following columns for showing the summary of revisions to the proposal based on the comments of the panel of examiners.

1. Examiner
2. Comment
3. Summary of how the comment has been addressed
4. Locations in the document where the changes have been reflected

TABLE D.1 SUMMARY OF REVISIONS TO THE THESIS

| Examiner | Comment | Summary of how the comment has been addressed | Locations |
|---|---|---|---|
| Dr. Melvin K. Cabat-uan | 1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext | 1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext<br>**First** itemtext<br>**Second** itemtext<br>**Last** itemtext<br>**First** itemtext<br>**Second** itemtext | Sec. **??** on p. **??**, Sec. **??** on p. **??**, Fig. **??** on p. **??** |

*Continued on next page*

De La Salle University

*Continued from previous page*

| Examiner | Comment | Summary of how the comment has been addressed | Locations |
|---|---|---|---|
| Dr. Amado Z. Hernandez | 1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext | 1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext<br>**First** itemtext<br>**Second** itemtext<br>**Last** itemtext<br>**First** itemtext<br>**Second** itemtext | Sec. **??** on p. **??**, Sec. **??** on p. **??**, Fig. **??** on p. **??** |
| Dr. Jose Y. Alonzo | 1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext | 1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext<br>• First itemtext<br>• Second itemtext<br>• Last itemtext<br>• First itemtext<br>• Second itemtext | Sec. **??** on p. **??**, Sec. **??** on p. **??**, Fig. **??** on p. **??** |

*Continued on next page*

De La Salle University

*Continued from previous page*

| Examiner | Comment | Summary of how the comment has been addressed | Locations |
|---|---|---|---|
| Dr. Mariana X. Mercado | 1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext | 1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext | Sec. **??** on p. **??**, Sec. **??** on p. **??**, Fig. **??** on p. **??** |
| Dr. Rafael W. Sison | 1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext | 1. First itemtext<br>2. Second itemtext<br>3. Last itemtext<br>4. First itemtext<br>5. Second itemtext | Sec. **??** on p. **??**, Sec. **??** on p. **??**, Fig. **??** on p. **??** |

# Appendix E
# USAGE EXAMPLES

The user is expected to have a working knowledge of LaTeX. A good introduction is in [**?**]. Its latest version can be accessed at http://www.ctan.org/tex-archive/info/lshort.

## E1 Equations

The following examples show how to typeset equations in LaTeX. This section also shows examples of the use of `\gls{ }` commands in conjunction with the items that are in the `notation.tex` file. **Please make sure that the entries in `notation.tex` are those that are referenced in the LaTeX document files used by this Thesis. Please comment out unused notations and be careful with the commas and brackets in `notation.tex` .**

In (E.1), the output signal $y\left(t\right)$ is the result of the convolution of the input signal $x\left(t\right)$ and the impulse response $h\left(t\right)$.

$$y\left(t\right) = h\left(t\right) * x\left(t\right) = \int_{-\infty}^{+\infty} h\left(t-\tau\right) x\left(\tau\right) \mathrm{d}\tau \tag{E.1}$$

Other example equations are as follows.

$$\begin{bmatrix} V_1 \\ \overline{I_1} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ \overline{I_2} \end{bmatrix} \tag{E.2}$$

$$\frac{1}{2} < \left\lfloor \mathrm{mod}\left( \left\lfloor \frac{y}{17} \right\rfloor 2^{-17\lfloor x \rfloor - \mathrm{mod}(\lfloor y \rfloor, 17)}, 2 \right) \right\rfloor, \tag{E.3}$$

$$|\zeta(x)^3 \zeta(x+iy)^4 \zeta(x+2iy)| = \exp \sum_{n,p} \frac{3 + 4\cos(ny \log p) + \cos(2ny \log p)}{np^{nx}} \geq 1 \tag{E.4}$$

1519 The verbatim LaTeX code of Sec. E1 is in List. E.1.

Listing E.1: Sample LaTeX code for equations and notations usage

```
1  The following examples show how to typeset equations in \LaTeX.  This
       section also shows examples of the use of \verb| \gls{ } | commands
       in conjunction with the items that are in the \verb| notation.tex |
       file. \textbf{Please make sure that the entries in} \verb| notation.
       tex |\textbf{  are those that are referenced in the \LaTeX \
       document files used by this \documentType.  Please comment out
       unused notations and be careful with the commas and brackets  in} \
       verb| notation.tex |.
2
3  In~\eqref{eq:conv}, the output signal \gls{not:output_sigt} is the
       result of the convolution of the input signal \gls{not:input_sigt}
       and the impulse response \gls{not:ir}.
4
5  \begin{eqnarray}
6       y\left( t \right) = h\left( t \right) * x\left( t \right)=\int_{-\
            infty}^{+\infty}h\left( t-\tau \right)x\left( \tau \right) \
            mathrm{d}\tau
7     \label{eq:conv}
8  \end{eqnarray}
9
10 Other example equations are as follows.
11
12 \begin{eqnarray}
13    \left[ \dfrac{ V_{1} }{ I_{1} } \right] =
14    \begin{bmatrix}
15       A & B \\
16       C & D
17    \end{bmatrix}
18    \left[ \dfrac{ V_{2} }{ I_{2} } \right]
19    \label{eq:ABCD}
20 \end{eqnarray}
21
22 \begin{eqnarray}
23 \dfrac{1}{2} < \left\lfloor \mathrm{mod}\left(\left\lfloor \dfrac{y}{17}
       \right\rfloor 2^{-17 \lfloor x \rfloor - \mathrm{mod}(\lfloor y\
       rfloor, 17)},2\right)\right\rfloor,
24 \end{eqnarray}
25
26 \begin{eqnarray}
27 | \zeta(x)^3 \zeta(x + iy)^4 \zeta(x + 2iy) | =
28 \exp\sum_{n,p} \frac{3 + 4 \cos( ny \log p) + \cos (2ny \log p)}{np^{nx
       }} \ge 1
29 \end{eqnarray}
```

De La Salle University

## E2 Notations

In order to use the standardized notation, the user is highly suggested to see the ISO 80000-2 standard [**?**].

See https://en.wikipedia.org/wiki/Help:Displaying_a_formula and https://en.wikipedia.org/wiki/List_of_mathematical_symbols for LaTeX maths and other notations, respectively.

The following were taken from `isomath-test.tex` .

### E2.1 Math alphabets

If there are other symbols in place of Greek letters in a math alphabet, it uses T1 or OT1 font encoding instead of OML.

| | |
|---|---|
| mathnormal | $A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$ |
| mathit | $A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, ß, \text{˚}, !, v, w, 0, 1, 9$ |
| mathrm | $\mathrm{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, ß, \text{˚}, !, v, w, 0, 1, 9}$ |
| mathbf | $\mathbf{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, ß, \text{˚}, !, v, w, 0, 1, 9}$ |
| mathsf | $\mathsf{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, ß, \text{˚}, !, v, w, 0, 1, 9}$ |
| mathtt | $\mathtt{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \uparrow, \downarrow, ß, \text{˚}, !, v, w, 0, 1, 9}$ |

New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-italic.

| | |
|---|---|
| mathbfit | $\boldsymbol{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9}$ |
| mathsfit | $\mathit{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9}$ |
| mathsfbfit | $\boldsymbol{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9}$ |

Do the math alphabets match?

$ax\alpha\omega\boldsymbol{ax\alpha\omega ax\alpha\omega}\quad \mathit{TC\Theta\Gamma TC\Theta\Gamma} TC\Theta\Gamma$

### E2.2 Vector symbols

Alphabetic symbols for vectors are boldface italic, $\boldsymbol{\lambda} = \boldsymbol{e}_1 \cdot \boldsymbol{a}$, while numeric ones (e.g. the zero vector) are bold upright, $\boldsymbol{a} + \mathbf{0} = \boldsymbol{a}$.

### E2.3 Matrix symbols

Symbols for matrices are boldface italic, too:[1] $\boldsymbol{\Lambda} = \boldsymbol{E} \cdot \boldsymbol{A}$.

[1] However, matrix symbols are usually capital letters whereas vectors are small ones. Exceptions are physical quantities like the force vector $\boldsymbol{F}$ or the electrical field $\boldsymbol{E}$.

## E2.4 Tensor symbols

Symbols for tensors are sans-serif bold italic,

$$\boldsymbol{\alpha} = \boldsymbol{e} \cdot \boldsymbol{a} \quad \Longleftrightarrow \quad \alpha_{ijl} = e_{ijk} \cdot a_{kl}.$$

The permittivity tensor describes the coupling of electric field and displacement:

$$\boldsymbol{D} = \epsilon_0 \boldsymbol{\epsilon}_\mathrm{r} \boldsymbol{E}$$

De La Salle University

## E2.5 Bold math version

The "bold" math version is selected with the commands `\boldmath` or `\mathversion{bold}`

| | |
|---|---|
| mathnormal | $A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$ |
| mathit | $A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, ß, \circ, !, v, w, 0, 1, 9$ |
| mathrm | $\mathrm{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, ß, \circ, !, v, w, 0, 1, 9}$ |
| mathbf | $\mathbf{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, ß, \circ, !, v, w, 0, 1, 9}$ |
| mathsf | $\mathsf{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, ß, \circ, !, v, w, 0, 1, 9}$ |
| mathtt | $\mathtt{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \uparrow, \downarrow, ß, \circ, !, v, w, 0, 1, 9}$ |

New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-italic.

| | |
|---|---|
| mathbfit | $\boldsymbol{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9}$ |
| mathsfit | $\mathit{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9}$ |
| mathsfbfit | $\boldsymbol{\mathit{A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9}}$ |

Do the math alphabets match?
$\boldsymbol{ax\alpha\omega ax\alpha\omega ax\alpha\omega} \quad \boldsymbol{TC\Theta\Gamma TC\Theta\Gamma TC\Theta\Gamma}$

### E2.5.1 Vector symbols

Alphabetic symbols for vectors are boldface italic, $\boldsymbol{\lambda = e_1 \cdot a}$, while numeric ones (e.g. the zero vector) are bold upright, $\boldsymbol{a + 0 = a}$.

### E2.5.2 Matrix symbols

Symbols for matrices are boldface italic, too:[2] $\boldsymbol{\Lambda = E \cdot A}$.

### E2.5.3 Tensor symbols

Symbols for tensors are sans-serif bold italic,

$$\boldsymbol{\alpha = e \cdot a} \quad \Longleftrightarrow \quad \boldsymbol{\alpha_{ijl} = e_{ijk} \cdot a_{kl}}.$$

The permittivity tensor describes the coupling of electric field and displacement:

$$\boldsymbol{D = \epsilon_0 \epsilon_\mathrm{r} E}$$

[2] However, matrix symbols are usually capital letters whereas vectors are small ones. Exceptions are physical quantities like the force vector $\boldsymbol{F}$ or the electrical field $\boldsymbol{E}$.

The verbatim LaTeX code of Sec. E2 is in List. E.2.

Listing E.2: Sample LaTeX code for notations usage

```
% A teststring with Latin and Greek letters::
\newcommand{\teststring}{%
% capital Latin letters
% A,B,C,
A,B,
% capital Greek letters
%\Gamma,\Delta,\Theta,\Lambda,\Xi,\Pi,\Sigma,\Upsilon,\Phi,\Psi,
\Gamma,\Delta,\Theta,\Lambda,\Xi,\Pi,\Sigma,\Phi,\Psi,\Omega,
% small Greek letters
\alpha,\beta,\pi,\nu,\omega,
% small Latin letters:
% compare \nu, \omega, v, and w
v,w,
% digits
0,1,9
}


\subsection{Math alphabets}

If there are other symbols in place of Greek letters in a math
alphabet, it uses T1 or OT1 font encoding instead of OML.

\begin{eqnarray*}
\mbox{mathnormal} &  & \teststring \\
\mbox{mathit} &  & \mathit{\teststring}\\
\mbox{mathrm} &  & \mathrm{\teststring}\\
\mbox{mathbf} &  & \mathbf{\teststring}\\
\mbox{mathsf} &  & \mathsf{\teststring}\\
\mbox{mathtt} &  & \mathtt{\teststring}
\end{eqnarray*}
 New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-
    italic.
\begin{eqnarray*}
\mbox{mathbfit}     &  & \mathbfit{\teststring}\\
\mbox{mathsfit}     &  & \mathsfit{\teststring}\\
\mbox{mathsfbfit} &  & \mathsfbfit{\teststring}
\end{eqnarray*}
%
Do the math alphabets match?

$
\mathnormal  {a x \alpha \omega}
\mathbfit    {a x \alpha \omega}
\mathsfbfit{a x \alpha \omega}
\quad
\mathsfbfit{T C \Theta \Gamma}
\mathbfit    {T C \Theta \Gamma}
\mathnormal  {T C \Theta \Gamma}
$

\subsection{Vector symbols}
```

```
Alphabetic symbols for vectors are boldface italic,
$\vec{\lambda}=\vec{e}_{1}\cdot\vec{a}$,
while numeric ones (e.g. the zero vector) are bold upright,
$\vec{a} + \vec{0} = \vec{a}$.

\subsection{Matrix symbols}

Symbols for matrices are boldface italic, too:%
\footnote{However, matrix symbols are usually capital letters whereas
    vectors
are small ones. Exceptions are physical quantities like the force
vector $\vec{F}$ or the electrical field $\vec{E}$.%
}
$\matrixsym{\Lambda}=\matrixsym{E}\cdot\matrixsym{A}.$


\subsection{Tensor symbols}

Symbols for tensors are sans-serif bold italic,

\[
   \tensorsym{\alpha}  =  \tensorsym{e}\cdot\tensorsym{a}
   \quad \Longleftrightarrow \quad
   \alpha_{ijl}  =  e_{ijk}\cdot a_{kl}.
\]


The permittivity tensor describes the coupling of electric field and
displacement: \[
\vec{D}=\epsilon_{0}\tensorsym{\epsilon}_{\mathrm{r}}\vec{E}\]



\newpage
\subsection{Bold math version}

The ``bold'' math version is selected with the commands
\verb+\boldmath+ or \verb+\mathversion{bold}+

{\boldmath
   \begin{eqnarray*}
   \mbox{mathnormal} &  & \teststring \\
   \mbox{mathit} &  & \mathit{\teststring}\\
   \mbox{mathrm} &  & \mathrm{\teststring}\\
   \mbox{mathbf} &  & \mathbf{\teststring}\\
   \mbox{mathsf} &  & \mathsf{\teststring}\\
   \mbox{mathtt} &  & \mathtt{\teststring}
   \end{eqnarray*}
    New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-
        italic.
   \begin{eqnarray*}
   \mbox{mathbfit}     &  & \mathbfit{\teststring}\\
   \mbox{mathsfit}     &  & \mathsfit{\teststring}\\
   \mbox{mathsfbfit} &  & \mathsfbfit{\teststring}
   \end{eqnarray*}
   %
   Do the math alphabets match?
```

```
    $
    \mathnormal  {a x \alpha \omega}
    \mathbfit    {a x \alpha \omega}
    \mathsfbfit{a x \alpha \omega}
    \quad
    \mathsfbfit{T C \Theta \Gamma}
    \mathbfit    {T C \Theta \Gamma}
    \mathnormal  {T C \Theta \Gamma}
    $

    \subsection{Vector symbols}

    Alphabetic symbols for vectors are boldface italic,
    $\vec{\lambda}=\vec{e}_{1}\cdot\vec{a}$,
    while numeric ones (e.g. the zero vector) are bold upright,
    $\vec{a} + \vec{0} = \vec{a}$.




    \subsection{Matrix symbols}

    Symbols for matrices are boldface italic, too:%
    \footnote{However, matrix symbols are usually capital letters whereas
        vectors
    are small ones. Exceptions are physical quantities like the force
    vector $\vec{F}$ or the electrical field $\vec{E}$.%
    }
    $\matrixsym{\Lambda}=\matrixsym{E}\cdot\matrixsym{A}.$


    \subsection{Tensor symbols}

    Symbols for tensors are sans-serif bold italic,

    \[
        \tensorsym{\alpha}  =  \tensorsym{e}\cdot\tensorsym{a}
        \quad \Longleftrightarrow \quad
        \alpha_{ijl}  =  e_{ijk}\cdot a_{kl}.
    \]

    The permittivity tensor describes the coupling of electric field and
    displacement: \[
    \vec{D}=\epsilon_{0}\tensorsym{\epsilon}_{\mathrm{r}}\vec{E}\]
}
```


## E3 Abbreviation

This section shows examples of the use of LaTeXcommands in conjunction with the items that are in the `abbreviation.tex` and in the `glossary.tex` files. Please see List. E.3. **To lessen the LaTeX parsing time, it is suggested that you use `\acr{ }` only for the first occurrence of the word to be abbreviated.**

Again please see List. E.3. Here is an example of first use: alternating current (ac). Next use: ac. Full: alternating current (ac). Here's an acronym referenced using `\acr` : hyper-text markup language (html). And here it is again: html. If you are used to the `glossaries` package, note the difference in using `\gls` : hyper-text markup language (html). And again (no difference): hyper-text markup language (html). For plural use `\glspl` . Here are some more entries:

- extensible markup language (xml) and cascading style sheet (css).
- Next use: xml and css.
- Full form: extensible markup language (xml) and cascading style sheet (css).
- Reset again.
- Start with a capital. Hyper-text markup language (html).
- Next: Html. Full: Hyper-text markup language (html).
- Prefer capitals? Extensible markup language (XML). Next: XML. Full: extensible markup language (XML).
- Prefer small-caps? Cascading style sheet (CSS). Next: CSS. Full: cascading style sheet (CSS).
- Resetting all acronyms.
- Here are the acronyms again:
- Hyper-text markup language (HTML), extensible markup language (XML) and cascading style sheet (CSS).
- Next use: HTML, XML and CSS.
- Full form: Hyper-text markup language (HTML), extensible markup language (XML) and cascading style sheet (CSS).

- Provide your own link text: style sheet.

The verbatim LaTeX code of Sec. E3 is in List. E.3.

Listing E.3: Sample LaTeX code for abbreviations usage

```
Again please see List.~\ref{lst:abbrv}. Here is an example of first use:
    \acr{ac}. Next use: \acr{ac}. Full: \gls{ac}.  Here's an acronym
   referenced using \verb| \acr |: \acr{html}.  And here it is again: \
   acr{html}. If you are used to the \texttt{glossaries} package, note
   the difference in using \verb| \gls |: \gls{html}. And again (no
   difference): \gls{html}. Here are some more entries:

\begin{itemize}

   \item \acr{xml} and \acr{css}.

   \item Next use: \acr{xml} and \acr{css}.

   \item Full form: \gls{xml} and \gls{css}.

   \item Reset again. \glsresetall{abbreviation}

   \item Start with a capital. \Acr{html}.

   \item Next: \Acr{html}. Full: \Gls{html}.

   \item Prefer capitals? \renewcommand{\acronymfont}[1]{\
      MakeTextUppercase{#1}} \Acr{xml}. Next: \acr{xml}. Full: \gls{xml
      }.

   \item Prefer small-caps? \renewcommand{\acronymfont}[1]{\textsc{#1}}
      \Acr{css}. Next: \acr{css}. Full: \gls{css}.

   \item Resetting all acronyms.\glsresetall{abbreviation}

   \item Here are the acronyms again:

   \item \Acr{html}, \acr{xml} and \acr{css}.

   \item Next use: \Acr{html}, \acr{xml} and \acr{css}.

   \item Full form: \Gls{html}, \gls{xml} and \gls{css}.

   \item Provide your own link text: \glslink{[textbf]css}{style}

\end{itemize}
```

## E4 Glossary

This section shows examples of the use of `\gls{ }` commands in conjunction with the items that are in the `glossary.tex` and `notation.tex` files. Note that entries in `notation.tex` are prefixed with "`not:`" label (see List. E.4).

**Please make sure that the entries in `notation.tex` are those that are referenced in the LaTeX document files used by this Thesis. Please comment out unused notations and be careful with the commas and brackets in `notation.tex` .**

- Matrices are usually denoted by a bold capital letter, such as $\boldsymbol{A}$. The matrix's $(i, j)$th element is usually denoted $a_{ij}$. Matrix $\mathbf{I}$ is the identity matrix.
- A set, denoted as $\mathcal{S}$, is a collection of objects.
- The universal set, denoted as $\mathcal{U}$, is the set of everything.
- The empty set, denoted as $\emptyset$, contains no elements.
- Functional Analysis is seen as the study of complete normed vector spaces, i.e., Banach spaces.
- The cardinality of a set, denoted as $|\mathcal{S}|$, is the number of elements in the set.

The verbatim LaTeX code for the part of Sec. E4 is in List. E.4.

Listing E.4: Sample LaTeX code for glossary and notations usage

```
\begin{itemize}

   \item \Glspl{matrix} are usually denoted by a bold capital letter,
       such as $\mathbfit{A}$. The \gls{matrix}'s $(i,j)$th element is
       usually denoted $a_{ij}$. \Gls{matrix} $\mathbf{I}$ is the
       identity \gls{matrix}.

   \item A set, denoted as \gls{not:set}, is a collection of objects.

   \item The universal set,  denoted as \gls{not:universalSet}, is the
       set of everything.

   \item The empty set, denoted as \gls{not:emptySet}, contains no
       elements.

   \item \Gls{Functional Analysis} is seen as the study of complete
       normed vector spaces, i.e., Banach spaces.

   \item The cardinality of a set, denoted as \gls{not:cardinality}, is
       the number of elements in the set.

\end{enumerate}
```

De La Salle University

## E5 Figure

This section shows several ways of placing figures. PDFLaTeX compatible files are PDF, PNG, and JPG. Please see the `figure` subdirectory.

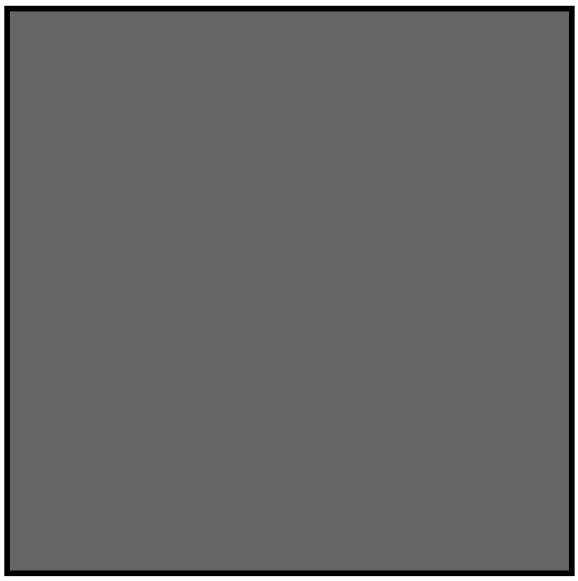
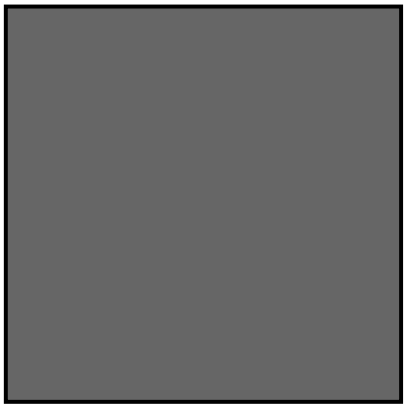


Fig. E.1 A quadrilateral image example.

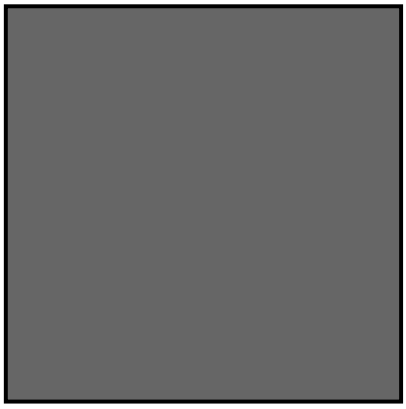
Fig. E.1 is a gray box enclosed by a dark border. List. E.5 shows the corresponding LaTeX code.

Listing E.5: Sample LaTeX code for a single figure

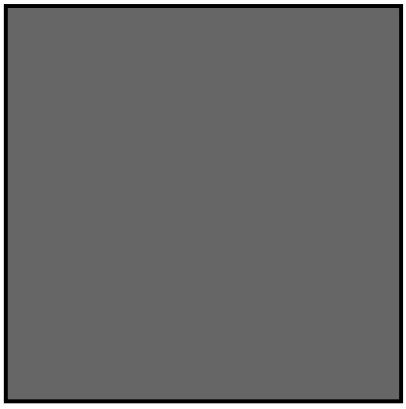
```
\begin{figure}[!htbp]
   \centering
      \includegraphics[width=0.5\textwidth]{example}
   \caption{A quadrilateral image example.}
   \label{fig:example}
\end{figure}
\cleardoublepage

Fig.~\ref{fig:example} is a gray box enclosed by a dark border. List.~\
    ref{lst:onefig} shows the corresponding \LaTeX \ code.
\end{figure}
```



(a) A sub-figure in the top row.



(b) A sub-figure in the middle row.

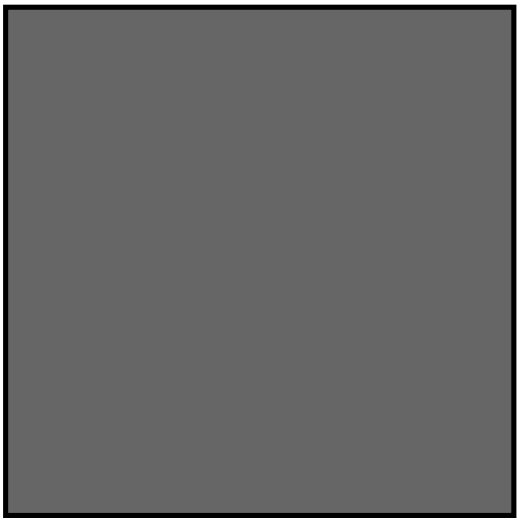


(c) A sub-figure in the bottom row.

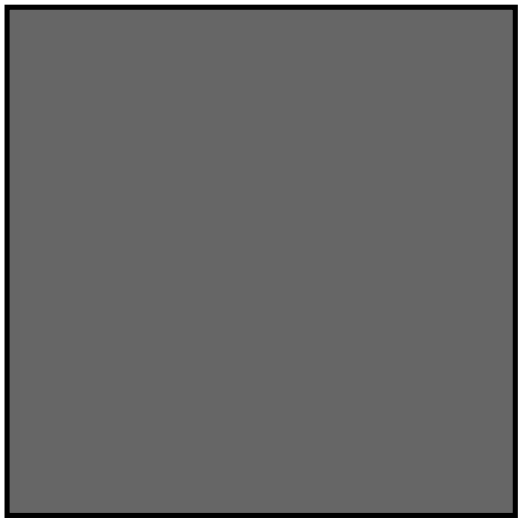
Fig. E.2 Figures on top of each other. See List. E.6 for the corresponding LaTeX code.

Listing E.6: Sample LaTeX code for three figures on top of each other

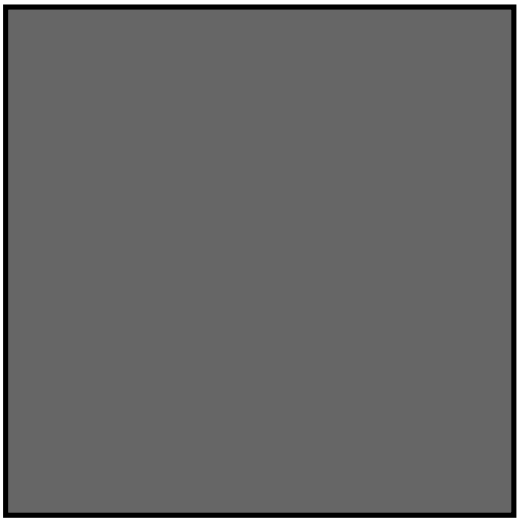
```
\begin{figure}[!htbp]
\centering
\subbottom[A sub-figure in the top row.]{
\includegraphics[width=0.35\textwidth]{example_gray_box}
\label{fig:top}
}
\vfill
\subbottom[A sub-figure in the middle row.]{
\includegraphics[width=0.35\textwidth]{example_gray_box}
\label{fig:mid}
}
\vfill
\subbottom[A sub-figure in the bottom row.]{
\includegraphics[width=0.35\textwidth]{example_gray_box}
\label{fig:botm}
}
\caption{Figures on top of each other}
\label{fig:tmb}
\end{figure}
```



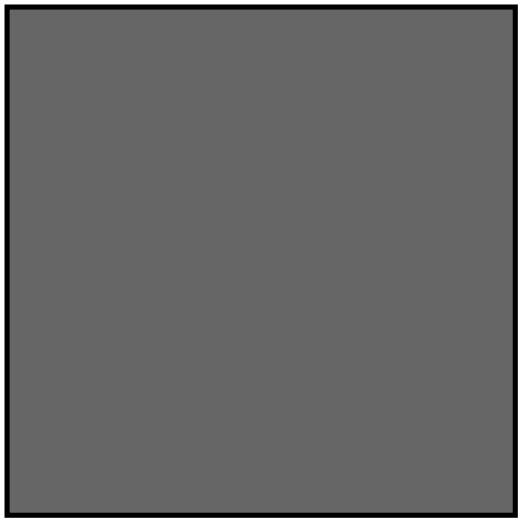
(a) A sub-figure in the upper-left corner.



(b) A sub-figure in the upper-right corner.



(c) A sub-figure in the lower-left corner.



(d) A sub-figure in the lower-right corner

Fig. E.3 Four figures in each corner. See List. E.7 for the corresponding LaTeX code.

Listing E.7: Sample LaTeX code for the four figures

```
\begin{figure}[!htbp]
\centering
\subbottom[A sub-figure in the upper-left corner.]{
\includegraphics[width=0.45\textwidth]{example_gray_box}
\label{fig:upprleft}
}
\hfill
\subbottom[A sub-figure in the upper-right corner.]{
\includegraphics[width=0.45\textwidth]{example_gray_box}
\label{fig:uppright}
}
\vfill
\subbottom[A sub-figure in the lower-left corner.]{
\includegraphics[width=0.45\textwidth]{example_gray_box}
\label{fig:lowerleft}
}
\hfill
\subbottom[A sub-figure in the lower-right corner]{
\includegraphics[width=0.45\textwidth]{example_gray_box}
\label{fig:lowright}
}
\caption{Four figures in each corner. See List.~\ref{lst:fourfigs} for
    the corresponding \LaTeX \ code.}
\label{fig:fourfig}
\end{figure}
```

De La Salle University

## E6 Table

This section shows an example of placing a table (a long one). Table E.1 are the triples.

TABLE E.1 FEASIBLE TRIPLES FOR HIGHLY VARIABLE GRID

| Time (s) | Triple chosen | Other feasible triples |
|---|---|---|
| 0 | (1, 11, 13725) | (1, 12, 10980), (1, 13, 8235), (2, 2, 0), (3, 1, 0) |
| 2745 | (1, 12, 10980) | (1, 13, 8235), (2, 2, 0), (2, 3, 0), (3, 1, 0) |
| 5490 | (1, 12, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 8235 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 10980 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 13725 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 16470 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 19215 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 21960 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 24705 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 27450 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 30195 | (2, 2, 2745) | (2, 3, 0), (3, 1, 0) |
| 32940 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 35685 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 38430 | (1, 13, 10980) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 41175 | (1, 12, 13725) | (1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 43920 | (1, 13, 10980) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 46665 | (2, 2, 2745) | (2, 3, 0), (3, 1, 0) |
| 49410 | (2, 2, 2745) | (2, 3, 0), (3, 1, 0) |
| 52155 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 54900 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 57645 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 60390 | (1, 12, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 63135 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 65880 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 68625 | (2, 2, 2745) | (2, 3, 0), (3, 1, 0) |
| 71370 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 74115 | (1, 12, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 76860 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 79605 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 82350 | (1, 12, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 85095 | (1, 12, 13725) | (1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 87840 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 90585 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 93330 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 96075 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 98820 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 101565 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 104310 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 107055 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 109800 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 112545 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 115290 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 118035 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 120780 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 123525 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |

*Continued on next page*

*Continued from previous page*

| Time (s) | Triple chosen | Other feasible triples |
|---|---|---|
| 126270 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 129015 | (2, 2, 2745) | (2, 3, 0), (3, 1, 0) |
| 131760 | (2, 2, 2745) | (2, 3, 0), (3, 1, 0) |
| 134505 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 137250 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 139995 | (2, 2, 2745) | (2, 3, 0), (3, 1, 0) |
| 142740 | (2, 2, 2745) | (2, 3, 0), (3, 1, 0) |
| 145485 | (1, 12, 16470) | (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 148230 | (2, 2, 2745) | (2, 3, 0), (3, 1, 0) |
| 150975 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 153720 | (1, 12, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 156465 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 159210 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 161955 | (1, 13, 16470) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |
| 164700 | (1, 13, 13725) | (2, 2, 2745), (2, 3, 0), (3, 1, 0) |

1766

List. E.8 shows the corresponding LaTeX code.

Listing E.8: Sample LaTeX code for making typical table environment

```
\begin{center}
{\scriptsize
\begin{tabularx}{\textwidth}{p{0.1\textwidth}|p{0.2\textwidth}|p{0.5\textwidth}}
\caption{Feasible triples for highly variable grid} \label{tab:triple_grid} \\
\hline
\hline
\textbf{Time (s)} &
\textbf{Triple chosen} &
\textbf{Other feasible triples} \\
\hline
\endfirsthead
\multicolumn{3}{c}%
{\textit{Continued from previous page}} \\
\hline
\hline
\textbf{Time (s)} &
\textbf{Triple chosen} &
\textbf{Other feasible triples} \\
\hline
\endhead
\hline
\multicolumn{3}{r}{\textit{Continued on next page}} \\
\endfoot
\hline
\endlastfoot
\hline

0 & (1, 11, 13725) & (1, 12, 10980), (1, 13, 8235), (2, 2, 0), (3, 1, 0) \\
2745 & (1, 12, 10980) & (1, 13, 8235), (2, 2, 0), (2, 3, 0), (3, 1, 0) \\
5490 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
8235 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
10980 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
13725 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
16470 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
19215 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
21960 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
24705 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
27450 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
30195 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
32940 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
35685 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
38430 & (1, 13, 10980) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
```

```
43 41175 & (1, 12, 13725) & (1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1,
       0) \\
44 43920 & (1, 13, 10980) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
45 46665 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
46 49410 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
47 52155 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1,
       0) \\
48 54900 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
49 57645 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
50 60390 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
51 63135 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
52 65880 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
53 68625 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
54 71370 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
55 74115 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
56 76860 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
57 79605 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
58 82350 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
59 85095 & (1, 12, 13725) & (1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1,
       0) \\
60 87840 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
61 90585 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
62 93330 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
63 96075 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
64 98820 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
65 101565 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
66 104310 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
67 107055 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
68 109800 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
69 112545 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3,
       1, 0) \\
70 115290 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
71 118035 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
72 120780 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
73 123525 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
74 126270 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3,
       1, 0) \\
75 129015 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
76 131760 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
77 134505 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
78 137250 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
79 139995 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
80 142740 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
81 145485 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3,
       1, 0) \\
82 148230 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
83 150975 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
84 153720 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
85 156465 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
86 159210 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
87 161955 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
88 164700 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
89 \end{tabularx}
90 }
91 \end{center}
```

De La Salle University

## E7 Algorithm or Pseudocode Listing

Table E.2 shows an example pseudocode. Note that if the pseudocode exceeds one page, it can mean that its implementation is not modular. List. E.9 shows the corresponding LaTeX code.

TABLE E.2 CALCULATION OF $y = x^n$

| | | |
|---|---|---|
| **Input(s):** | | |
| $n$ | : | $n$th power; $n \in \mathbb{Z}^+$ |
| $x$ | : | base value; $x \in \mathbb{R}^+$ |
| **Output(s):** | | |
| $y$ | : | result; $y \in \mathbb{R}^+$ |

**Require:** $n \geq 0 \vee x \neq 0$
**Ensure:** $y = x^n$
1: $y \Leftarrow 1$
2: **if** $n < 0$ **then**
3: $X \Leftarrow 1/x$
4: $N \Leftarrow -n$
5: **else**
6: $X \Leftarrow x$
7: $N \Leftarrow n$
8: **end if**
9: **while** $N \neq 0$ **do**
10: **if** $N$ is even **then**
11: $X \Leftarrow X \times X$
12: $N \Leftarrow N/2$
13: **else** $\{N$ is odd$\}$
14: $y \Leftarrow y \times X$
15: $N \Leftarrow N - 1$
16: **end if**
17: **end while**

Listing E.9: Sample LaTeX code for algorithm or pseudocode listing usage

```
\begin{table}[!htbp]
   \caption{Calculation of $y = x^n$}
   \label{tab:calcxn}
   {\footnotesize
   \begin{tabular}{lll}
   \hline
   \hline
   {\bfseries Input(s):} & & \\
   $n$ & : & $n$th power; $n \in \mathbb{Z}^{+}$ \\
   $x$ & : & base value; $x \in \mathbb{R}^{+}$ \\
   \hline
   {\bfseries Output(s):} & & \\
   $y$ & : & result; $y \in \mathbb{R}^{+}$  \\
   \hline
   \hline
   \\
   \end{tabular}
   }
   \begin{algorithmic}[1]
   {\footnotesize
      \REQUIRE $n \geq 0 \vee x \neq 0$
      \ENSURE $y = x^n$
      \STATE $y \Leftarrow 1$
      \IF{$n < 0$}
            \STATE $X \Leftarrow 1 / x$
            \STATE $N \Leftarrow -n$
      \ELSE
            \STATE $X \Leftarrow x$
            \STATE $N \Leftarrow n$
      \ENDIF
      \WHILE{$N \neq 0$}
            \IF{$N$ is even}
                  \STATE $X \Leftarrow X \times X$
                  \STATE $N \Leftarrow N / 2$
            \ELSE[$N$ is odd]
                  \STATE $y \Leftarrow y \times X$
                  \STATE $N \Leftarrow N - 1$
            \ENDIF
      \ENDWHILE
   }
   \end{algorithmic}
\end{table}
```

De La Salle University

## E8 Program/Code Listing

List. E.10 is a program listing of a C code for computing Fibonacci numbers by calling the actual code. Please see the `code` subdirectory.

Listing E.10: Computing Fibonacci numbers in C (./code/fibo.c)

```
/* fibo.c -- It prints out the first N Fibonacci
 *           numbers.
 */

#include <stdio.h>

int main(void) {
    int n;        /* Number of fibonacci numbers we will print */
    int i;        /* Index of fibonacci number to be printed next */
    int current;  /* Value of the (i)th fibonacci number */
    int next;     /* Value of the (i+1)th fibonacci number */
    int twoaway;  /* Value of the (i+2)th fibonacci number */

    printf("How many Fibonacci numbers do you want to compute? ");
    scanf("%d", &n);
    if (n<=0)
       printf("The number should be positive.\n");
    else {
      printf("\n\n\tI \t Fibonacci(I) \n\t=====================\n");
      next = current = 1;
      for (i=1; i<=n; i++) {
   printf("\t%d \t\t\t%d\n", i, current);
   twoaway = current+next;
   current = next;
   next    = twoaway;
      }
    }
}

/* The output from a run of this program was:

How many Fibonacci numbers do you want to compute? 9

   I   Fibonacci(I)
   =====================
   1      1
   2      1
   3      2
   4      3
   5      5
   6      8
   7      13
   8      21
   9      34

*/
```

List. E.11 shows the corresponding LaTeX code.

Listing E.11: Sample LaTeX code for program listing

```
List.~\ref{lst:fib_c} is a program listing of a C code for computing
    Fibonacci numbers by calling the actual code. Please see the \verb|
    code | subdirectory.
```

De La Salle University

## E9 Referencing

Referencing chapters: This appendix is in Appendix E, which is about examples in using various LaTeX commands.

Referencing sections: This section is Sec. E9, which shows how to refer to the locations of various labels that have been placed in the LaTeX files. List. E.12 shows the corresponding LaTeX code.

Listing E.12: Sample LaTeX code for referencing sections

```
Referencing sections: This section is Sec.~\ref{sec:ref}, which shows
    how to refer to the locations of various labels that have been
    placed in the \LaTeX \ files. List.~\ref{lst:refsec} shows the
    corresponding \LaTeX \ code.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

De La Salle University

## E9.1 A subsection

Referencing subsections: This section is Sec. E9.1, which shows how to refer to a subsection. List. E.13 shows the corresponding LaTeX code.

Listing E.13: Sample LaTeX code for referencing subsections

```
Referencing subsections: This section is Sec.~\ref{sec:subsec}, which
    shows how to refer to a subsection. List.~\ref{lst:refsub} shows the
     corresponding \LaTeX \ code.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

### E9.1.1 A sub-subsection

Referencing sub-subsections: This section is Sec. E9.1.1, which shows how to refer to a sub-subsection. List. E.14 shows the corresponding LaTeX code.

Listing E.14: Sample LaTeX code for referencing sub-subsections

```
Referencing sub-subsections: This section is Sec.~\ref{sec:subsubsec},
    which shows how to refer to a sub-subsection. List.~\ref{lst:
    refsubsub} shows the corresponding \LaTeX \ code.
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

# E10 Citing

Citing bibliography content is done using BibTeX. It requires the creation of a BibTeX file (.bib extension name), and then added in the argument of `\bibliography{ }` . For each .bib file, separate them by a comma in the argument of `\bibliography{ }` without the extension name. Building your BibTeX file (references.bib) can be done easily with a tool called JabRef (www.jabref.org).

The following subsections are examples of citations.

## E10.1 Books

- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]

De La Salle University

1951 • [?]

1952 • [?]

1953 • [?]

1954 • [?]

1955 • [?]

1956 • [?]

1957 • [?]

1958 • [?]

1959 • [?]

1960 • [?]

1961 • [?]

1962 • [?]

1963 • [?]

1964 • [?]

1965 • [?]

1966 • [?]

1967 • [?]

1968 • [?]

1969 • [?]

1970 • [?]

1971 • [?]

1972 • [?]

1973 • [?]

1974 • [?]

1975 • [?]

1976 • [?]

## E10.2 Booklets

- [?]

## E10.3 Proceedings

- [?]

## E10.4 In books

- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]

- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]

## E10.5 In proceedings

- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]

## E10.6 Journals

- [?]
- [?]
- [?]
- [?]
- [?]
- [?]

2023 • [?]

2024 • [?]

2025 • [?]

2026 • [?]

2027 • [?]

2028 • [?]

2029 • [?]

2030 • [?]

2031 • [?]

2032 • [?]

2033 • [?]

2034 • [?]

2035 • [?]

2036 • [?]

2037 • [?]

2038 • [?]

2039 • [?]

2040 • [?]

2041 • [?]

2042 • [?]

2043 • [?]

2044 • [?]

2045 • [?]

2046 • [?]

- [?]
- [?]
- [?]
- [?]
- [?]

## E10.7 Theses/dissertations

- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]

## E10.8 Technical Reports and Others

- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]

- [?]
- [?]
- [?]
- [?]
- [?]
- [?]

## E10.9 Miscellaneous

- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]
- [?]

## E11 Index

For key words or topics that are expected (or the user would like) to appear in the Index, use `index{key}` , where `key` is an example keyword to appear in the Index. For example, Fredholm integral and Fourier operator of the following paragraph are in the Index.

If we make a very large matrix with complex exponentials in the rows (i.e., cosine real parts and sine imaginary parts), and increase the resolution without bound, we approach the kernel of the Fredholm integral equation of the 2nd kind, namely the Fourier operator that defines the continuous Fourier transform.

List. E.15 is a program listing of the above-mentioned paragraph.

Listing E.15: Sample LaTeX code for Index usage

```
1 If we make a very large matrix with complex exponentials in the rows (i.
    e., cosine real parts and sine imaginary parts), and increase the
    resolution without bound, we approach the kernel of the \index{
    Fredholm integral} Fredholm integral equation of the 2nd kind,
    namely the \index{Fourier} Fourier operator that defines the
    continuous Fourier transform.
```

## E12 Adding Relevant PDF Pages

Examples of such PDF pages are Standards, Datasheets, Specification Sheets, Application Notes, etc. Selected PDF pages can be added (see List. E.16), but note that the options must be tweaked. See the manual of `pdfpages` for other options.

Listing E.16: Sample LaTeX code for including PDF pages

```
\includepdf[pages={8-10},%
offset=3.5mm -10mm,%
scale=0.73,%
frame,%
pagecommand={},]
{./reference/Xilinx2015-UltraScale-Architecture-Overview.pdf}
```

# Virtex UltraScale FPGA Feature Summary

*Table 6:* **Virtex UltraScale FPGA Feature Summary**

| | VU065 | VU080 | VU095 | VU125 | VU160 | VU190 | VU440 |
|---|---|---|---|---|---|---|---|
| Logic Cells | 626,640 | 780,000 | 940,800 | 1,253,280 | 1,621,200 | 1,879,920 | 4,432,680 |
| CLB Flip-Flops | 716,160 | 891,424 | 1,075,200 | 1,432,320 | 1,852,800 | 2,148,480 | 5,065,920 |
| CLB LUTs | 358,080 | 445,712 | 537,600 | 716,160 | 926,400 | 1,074,240 | 2,532,960 |
| Maximum Distributed RAM (Mb) | 4.8 | 3.9 | 4.8 | 9.7 | 12.7 | 14.5 | 28.7 |
| Block RAM/FIFO w/ECC (36Kb each) | 1,260 | 1,421 | 1,728 | 2,520 | 3,276 | 3,780 | 2,520 |
| Total Block RAM (Mb) | 44.3 | 50.0 | 60.8 | 88.6 | 115.2 | 132.9 | 88.6 |
| CMT (1 MMCM, 2 PLLs) | 10 | 16 | 16 | 20 | 30 | 30 | 30 |
| I/O DLLs | 40 | 64 | 64 | 80 | 120 | 120 | 120 |
| Fractional PLLs | 5 | 8 | 8 | 10 | 15 | 15 | 0 |
| Maximum HP I/Os(1) | 468 | 780 | 780 | 780 | 650 | 650 | 1,404 |
| Maximum HR I/Os(2) | 52 | 52 | 52 | 104 | 52 | 52 | 52 |
| DSP Slices | 600 | 672 | 768 | 1,200 | 1,560 | 1,800 | 2,880 |
| System Monitor | 1 | 1 | 1 | 2 | 3 | 3 | 3 |
| PCIe Gen3 x8 | 2 | 4 | 4 | 4 | 5 | 6 | 6 |
| 150G Interlaken | 3 | 6 | 6 | 6 | 8 | 9 | 0 |
| 100G Ethernet | 3 | 4 | 4 | 6 | 9 | 9 | 3 |
| GTH 16.3Gb/s Transceivers | 20 | 32 | 32 | 40 | 52 | 60 | 48 |
| GTY 30.5Gb/s Transceivers | 20 | 32 | 32 | 40 | 52 | 60 | 0 |

**Notes:**
1. HP = High-performance I/O with support for I/O voltage from 1.0V to 1.8V.
2. HR = High-range I/O with support for I/O voltage from 1.2V to 3.3V.

## Virtex UltraScale Device-Package Combinations and Maximum I/Os

*Table 7:* **Virtex UltraScale Device-Package Combinations and Maximum I/Os**

| Package[1][2][3] | Package Dimensions (mm) | VU065 | VU080 | VU095 | VU125 | VU160 | VU190 | VU440 |
|---|---|---|---|---|---|---|---|---|
| | | HR, HP GTH, GTY | HR, HP GTH, GTY | HR, HP GTH, GTY | HR, HP GTH, GTY | HR, HP GTH, GTY | HR, HP GTH, GTY | HR, HP GTH, GTY |
| FFVC1517 | 40x40 | 52, 468 20, 20 | 52, 468 20, 20 | 52, 468 20, 20 | | | | |
| FFVD1517 | 40x40 | | 52, 286 32, 32 | 52, 286 32, 32 | | | | |
| FLVD1517 | 40x40 | | | | 52, 286 40, 32 | | | |
| FFVB1760 | 42.5x42.5 | | 52, 650 32, 16 | 52, 650 32, 16 | | | | |
| FLVB1760 | 42.5x42.5 | | | | 52, 650 36, 16 | | | |
| FFVA2104 | 47.5x47.5 | | 52, 780 28, 24 | 52, 780 28, 24 | | | | |
| FLVA2104 | 47.5x47.5 | | | | 52, 780 28, 24 | | | |
| FFVB2104 | 47.5x47.5 | | 52, 650 32, 32 | 52, 650 32, 32 | | | | |
| FLVB2104 | 47.5x47.5 | | | | 52, 650 40, 36 | | | |
| FLGB2104 | 47.5x47.5 | | | | | 52, 650 40, 36 | 52, 650 40, 36 | |
| FFVC2104 | 47.5x47.5 | | | 52, 364 32, 32 | | | | |
| FLVC2104 | 47.5x47.5 | | | | 52, 364 40, 40 | | | |
| FLGC2104 | 47.5x47.5 | | | | | 52, 364 52, 52 | 52, 364 52, 52 | |
| FLGB2377 | 50x50 | | | | | | | 52, 1248 36, 0 |
| FLGA2577 | 52.5x52.5 | | | | | | 0, 448 60, 60 | |
| FLGA2892 | 55x55 | | | | | | | 52, 1404 48, 0 |

**Notes:**

1. Go to Ordering Information for package designation details.
2. All packages have 1.0mm ball pitch.
3. Packages with the same last letter and number sequence, e.g., A2104, are footprint compatible with all other UltraScale architecture-based devices with the same sequence. The footprint compatible devices within this family are outlined. See the UltraScale Architecture Product Selection Guide for details on inter-family migration.

# Virtex UltraScale+ FPGA Feature Summary

*Table 8:* **Virtex UltraScale+ FPGA Feature Summary**

| | VU3P | VU5P | VU7P | VU9P | VU11P | VU13P |
|---|---|---|---|---|---|---|
| Logic Cells | 689,640 | 1,051,010 | 1,379,280 | 2,068,920 | 2,147,040 | 2,862,720 |
| CLB Flip-Flops | 788,160 | 1,201,154 | 1,576,320 | 2,364,480 | 2,453,760 | 3,271,680 |
| CLB LUTs | 394,080 | 600,577 | 788,160 | 1,182,240 | 1,226,880 | 1,635,840 |
| Max. Distributed RAM (Mb) | 12.0 | 18.3 | 24.1 | 36.1 | 34.8 | 46.4 |
| Block RAM/FIFO w/ECC (36Kb each) | 720 | 1,024 | 1,440 | 2,160 | 2,016 | 2,688 |
| Block RAM (Mb) | 25.3 | 36.0 | 50.6 | 75.9 | 70.9 | 94.5 |
| UltraRAM Blocks | 320 | 470 | 640 | 960 | 1,152 | 1,536 |
| UltraRAM (Mb) | 90.0 | 132.2 | 180.0 | 270.0 | 324.0 | 432.0 |
| CMTs (1 MMCM and 2 PLLs) | 10 | 20 | 20 | 30 | 12 | 16 |
| Max. HP I/O(1) | 520 | 832 | 832 | 832 | 624 | 832 |
| DSP Slices | 2,280 | 3,474 | 4,560 | 6,840 | 8,928 | 11,904 |
| System Monitor | 1 | 2 | 2 | 3 | 3 | 4 |
| GTY Transceivers 32.75Gb/s | 40 | 80 | 80 | 120 | 96 | 128 |
| PCIe Gen3 x16 and Gen4 x8 | 2 | 4 | 4 | 6 | 3 | 4 |
| 150G Interlaken | 3 | 4 | 6 | 9 | 9 | 12 |
| 100G Ethernet w/RS-FEC | 3 | 4 | 6 | 9 | 6 | 8 |

**Notes:**
1. HP = High-performance I/O with support for I/O voltage from 1.0V to 1.8V.

## Virtex UltraScale+ Device-Package Combinations and Maximum I/Os

*Table 9:* **Virtex UltraScale+ Device-Package Combinations and Maximum I/Os**

| Package (1)(2)(3) | Package Dimensions (mm) | VU3P | VU5P | VU7P | VU9P | VU11P | VU13P |
|---|---|---|---|---|---|---|---|
| | | HP, GTY | HP, GTY | HP, GTY | HP, GTY | HP, GTY | HP, GTY |
| FFVC1517 | 40x40 | 520, 40 | | | | | |
| FLVF1924 | 45x45 | | | | | 624, 64 | |
| FLVA2104 | 47.5x47.5 | | 832, 52 | 832, 52 | 832, 52 | | |
| FHVA2104 | 52.5x52.5(4) | | | | | | 832, 52 |
| FLVB2104 | 47.5x47.5 | | 702, 76 | 702, 76 | 702, 76 | 624, 76 | |
| FHVB2104 | 52.5x52.5(4) | | | | | | 702, 76 |
| FLVC2104 | 47.5x47.5 | | 416, 80 | 416, 80 | 416, 104 | 416, 96 | |
| FHVC2104 | 52.5x52.5(4) | | | | | | 416, 104 |
| FLVA2577 | 52.5x52.5 | | | | 448, 120 | 448, 96 | 448, 128 |

**Notes:**
1. Go to Ordering Information for package designation details.
2. All packages have 1.0mm ball pitch.
3. Packages with the same last letter and number sequence, e.g., A2104, are footprint compatible with all other UltraScale devices with the same sequence. The footprint compatible devices within this family are outlined.
4. These 52.5x52.5mm overhang packages have the same PCB ball footprint as the corresponding 47.5x47.5mm packages (i.e., the same last letter and number sequence) and are footprint compatible.

# Appendix F
# VITA

John Carlo Theo S. Dela Cruz received the B.Sc., M.Sc., and Ph.D. degrees in chemistry all from the Pamantasan ng Pilipinas, San Juan, Metro Manila, Philippines, in 2020, 2022 and 2025 respectively. He is currently taking up his B.Sc. Computer Engineering studies. He has developed several high-speed packet-switched network systems and node modules. His research interests include high-speed packet-switched networks, high speed radio interface design, discrete simulation and statistical models for packet switches.

Pierre Justine P. Parel received the B.Sc., M.Sc., and Ph.D. degrees in chemistry all from the Pamantasan ng Pilipinas, San Juan, Metro Manila, Philippines, in 2020, 2022 and 2025 respectively. He is currently taking up his B.Sc. Computer Engineering studies. He has developed several high-speed packet-switched network systems and node modules. His research interests include high-speed packet-switched networks, high speed radio interface design, discrete simulation and statistical models for packet switches.

Jiro Renzo D. Tabiolo received the B.Sc., M.Sc., and Ph.D. degrees in chemistry all from the Pamantasan ng Pilipinas, San Juan, Metro Manila, Philippines, in 2020, 2022 and 2025 respectively. He is currently taking up his B.Sc. Computer Engineering studies. He has developed several high-speed packet-switched network systems

and node modules. His research interests include high-speed packet-switched networks, high speed radio interface design, discrete simulation and statistical models for packet switches.

Ercid Bon B. Valencerina received the B.Sc., M.Sc., and Ph.D. degrees in chemistry all from the Pamantasan ng Pilipinas, San Juan, Metro Manila, Philippines, in 2020, 2022 and 2025 respectively. He is currently taking up his B.Sc. Computer Engineering studies. He has developed several high-speed packet-switched network systems and node modules. His research interests include high-speed packet-switched networks, high speed radio interface design, discrete simulation and statistical models for packet switches.

# Appendix G
# ARTICLE PAPER(S)

# Article/Forum Paper Format (IEEE LaTeX format)

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

2138

***Abstract*—The abstract goes here.** Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

***Index Terms*—Computer Society, IEEE, IEEEtran, journal, LaTeX, paper, template.**

## I. Introduction

THIS demo file is intended to serve as a "starter file" for IEEE article papers produced under LaTeX using IEEEtran.cls version 1.8b and later. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

### A. Subsection Heading Here

Subsection text here. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin.

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332.
E-mail: see http://www.michaelshell.org/contact.html
J. Doe and J. Doe are with Anonymous University.

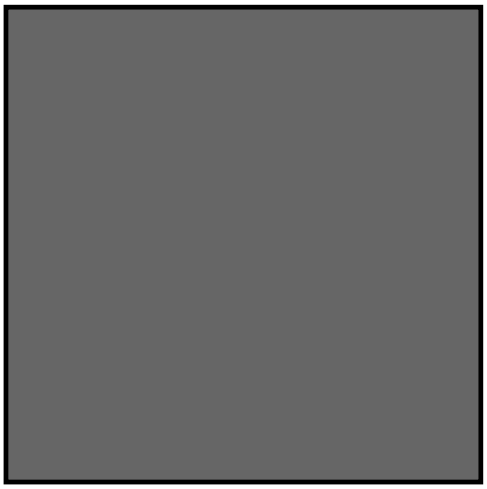


Fig. 1. Simulation results for the network.

TABLE I
AN EXAMPLE OF A TABLE

| One | Two |
|---|---|
| Three | Four |

Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

*1) Subsubsection Heading Here:* Subsubsection text here. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

## II. Conclusion

The conclusion goes here.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue,


(a) Case I (b) Case II

Fig. 2. Simulation results for the network.

a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

## APPENDIX A
## PROOF OF THE FIRST ZONKLAR EQUATION

Appendix one text goes here.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

## APPENDIX B

Appendix two text goes here. [1].

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

## ACKNOWLEDGMENT

The authors would like to thank...

## REFERENCES

[1] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl, *The Not So Short Introduction to LaTeX 2ε Or LaTeX 2ε in 157 minutes*. n.a., 2014.