



1

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

4

5 A Thesis
6 Presented to the Faculty of the
7 Department of Electronics and Computer Engineering
8 Gokongwei College of Engineering
9 De La Salle University

10

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

14

15 by

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

20 September, 2025



De La Salle University

21

ORAL DEFENSE RECOMMENDATION SHEET

22

23

24

25

26

27

28

29

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.

30

31

32

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**.

33

34

35

36

Dr. Melvin K. Cabatuan
Adviser

37

September 26, 2025



38

ABSTRACT

39 With the growing demand for coffee, the Philippine coffee industry faces the challenge of
40 inefficient and utilizing traditional manual sorting methods which hinders quality consistency
41 and competitiveness. This study proposes to develop a two-stage automated green
42 coffee bean (GCB) sorter designed to identify and segregate defective beans from the batch,
43 and the good classification would proceed to the second stage which is the density-analysis
44 stage wherein it sorts the dense and less-dense beans. The system integrates machine vision
45 that utilizes the RF-DETR model for detection and uses YOLOv12 and Vision Transformer
46 (ViT) models for classification. YOLOv12 achieved an average accuracy of 94.8% across
47 all trials with varying test conditions, while the ViT model achieved an accuracy of 98%,
48 which was similar to its per-classification accuracy. Overall, the ViT model was superior
49 across all metrics in both training and testing phases. The density-analysis stage achieved
50 an average sorting accuracy of 90.89%, demonstrating its effectiveness in distinguishing
51 dense beans from less-dense ones with high reliability. Overall, this automated sorter pro-
52 vides a reliable alternative to manual processing, which significantly reduces human labor
53 and improves GCB quality, thereby strengthening the local coffee farmers and industry's
54 foundation and competitiveness.

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



57 TABLE OF CONTENTS

58	Oral Defense Recommendation Sheet	ii
59	Abstract	iii
60	Table of Contents	iv
61	List of Figures	ix
62	List of Tables	xi
63	Abbreviations and Acronyms	xii
64	Notations	xiii
65	Glossary	xiv
66	Chapter 1 INTRODUCTION	1
67	1.1 Background of the Study	2
68	1.2 Prior Studies	3
69	1.3 Problem Statement	7
70	1.4 Objectives and Deliverables	8
71	1.4.1 General Objective (GO)	8
72	1.4.2 Specific Objectives (SOs)	8
73	1.4.3 Expected Deliverables	9
74	1.5 Significance of the Study	11
75	1.5.1 Technical Benefit	11
76	1.5.2 Impact to the Coffee Industry	11
77	1.6 Assumptions, Scope, and Delimitations	12
78	1.6.1 Assumptions	12
79	1.6.2 Scope	12
80	1.6.3 Delimitations	13
81	1.6.4 Overview of the Methodology	13
82	1.6.5 Estimated Work Schedule & Budget	15
83	Chapter 2 LITERATURE REVIEW	17
84	2.1 Existing Work	18



85	2.2 Lacking in the Approaches	27
86	2.3 Summary	27
87	Chapter 3 THEORETICAL CONSIDERATIONS	29
88	3.1 Theoretical Framework	30
89	3.2 Conceptual Framework	30
90	3.3 Quality Assurance Theory	32
91	3.4 Artificial Intelligence Theory	33
92	3.5 Computer Vision Theory	34
93	3.6 Performance Evaluation	35
94	3.7 Existing Technologies and Approaches	36
95	3.8 Density Measurement	37
96	3.9 Summary	37
97	Chapter 4 DESIGN CONSIDERATIONS	39
98	4.1 Mechanical Design	40
99	4.1.1 Screw Feeder	40
100	4.1.2 Rotating Conveyor Table	41
101	4.1.3 Inspection Tray (1st Stage)	42
102	4.1.4 Density Sorter (2nd Stage)	43
103	4.2 Embedded Systems	43
104	4.2.1 Microcontroller	43
105	4.2.2 Sensors	45
106	4.2.3 Motor control	47
107	4.2.4 Operating Voltage	49
108	4.3 Computer Vision System	50
109	4.3.1 Image Processing	50
110	4.4 Serial Communication	51
111	4.5 Graphical User Interface (GUI)	51
112	4.6 Density Analysis	52
113	4.7 Technical Standards	53
114	4.7.1 Hardware	53
115	4.7.2 Software	54
116	4.7.3 Green Coffee Bean Sorting	54
117	Chapter 5 METHODOLOGY	56
118	5.1 Description of the System	59
119	5.2 Research Design	62
120	5.3 Dataset Collection	63
121	5.3.1 Dataset Collection and Model Training	64



122	5.3.2 Utilization of Open-Source Database	65
123	5.3.3 First Iteration of Dataset Collection	66
124	5.3.4 Second Iteration of Dataset Collection	68
125	5.4 Dataset Preprocessing	69
126	5.4.1 Dataset Splitting	69
127	5.4.2 Image Annotation	69
128	5.4.3 Dataset Augmentation Techniques	70
129	5.5 Density Analysis	70
130	5.5.1 Density Threshold Calibration Using Water Displacement Method	70
131	5.5.2 Density Threshold Calibration	71
132	5.6 Model Training	71
133	5.6.1 Image Detection Models	71
134	5.6.2 Image Classification Models	72
135	5.6.3 Bean Classification Logic	73
136	5.6.4 Model Evaluation	74
137	5.6.5 Model Benchmarking and Selection	76
138	5.7 Hardware Development	76
139	5.7.1 Screw Feeder	77
140	5.7.2 Rotating Conveyor Table	78
141	5.7.3 Stage 1: Defect Sorting (Machine Vision and Inspection Tray) . .	84
142	5.7.4 Stage 2: Density Sorting (Precision Scale and Servo Mechanism) .	85
143	5.8 Hardware and Software Integration	87
144	5.8.1 Serial Communication	87
145	5.8.2 Recommended Standard 232 (RS-232)	89
146	5.9 Prototype Setup	90
147	5.9.1 Actual Setup	90
148	5.9.2 Lighting Setup for Inspection Tray	92
149	5.9.3 System Operation	96
150	5.10 Prototype Testing	98
151	5.10.1 Sorting Speed	98
152	5.10.2 Defect Sorting Accuracy	99
153	5.10.3 Density Sorting Accuracy	102
154	Chapter 6 RESULTS AND DISCUSSIONS	104
155	6.1 Description of the New Custom Dataset	108
156	6.2 Performance of Classification Models on Custom Dataset	109
157	6.2.1 EfficientNetV2S	110
158	6.2.2 YOLOv8	112
159	6.2.3 YOLOv11-cls	114



160	6.2.4 YOLOv12-cls	116
161	6.2.5 Vision Transformer (ViT)	118
162	6.3 Actual Performance of Classification Models in the System	119
163	6.4 Sorting Speed	126
164	Chapter 7 CONCLUSIONS, RECOMMENDATIONS, AND FUTURE DIRECTIVES	128
166	7.1 Concluding Remarks	129
167	7.2 Contributions	129
168	7.3 Recommendations	130
169	7.4 Future Prospects	130
170	References	131
171	Appendix A STUDENT RESEARCH ETHICS CLEARANCE	135
172	Appendix B ANSWERS TO QUESTIONS TO THIS THESIS	137
173	Appendix C REVISIONS TO THE PROPOSAL	140
174	Appendix D REVISIONS TO THE FINAL	146
175	Appendix E USAGE EXAMPLES	150
176	E1 Equations	151
177	E2 Notations	153
178	E2.1 Math alphabets	153
179	E2.2 Vector symbols	153
180	E2.3 Matrix symbols	153
181	E2.4 Tensor symbols	154
182	E2.5 Bold math version	155
183	E2.5.1 Vector symbols	155
184	E2.5.2 Matrix symbols	155
185	E2.5.3 Tensor symbols	155
186	E3 Abbreviation	159
187	E4 Glossary	161
188	E5 Figure	163
189	E6 Table	169
190	E7 Algorithm or Pseudocode Listing	173
191	E8 Program/Code Listing	175
192	E9 Referencing	177
193	E9.1 A subsection	178



De La Salle University

194	E9.1.1	A sub-subsection	179
195	E10	Citing	180
196		E10.1 Books	180
197		E10.2 Booklets	182
198		E10.3 Proceedings	182
199		E10.4 In books	182
200		E10.5 In proceedings	183
201		E10.6 Journals	183
202		E10.7 Theses/dissertations	185
203		E10.8 Technical Reports and Others	185
204		E10.9 Miscellaneous	186
205	E11	Index	187
206	E12	Adding Relevant PDF Pages	188
207	Appendix F VITA		192
208	Appendix G ARTICLE PAPER(S)		194



209 LIST OF FIGURES

210	1.1	Gantt Chart for John Carlo Dela Cruz	15
211	1.2	Gantt Chart for Pierre Parel	16
212	1.3	Gantt Chart for Jiro Tabiolo	16
213	1.4	Gantt Chart for Ercid Bon Valencerina	16
214	3.1	Theoretical Framework	30
215	3.2	Conceptual Framework	31
216	4.1	Screw Feeder Diagram	40
217	4.2	Rotating Conveyor Table 3D Design, 32-inch Rotary Table Accumulator (RTA)	41
218	4.3	Inspector Tray 3D Design	42
219	4.4	Arduino UNO Microcontroller	43
220	4.5	Arduino Nano Microcontroller	44
221	4.6	Infrared Sensor	45
222	4.7	TOF10120	46
223	4.8	12V NEMA 17 Stepper Motor	47
224	4.9	6V DC Motor	48
225	4.10	TB6612FNG Motor Driver	48
226	4.11	12V Power Supply	49
227	4.12	MT3608 Step-Up Module	50
228	4.13	C920 Camera	50
229	4.14	Graphical User Interface	52
230	5.1	System Block Diagram	59
231	5.2	Schematic Diagram of the System	60
232	5.3	Design Overview of the System	61
233	5.4	Design and Development Research (DDR) Methodology	62
234	5.5	Data Collection Process	63
235	5.6	Manual Sorting Process	64
236	5.7	First Iteration of Data Collection Setup	66
237	5.8	Sample Images from the First Iteration of Dataset Collection	67
238	5.9	Sample Images from the Second Iteration of Dataset Collection	68
239	5.10	Screw Feeder 3D Design	77
240	5.11	Rotating Conveyor Table 3D Design	78
241	5.12	Rotating Conveyor Table with Aluminum Guides	79
242	5.13	Revised Rotating Conveyor Table	80



243	5.14 3D Printed Guides	81
244	5.15 IR Sensor Placement	82
245	5.16 Rotating Conveyor Table with IR Sensor	83
246	5.17 Inspection Tray 3D Design	84
247	5.18 Density Sorter Mechanism Design	85
248	5.19 Precision Scale	86
249	5.20 Serial Communication Flow for Stage 1 Classification	87
250	5.21 Precision Scale Integration with RS232 for Stage 2 Classification	89
251	5.22 Actual System Setup	90
252	5.23 First Iteration of Lighting Setup	93
253	5.24 Second Iteration of Lighting Setup	94
254	5.25 Final Iteration of Lighting Setup	95
255	5.26 Top and Bottom View of the Cameras	96
256	5.27 Per-Classification Test Dataset	100
257	5.28 Good vs. Defective Test Dataset	101
258	6.1 Normalized Confusion Matrix for EfficientNetV2S on Test Dataset	110
259	6.2 Normalized Confusion Matrix for YOLOv8 on Test Dataset	112
260	6.3 Normalized Confusion Matrix for YOLOv11 on Test Dataset	114
261	6.4 Normalized Confusion Matrix for YOLOv12 on Test Dataset	116
262	6.5 Normalized Confusion Matrix for ViT on Test Dataset	118
263	6.6 Actual Performance of YOLOv12 in the System (Per-Classification)	122
264	6.7 Actual Performance of ViT in the System (Per-Classification)	124
265	6.8 Actual Density Sorting Performance	127
266	E.1 A quadrilateral image example.	163
267	E.2 Figures on top of each other. See List. E.6 for the corresponding L ^A T _E X code.	165
268	E.3 Four figures in each corner. See List. E.7 for the corresponding L ^A T _E X code. .	167



269 LIST OF TABLES

270	1.1	Summary of the Literature Review	4
271	1.2	Comparison Table on Existing Studies	6
272	1.3	Expected Deliverables per Objective	10
273	1.4	Budget Plan	15
274	2.1	Review of Related Literature	18
275	2.2	Comparing Proposed Study and Existing Studies	27
276	5.1	Summary of methods for reaching the objectives	57
277	5.2	Classification Algorithm for Coffee Beans	74
278	5.3	Sorting Speed Testing Table	98
279	5.4	Dataset Distribution for Overall Testing	102
280	6.1	Summary of results for achieving the objectives	105
281	6.2	Class Distribution Summary	108
282	6.3	Dataset Split Summary	108
283	6.4	Model Performance Comparison	119
284	6.5	Actual Performance of YOLOv12 in the System (Good vs. Defect)	125
285	6.7	Actual Performance of ViT in the System (Good vs. Defect)	125
286	6.9	Sorting Speed Test Conditions	126
287	C.1	Summary of Revisions to the Proposal	141
288	D.1	Summary of Revisions to the Thesis	147
289	E.1	Feasible triples for highly variable grid	169
290	E.2	Calculation of $y = x^n$	173



291

ABBREVIATIONS

292	AC	Alternating Current	159
293	HTML	Hyper-text Markup Language	159
294	CSS	Cascading Style Sheet	159
295	XML	eXtensible Markup Language	159



296 NOTATION

297	\mathcal{S}	a collection of distinct objects	161
298	\mathcal{U}	the set containing everything	161
299	\emptyset	the set with no elements.....	161
300	$ \mathcal{S} $	the number of elements in the set \mathcal{S}	161
301	$h(t)$	impulse response	151
302	$x(t)$	input signal represented in the time domain	151
303	$y(t)$	output signal represented in the time domain	151

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



310 **GLOSSARY**

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



313

LISTINGS

314	E.1 Sample L ^A T _E X code for equations and notations usage	152
315	E.2 Sample L ^A T _E X code for notations usage	156
316	E.3 Sample L ^A T _E X code for abbreviations usage	160
317	E.4 Sample L ^A T _E X code for glossary and notations usage	162
318	E.5 Sample L ^A T _E X code for a single figure	164
319	E.6 Sample L ^A T _E X code for three figures on top of each other	166
320	E.7 Sample L ^A T _E X code for the four figures	168
321	E.8 Sample L ^A T _E X code for making typical table environment	171
322	E.9 Sample L ^A T _E X code for algorithm or pseudocode listing usage	174
323	E.10 Computing Fibonacci numbers	175
324	E.11 Sample L ^A T _E X code for program listing	176
325	E.12 Sample L ^A T _E X code for referencing sections	177
326	E.13 Sample L ^A T _E X code for referencing subsections	178
327	E.14 Sample L ^A T _E X code for referencing sub-subsections	179
328	E.15 Sample L ^A T _E X code for Index usage	187
329	E.16 Sample L ^A T _E X code for including PDF pages	188



De La Salle University

330

Chapter 1

331

INTRODUCTION



332 **1.1 Background of the Study**

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

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



355 among 3 categories: Category 1 includes the most severe issues such as foreign matter
356 and black beans, Category 2 includes less severe defects like broken beans, and Category
357 3 includes minor defects like slight discoloration. Determining the quality of the coffee
358 beans in relation to their defect values is based on quality standards and grading systems
359 such as SCAA protocols guidance or the Philippine National Standard on Green Coffee
360 Bean [Bureau of Agriculture and Fisheries Standards, 2012].

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

374 **1.2 Prior Studies**

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



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

TABLE 1.1 SUMMARY OF THE LITERATURE REVIEW

Existing Literature	Description
Defect Detection	<p>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.</p>

**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	<p>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.</p>
---	--

385

TABLE 1.2 COMPARISON TABLE ON EXISTING STUDIES

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



- | | | |
|---|--|---|
| <ul style="list-style-type: none"> • Defect sorting using RFDETR, YOLOv12, and Vision Transformer models. • Considers classification of 7 defect types. • The system considers density parameters to sort out less-dense beans. • The system includes a graphical user interface for farmers to visualize the cumulative data of the defects present in the batch. • The system also includes AI-generated recommendations on the possible interventions for the farmers based on the data gathered from the sorting system. | <ul style="list-style-type: none"> • Defect sorting using YOLOv8 • The study considered only 6 types of defects. | <ul style="list-style-type: none"> • Defect sorting using YOLOv2 and InceptionV3. • The study considered only 2 types of defects. |
|---|--|---|

386

1.3 Problem Statement

387

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

388

389

390

391

392



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

404 **1.4 Objectives and Deliverables**

405 **1.4.1 General Objective (GO)**

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

410 **1.4.2 Specific Objectives (SOs)**

- 411 • SO1: To gather and create a dataset consisting of 500 high-resolution images of
412 good Arabica green coffee beans and 200 high-resolution images per classification



- 413 of defective beans (Category 1 & Category 2).;
- 414 • SO2: To improve the synchronization between the machine vision system and the
415 embedded sorting mechanism, ensuring defect sorting of at least 20 beans per minute
416 for stage one, solving issues such as non-synchronization of the system.;
- 417 • SO3: To achieve an accuracy of at least 85% in classifying defective green coffee
418 beans using computer vision;
- 419 • SO4: To achieve an accuracy of at least 85% in filtering out less-dense green coffee
420 beans;

421 **1.4.3 Expected Deliverables**

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



TABLE 1.3 EXPECTED DELIVERABLES PER OBJECTIVE

Objectives	Expected Deliverables
GO: The study aims 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).	<ul style="list-style-type: none"> • Data Gathering • 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.	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Computer Vision Program • Sorting Mechanism
SO4: To achieve an accuracy of at least 85% in filtering out less-dense green coffee beans	<ul style="list-style-type: none"> • Density-based Analysis • Sorting Mechanism



424 **1.5 Significance of the Study**

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

432 **1.5.1 Technical Benefit**

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

441 **1.5.2 Impact to the Coffee Industry**

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



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

448 **1.6 Assumptions, Scope, and Delimitations**

449 **1.6.1 Assumptions**

- 450 1. There would be a defective coffee bean from the green coffee bean test batch;
- 451 2. Identifying the defective coffee beans using the machine vision and density-based
452 analysis would be much more efficient and accurate than manually sorting them;
- 453 3. During testing, test batches will contain 50% good beans and 50% defective beans,
454 60% good beans and 40% defective beans, 70% good beans and 30% defective beans,
455 80% good beans and 20% defective beans, 90% good beans and 10% defective beans,
456 100% good beans;

457 **1.6.2 Scope**

- 458 1. The study only focuses on Arabica green coffee beans;
- 459 2. The study has two stages, the first stage would segregate the defective green coffee
460 beans from the batch, then the second stage would identify the specialty-grade green
461 coffee beans depending on its density;
- 462 3. The study will only consider the following metrics for classification: Color, Shape,
463 Physical Characteristics, Density, and Texture.



464 4. The system is not designed to evaluate other coffee bean varieties or factors such
465 as moisture content, different coffee bean origins, bean size beyond the allowed
466 deviation, or post-roasting characteristics.

467 **1.6.3 Delimitations**

- 468 1. The batch of coffee beans to be used for testing and dataset collection will consist
469 solely of Arabica beans from the same origin, farmer, and processed in the same way;
- 470 2. The system is only limited to unroasted green coffee beans;
- 471 3. The batch of coffee beans to be used should only be dehulled and not sorted visually
472 and by density;
- 473 4. Since the system is considering several types of defects and density parameter, sorting
474 time is compromised;
- 475 5. The system is designed to perform individual scanning of each coffee bean;
- 476 6. Sizing will not be considered, as it may cause jamming in the system. However, a
477 maximum size deviation of 20% among beans will be allowed.
- 478 7. This study is limited to detecting only the most evident defect in a green coffee bean
479 when multiple defects are present. The system prioritizes the most prominent defect
480 based on visual characteristics and does not classify multiple defects per bean.

481 **1.6.4 Overview of the Methodology**

482 The rapid advancements in computer engineering have led to innovative solutions across
483 various domains, including artificial intelligence and embedded systems. This thesis



484 presents a novel approach to automated coffee bean sorting, addressing key challenges in
485 manual classification and proposing a two-stage system that integrates machine vision and
486 density-based analysis.

487 The study explores fundamental principles and technologies relevant to coffee bean
488 classification, including machine vision, deep learning, and density-based sorting. By
489 combining theoretical analysis with practical implementation, the proposed system aims to
490 enhance sorting accuracy, reduce human labor, and improve efficiency and quality output
491 in coffee production.

492 The study's objective is to create a dataset consisting of 500 high-resolution images of
493 Arabica specialty-grade green coffee beans from a certain origin, to ensure sufficient data
494 for training and validation of the algorithm in the machine vision system. The system will
495 capture real-time data on the visual and density characteristics of each green coffee beans,
496 to aid coffee farmers from sorting each coffee bean efficiently. In the succeeding chapters,
497 the study will delve into existing research on automated coffee bean sorting systems, with a
498 focus on machine vision, density analysis, and their applications in the coffee industry. The
499 study would explore gaps in current manual sorting practices and demonstrate the system's
500 ability to improve overall efficiency through implementing advanced technologies.



501

1.6.5 Estimated Work Schedule & Budget

TABLE 1.4 BUDGET PLAN

Item	Quantity	Price	Total
IP Camera	2	999	1,998
Arduino Nano	2	300	600
ToF10120	1	400	400
Acrylic Platform	1	67	67
12 NEMA Stepper Motor	1	405	405
6V DC Motor	1	150	150
TB6612FNG Motor Driver	1	50	50
MT3608 Step-Up Module	1	50	50
Lighting Equipment (LED Bar, Ring, Spotlight)	3	80	240
Precision Weighing Scale	1	2,000	2,000
HX711	1	69	69
Rotating Conveyor Table 3D Printing	1	5,000	5,000
Inspection Tray 3D Printing	1	4,700	4,700
Density Sorter 3D Printing	1	2,500	2,500
Screw Feeder 3D Printing	1	3,500	3,500
<i>Other Hardware Components/Wires</i>	<i>Approximate</i>	<i>Approximate</i>	<i>Approximate</i>
Total Estimate			21,879 + approximate additional hardware components

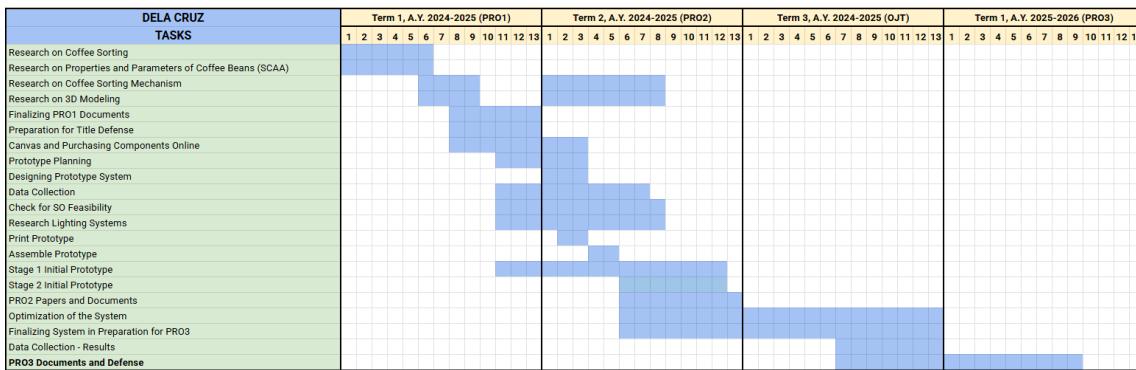


Fig. 1.1 Gantt Chart for John Carlo Dela Cruz

1. Introduction



De La Salle University

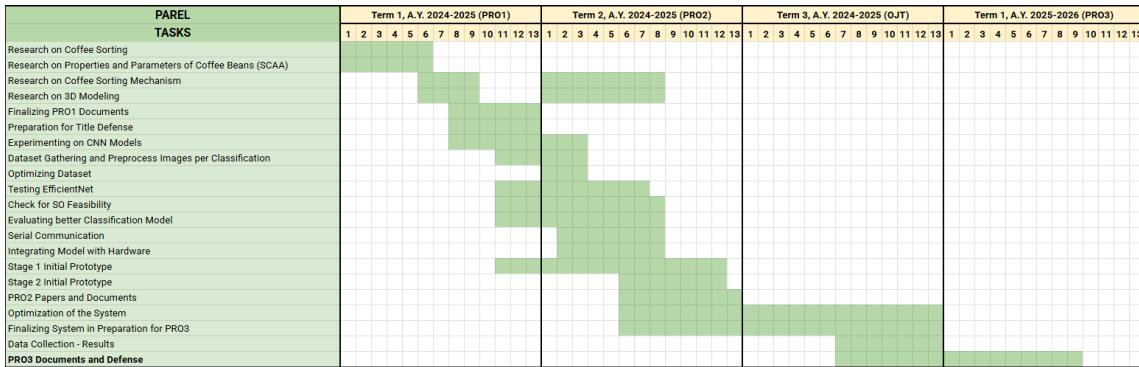


Fig. 1.2 Gantt Chart for Pierre Parel

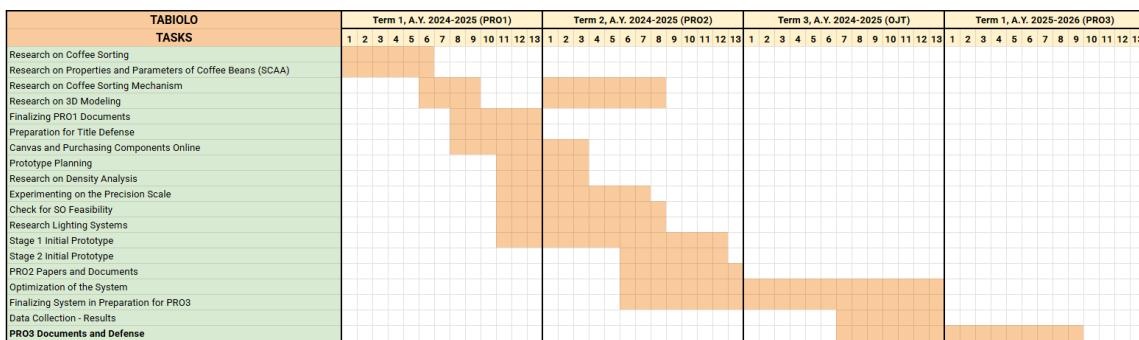


Fig. 1.3 Gantt Chart for Jiro Tabiolo

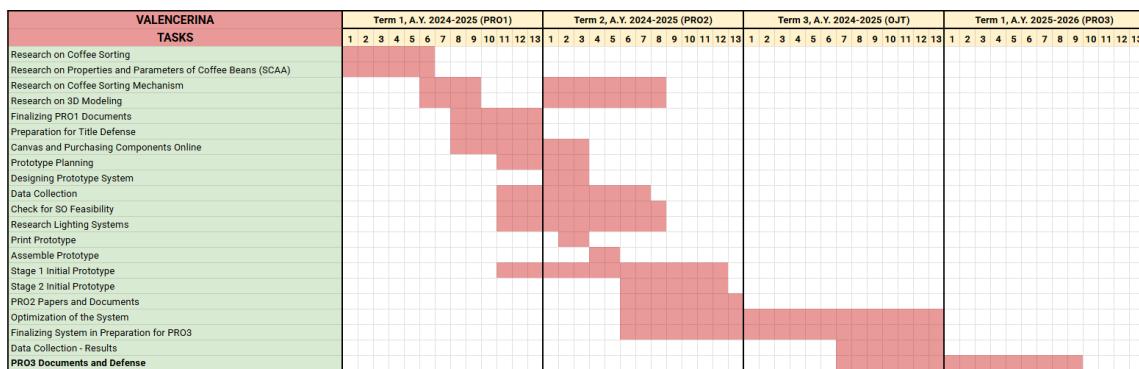


Fig. 1.4 Gantt Chart for Ercid Bon Valencerina



De La Salle University

502

Chapter 2

503

LITERATURE REVIEW



504

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.



[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.
[Muchtar et al., 2025]	This study focuses on defect detection in coffee beans using nine image detection models, consisting of five CNN-based models and four transformer-based models. The system is limited to identifying whether a bean is defective or not, without classifying the specific type of defect. Furthermore, the scope does not cover any sorting mechanism, as it only deals with defect identification.



505

506

2.2 Lacking in the Approaches

TABLE 2.2 COMPARING PROPOSED STUDY AND EXISTING STUDIES

Existing Studies	Proposed Study
<ul style="list-style-type: none"> Uses computer vision to classify green coffee bean grade based on its visual characteristics such as size, color, and shape. Most related studies classify defective and non-defective beans only. The density parameter of the green coffee beans is not considered. Similar study [Lualhati et al., 2022] only considered three classifications of GCBs: Good, Black, and Irregular-Shaped beans. Similar automated GCB sorter [Balay et al., 2024] only considered one side of the bean. Existing classification of GCBs with automated sorters do not have an integrated graphical user interface (GUI) for data analytics. 	<ul style="list-style-type: none"> Computer vision will be used to analyze the physical characteristics of the bean, including its volume. Density parameters will be considered by implementing a weighing scale to the system. The system will implement two stages of sorting: <ul style="list-style-type: none"> The first stage sorts out the defective beans. The second stage sorts out the potential specialty-grade beans based on their density and size. The system is designed to inspect both sides, utilizing two cameras. The system is designed with a GUI for farmers to visualize the cumulative data of the defects present in the batch.

507

2.3 Summary

508

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

509



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



524

Chapter 3

525

THEORETICAL CONSIDERATIONS



526

3.1 Theoretical Framework

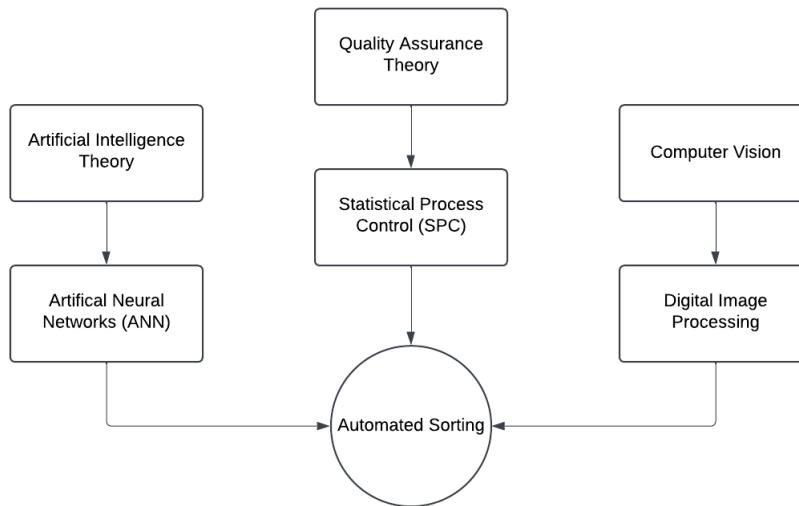


Fig. 3.1 Theoretical Framework

527

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.

531

3.2 Conceptual Framework

532

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

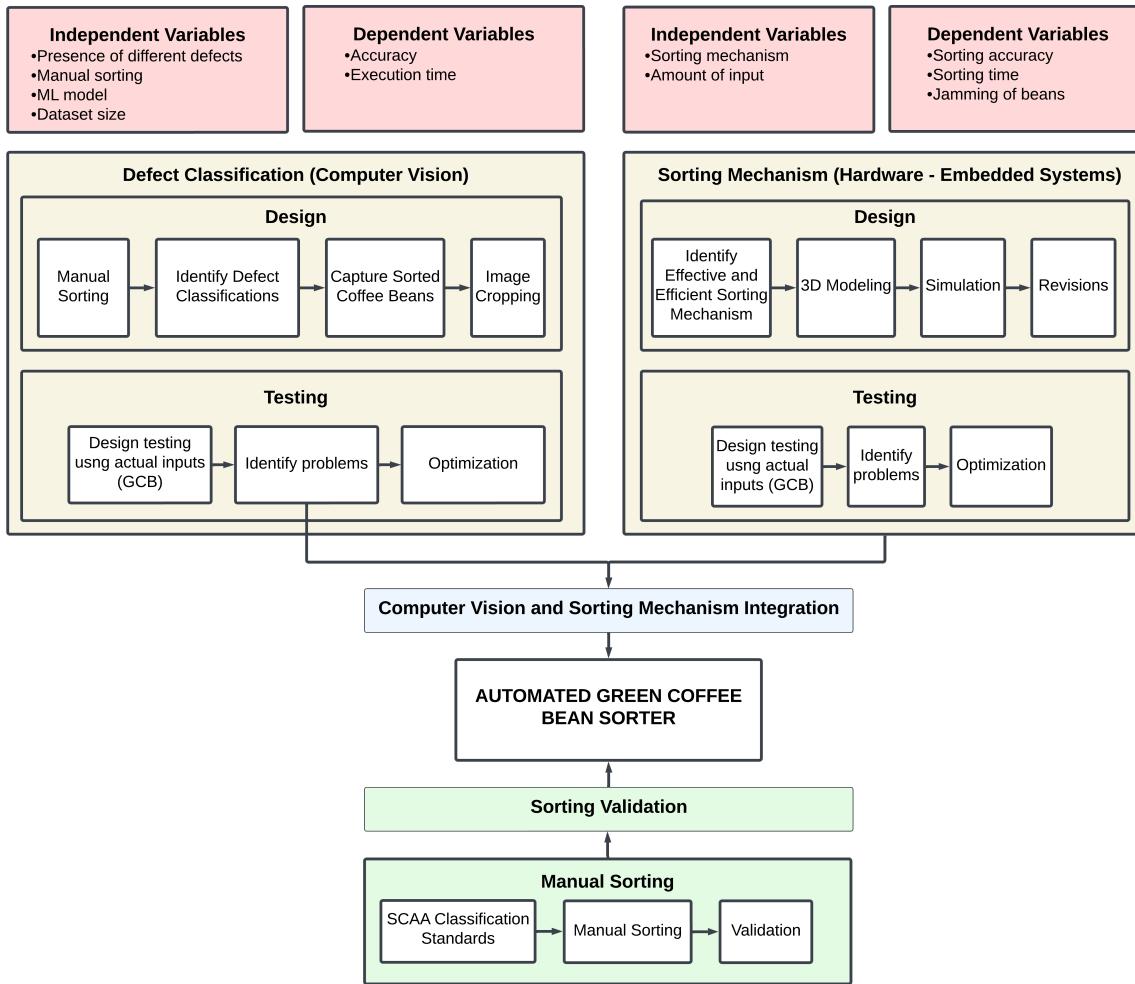


Fig. 3.2 Conceptual Framework

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



539 **3.3 Quality Assurance Theory**

540 Quality assurance theory refers to the set of principles and practices that focuses on estab-
541 lishing a systematic process to ensure that a product or service conforms to a predetermined
542 standard. In the aspect of food and agriculture, there are a number of practices and prin-
543 ciples that ensure the safety and quality of food products. According to [da Cruz et al.,
544 2006], there are a number of practices in place that must be followed, one of which is
545 Good Agricultural Practices, where these procedures are aimed to reduce hazards related to
546 product safety at the farm level. Another one of said practices is the Good manufacturing
547 practice, which were formerly called support programs that provide foundations to the
548 overall food safety management programme. This includes cleaning, maintenance, person-
549 nel training, calibration equipment, quality control, and pest control. Industries that adopt
550 such practices produce the following results, better quality products, greener initiatives
551 and better productivity within a department. Lastly, hazard analysis and critical control
552 points (HACCP), is a science-based system that was created to identify potential hazards
553 and actions to control said hazards. This practice is used to ensure food safety.

554 In the context of coffee beans, there are a number of systems in place to ensure that
555 quality beans are being provided to the consumer market. The governing body known as
556 the Specialty Coffee Association (SCAA) has implemented grades to green coffee beans
557 to provide a better way to classify said beans. These grades can be differentiated into 5
558 grades namely, Specialty Grade, Premium Coffee Grade, Exchange Coffee Grade, Below
559 Standard Coffee Grade, and Off grade Coffee. They are classified according to the number
560 of defects found in a sample batch of 300 grams and according to their size. Specialty
561 grade coffee beans are supposed to contain less than 5 defects in a sample batch while also



562 not allowing any primary defects to be present; it should only have less than 5% difference
563 between its sizes. Coffee beans in this grade should also contain a special attribute whether
564 in its body, flavor, aroma, or acidity, and its moisture content should only be in the range
565 of 9-13%. Premium Coffee grade beans should only contain 8 full defects in a sample
566 batch but primary defects are allowed in the sample batch. Similarly to specialty grade
567 coffee beans, its sizes should only contain a 5% difference to one another; it should also
568 contain a special attribute and moisture content should also be similar to its specialty grade
569 counterpart. Exchange coffee grade should contain defects ranging from 9-23 beans in a
570 sample batch, with sizes that can vary up to 50% difference in weight but also only 5% in
571 its sizes. Below standard and off grade coffee beans are classified according to the number
572 of defects present in a sample batch; 24-86 beans for below standard while more than 86
573 beans for off grade. These gradings are used to ensure that quality green coffee beans are
574 produced and ensure that consumers are provided with the best quality available.

575 **3.4 Artificial Intelligence Theory**

576 Artificial Intelligence in defect classification are widely used in this industry which are
577 commonly used in manufacturing and industrial applications. Several deep learning tech-
578 niques are used in order to achieve an effective defect classification. Models such as
579 convolutional neural networks (CNNs) and You Only Look Once (YOLO) are widely used
580 for classification. CNN utilizes an image based analysis and feature extraction approach
581 to identify different classifications. CNN is more effective in analyzing grid-like data like
582 images, making it suitable for defect classification [Das et al., 2019]. One of its major
583 advantages is its ability to automatically detect important features such as shape, patterns,



584 and edges. Although it may have its own advantages, there are also disadvantages that need
585 to be taken into account, mainly in scenarios that involve class imbalance and complex
586 backgrounds (Moon, 2021) . YOLO is another model that is suitable for defect classifica-
587 tion, its ability to provide real-time defect classification while also providing high accuracy
588 is essential in some industries. In YOLO, there are several versions that are developed over
589 the years, which are supposed to bring several improvements in terms of speed, accuracy,
590 and computational efficiency. Combining different models is also effective, in the case
591 of [Deepti and Prabadevi, 2024], they combined transformer architecture with YOLOv7
592 to enhance its feature extraction, this resulted in an increase of 5.4% in mean average
593 precision and F1 score.

594 **3.5 Computer Vision Theory**

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

602 For defect classification, feature extraction is important to identify the relevant features
603 then extracting said features to help indicate specific defects, this utilizes the edges,
604 textures, and shapes to help in defect classification [Wu et al., 2024]. By utilizing OpenCV
605 and deep learning models is advisable for automatic feature extraction. Models like CNN,



606 can automatically extract features from images, which greatly reduces the need for manual
607 extraction, this helps in a more robust and scalable solution [Bali and Tyagi, 2020]. The
608 versatility of OpenCV library which allows support for multiple image pre-processing tasks,
609 when combined with deep learning models can be applied to different fields.

610 **3.6 Performance Evaluation**

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



625 3.7 Existing Technologies and Approaches

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

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

640 The study of [Muchtar et al., 2025] explored the use of 9 image classification models
641 with varying architectures such as CNN-based (EfficientNetB7, DenseNet121, InceptionV4,
642 etc.) and Transformer-based models (ViT, Swin Transformer, FocalNet, etc.) in detecting
643 defective coffee beans. The study focuses on differentiating defective from good coffee
644 beans, but doesn't distinguish between defects. The results of the study suggest that
645 FocalNet outperforms all other models significantly in both training and testing phases for
646 detecting defects in coffee beans.



647 3.8 Density Measurement

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

$$d = \frac{m_2 - m_1}{V} \quad (3.1)$$

654 where m_2 is the mass of the green coffee bean, m_1 is the mass of the empty con-
 655 tainer, and V is the capacity (in liters) of the container [International Organization for
 656 Standardization, 1995].

657 3.9 Summary

658 This chapter gives the theoretical and conceptual backgrounds of an automated green
 659 coffee bean sorter using Artificial Intelligence (AI), Quality Assurance, and Computer
 660 Vision. The theoretical background focuses on key concepts like deep learning models
 661 (CNNs, YOLO, ViT) used for defect classification, quality assurance principles (GAP,
 662 GMP, HACCP) ensuring food safety, and computer vision algorithms (preprocessing,
 663 segmentation, and feature extraction) used for image analysis. The conceptual background
 664 explains the integration of machine vision for defect detection with embedded systems for
 665 sorting, thus conforming to the SCAA coffee grading standards. Performance metrics like



De La Salle University

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.



670

Chapter 4

671

DESIGN CONSIDERATIONS



672 **4.1 Mechanical Design**

673 **4.1.1 Screw Feeder**

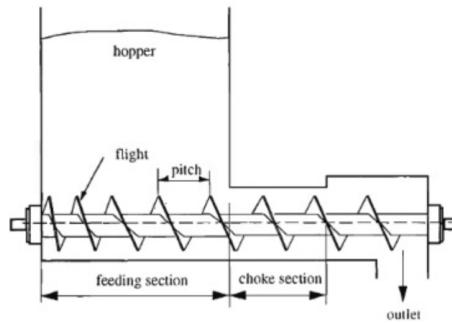


Fig. 4.1 Screw Feeder Diagram

674 Figure 4.1 shows the diagram of a screw feeder. Screw feeders are usually used in
675 industrial fields like agriculture, chemicals, plastics, cements, poultry and food processing.
676 According to [Minglani et al., 2020], screw feeders are specifically used to transport or
677 move granular materials at a controlled rate like corn and wheat. It consists of a rotating
678 screw and small feeding section or the hopper. Despite having big batches of a certain
679 material, screw feeders can control the rate of which these materials are dispensed. With
680 this concept, the group decided to utilize a screw feeder as the input mechanism for the
681 system. This mechanism allows a controlled rate of coffee bean dispensing, which is a
682 significant factor to avoid overcrowding in the rotating conveyor table causing the beans to
683 jam. In addition, batches of coffee beans can be put at once instead of just adding a certain
684 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 come out from the screw feeder, the coffee beans would then be placed in the rotating conveyor table. According to the study of [Dabek et al., 2022]. The conveyor table is used as a transportation system 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.

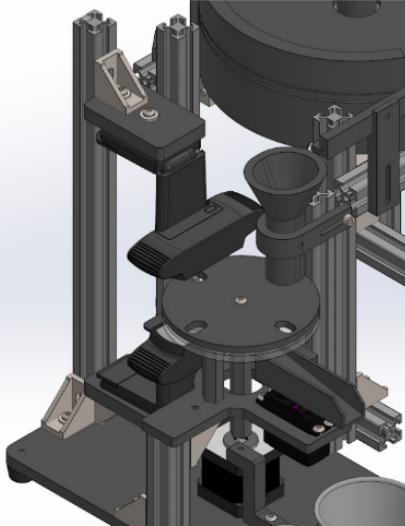


Fig. 4.3 Inspector Tray 3D Design

695 **4.1.3 Inspection Tray (1st Stage)**

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



705 **4.1.4 Density Sorter (2nd Stage)**

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

712 **4.2 Embedded Systems**

713 **4.2.1 Microcontroller**



Fig. 4.4 Arduino UNO Microcontroller



Fig. 4.5 Arduino Nano Microcontroller

714 Since the system is composed of two stages of sorting: defect sorting through computer
715 vision and density-based analysis—the group decided to utilize two Arduino Nano micro-
716 controllers to modularize the control process. The first Arduino Nano microcontroller is
717 tasked to handle the computer vision-based defect sorting through serial communication
718 with pyserial operating in Python. In addition, it handles the operation of defect sorting
719 consisting of a stepper motor for the rotation of the inspection tray and a servo motor for the
720 slider, which directs the beans to the designated bin (defect or good bin). On the other hand,
721 the second Arduino Nano microcontroller manages the density-based analysis and sorting,
722 which consists of another stepper motor to direct the beans to its respective bin (dense
723 and less-dense bin), the precision scale which is interfaced through RS232, and the top
724 feeder where the input beans are poured. The use of separate Arduino microcontrollers is
725 advantageous when it comes to the computer vision-based sorting of beans. This is because
726 serial communication is much faster when code complexity is significantly reduced. With
727 this, a designated microcontroller handles the computer vision part and two-way serial
728 communication between the microcontroller and the computer vision algorithm running in
729 Python. Most importantly, the use of two microcontrollers allowed the system to not rely
730 solely on a sequential approach. This means that the two stages of sorting are not relying



731 on the timing of each other, allowing the inspection tray and the top feeder to operate
732 independently. Thus, resulting in a much faster and efficient sorting process.

733 **4.2.2 Sensors**



Fig. 4.6 Infrared Sensor

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



Fig. 4.7 TOF10120

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



746

4.2.3 Motor control



Fig. 4.8 12V NEMA 17 Stepper Motor

747

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. 4.9 6V DC Motor

756 For the rotating conveyor table (top feeder), where the beans are initially poured, a
757 6V DC motor is used. The group decided to use this motor due to its high RPM, which
758 is needed for a fast rotation of the rotating conveyor table. The speed of the feeder is
759 regulated to prevent clogging and ensure that the beans are evenly spaced before they
760 enter the inspection tray. The motor speed is fine-tuned through pulse-width modulation
761 (PWM) to synchronize with the stepper motor-driven inspection tray, ensuring a steady
762 input without overwhelming the system.

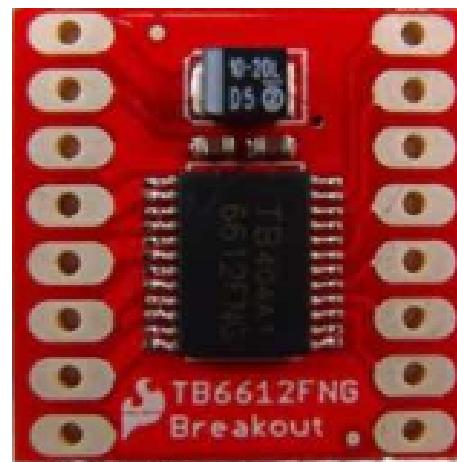


Fig. 4.10 TB6612FNG Motor Driver



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

766 **4.2.4 Operating Voltage**



Fig. 4.11 12V Power Supply

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



Fig. 4.12 MT3608 Step-Up Module

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

779 4.3 Computer Vision System

780 4.3.1 Image Processing



Fig. 4.13 C920 Camera



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

786 **4.4 Serial Communication**

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

794 **4.5 Graphical User Interface (GUI)**

795 The proposed system would be integrating a graphical user interface developed using PyGui
796 and ChatGPT API. The GUI would serve as the control center platform for the system. This
797 would provide real-time feedback and insights for users. As shown in Figure 8, a concept
798 of how the GUI would interact with the system would be a start button, once the button
799 is executed the system would then be expecting inputs and start sorting. There would be
800 real-time feedback during the sorting process, then some visual markers to indicate their



Fig. 4.14 Graphical User Interface

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.



815 **4.7 Technical Standards**

816 **4.7.1 Hardware**

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

824 On top of this, ISO 14121-1 – Risk Assessment for Machinery was also followed to
825 further assess the potential risks throughout the system. The standard includes identify-
826 ing and quantifying hazards such as electrical short circuits, faulty wirings, and motor
827 overheating [International Organization for Standardization, 2007]. With this, the system
828 included protective enclosures for the electrical wirings, proper grounding of the circuits,
829 and controlled motor actuation. More specifically, for motors, it was made sure that the
830 design has sufficient voltage and ampere to power the different kinds of motors used with
831 the use of L298N, and MT3608 modules. These are the main components for adjusting
832 motor speeds dynamically during the sorting process.

833 Lastly, ISO 30071-1 was standard used to provide sufficient lighting during data
834 collection, and real time bean inspection during sorting process. This standard helps ensure
835 consistent and non-glare lighting conditions, which are essential for the machine vision
836 cameras to accurately capture bean features [International Organization for Standardization,
837 2019]. Uniform illumination improves the reliability of image classification by reducing



838 shadow artifacts and reflections, thereby enhancing overall detection performance.

839 **4.7.2 Software**

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

847 Lastly, ISO/IEC 23053 – Framework for Artificial Intelligence (AI) offers a reference
848 architecture to build and integrate machine learning building blocks [International Organiza-
849 tion for Standardization, 2022]. This standard was highly applicable in determining the
850 design of the machine vision module, where a pre-trained deep learning model is utilized
851 for the classification of bean defects. This standard provides guidelines on best practice for
852 the overall machine learning cycle, ranging from data acquisition, feature extraction, and
853 model training through to model evaluation, deployment, and monitoring.

854 **4.7.3 Green Coffee Bean Sorting**

855 For sorting green coffee beans, Specialty Coffee Association of America (SCAA) Standards
856 for Green Coffee Bean Sorting was incorporated to maintain conformity. The standards
857 set the definition for the classification of primary and secondary defects (i.e., black, sour,
858 insect-damaged, broken, and floater beans) and sets the maximum allowable defect counts



De La Salle University

859 for specialty-grade coffee. The SCAA standards were applied to mark the training set of the
860 machine vision model and also to set up the thresholds of defect classification, so visually
861 defective beans can be correctly classified and rejected. Also, the sorting mechanism based
862 on density points towards SCAA bean weight and volume guidelines using a precision
863 scale and ToF sensor to sort beans based on within-acceptability density limits.

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



De La Salle University

873

Chapter 5

874

METHODOLOGY



TABLE 5.1 SUMMARY OF METHODS FOR REACHING THE OBJECTIVES

Objectives	Methods	Locations
GO: The study aims 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.	<ul style="list-style-type: none"> • DDR Methodology • Description of the System 	Sec. 5.1 on p. 59 Sec. 5.2 on p. 62
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).	<ul style="list-style-type: none"> • Dataset Collection • Manual Sorting 	Sec. 5.3 on p. 63

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.	<ul style="list-style-type: none"> • Data Collection • Dataset preprocessing • Model Training • Serial Communication 	Sec. 5.3 on p. 63 Sec. ?? on p. ??
Sec. 5.8.1 on p. 87 SO3: To achieve an accuracy of at least 85% in classifying defective green coffee beans using computer vision	<ul style="list-style-type: none"> • Dataset preprocessing • Model Training 	Sec. ?? on p. ??
SO4: To achieve an accuracy of at least 85% in filtering out less-dense green coffee beans	<ul style="list-style-type: none"> • Density Threshold Calibration Using Water Displacement Method • Density Sorter 	Sec. 5.5.1 on p. 70 Sec. 5.7.4 on p. 85



875

5.1 Description of the System

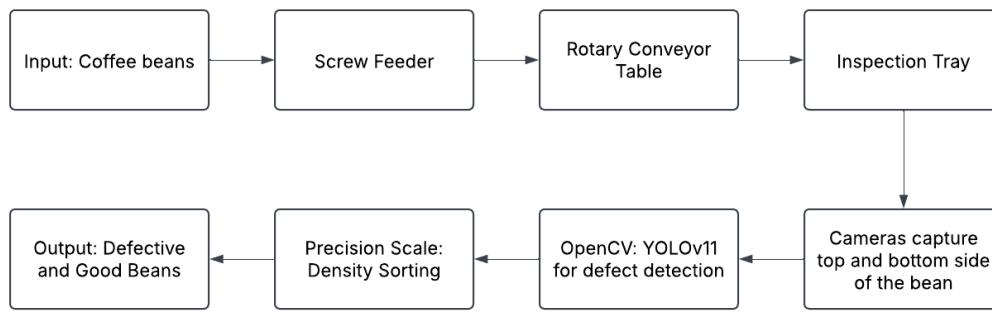


Fig. 5.1 System Block Diagram

876

The system is an automated green coffee bean sorting machine, utilizing machine vision. Firstly, the coffee beans are introduced into the system through a funnel, which directs them to a conveyor belt mechanism. The green coffee beans are sorted depending on their visual characteristics. In this process, the physical qualities of the bean are 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 a precision scale. With the volume and mass of the bean in hand, the density of the bean can be estimated. Depending on the density threshold set by the user, the bean will be classified whether it is good or not.

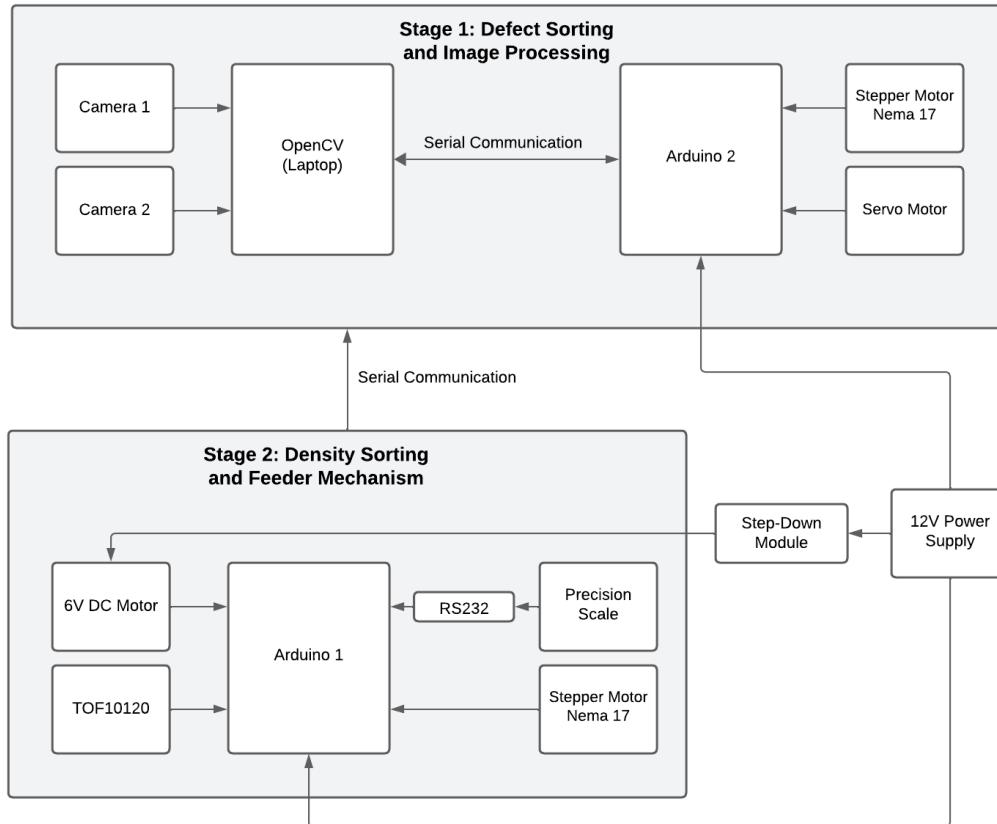


Fig. 5.2 Schematic Diagram of the System

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



893 communication, ensuring smooth coordination.

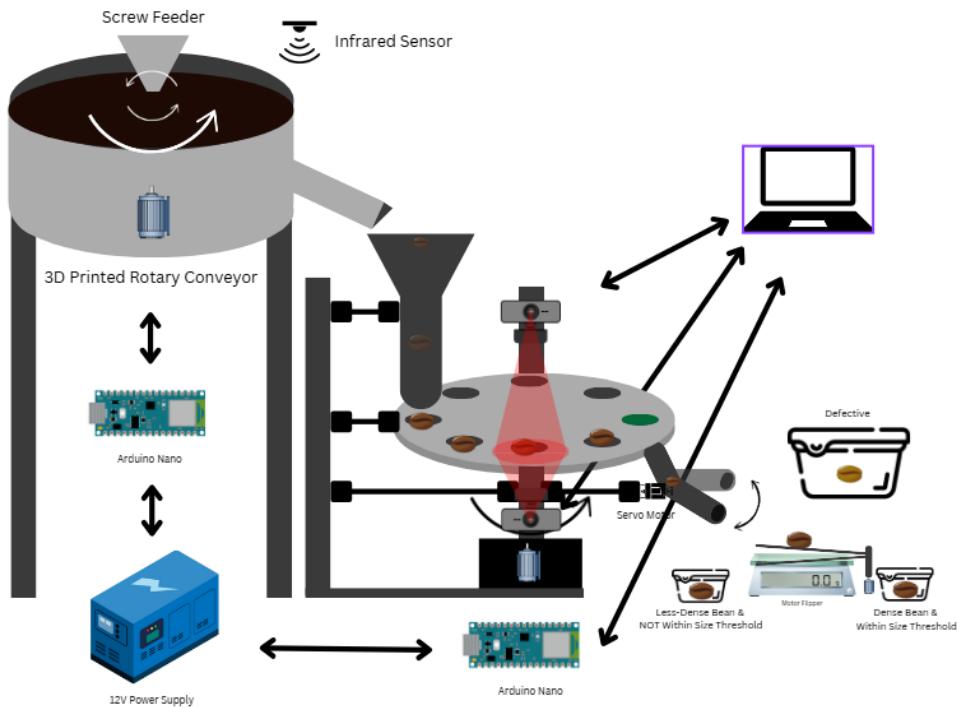


Fig. 5.3 Design Overview of the System

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



901

5.2 Research Design

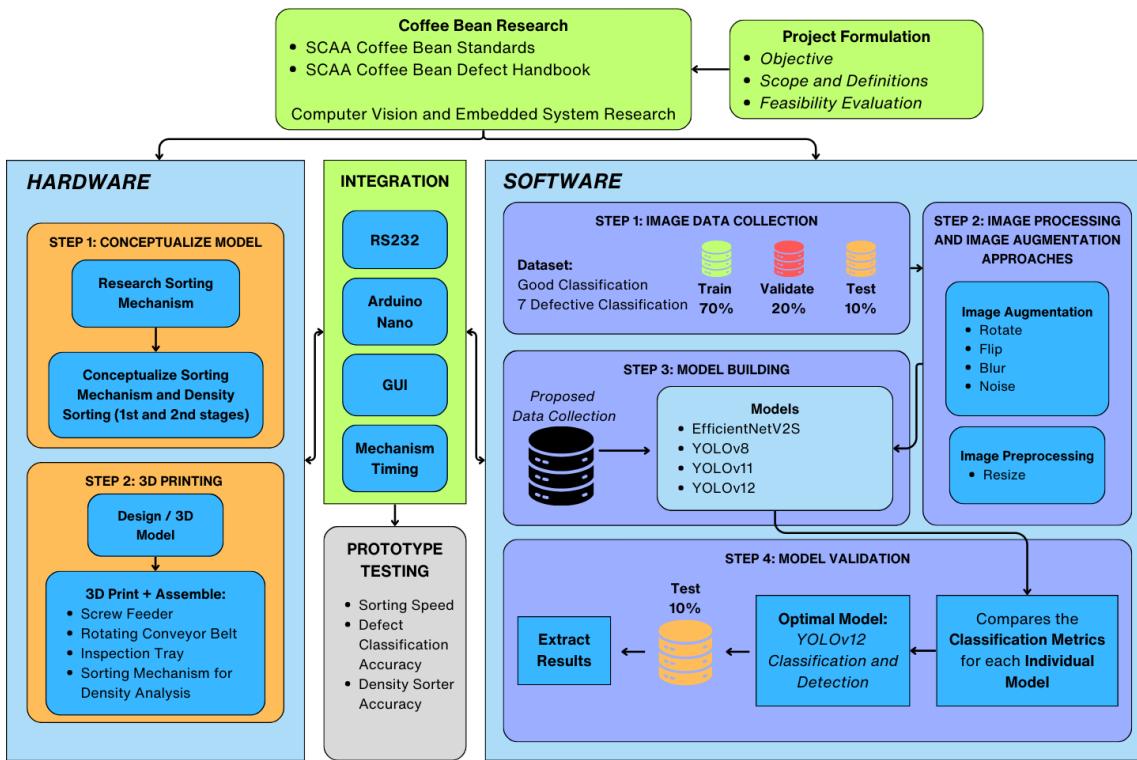


Fig. 5.4 Design and Development Research (DDR) Methodology

902

The researchers opted for a Design and Development Research model for the research.

903

As shown in Figure 5.4, there are multiple levels that were needed in order to develop a

904

working prototype for the system.

905

5.3 Dataset Collection

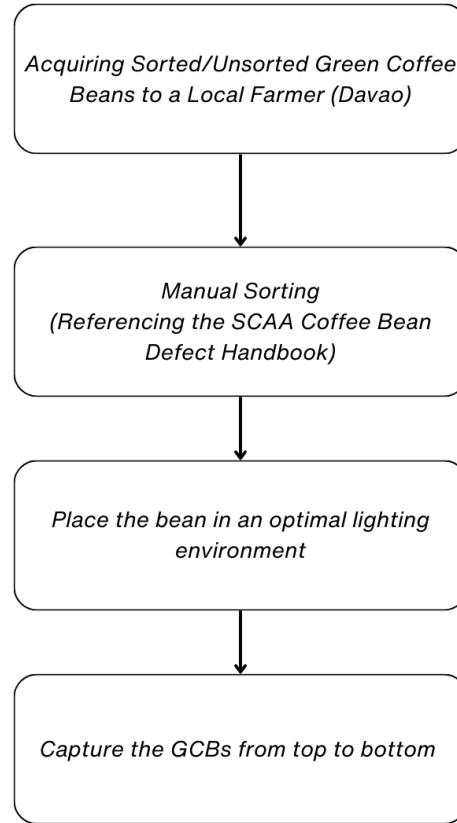


Fig. 5.5 Data Collection Process

906 For dataset collection, Arabica green beans from a farm will be used. Each bean will be
907 captured by a high-resolution camera under sufficient and consistent lighting. Proper light-
908 ing is crucial, as it directly affects the visibility of the bean's physical features, minimizing
909 shadows, grain, and other noise that could result from inconsistent illumination. The top
910 and bottom side pictures of the beans are to be collected. In addition, defective beans of
911 the same type and origin will be gathered to identify the different classification of defects
912 (primary and secondary). This study focuses on defects such as Broken, Dried Cherry,



913 Floater, Full Black, Full Sour, Fungus Damage, and Insect Damage. The dataset will
 914 include at least 500 images of good beans and a minimum of 200 images for each defect
 915 category. To expand the dataset and enhance model training, augmentation techniques such
 916 as scaling, rotation, and mirroring will be applied.

917 5.3.1 Dataset Collection and Model Training

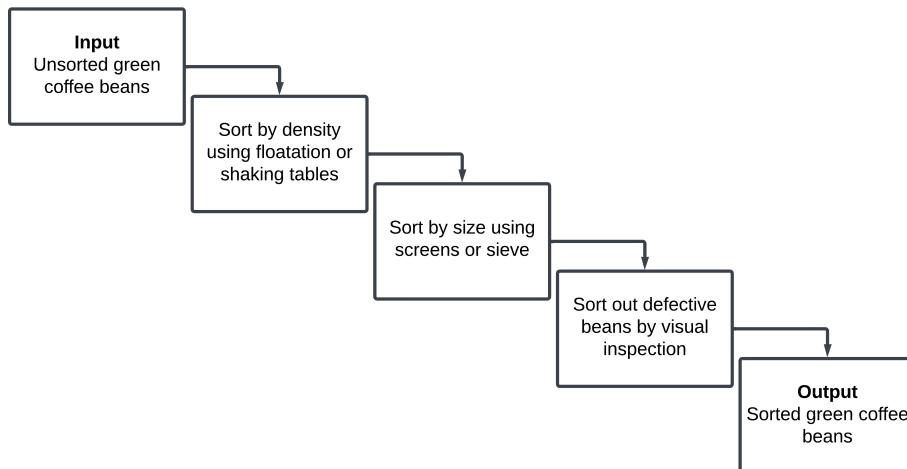


Fig. 5.6 Manual Sorting Process

918 The diagram in Figure 5.6 depicts the representation of the process of manual sorting of
 919 unsorted green coffee beans through a series of steps. First, the beans are sorted by density
 920 using methods such as floatation or shaking tables. This helps in separating the denser
 921 beans, usually pertaining to a more developed and higher quality bean. Then, the beans are
 922 sorted by size using screens and sieves with specific dimensions depending on the variety
 923 of the beans. After this, a thorough visual inspection is performed by the sorters to identify
 924 and remove the defective beans from the batch. To ensure consistency and accuracy, the
 925 group follows the Specialty Coffee Association of America (SCAA) Standards Defect



926 Handbook, which provide documentation and guidelines for identifying and classifying
927 defective beans. Finally, the process results in the output of sorted green coffee beans,
928 ready for further processing or sale. To ensure the dataset reflects real-world conditions, the
929 group acquired Arabica green coffee beans from Davao. These beans were manually sorted
930 to properly classify defective characteristics before capturing images for dataset creation.
931 This step was crucial for improving the efficiency of batch image capture and ensuring
932 accurate model training, making the system more applicable to Philippine coffee producers.

933 **5.3.2 Utilization of Open-Source Database**

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



941

5.3.3 First Iteration of Dataset Collection



Fig. 5.7 First Iteration of Data Collection Setup



De La Salle University

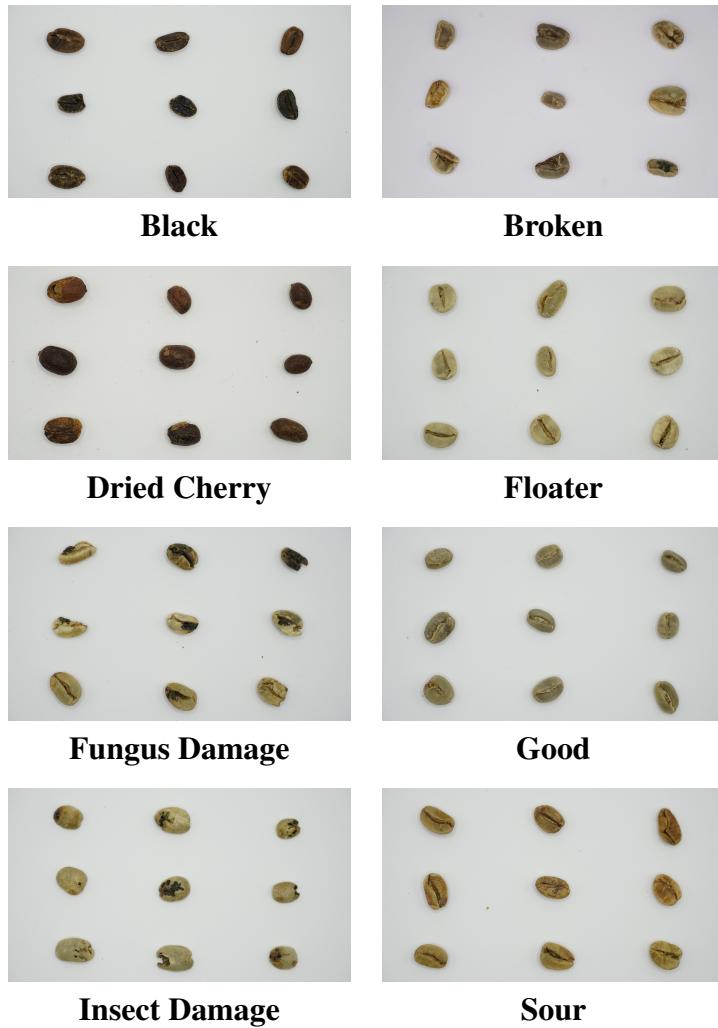


Fig. 5.8 Sample Images from the First Iteration of Dataset Collection

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



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

951 **5.3.4 Second Iteration of Dataset Collection**



Fig. 5.9 Sample Images from the Second Iteration of Dataset Collection



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

957 **5.4 Dataset Preprocessing**

958 **5.4.1 Dataset Splitting**

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

966 **5.4.2 Image Annotation**

967 Roboflow Annotate was used to label images of coffee beans. The platform was used for
968 two separate datasets: one for the detection model, the other for the classification model.
969 In the detection dataset, bounding boxes were drawn around individual coffee beans and
970 labeled accordingly. For the classification dataset, the trained detection model was used
971 to crop individual coffee beans from the raw dataset, which were then categorized into the



972 eight different classifications. Roboflow was chosen for its ability to store datasets in the
973 cloud and its support for different annotation formats, such as COCO and YOLO, ensuring
974 compatibility with different deep learning models during experimentation.

975 **5.4.3 Dataset Augmentation Techniques**

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

981 **5.5 Density Analysis**

982 **5.5.1 Density Threshold Calibration Using Water Displacement 983 Method**

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

990 The total volume of the batch of beans was measured by the water displacement
991 technique, a commonly used method to measure the volume of solids that are irregularly



992 shaped. The beans were fully immersed in a water-filled graduated cylinder, and the rise in
 993 water level was measured. The volume of water displaced is equivalent to the combined
 994 volume of the batch of beans, measured in cubic centimeters (cm^3).

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

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

998 This computed average density served as the threshold value in the system. During
 999 automated classification, individual bean density is calculated using estimated volume (from
 1000 image analysis) and actual weight (from the precision scale via RS232 communication).
 1001 Beans with a density lower than the threshold are classified as less dense, while those
 1002 meeting or exceeding the threshold are considered dense, indicating higher quality.

1003 **5.5.2 Density Threshold Calibration**

1004 **5.6 Model Training**

1005 **5.6.1 Image Detection Models**

1006 The object detection model identifies and isolates the coffee beans from the background.
 1007 For this task, different models were explored:

1008 **1. RF-DETR**



1009 A transformer-based object detection model that eliminates the need for anchor boxes,
1010 improving small object detection.

1011 **2. YOLOv11**

1012 A CNN-based YOLO variant that incorporates the C3k2 block, SPPF, and C2PSA
1013 components to enhance feature extraction and detection accuracy.

1014 **3. YOLOv12**

1015 The latest YOLO version and attention-centric model that integrates transformer-
1016 based components to enhance performance while maintaining real-time efficiency.

1017 **5.6.2 Image Classification Models**

1018 Following detection, each identified coffee bean was cropped and classified based on its
1019 defect type. The classification models used included:

1020 **1. EfficientNetV2**

1021 A convolutional neural network (CNN) designed for high efficiency and accuracy,
1022 balancing computational cost and performance.

1023 **2. YOLOv8**

1024 A lightweight yet highly accurate model that supports both object detection and
1025 classification, making it suitable for real-time applications.

1026 **3. YOLOv11**

1027 A classification-specific adaptation of YOLOv11, leveraging enhanced feature ex-
1028 traction techniques for defect recognition.

1029 **4. YOLOv12**

1030 A classification variant of YOLOv12, incorporating advanced attention mechanisms
1031 to improve accuracy.

1032 **5. Vision Transformer (ViT)**

1033 The Vision Transformer (ViT) processes images as sequences of tokens for classifica-
1034 tion [Dosovitskiy et al., 2021]. A learnable “class token” is added to the sequence to
1035 support classification. Through the attention mechanism, it captures dependencies be-
1036 tween image tokens. The encoder is composed of repeated multi-head self-attention
1037 and feedforward layers, with self-attention computing weighted sums of sequence
1038 elements.

1039 **5.6.3 Bean Classification Logic**

1040 To ensure proper classification, conditions are set based on how the top and bottom-side of
1041 the beans are categorized. For a bean to be labeled Good, both sides need to be classified
1042 as such. Similarly, Dried Cherry requires both sides to be classified as such, since Black
1043 beans can resemble Dried Cherry on its outer side. Moreover, both sides must be classified
1044 as Fungus Damage to be considered as such. If one side is classified as Black and the other
1045 Dried Cherry, or both are Black, then it will be labeled as Black. The rest of the defect
1046 types are sufficient for single-side detection, as seen in Table 5.2.



TABLE 5.2 CLASSIFICATION ALGORITHM FOR COFFEE BEANS

Input(s):

top_class : classification result from top camera; $top_class \in \mathbb{Z}^+$
bottom_class : classification result from bottom camera; $bottom_class \in \mathbb{Z}^+$

Output(s):

class : final bean classification (Good, Defective, or specific defect)

```

1: if top_class = 5  $\wedge$  bottom_class = 5 then
2:   class  $\Leftarrow$  "Good"
3: else if top_class = 2  $\wedge$  bottom_class = 2 then
4:   class  $\Leftarrow$  "Dried Cherry"
5: else if (top_class = 0  $\wedge$  bottom_class = 0)  $\vee$  (top_class = 0  $\wedge$  bottom_class = 2)  $\vee$  (top_class = 2  $\wedge$  bottom_class = 0) then
6:   class  $\Leftarrow$  "Black"
7: else if top_class = 4  $\wedge$  bottom_class = 4 then
8:   class  $\Leftarrow$  "Fungus Damage"
9: else if top_class = 6  $\wedge$  bottom_class = 6 then
10:  class  $\Leftarrow$  "Insect Damage"
11: else if top_class = 1  $\vee$  bottom_class = 1 then
12:  class  $\Leftarrow$  "Broken"
13: else if top_class = 7  $\vee$  bottom_class = 7 then
14:  class  $\Leftarrow$  "Sour"
15: else if top_class = 3  $\vee$  bottom_class = 3 then
16:  class  $\Leftarrow$  "Floater"
17: else
18:  class  $\Leftarrow$  "Defective"
19: end if
20: return class

```

5.6.4 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.



De La Salle University

1054 • **False Positives (FP)** - The number of times a different category was incorrectly
 1055 classified as this defect type.

1056 • **True Negatives (TN)** - All correctly classified instances excluding the defect category
 1057 in question.

1058 • **False Negatives (FN)** - The number of times this defect type was classified as
 1059 something else.

1060 Through these parameters, key performance metrics such as accuracy, precision, recall,
 1061 and F1-score were computed to evaluate the system's performance in different classifica-
 1062 tions as shown below. This test will assist in determining what types of defects the system
 1063 correctly classifies and which types might need improvements in image preprocessing,
 1064 dataset expansion, or optimization of the machine learning model. The outcome will be
 1065 applied to optimize the sorting algorithm for minimal misclassifications to ensure greater
 1066 reliability in real-world defect detection.

1067 1. **Accuracy** measures overall correctness of the classification model

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

1068 2. **Precision** measures how many of the predicted positive classifications were actually
 1069 correct

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

1070 3. **Recall** evaluates how well the model identifies actual positive cases

$$Recall = \frac{TP}{TP + FN} \quad (5.3)$$



1071 4. **F1-score** represents the harmonic mean of precision and recall

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

1072 **5.6.5 Model Benchmarking and Selection**

1073 Several models were trained and tested within the actual system to determine the most
1074 effective one. These models trained and evaluated include EfficientNetV2, YOLOv8,
1075 YOLOv11, and YOLOv12. Each model was assessed using the defined performance
1076 metrics and compared accordingly. The model with the highest overall performance will be
1077 selected for deployment in the system.

1078 **5.7 Hardware Development**

1079 The hardware elements of the system, two-stage automated coffee bean sorter, are devel-
1080 oped to provide effective and precise sorting using a mix of mechanical and electronic
1081 components. Each element is designed and tested to maximize the sorting process while
1082 providing system reliability.



1083

5.7.1 Screw Feeder

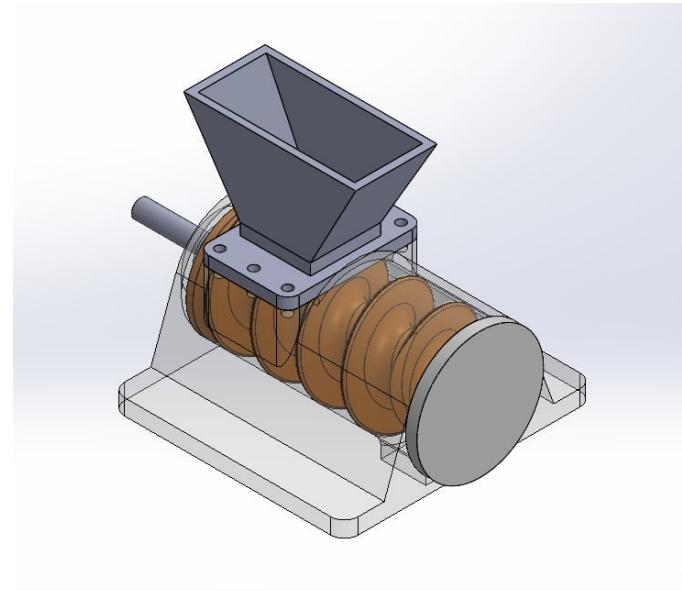


Fig. 5.10 Screw Feeder 3D Design

1084

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 5.10 shows the actual 3D model design of the screw feeder used in the system.



1092

5.7.2 Rotating Conveyor Table

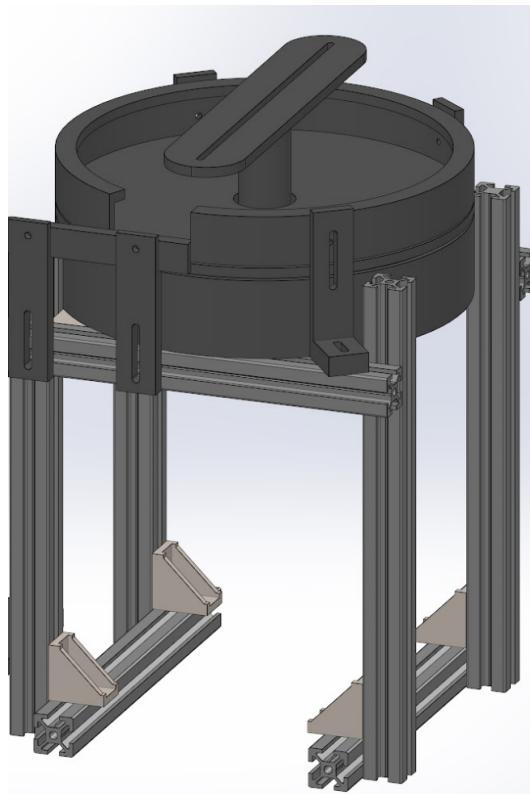


Fig. 5.11 Rotating Conveyor Table 3D Design

1093

The conveyor table, as shown in Figure 5.11, 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.

1094

1095

1096

1097

1098

1099



Fig. 5.12 Rotating Conveyor Table with Aluminum Guides

1100 As shown in Figure 5.12, there are aluminum guides on the rotating conveyor table that
1101 ensures coffee beans to be linearly arranged. This linear arrangement of beans significantly
1102 helped the system to ensure that coffee beans are dropped onto the slide, which connects the
1103 conveyor table to the inspection try, in a one-by-one manner. In addition, the guides are also
1104 installed to keep the beans from accumulating in one area, which can cause the jamming of
1105 beans. The researchers tested the different motor speeds to observe the optimal settings that
1106 will not cause bean jamming and meet the minimum sorting speed of the system. However,
1107 while the aluminum guides are effective in arranging the beans linearly, it was hard to
1108 re-calibrate or adjust. Another problem was it was easy to be deformed whenever there



De La Salle University

1109 was jamming of beans. Upon printing the model, the bearing of the rotating mechanism
1110 was also not robust enough to hold the motor in higher RPM. There are instances where
1111 the gearing mechanism between the motor and the conveyor table experiences jamming,
1112 resulting in the conveyor table temporarily ceasing rotation. This issue typically arises from
1113 misalignment or friction within the gear interface, which hinders the smooth transfer of
1114 torque from the motor to the conveyor table.

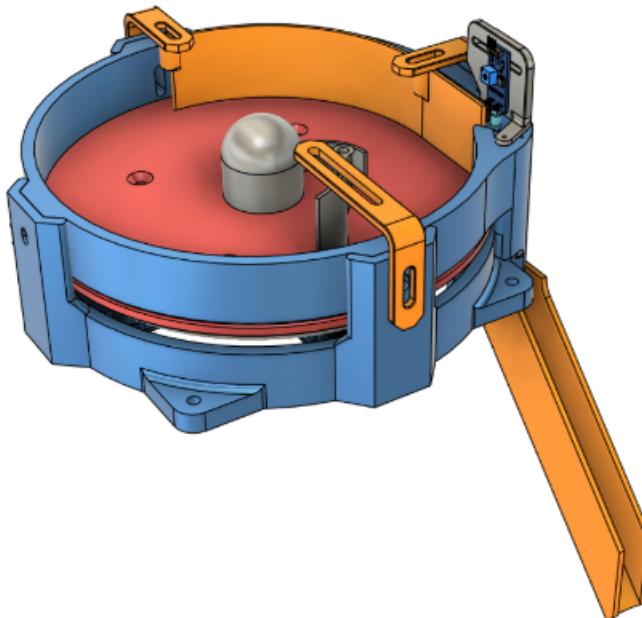


Fig. 5.13 Revised Rotating Conveyor Table

1115 Figure 5.13 shows the re-designed conveyor table. In this revision, calibration of the
1116 guides was easier since it can be moved by only loosening the screws. To avoid the issue
1117 caused by the misalignment within the gear of the previous rotating table mechanism, the
1118 researchers opted to revise the design. The new iteration of the rotating conveyor table
1119 was redesigned using a Lazy Susan bearing. In this modification, smoother rotation was



1120 observed.

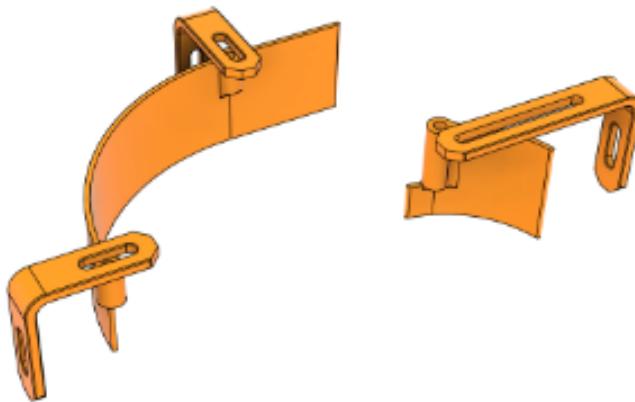


Fig. 5.14 3D Printed Guides

1121 In addition, the previously aluminum guides were replaced with 3D-printed guides that
1122 were screwed onto the table walls. The revised design allowed for easier calibration, as
1123 the guides could be repositioned simply by loosening the screws, enabling more precise
1124 alignment of beans.

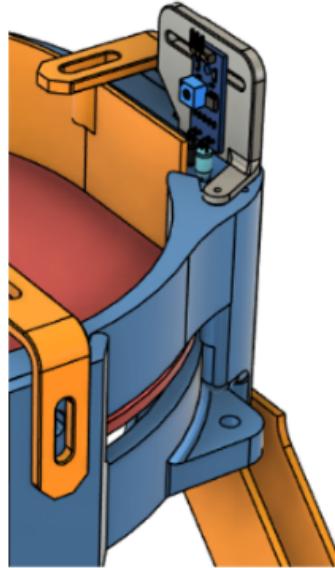


Fig. 5.15 IR Sensor Placement

Another point of improvement was the placement of the IR sensor, which was redesigned to be adjustable. This adjustability allowed faster calibration and ensured accurate detection of beans at the table's edge, further enhancing the system's reliability and consistency. 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 X. 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 in a faster sorting time for the



1138 first stage.

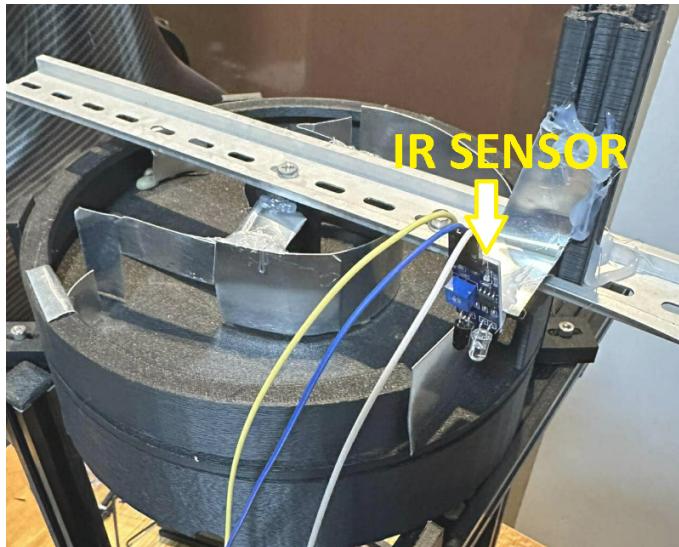


Fig. 5.16 Rotating Conveyor Table with IR Sensor

1139 Initially, the rotating conveyor table is set at a fixed and slow speed to ensure that coffee
1140 beans are dropped into the inspection tray one-by-one. However, at this rate, the time travel
1141 time of the first bean dropped from the center of the table is very long. Thus, the group
1142 decided to add an IR sensor at the edge of the rotating table as seen in Figure 5.16. The
1143 sensor's responsibility is to detect if there is a bean at the edge. If there is no bean detected,
1144 the rotating table is set to a higher speed to expedite the process. On the other hand, if a
1145 bean is detected by the sensor, the rotation of the table is adjusted in such a way that it is
1146 able to drop the beans one-by-one onto the inspection tray. With this sensor integrated into
1147 the system, a higher speed can be set for the rotating table, minimizing the time travel of
1148 the beans from the center to the inspection tray, resulting to a faster sorting time for the
1149 first stage.



5.7.3 Stage 1: Defect Sorting (Machine Vision and Inspection Tray)

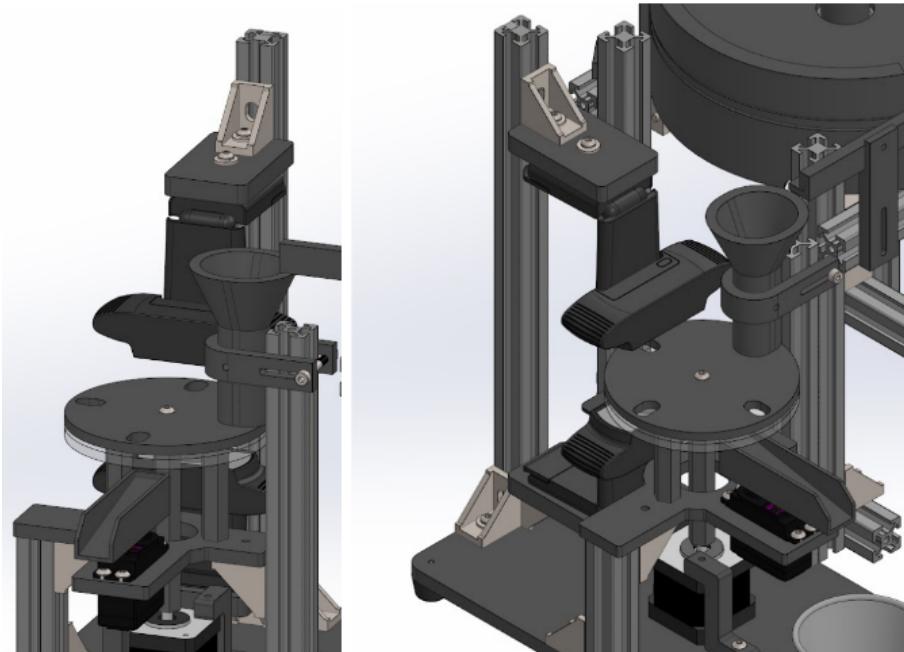


Fig. 5.17 Inspection Tray 3D Design

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



1161 defective beans.

1162 **5.7.4 Stage 2: Density Sorting (Precision Scale and Servo
1163 Mechanism)**

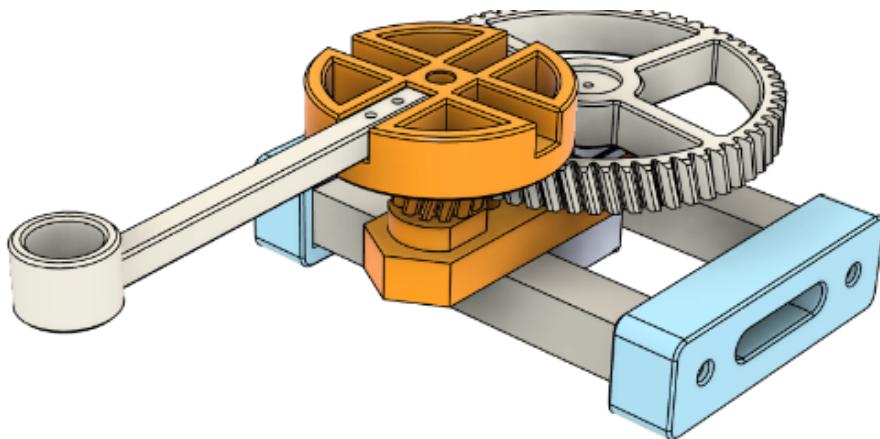


Fig. 5.18 Density Sorter Mechanism Design

1164 The density sorter is the second-stage sorting system, tasked with sorting coffee beans
1165 according to their measured density. This is achieved by initially measuring each bean's
1166 mass using a precision weighing scale and volume using the computer vision. After
1167 calculating the density, the system triggers a sorting system powered by a geared 5V
1168 servomotor, which sorts beans into various collection bins according to their classification.
1169 This sorting operation is such that high-density beans are kept separate from low-density
1170 beans. The density sorter's accuracy is verified by comparing the results of its classification
1171 to manual weighing measurements (ground truth data).



Fig. 5.19 Precision Scale

1172 The U.S. Solid Electronic Precision Balance (0.01g, 1200g capacity, RS232 port,
1173 AC/DC power) was selected for the density sorting mechanism because it is highly accurate,
1174 transmits data in real-time, and is well-calibrated. Its 0.01g precision guarantees accurate
1175 mass readings, which are critical to precise density calculations in sorting coffee beans.
1176 The RS232 port facilitates smooth integration with the microcontroller for automatic data
1177 processing and sorting decisions, minimizing manual errors. Its dual power source (AC
1178 and battery) also guarantees uninterrupted operation in different environments, making it a
1179 dependable and efficient part of the coffee bean sorting system.



5.8 Hardware and Software Integration

5.8.1 Serial Communication

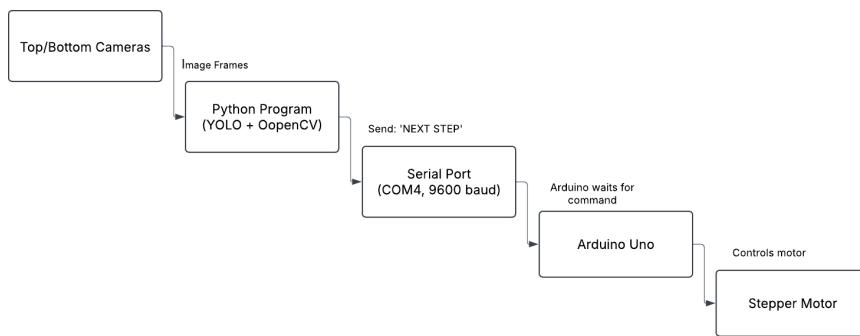


Fig. 5.20 Serial Communication Flow for Stage 1 Classification

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



De La Salle University

1195 interface using the pyserial library in Python. In addition, this is configured at a baud rate
1196 of 9600.

1197 The system, specifically at the inspection tray mechanism where the YOLO detection
1198 and classification is implemented, has a function move_stepper() responsible for sending
1199 the command from the Python code to the Arduino microcontroller. When the Arduino
1200 receive this command, it executes motor movement that allows the stepper motor to move
1201 at a certain angle that allows the camera to capture the bean. This function is crucial for the
1202 system as this is how each bean in the inspection tray is fed to the image processing side of
1203 the system. This movement rotates the mechanism holding the coffee beans, positioning the
1204 next bean beneath the top and bottom cameras for inspection. After the motor completes
1205 the movement, the Arduino will send back a message to the program running Python,
1206 signalling that the bean is ready for image capture and further processing. In addition, the
1207 Python script is continuously or constantly waiting for the Arduino's message through the
1208 arduino.readline() function, ensuring seamless communication and faster processing.



1209

5.8.2 Recommended Standard 232 (RS-232)

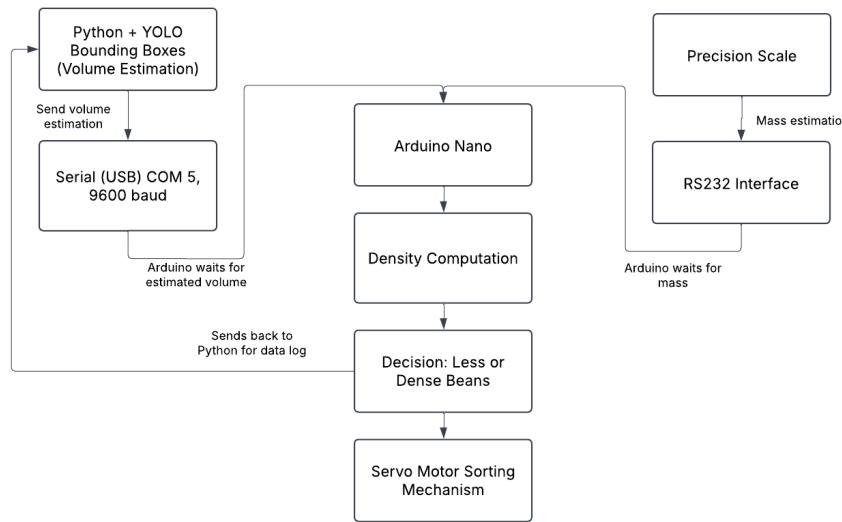


Fig. 5.21 Precision Scale Integration with RS232 for Stage 2 Classification

1210

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.

1216

The stage 2 classification, density-based sorting, is implemented using a combination of RS232 and USB serial communication. In this stage, each bean that has been classified as 'Good' from stage 1 is again sorted based on the density. The RS232's main responsibility is to simultaneously record and pass the values to the Arduino Nano to compute the density. Subsequently, the data from the computations are the deciding factor whether to sort out



1221 the bean or not, depending on the predefined threshold for the system.

1222 **5.9 Prototype Setup**

1223 **5.9.1 Actual Setup**

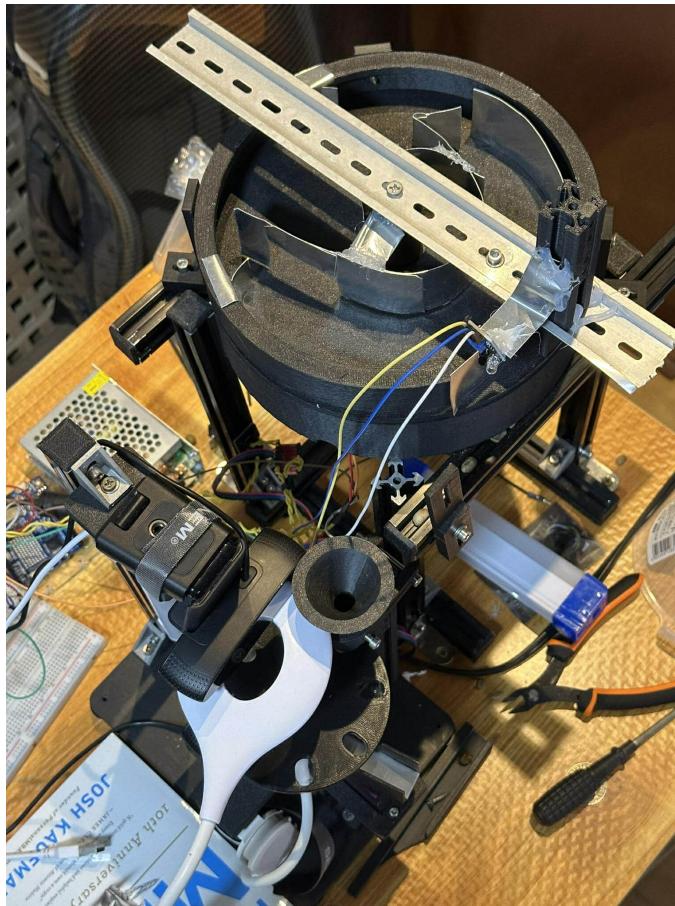


Fig. 5.22 Actual System Setup

1224 Physical integration of the automatic coffee bean sorter system comprises various integrated
1225 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



1250 regulated by a NEMA 17 stepper motor.

1251 **5.9.2 Lighting Setup for Inspection Tray**

1252 Lighting has a key importance in the image-based detection and classification system,
1253 specifically for the inspection tray. For the model to be more accurate and precise in
1254 classifying good and defective beans, correct lighting is important such that details like
1255 surface texture, color difference, and defects are properly rendered by the imaging system.
1256 Asymmetrical, unsteady, or low-quality lighting can create shadows, reflections, or over-
1257 exposure, all of which lower the quality of input images and thus decrease the accuracy
1258 of object detection and classification models like YOLO. To improve the consistency and
1259 definition of images taken during inspection, the lighting arrangement above the inspection
1260 tray was refined incrementally throughout development. The refinements were intended to
1261 maximize the illumination conditions for both the top and bottom camera modules.



De La Salle University



Fig. 5.23 First Iteration of Lighting Setup

Figure 5.23 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.

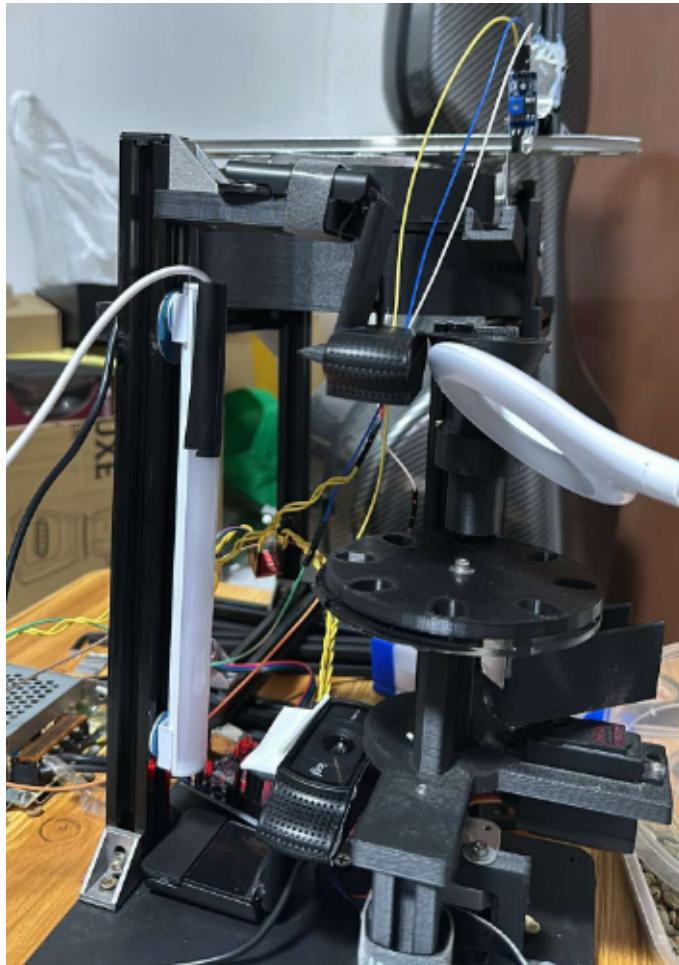


Fig. 5.24 Second Iteration of Lighting Setup

1269 For the second iteration of the lighting setup, the researchers decided to add another
1270 LED strip lighting at the side of the inspection tray, while keeping the LED lighting
1271 mounted at the top. This provided good lighting for both top and bottom cameras. However,
1272 the view of the bottom camera is still a bit dark.

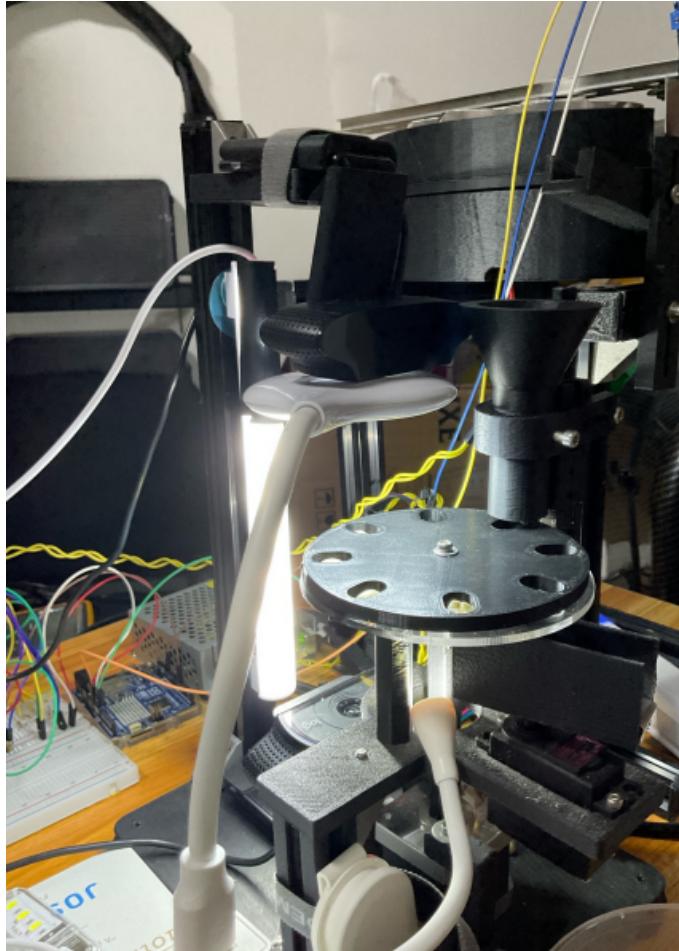


Fig. 5.25 Final Iteration of Lighting Setup

1273 To ensure that both camera views have sufficient lighting and avoid shadows, the
1274 researchers decided to use a total of three LED lights. One is a small ring light placed
1275 exactly above the inspection tray. Another LED light is a strip light placed at the side of
1276 the inspection tray to improve lighting at the side of each bean. Another small LED light
1277 is placed under the inspection tray to ensure that the bottom camera has enough lighting.
1278 Electrical tape was applied to the side of the acrylic plate to block excess lighting, thereby
1279 optimizing the lighting setup of the system.

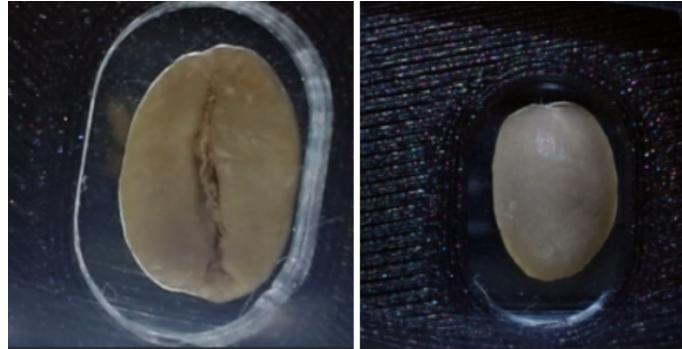


Fig. 5.26 Top and Bottom View of the Cameras

5.9.3 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 begins 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 detect 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



1295 of top and bottom cameras. The system captures high-resolution images of the bean and
1296 processes the data to determine which classification it belongs. If the bean is identified as
1297 defective, a signal is sent to the servo motor, which redirects the bean into the defective bin
1298 for disposal, if the bean is classified as good, it then proceeds to the second phase of the
1299 system

1300 The second stage involves density-based sorting, where each GCB's weight is measured
1301 using a precision scale, while its volume is determined by the ToF10120 infrared sensor.
1302 The system then calculates the density and classifies the bean accordingly.

1303 The sorting mechanism activates, directing beans into designated collection bins based
1304 on their density. High-density beans, often associated with specialty-grade quality, are
1305 separated from low-density, commercial-grade, or defective beans.



1306 **5.10 Prototype Testing**

1307 **5.10.1 Sorting Speed**

TABLE 5.3 SORTING SPEED TESTING TABLE

Test Condition	Conveyor Table Speed (RPM)	Inspection Tray Speed (RPM)	Sorting Speed (Beans per Minute)
100% Good Beans			
75% Good, 25% Defective Beans			
50% Good, 50% Defective Beans			
25% Good, 75% Defective Beans			
100% Defective Beans			

1308 The sorting speed of the system will be determined by conducting at least five trials.
 1309 Each trial will be exactly conducted for one minute. The number of beans sorted out within
 1310 the time frame are considered as the sorting speed in beans per minute. Then, the average
 1311 sorting speed from the five trials is computed. In each trial session, controlled variables
 1312 such as motor speed of the inspection tray and rotating conveyor table are varied to observe
 1313 the optimal setting for the system, ensuring that there are no beans jamming in the tray and
 1314 fast enough to meet the minimum sorting speed. Table 5.3 shows the different conditions
 1315 for each trial to ensure that the sorting speed across different type of beans are considered.

**1316 5.10.2 Defect Sorting Accuracy**

1317 To measure the system's performance on defect sorting, two separate tests were performed.
1318 The first test was performed to measure the system's accuracy on classifying the seven
1319 defect types including good beans. On the other hand, a second test was conducted to
1320 measure the system's reliability on sorting out defective beans from good beans. In this
1321 test, there are only two classifications – defective beans and good beans.



De La Salle University



Black - 30 Beans



Broken - 30 Beans



Dried Cherry - 30 Beans



Floater - 30 Beans



Fungus Damage - 30 Beans



Good - 150 Beans



Insect Damage - 30 Beans



Sour - 30 Beans

Fig. 5.27 Per-Classification Test Dataset

1322 Table 5.27 shows the actual test dataset used in the first test. This test was conducted
1323 using the top two performing models after training. Based on the 70-20-10 training,
1324 validation, test split, the researchers gathered 30 beans for each defect type, and 150 for



1325 good beans. The test consists of 5 trials with the same test set. Each trial was divided
 1326 into 8 parts, corresponding to each classification. In this test, if one of the cameras detects
 1327 the specific defect type being tested, it is considered as correctly classified. However, for
 1328 good beans, both cameras should be able to classify it as good to be considered as a correct
 1329 classification.

**100% Good****75% Good, 25% Defects****50% Good, 50% Defects****25% Good, 75% Defects****100% Defects**

Fig. 5.28 Good vs. Defective Test Dataset

1330 The defect sorting accuracy by feeding 100 beans on each trial. For testing its accuracy
 1331 for detecting good beans and defective beans, five trials are conducted containing 100 beans
 1332 of good beans for the first trial, 75 good and 25 defects for the second trial, 50 good and 50



1333 defects for the third trial, 25 good and 75 defects for the fourth trial, and 100 defects for the
 1334 last trial. With these, the number of correctly classified and misclassified beans are logged
 1335 into the system to compute for accuracy.

TABLE 5.4 DATASET DISTRIBUTION FOR OVERALL TESTING

Bean Classification	Bean Count
Good	150
Black	30
Broken	30
Dried Cherry	30
Floater	30
Fungus Damage	30
Insect Damage	30
Sour	30
Total Beans	360

1336 Lastly, to assess the overall accuracy and reliability of the first stage, machine vision-
 1337 based defect classification, a trial consisting of a predefined dataset of 360 coffee beans
 1338 was conducted. The good classification consists of 150 beans, while each defect, such as
 1339 black, dried cherry, fungus, insect damage, sour, floater, and broken beans, consists of 30
 1340 beans as shown in Table 5.4.

5.10.3 Density Sorting Accuracy

1341 To assess the accuracy of the mechanism, it will rely on measuring the accuracy and the
 1342 reliability of the density sorting mechanism in sorting out the dense beans to the less dense
 1343 beans. To successfully determine the accuracy of the system, the basis will be the scale,
 1344 where the system should be able to sort the dense beans to the less dense bean in relation



1346 to the detected weight in the scale. A successful system should be able to sort with an
1347 accuracy of 85%.



De La Salle University

1348

Chapter 6

1349

RESULTS AND DISCUSSIONS



TABLE 6.1 SUMMARY OF RESULTS FOR ACHIEVING THE OBJECTIVES

Objectives	Results	Locations
GO: The study aims 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.	<ul style="list-style-type: none"> Achieved to gather and create a unique dataset consisting of 500 good and 200 defective beans Achieved improvisation of the synchronization between the machine vision and embedded system. 	Sec. 6.1 on p. 108
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).	<ul style="list-style-type: none"> Acquired 391 images of Black coffee beans Gathered 259 images of Broken coffee beans Gathered 359 images of Dried Cherry coffee beans Acquired 260 images of Floater coffee beans Acquired 255 images of Fungus Damage coffee beans Gathered 1513 images of Good coffee beans Acquired 370 images of Insect Damage coffee beans Gathered 404 images of Sour coffee beans 	Sec. 6.1 on p. 108

Continued on next page

6. Results and Discussions



De La Salle University

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.	<ul style="list-style-type: none">Achieved 22 beans per minute for stage one of the system	Sec. ?? on p. ??

Continued on next page

6. Results and Discussions



De La Salle University

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	<ul style="list-style-type: none"> • Achieved 90.07% testing accuracy in classifying Black coffee beans. • Achieved 90.07% testing accuracy in identifying Broken coffee beans. • Attained 90.65% testing accuracy in recognizing Dried Cherry coffee beans. • Recorded 87.78% testing accuracy in detecting Floater coffee beans. • Achieved 90.65% testing accuracy in classifying Fungus Damage coffee beans. • Reached 90.07% testing accuracy in identifying Good coffee beans. • Attained 90.07% testing accuracy in detecting Insect Damage coffee beans. • Achieved 90.65% testing accuracy in classifying Sour coffee beans. • Achieved 90.00% overall testing accuracy of the system. 	Sec. ?? on p. ??
SO4: To achieve an accuracy of at least 85% in filtering out less-dense green coffee beans	<ul style="list-style-type: none"> • 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	391
Broken	259
Dried Cherry	359
Floater	260
Fungus Damage	255
Good	1513
Insect Damage	370
Sour	404
Total	3811

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 and a minimum of 1500 images for Good beans; for instance, Black has 391 images and Good has 1513 images. The table confirms a total of 3811 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	70%	2668	Original training images are augmented three times
Validation	20%	762	Non-augmented
Test	10%	381	Non-augmented

Table 6.3 outlines the dataset split into training, validation, and test sets. The training set comprises 70% (2,668 images), while the validation and test sets account for 20% (762



1358 images) and 10% (381 images) respectively, with the training images later augmented 3×
1359 per image.

1360 **6.2 Performance of Classification Models on Cus- 1361 tom Dataset**

1362 Five different classification models, such as EfficientNet, YOLOv8, YOLOv11, YOLOv12,
1363 and ViT were benchmarked to determine the most optimal model to be used for the system.
1364 Each model was trained using a custom dataset manually gathered by the researchers. In
1365 addition, augmentations such as rotation, flip, blur and noise, were applied.



1366

6.2.1 EfficientNetV2S

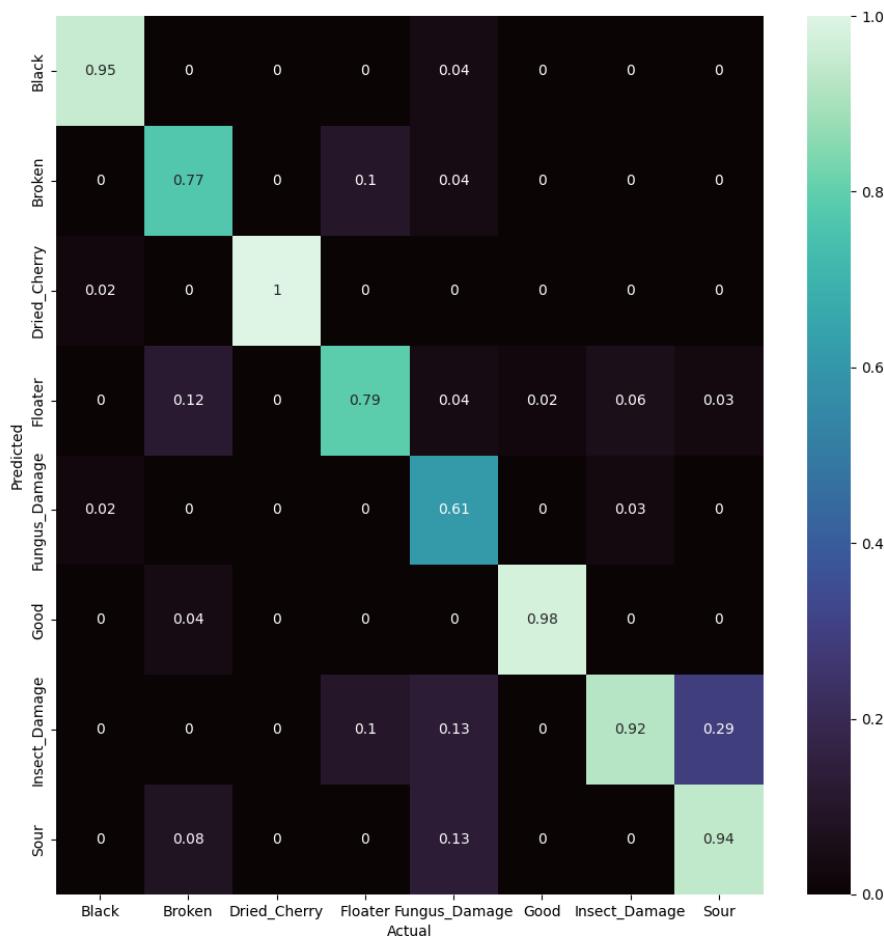


Fig. 6.1 Normalized Confusion Matrix for EfficientNetV2S on Test Dataset

1367

EfficientNetV2 maintained competitive recognition for Black (95%), Dried Cherry (100%),

1368

and Good (98%), but defect categories performed poorly. Fungus Damage was the weakest

1369

across all models (61%), with extensive misclassification into Insect Damage and Sour.



- 1370 Broken scored only 77%, spilling into Floater and Sour. Sour beans had 94% recognition
1371 but with 29% misclassified as Insect Damage. While good at separating distinct classes,
1372 this model struggles most with subtle defect patterns.



1373

6.2.2 YOLOv8

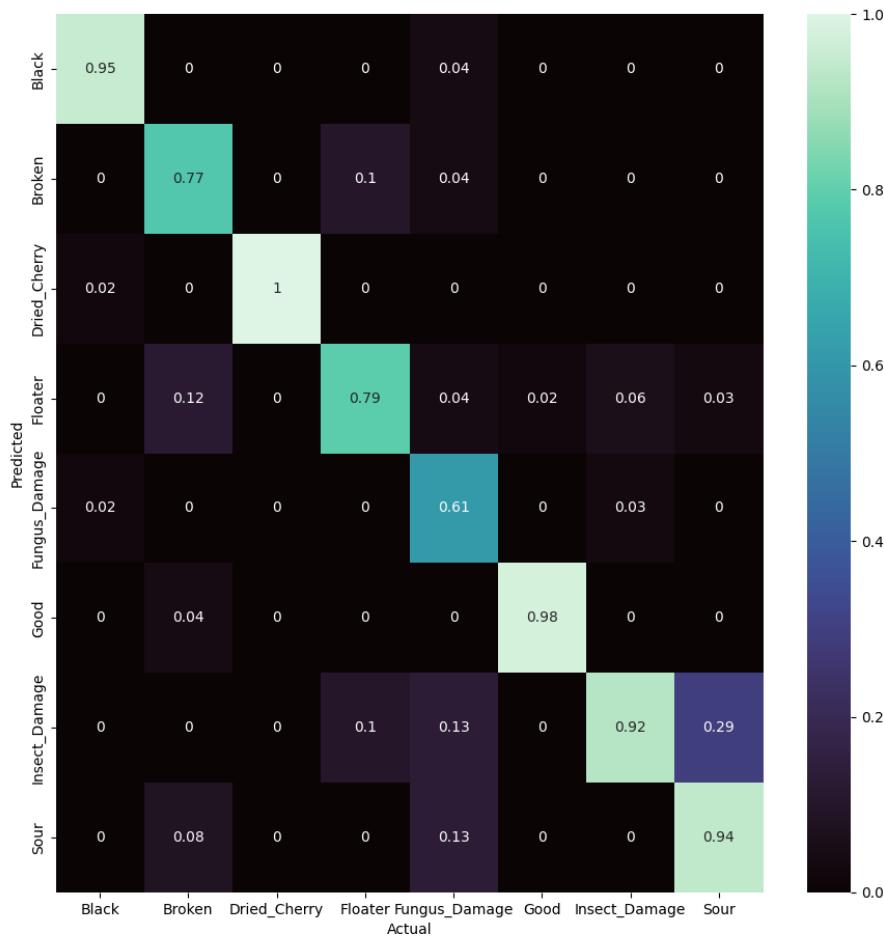


Fig. 6.2 Normalized Confusion Matrix for YOLOv8 on Test Dataset

1374

YOLOv8 achieved strong recognition for Black (98%), Dried Cherry (100%), Good (99%),

1375

and Sour (100%). However, it struggled with Fungus Damage (61%), where a large share of

1376

samples were confused with Broken and Insect Damage. Broken was moderately accurate



- 1377 (85%) but often mistaken as Floater. Insect Damage was fairly strong (94%) yet confused
1378 back into Fungus Damage. The model excels at highly distinctive classes but struggles with
1379 visually similar defects.



1380

6.2.3 YOLOv11-cls

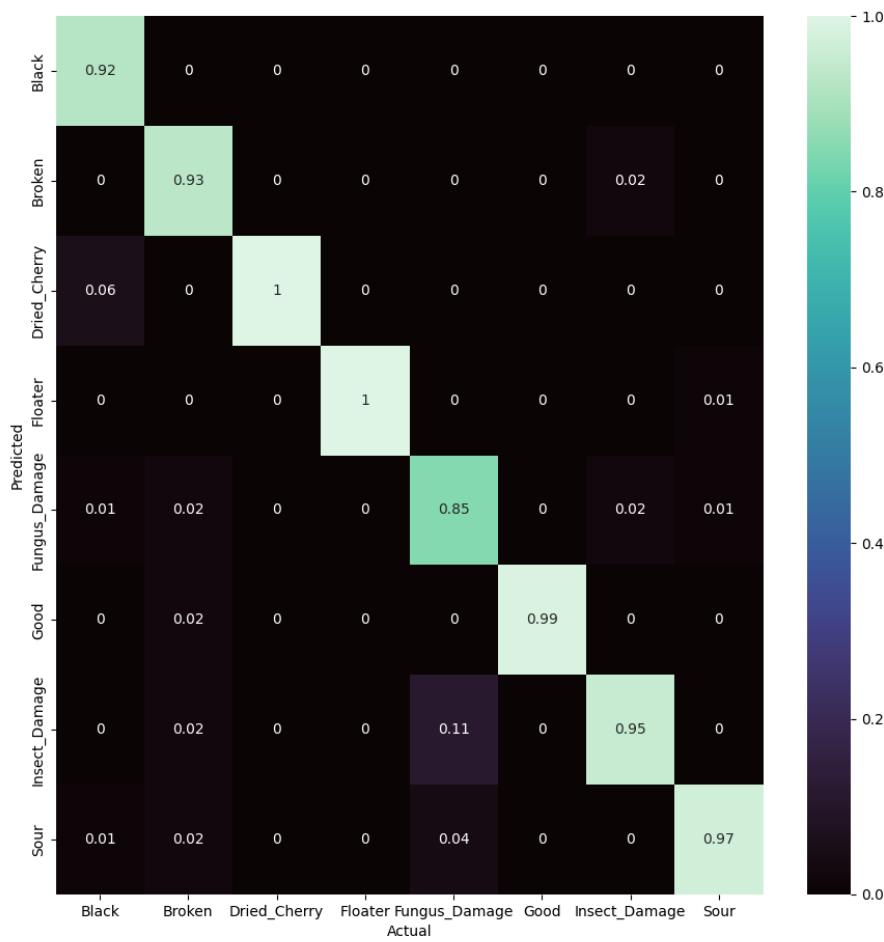


Fig. 6.3 Normalized Confusion Matrix for YOLOv11 on Test Dataset

1381

YOLOv11 improved class balance compared to YOLOv8, with near-perfect results in Dried

1382

Cherry, Floater, and Good (all 99–100%). Broken rose to 93%, reducing cross-class errors.

1383

Fungus Damage remained difficult at 85%, misclassified into Broken, Insect Damage, and



De La Salle University

- 1384 Sour. Insect Damage achieved 95% but still leaked into Fungus Damage. This model
1385 demonstrates more stability in defect-prone categories while maintaining high precision in
1386 separable classes.



1387

6.2.4 YOLOv12-cls

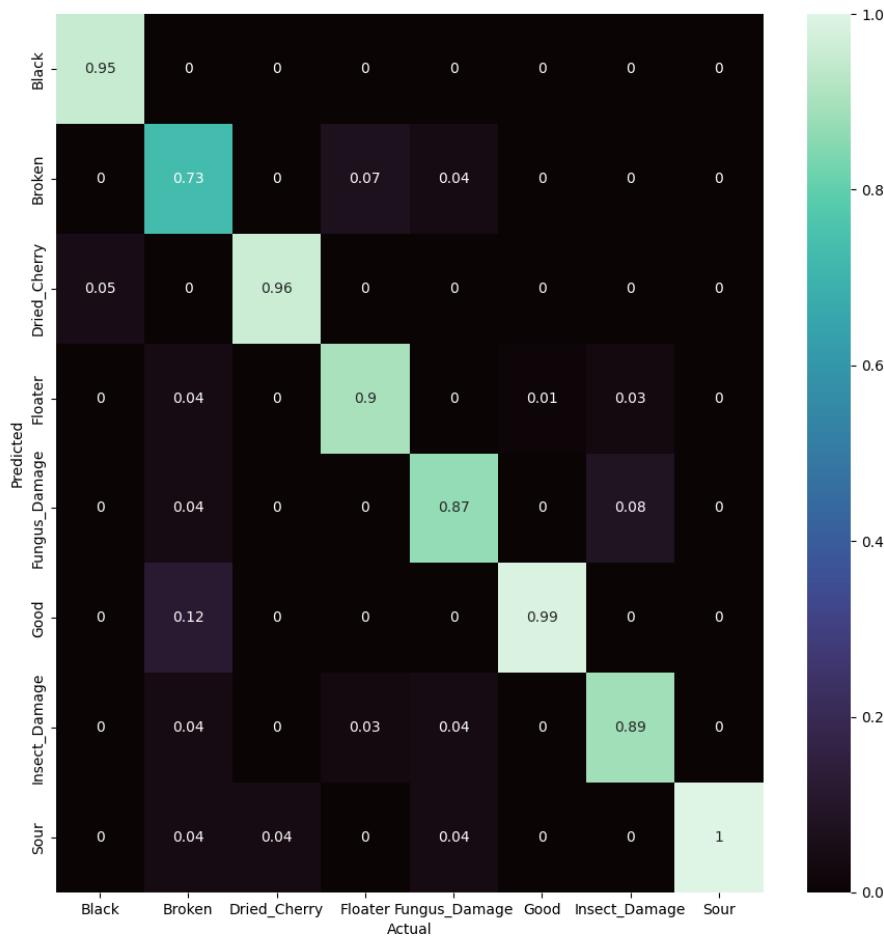


Fig. 6.4 Normalized Confusion Matrix for YOLOv12 on Test Dataset

1388

YOLOv12 presented mixed performance: Dried Cherry (96%), Good (99%), and Sour

1389

(100%) stayed strong, but Broken collapsed to 73%, with heavy confusion into Floater and

1390

Fungus Damage. Fungus Damage was moderate (87%) but overlapped with Insect Damage



1391 (8%). Insect Damage itself dropped to 89%, reflecting this reciprocal confusion. Compared
1392 to YOLOv11, YOLOv12 was less stable on defect-heavy categories, showing variability
1393 despite strong results in clearer classes.



1394

6.2.5 Vision Transformer (ViT)

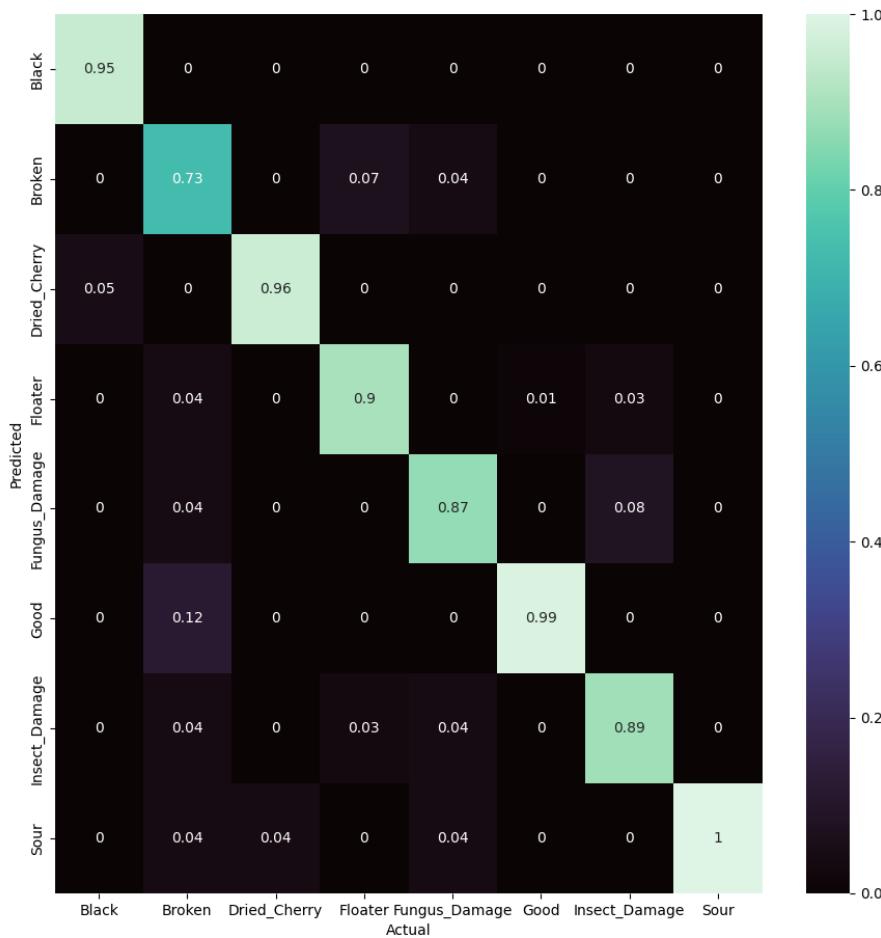


Fig. 6.5 Normalized Confusion Matrix for ViT on Test Dataset

1395

The ViT model delivered the most consistent results overall. Black and Dried Cherry were perfectly classified (100%), while Good, Insect Damage, and Sour exceeded 97%. Broken

1396



1397 reached 92%, with minor leakage into Floater and Fungus Damage. Floater held 93%
 1398 accuracy, though with 7% misclassified as Broken. Fungus Damage was stronger than
 1399 in other models (87%) but still overlapped with Insect Damage (9%). ViT demonstrates
 1400 the best balance, minimizing confusion between defects while maintaining near-perfect
 1401 performance in distinct classes.

TABLE 6.4 MODEL PERFORMANCE COMPARISON

Model	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
EfficientNetV2	87.90	87.08	87.08	98.02
YOLOv8	91.14	90.41	90.55	98.65
YOLOv11	92.97	93.71	93.19	98.90
YOLOv12	91.97	91.20	91.47	98.64
ViT	95.85	95.76	95.77	99.35

1402 Table 6.4 shows that EfficientNetV2 had the weakest performance, with the lowest
 1403 precision, recall, F1-score, and accuracy. The YOLO models improved on these results,
 1404 with YOLOv11 performing slightly better than YOLOv8 and YOLOv12. Among all, ViT
 1405 achieved the highest scores across all metrics, showing the best ability to classify the test
 1406 dataset with minimal errors. Overall, performance increases from EfficientNetV2 to YOLO,
 1407 with ViT giving the most reliable results.

1408 **6.3 Actual Performance of Classification Models in** 1409 **the System**

1410 Among the 5 models, YOLOv12 and ViT achieved the highest performance. Thus, these
 1411 two models were tested and deployed on the actual system. To measure the performance of



1412 the two models, two types of tests were conducted. The first test was conducted to measure
1413 the models' performances on classifying different defect types, and the other test was
1414 dedicated to measuring the models' performances when classifying Good and Defective
1415 beans . The first test set was composed of 30 beans per defect and 150 good beans, with
1416 a total of 360 beans. For each trial, the testing was divided into 8, corresponding to each
1417 classification. True Positives (TP) are the number of samples from a defect type that were
1418 correctly classified as that defect. False Negatives (FN) are the number of samples from a
1419 defect type that were misclassified as something else. Per-class accuracy was computed
1420 for each trial and the average accuracy across all trials. On the other hand, the second test
1421 was composed of 100 beans per trial, wherein each trial had a varying distribution of good
1422 and defective beans. In this test, TP is the number of good beans classified as good, TN is
1423 the number of defective beans classified as defects, FP is the number of defective beans
1424 misclassified as good, and FN is the number of good beans misclassified as defects.

6. Results and Discussions



De La Salle University

Trial	Defect Type	Number of Samples	TP	FN	Accuracy	Average Accuracy Per Trial
1	Black	30	24	6	80.00%	86.67%
	Broken	30	22	8	73.33%	
	Dried Cherry	30	22	8	73.33%	
	Floater	30	28	2	93.33%	
	Fungus Damage	30	22	8	73.33%	
	Good	150	144	6	96.00%	
	Insect Damage	30	30	30	100.00%	
	Sour	30	30	30	100.00%	
2	Black	30	25	5	83.33%	86.67%
	Broken	30	22	8	73.33%	
	Dried Cherry	30	23	7	76.67%	
	Floater	30	27	3	90.00%	
	Fungus Damage	30	23	7	76.67%	
	Good	150	145	5	96.67%	
	Insect Damage	30	29	1	96.67%	
	Sour	30	30	0	100.00%	
3	Black	30	24	6	80.00%	85.75%
	Broken	30	23	7	76.67%	
	Dried Cherry	30	21	9	70.00%	
	Floater	30	28	2	93.33%	
	Fungus Damage	30	22	8	73.33%	
	Good	150	144	6	96.00%	
	Insect Damage	30	30	30	100.00%	
	Sour	30	29	1	96.67%	



4	Black	30	26	4	86.67%	87.09%
	Broken	30	21	9	70.00%	
	Dried Cherry	30	23	7	76.67%	
	Floater	30	28	2	93.33%	
	Fungus Damage	30	23	7	76.67%	
	Good	150	145	5	96.67%	
	Insect Damage	30	29	1	96.67%	
	Sour	30	30	0	100.00%	
5	Black	30	24	6	80.00%	85.83%
	Broken	30	22	8	73.33%	
	Dried Cherry	30	22	8	73.33%	
	Floater	30	27	3	90.00%	
	Fungus Damage	30	22	8	73.33%	
	Good	150	145	5	96.67%	
	Insect Damage	30	30	0	100.00%	
	Sour	30	30	0	100.00%	
Overall Accuracy Across All Trials						86.40%

Fig. 6.6 Actual Performance of YOLOv12 in the System (Per-Classification)

1425 Table 6.6 represents the actual performance of YOLOv12 when deployed into the
 1426 system. Across five trials, the results showed that the model achieved promising accuracy in
 1427 certain categories like Insect Damage and Sour, where an accuracy of 100% in some trials
 1428 were recorded. Most importantly, the model's performance on Good beans classification
 1429 was highly reliable achieving 96-97% across the different trials. However, it was observed
 1430 that the model struggled with other categories such as Black, Broken, Fungus Damage, and
 1431 Dried Cherry, achieving an accuracy score of only around 70-80%. The overall accuracy of
 1432 YOLOv12 across all the trials was 86.4%, which is already reliable especially for detecting
 1433 Good vs Defective beans.

6. Results and Discussions



De La Salle University

Trial	Defect Type	Number of Samples	TP	FN	Accuracy	Average Accuracy Per Trial
1	Black	30	30	0	100.00%	95.83%
	Broken	30	26	4	86.67%	
	Dried Cherry	30	30	0	100.00%	
	Floater	30	30	0	100.00%	
	Fungus Damage	30	27	3	90.00%	
	Good	150	150	0	100.00%	
	Insect Damage	30	28	2	93.33%	
	Sour	30	29	1	96.67%	
2	Black	30	30	0	100.00%	96.42%
	Broken	30	27	3	90.00%	
	Dried Cherry	30	29	1	96.67%	
	Floater	30	30	0	100.00%	
	Fungus Damage	30	27	3	90.00%	
	Good	150	149	1	99.33%	
	Insect Damage	30	29	1	96.67%	
	Sour	30	29	1	96.67%	
3	Black	30	29	1	96.67%	96.00%
	Broken	30	26	4	86.67%	
	Dried Cherry	30	30	0	100.00%	
	Floater	30	30	0	100.00%	
	Fungus Damage	30	28	2	93.33%	
	Good	150	150	0	100.00%	
	Insect Damage	30	29	1	96.67%	
	Sour	30	29	1	96.67%	



4	Black	30	30	0	100.00%	96.25%
	Broken	30	27	3	90.00%	
	Dried Cherry	30	30	0	100.00%	
	Floater	30	29	1	96.67%	
	Fungus Damage	30	27	3	90.00%	
	Good	150	150	0	100.00%	
	Insect Damage	30	28	2	93.33%	
	Sour	30	30	0	100.00%	
5	Black	30	30	0	100.00%	96.33%
	Broken	30	26	4	86.67%	
	Dried Cherry	30	30	0	100.00%	
	Floater	30	30	0	100.00%	
	Fungus Damage	30	28	2	93.33%	
	Good	150	149	1	99.33%	
	Insect Damage	30	29	1	96.67%	
	Sour	30	29	1	96.67%	
Overall Accuracy Across All Trials						96.17%

Fig. 6.7 Actual Performance of ViT in the System (Per-Classification)

1434 In Table 6.7, the performance of the Vision Transformer (ViT) model across all five trials
 1435 were presented. The recorded data shows that ViT was more consistent than the YOLOv12,
 1436 achieving very high accuracy scores in classifying Black, Dried Cherry, Floater, Sour, and
 1437 Good beans. It was observed that the model even achieved perfect accuracies on these
 1438 classifications on some trials. On the other hand, the model's performance on classifying
 1439 Broken beans was slightly lower but still reliable, achieving an accuracy between 86-90

1440 Table 6.5 presents the performance of YOLOv12 on classifying good and defective
 1441 beans under varying test conditions. Compared to the model's per-classification perfor-
 1442 mance, the results demonstrated higher accuracy when classifying good and defective
 1443 beans. The model achieved an average accuracy of 94.8% across all trials with varying test



TABLE 6.5 ACTUAL PERFORMANCE OF YOLOv12 IN THE SYSTEM (GOOD VS. DEFECT)

Test Condition	TP	TN	FP	FN	Accuracy (%)
100% Good Beans	89	0	0	11	89
75% Good, 25% Defective	71	24	1	4	95
50% Good, 50% Defective	49	49	1	1	98
25% Good, 75% Defective	23	72	3	2	95
100% Defective Beans	0	97	3	0	97
Average Accuracy					94.8

1444 conditions. These findings indicate that YOLOv12 can be a reliable model when simply
 1445 sorting between the two classifications.

TABLE 6.7 ACTUAL PERFORMANCE OF ViT IN THE SYSTEM (GOOD VS. DEFECT)

Test Condition	TP	TN	FP	FN	Accuracy (%)
100% Good Beans	100	0	0	0	100
75% Good, 25% Defective	72	24	1	3	96
50% Good, 50% Defective	48	50	0	2	98
25% Good, 75% Defective	23	75	0	2	98
100% Defective Beans	0	98	2	0	98
Average Accuracy					98

1446 On the other hand, Table 6.7 presents the data on the performance of ViT model under
 1447 varying proportions of good and defective beans. The model achieved an average accuracy
 1448 of 98% across all five trials, which was similar to its per-classification accuracy. Thus,
 1449 showing reliability and consistency on its accuracy on both per-classification testing (defect
 1450 types) and binary testing (good and defect).



1451

6.4 Sorting Speed

TABLE 6.9 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

1452

Table 6.9 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% good beans, 80:20, and 100% defective beans. The minimal drop to 21 beans per minute under the 70% good and 30% defective condition could be due to the long wait time for the beans to fall onto the inspection tray. On the other hand, the peak sorting rate of 24 beans per minute under the 50:50 condition indicates that the system's classification and actuations were synchronized. Overall, the prototype proves to have stable and consistent sorting throughput. The results confirm that the system can work at a steady speed appropriate for small-scale processing operations without degradation in performance by the number of defective beans.



Trial	Category	Samples	Correct	Accuracy	Overall Accuracy
1	Dense	53	50	94.34%	92.06%
	Less Dense	10	8	80.00%	
2	Dense	53	49	92.45%	90.48%
	Less Dense	10	8	80.00%	
3	Dense	53	48	90.57%	90.48%
	Less Dense	10	9	90.00%	
4	Dense	53	49	92.45%	90.48%
	Less Dense	10	8	80.00%	
5	Dense	53	50	94.34%	90.48%
	Less Dense	10	7	70.00%	
Average Accuracy					90.89%

Fig. 6.8 Actual Density Sorting Performance

1465 Table 6.8 presents the results of five trials evaluating the density sorting mechanism.
 1466 The system consistently classified dense beans with high accuracy, ranging from 90.57% to
 1467 94.34%, showing that the sorter was generally reliable in distinguishing beans with higher
 1468 density. In contrast, the classification of less dense beans fluctuated more significantly,
 1469 with accuracies between 70.00% and 90.00%, indicating that this category posed greater
 1470 difficulty for the system. Overall accuracies across the trials remained relatively stable,
 1471 ranging from 90.48% to 92.06% with an average accuracy of 90.89%, which demonstrates
 1472 consistent sorting performance but also highlights that occasional misclassifications pre-
 1473 vented the system from reaching perfect accuracy. These findings suggest that while the
 1474 sorter is effective in identifying dense beans, improvements may be needed in refining the
 1475 sensitivity of the system to reliably classify less dense beans.



1476

Chapter 7

1477

CONCLUSIONS, RECOMMENDATIONS, AND FUTURE DIRECTIVES

1478



1479 7.1 Concluding Remarks

1480 The study was able to present the design, development, and actual implementation of a two-
1481 staged automated green coffee bean sorting system, utilizing computer vision and embedded
1482 systems. The design is composed of a rotating conveyor table, a dual-camera inspection
1483 tray, defect sorting mechanism, and density-based sorting mechanism. In addition, five
1484 deep learning-based classification models such as EfficientNetV2, YOLOv8, YOLOv11,
1485 YOLOv12, and ViT were benchmarked. These models were deployed and tested into the
1486 actual defect sorting system with a test dataset of 20 beans per classification, where the
1487 ViT achieved the highest accuracy of 98%. In terms of the sorting speed, the system was
1488 tested in 5 trials, where it achieved an average sorting speed of 22.2 beans per minute. The
1489 system was tested under varying quality distributions and maintained consistent sorting
1490 speeds, thereby confirming its practical viability. Overall, the results indicate that the
1491 integration of deep learning and embedded automation offers a robust and scalable solution
1492 for post-harvest coffee bean quality assessment.

1493 7.2 Contributions

1494 This study contributed to the coffee industry in the Philippines by introducing a two-
1495 stage automated coffee bean sorter that enhances coffee quality assessment by segregating
1496 defective beans and sorting dense and less-dense beans. This system integrates machine
1497 vision and density-based sorting, ensuring that high-quality, dense beans and potential
1498 specialty-grade coffee are selected for further processing. This system can support the
1499 Philippine coffee industry's efforts to enhance product quality and meet global specialty
1500 coffee standards to improve market competitiveness.



7.3 Recommendations

The following are the recommendations for further study of this design:

- Optimize the density-based sorting mechanism
- Improvement of system portability by reducing the overall size and weight of the system

7.4 Future Prospects

This study offers a building block for future innovation in intelligent post-harvest coffee processing. A potential extension is combining cloud-based data storage and analytics for traceability at the batch level and remote monitoring. Another would be the deployment of light inference models on microcontroller units (MCUs) to facilitate real-time, on-device computation, thus minimizing system latency and increasing portability. Additional research might also investigate the use of unsupervised or semi-supervised learning methods to identify new or infrequent defects without depending solely on labeled data. Commercially, the system can be scaled to process greater volumes using modular conveyor lines and parallel sorting stations. These developments would greatly benefit coffee producers by providing consistent, efficient, and objective bean quality assessment.



1517

REFERENCES

- 1518 [Akbar et al., 2021] Akbar, M., Rachmawati, E., and Sthevanie, F. (2021). Visual feature and
1519 machine learning approach for arabica green coffee beans grade determination. In *Proceedings*
1520 of the 6th International Conference on Communication and Information Processing, ICCIP '20,
1521 page 97–104, New York, NY, USA. Association for Computing Machinery.
- 1522 [Amadea et al., 2024] Amadea, V., Rachmawati, E., Ferdian, E., and Akbar, M. N. S. (2024).
1523 Defect detection in arabica green coffee beans based on grade quality. In *Proceedings of the*
1524 2024 10th International Conference on Computing and Artificial Intelligence, ICCAI '24, pages
1525 103–110, New York, NY, USA. Association for Computing Machinery.
- 1526 [Arboleda et al., 2020] Arboleda, E. R., Fajardo, A. C., and Medina, R. P. (2020). Green coffee
1527 beans feature extractor using image processing. *TELKOMNIKA (Telecommunication Computing*
1528 *Electronics and Control)*, 18(4):2027–2034.
- 1529 [Balay et al., 2024] Balay, D., Cabrera, R., Jensen, J., and Mayuga, K. (2024). *Automatic Sorting*
1530 *of Defective Coffee Beans through Computer Vision*. PhD thesis, De La Salle University.
- 1531 [Balbin et al., 2020] Balbin, J. R., Del Valle, C. D., Lopez, V. J. L. G., and Quiambao, R. F. (2020).
1532 Grading and profiling of coffee beans for international standards using integrated image
1533 processing algorithms and back-propagation neural network. In *2020 IEEE 12th International*
1534 *Conference on Humanoid, Nanotechnology, Information Technology, Communication and Con-*
1535 *trol, Environment, and Management (HNICEM)*, pages 1–6.
- 1536 [Bali and Tyagi, 2020] Bali, S. and Tyagi, S. S. (2020). Evaluation of transfer learning techniques
1537 for classifying small surgical dataset. In *2020 10th International Conference on Cloud Computing,*
1538 *Data Science & Engineering (Confluence)*, pages 744–750.
- 1539 [Barbosa et al., 2019] Barbosa, M. d. S. G., Scholz, M. B. d. S., Kitzberger, C. S. G., and Benassi,
1540 M. d. T. (2019). Correlation between the composition of green arabica coffee beans and the
1541 sensory quality of coffee brews. *Food Chemistry*, 292:275–280.
- 1542 [Bureau of Agriculture and Fisheries Standards, 2012] Bureau of Agriculture and Fisheries Stan-
1543 dards (2012). Green coffee beans – specifications.
- 1544 [Bureau of Agriculture and Fisheries Standards, 2022] Bureau of Agriculture and Fisheries Stan-
1545 dards (2022). Green coffee bean sorter — specifications.
- 1546 [Córdoba et al., 2021] Córdoba, N., Moreno, F. L., Osorio, C., Velásquez, S., Fernandez-Alduenda,
1547 M., and Ruiz-Pardo, Y. (2021). Specialty and regular coffee bean quality for cold and hot
1548 brewing: Evaluation of sensory profile and physicochemical characteristics. *LWT*, 145:111363.
- 1549 [da Cruz et al., 2006] da Cruz, A. G., Cenci, S. A., and Maia, M. C. A. (2006). Quality assurance
1550 requirements in produce processing. *Trends in Food Science & Technology*, 17(8):406–411.
- 1551 [Dabek et al., 2022] Dabek, P., Krot, P., Wodecki, J., Zimroz, P., Szrek, J., and Zimroz, R. (2022).



- 1552 Measurement of idlers rotation speed in belt conveyors based on image data analysis for diagnostic purposes. *Measurement*, 202:111869.
- 1553
- 1554 [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).
- 1555
- 1556 [Datov and Lin, 2019] Datov, A. and Lin, Y.-C. (2019). Classification and grading of green coffee beans in asia.
- 1557
- 1558 [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.
- 1559
- 1560
- 1561 [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 manufaturing (ijidem).
- 1562
- 1563
- 1564 [Dosovitskiy et al., 2021] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. (arXiv:2010.11929). arXiv:2010.11929 [cs].
- 1565
- 1566
- 1567
- 1568 [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.
- 1569
- 1570
- 1571 [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 (*coffeea arabica* l.). *International Journal of Food Properties*, 22(1):758–767.
- 1572
- 1573
- 1574 [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.
- 1575
- 1576
- 1577 [International Coffee Association, 2023] International Coffee Association (2023). Summary coffee report & outlook december 2023.
- 1578
- 1579 [International Organization for Standardization, 1995] International Organization for Standardization (1995). Green and roasted coffee — determination of free-flow bulk density of whole beans (routine method).
- 1580
- 1581
- 1582 [International Organization for Standardization, 2007] International Organization for Standardization (2007). Safety of machinery – risk assessment – part 2: Practical guidance and examples of methods.
- 1583
- 1584
- 1585 [International Organization for Standardization, 2010] International Organization for Standardization (2010). Safety of machinery – general principles for design – risk assessment and risk reduction.
- 1586
- 1587
- 1588 [International Organization for Standardization, 2015] International Organization for Standardiza-



- 1589 tion (2015). Systems and software engineering – systems and software quality requirements and
 1590 evaluation (square) – measurement of data quality.
- 1591 [International Organization for Standardization, 2019] International Organization for Standardiza-
 1592 tion (2019). Information technology — development of user interface accessibility part 1: Code
 1593 of practice for creating accessible ict products and services.
- 1594 [International Organization for Standardization, 2022] International Organization for Standardiza-
 1595 tion (2022). Framework for artificial intelligence (ai) systems using machine learning (ml).
- 1596 [Lee and Tai, 2020] Lee, W.-C. and Tai, P.-L. (2020). Defect detection in striped images using a
 1597 one-dimensional median filter. *Applied Sciences*, 10(3):1012.
- 1598 [Lualhati et al., 2022] Lualhati, A. J. N., Mariano, J. B., Torres, A. E. L., and Fenol, S. D. (2022).
 1599 Development and testing of green coffee bean quality sorter using image processing and artificial
 1600 neural network. *Mindanao Journal of Science and Technology*, 20(1).
- 1601 [Luis et al., 2022] Luis, V. A. M., Quinones, M. V. T., and Yumang, A. N. (2022). Classification of
 1602 defects in robusta green coffee beans using yolo. In *ResearchGate*.
- 1603 [Minglani et al., 2020] Minglani, D., Sharma, A., Pandey, H., Dayal, R., Joshi, J. B., and Subrama-
 1604 niam, S. (2020). A review of granular flow in screw feeders and conveyors. *Powder Technology*,
 1605 366:369–381.
- 1606 [Muchtar et al., 2025] Muchtar, K., Hafifah, Y., Febriana, A., Dawood, R., Ahmadiar, A., Bahri,
 1607 A., Lin, C.-Y., and Yohannes, E. (2025). Edge ai-based detection for defective coffee beans using
 1608 deep learning and streamlit framework. *IEEE Access*, 13:67977–67992.
- 1609 [Pragathi and Jacob, 2024] Pragathi, S. P. and Jacob, L. (2024). Arabica coffee bean grading into
 1610 specialty and commodity type based on quality using visual inspection. *ResearchGate*.
- 1611 [Santos and Baltazar, 2022] Santos, D. T. and Baltazar, M. D. (2022). *The Philippine Coffee*
 1612 *Industry Roadmap, 2021-2025*. Department of Agriculture, Bureau of Agricultural Research.
 1613 Google-Books-ID: QkBT0AEACAAJ.
- 1614 [Santos et al., 2020] Santos, F., Rosas, J., Martins, R., Araújo, G., Viana, L., and Gonçalves, J.
 1615 (2020). Quality assessment of coffee beans through computer vision and machine learning
 1616 algorithms. *Coffee Science - ISSN 1984-3909*, 15:e151752–e151752.
- 1617 [Srisang et al., 2019] Srisang, N., Chanpaka, W., and Chungcharoen, T. (2019). The performance
 1618 of size grading machine of robusta green coffee bean using oscillating sieve with swing along
 1619 width direction. *IOP Conference Series: Earth and Environmental Science*, 301(1):012037.
- 1620 [Susanibar et al., 2024] Susanibar, G., Ramirez, J., Sanchez, J., and Ramirez, R. (2024). Devel-
 1621 opment of an automated machine for green coffee beans classification by size and defects —
 1622 request pdf. *ResearchGate*.
- 1623 [Tampon, 2023] Tampon, V. (2023). 63 coffee statistics you need to know for 2024 and beyond.
- 1624 [Wu et al., 2024] Wu, L., Hao, H.-Y., and Song, Y. (2024). A review of metal surface defect
 1625 detection based on computer vision. *Acta Automatica Sinica*.



De La Salle University

1626

Produced: September 26, 2025, 23:53



De La Salle University

1627

1628

Appendix A STUDENT RESEARCH ETHICS CLEARANCE



De La Salle University

1629

RESEARCH ETHICS CLEARANCE FORM¹

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.

Dr. Francisco D. Baltasar

Name and Signature of Adviser/Mentor:

Date: April 8, 2017

Noted by:

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.



De La Salle University

1630

1631

Appendix B ANSWERS TO QUESTIONS TO THIS THESIS





- 1632 **B1 How important is the problem to practice?**
- 1633 **B2 How will you know if the solution/s that you will achieve would be better than existing ones?**
- 1635 **B2.1 How will you measure the improvement/s?**
- 1636 **B2.1.1 What is/are your basis/bases for the improvement/s?**
- 1637 **B2.1.2 Why did you choose that/those basis/bases?**
- 1638 **B2.1.3 How significant are your measure/s of the improvement/s?**
- 1639 **B3 What is the difference of the solution/s from existing ones?**
- 1640
- 1641 **B3.1 How is it different from previous and existing ones?**
- 1642 **B4 What are the assumptions made (that are behind for your proposed solution to work)?**
- 1643
- 1644 **B4.1 Will your proposed solution/s be sensitive to these assumptions?**
- 1645
- 1646 **B4.2 Can your proposed solution/s be applied to more general cases when some assumptions are eliminated? If so, how?**
- 1647
- 1648 **B5 What is the necessity of your approach / proposed solution/s?**
- 1649
- 1650 **B5.1 What will be the limits of applicability of your proposed solution/s?**
- 1651
- 1652 **B5.2 What will be the message of the proposed solution to technical people? How about to non-technical managers and business people?**
- 1653
- 1654
- 1655 **B6 How will you know if your proposed solution/s is/are correct?**
- 1656
- 1657 **B6.1 Will your results warrant the level of mathematics used (i.e., will the end justify the means)?**
- 1658



De La Salle University

1671

Appendix C REVISIONS TO THE PROPOSAL

1672



De La Salle University

- 1673 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.
- 1674
- 1675 1. Examiner
- 1676 2. Comment
- 1677 3. Summary of how the comment was addressed
- 1678 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	<p>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.</p> <p>2. First itemtext</p> <p>3. Second itemtext</p> <p>4. Last itemtext</p> <p>5. First itemtext</p> <p>6. Second itemtext</p>	<p>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.</p> <p>2. First itemtext</p> <p>3. Second itemtext</p> <p>4. Last itemtext</p> <p>5. First itemtext</p> <p>6. Second itemtext</p>	<p>Sec. ?? on p. ??, Sec. ?? on p. ??, Fig. ?? on p. ??</p>

Continued on next page

C. Revisions to the Proposal



De La Salle University

Continued from previous page

Examiner	Comment	Summary of how the comment was addressed	Locations
Dr. Amado Z. Hernandez	<p>Dr. Amado Z. Hernandez's comment is a long, dense paragraph of Latin placeholder text (Lorem ipsum). It discusses various aspects of a document, such as the introduction of new concepts, the development of arguments, and the conclusion of sections. The text is structured into several paragraphs separated by periods.</p> <p>The summary of how the comment was addressed is also a long, dense paragraph of Latin placeholder text. It provides a detailed response to Dr. Hernandez's points, likely involving multiple pages of analysis and synthesis. The summary is presented in a single, continuous block of text.</p>	<p>First itemtext</p> <p>Second itemtext</p> <p>Last itemtext</p> <p>First itemtext</p> <p>Second itemtext</p>	<p>Sec. ?? on p. ??, Sec. ?? on p. ??, Fig. ?? on p. ??</p>

Continued on next page



Continued from previous page

Examiner	Comment	Summary of how the comment was addressed	Locations
Dr. Jose Y. Alonzo	<p>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.</p> <ul style="list-style-type: none"> • First itemtext • Second itemtext • Last itemtext • First itemtext • Second itemtext 	<p>Sec. ?? on p. ??, Sec. ?? on p. ??, Fig. ?? on p. ??</p>	

Continued on next page

C. Revisions to the Proposal



De La Salle University

Continued from previous page

Examiner	Comment	Summary of how the comment was addressed	Locations
Dr. Mariana X. Mercado	<p> 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.</p>	<p>1. First itemtext 2. Second itemtext 3. Last itemtext 4. First itemtext 5. Second itemtext</p>	<p>Sec. ?? on p. ??, Sec. ?? on p. ??, Fig. ?? on p. ??</p>

Continued on next page



De La Salle University

Continued from previous page



De La Salle University

1679

1680

Appendix D REVISIONS TO THE FINAL



- 1681 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.
- 1682
- 1683 1. Examiner
- 1684 2. Comment
- 1685 3. Summary of how the comment has been addressed
- 1686 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. Cabatuan	<p>1. First itemtext</p> <p>2. Second itemtext</p> <p>3. Last itemtext</p> <p>4. First itemtext</p> <p>5. Second itemtext</p> <p>First itemtext</p> <p>Second itemtext</p> <p>Last itemtext</p> <p>First itemtext</p> <p>Second itemtext</p>	<p>1. First itemtext</p> <p>2. Second itemtext</p> <p>3. Last itemtext</p> <p>4. First itemtext</p> <p>5. Second itemtext</p>	<p>Sec. ?? on p. ??, Sec. ?? on p. ??, Fig. ?? on p. ??</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 2. Second itemtext 3. Last itemtext 4. First itemtext 5. Second itemtext	1. First itemtext 2. Second itemtext 3. Last itemtext 4. First itemtext 5. Second itemtext First itemtext Second itemtext Last itemtext First itemtext Second itemtext	Sec. ?? on p. ??, Sec. ?? on p. ??, Fig. ?? on p. ???
Dr. Jose Y. Alonzo	1. First itemtext 2. Second itemtext 3. Last itemtext 4. First itemtext 5. Second itemtext	1. First itemtext 2. Second itemtext 3. Last itemtext 4. First itemtext 5. Second itemtext • First itemtext • Second itemtext • Last itemtext • First itemtext • 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 2. Second itemtext 3. Last itemtext 4. First itemtext 5. Second itemtext	1. First itemtext 2. Second itemtext 3. Last itemtext 4. First itemtext 5. Second itemtext	Sec. ?? on p. ??, Sec. ?? on p. ??, Fig. ?? on p. ???
Dr. Rafael W. Sison	1. First itemtext 2. Second itemtext 3. Last itemtext 4. First itemtext 5. Second itemtext	1. First itemtext 2. Second itemtext 3. Last itemtext 4. First itemtext 5. Second itemtext	Sec. ?? on p. ??, Sec. ?? on p. ??, Fig. ?? on p. ???



De La Salle University

1687

Appendix E USAGE EXAMPLES

1688



1689 The user is expected to have a working knowledge of L^AT_EX. A good introduction is
 1690 in [?]. Its latest version can be accessed at <http://www.ctan.org/tex-archive/info/lshort>.

1691 E1 Equations

1692 The following examples show how to typeset equations in L^AT_EX. This section also shows
 1693 examples of the use of `\gls{ }` commands in conjunction with the items that are in
 1694 the `notation.tex` file. **Please make sure that the entries in `notation.tex` are**
 1695 **those that are referenced in the L^AT_EX document files used by this Thesis. Please**
 1696 **comment out unused notations and be careful with the commas and brackets in**
 1697 **`notation.tex` .**

1698 In (E.1), the output signal $y(t)$ is the result of the convolution of the input signal $x(t)$
 1699 and the impulse response $h(t)$.

$$y(t) = h(t) * x(t) = \int_{-\infty}^{+\infty} h(t - \tau) x(\tau) d\tau \quad (\text{E.1})$$

1700 Other example equations are as follows.

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \quad (\text{E.2})$$

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

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



1701

The verbatim L^AT_EX code of Sec. E1 is in List. E.1.

Listing E.1: Sample L^AT_EX code for equations and notations usage

```

1 The following examples show how to typeset equations in \LaTeX. This
2 section also shows examples of the use of \verb| \gls{ } | commands
3 in conjunction with the items that are in the \verb| notation.tex |
4 file. \textbf{Please make sure that the entries in} \verb| notation.tex |
5 \textbf{| are those that are referenced in the \LaTeX \
6 document files used by this \documentType. Please comment out
7 unused notations and be careful with the commas and brackets in} \verb|
8 \verb| notation.tex |.
9
10 In \eqref{eq:conv}, the output signal \gls{not:output_sigt} is the
11 result of the convolution of the input signal \gls{not:input_sigt}
12 and the impulse response \gls{not:ir}.
13
14 \begin{eqnarray}
15     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
16 \label{eq:conv}
17 \end{eqnarray}
18 Other example equations are as follows.
19
20 \begin{eqnarray}
21     \left[ \frac{V_1}{I_1} \right] = \begin{bmatrix} A & B \\ C & D \end{bmatrix}
22 \label{eq:ABCD}
23 \end{eqnarray}
24
25 \begin{eqnarray}
26 \frac{1}{2} < \left\lfloor \mod{\left\lfloor \frac{y}{17} \right\rfloor}{2^{17}} \right\rfloor - \left\lfloor \mod{\left\lfloor \frac{y}{17} \right\rfloor}{2} \right\rfloor,
27 \end{eqnarray}
28
29 \begin{eqnarray}
30 |\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}}
31 \geq 1
32 \end{eqnarray}
```



1702 E2 Notations

1703 In order to use the standardized notation, the user is highly suggested to see the ISO 80000-2
 1704 standard [?].

1705 See https://en.wikipedia.org/wiki/Help:Displaying_a_formula and https://en.wikipedia.org/wiki/List_of_mathematical_symbols for L^AT_EX maths and other notations, respectively.

1706 The following were taken from `isomath-test.tex`.

1708 E2.1 Math alphabets

1709 If there are other symbols in place of Greek letters in a math alphabet, it uses T1 or OT1
 1710 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	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$
mathrm	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$
mathbf	$\mathbf{A}, \mathbf{B}, \mathbf{\Gamma}, \mathbf{\Delta}, \mathbf{\Theta}, \mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Pi}, \mathbf{\Sigma}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Omega}, ff, fi, \mathbf{\beta}, ^!, \mathbf{v}, \mathbf{w}, 0, 1, 9$
mathsf	$\mathsf{A}, \mathsf{B}, \mathsf{\Gamma}, \mathsf{\Delta}, \mathsf{\Theta}, \mathsf{\Lambda}, \mathsf{\Xi}, \mathsf{\Pi}, \mathsf{\Sigma}, \mathsf{\Phi}, \mathsf{\Psi}, \mathsf{\Omega}, ff, fi, \mathsf{\beta}, ^!, \mathsf{v}, \mathsf{w}, 0, 1, 9$
mathtt	$\mathtt{A}, \mathtt{B}, \mathtt{\Gamma}, \mathtt{\Delta}, \mathtt{\Theta}, \mathtt{\Lambda}, \mathtt{\Xi}, \mathtt{\Pi}, \mathtt{\Sigma}, \mathtt{\Phi}, \mathtt{\Psi}, \mathtt{\Omega}, \mathtt{ff}, \mathtt{fi}, \mathtt{\beta}, ^!, \mathtt{v}, \mathtt{w}, 0, 1, 9$

1711 New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-italic.

mathbfit	$\mathbf{\textit{A}}, \mathbf{\textit{B}}, \mathbf{\textit{\Gamma}}, \mathbf{\textit{\Delta}}, \mathbf{\textit{\Theta}}, \mathbf{\textit{\Lambda}}, \mathbf{\textit{\Xi}}, \mathbf{\textit{\Pi}}, \mathbf{\textit{\Sigma}}, \mathbf{\textit{\Phi}}, \mathbf{\textit{\Psi}}, \mathbf{\textit{\Omega}}, \mathbf{\textit{\alpha}}, \mathbf{\textit{\beta}}, \mathbf{\textit{\pi}}, \mathbf{\textit{\nu}}, \mathbf{\textit{\omega}}, \mathbf{\textit{v}}, \mathbf{\textit{w}}, \mathbf{\textit{o}}, \mathbf{\textit{1}}, \mathbf{\textit{9}}$
mathsfit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$
mathsfbf	$\mathsf{\textit{A}}, \mathsf{\textit{B}}, \mathsf{\textit{\Gamma}}, \mathsf{\textit{\Delta}}, \mathsf{\textit{\Theta}}, \mathsf{\textit{\Lambda}}, \mathsf{\textit{\Xi}}, \mathsf{\textit{\Pi}}, \mathsf{\textit{\Sigma}}, \mathsf{\textit{\Phi}}, \mathsf{\textit{\Psi}}, \mathsf{\textit{\Omega}}, \mathsf{\textit{\alpha}}, \mathsf{\textit{\beta}}, \mathsf{\textit{\pi}}, \mathsf{\textit{\nu}}, \mathsf{\textit{\omega}}, \mathsf{\textit{v}}, \mathsf{\textit{w}}, \mathbf{\textit{o}}, \mathbf{\textit{1}}, \mathbf{\textit{9}}$

1712 Do the math alphabets match?

1713 $ax\alpha\omega ax\alpha\omega ax\alpha\omega \quad TC\Theta\Gamma TC\Theta\Gamma TC\Theta\Gamma$

1714 E2.2 Vector symbols

1715 Alphabetic symbols for vectors are boldface italic, $\lambda = e_1 \cdot a$, while numeric ones (e.g.
 1716 the zero vector) are bold upright, $a + 0 = a$.

1717 E2.3 Matrix symbols

1718 Symbols for matrices are boldface italic, too:¹ $\Lambda = E \cdot A$.

¹However, matrix symbols are usually capital letters whereas vectors are small ones. Exceptions are physical quantities like the force vector F or the electrical field E .



1719 **E2.4 Tensor symbols**

1720 Symbols for tensors are sans-serif bold italic,

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

1721 The permittivity tensor describes the coupling of electric field and displacement:

$$\mathbf{D} = \epsilon_0 \epsilon_r \mathbf{E}$$



1722	E2.5 Bold math version												
1723	The “bold” math version is selected with the commands <code>\boldmath</code> or <code>\mathversion{bold}</code>												
	<table> <tr> <td>mathnormal</td><td>$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$</td></tr> <tr> <td>mathit</td><td>$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$</td></tr> <tr> <td>mathrm</td><td>$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$</td></tr> <tr> <td>mathbf</td><td>$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$</td></tr> <tr> <td>mathsf</td><td>$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$</td></tr> <tr> <td>mathtt</td><td>$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$</td></tr> </table>	mathnormal	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$	mathit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$	mathrm	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$	mathbf	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$	mathsf	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$	mathtt	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$
mathnormal	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$												
mathit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$												
mathrm	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$												
mathbf	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$												
mathsf	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$												
mathtt	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$												
1724	<p>New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-italic.</p> <table> <tr> <td>mathbfit</td><td>$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$</td></tr> <tr> <td>mathsfit</td><td>$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$</td></tr> <tr> <td>mathsfbfit</td><td>$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$</td></tr> </table>	mathbfit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$	mathsfit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$	mathsfbfit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$						
mathbfit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$												
mathsfit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$												
mathsfbfit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$												
1725	Do the math alphabets match?												
1726	$a x \alpha \omega a x \alpha \omega a x \alpha \omega \quad T C \Theta \Gamma T C \Theta \Gamma T C \Theta \Gamma$												
1727	E2.5.1 Vector symbols												
1728	Alphabetic symbols for vectors are boldface italic, $\lambda = e_1 \cdot a$, while numeric ones (e.g.												
1729	the zero vector) are bold upright, $a + 0 = a$.												
1730	E2.5.2 Matrix symbols												
1731	Symbols for matrices are boldface italic, too: ² $\Lambda = E \cdot A$.												
1732	E2.5.3 Tensor symbols												
1733	Symbols for tensors are sans-serif bold italic,												
	$\alpha = e \cdot a \iff \alpha_{ijl} = e_{ijk} \cdot a_{kl}.$												
1734	The permittivity tensor describes the coupling of electric field and displacement:												
	$D = \epsilon_0 \epsilon_r E$												

²However, matrix symbols are usually capital letters whereas vectors are small ones. Exceptions are physical quantities like the force vector F or the electrical field E .



1735 The verbatim L^AT_EX code of Sec. E2 is in List. E.2.

Listing E.2: Sample L^AT_EX code for notations usage

```

1  % A teststring with Latin and Greek letters::
2  \newcommand{\teststring}{%
3    % capital Latin letters
4    % A,B,C,
5    A,B,
6    % capital Greek letters
7    %\Gamma,\Delta,\Theta,\Lambda,\Xi,\Pi,\Sigma,\Upsilon,\Phi,\Psi,
8    \Gamma,\Delta,\Theta,\Lambda,\Xi,\Pi,\Sigma,\Upsilon,\Phi,\Psi,\Omega,
9    % small Greek letters
10   \alpha,\beta,\pi,\nu,\omega,
11   % small Latin letters:
12   % compare \nu, \omega, v, and w
13   v,w,
14   % digits
15   0,1,9
16 }

17
18
19 \subsection{Math alphabets}
20
21 If there are other symbols in place of Greek letters in a math
22 alphabet, it uses T1 or OT1 font encoding instead of OML.
23
24 \begin{eqnarray*}
25 \mbox{\rmfamily} & & \teststring \\
26 \mbox{\itshape} & & \mathit{\teststring} \\
27 \mbox{\rmrm} & & \mathrm{\teststring} \\
28 \mbox{\rmbf} & & \mathbf{\teststring} \\
29 \mbox{\rmss} & & \mathsf{\teststring} \\
30 \mbox{\rmtt} & & \mathsf{\teststring} \\
31 \end{eqnarray*}
32 New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-
33 italic.
34 \begin{eqnarray*}
35 \mathbf{\teststring} & & \mathbf{\teststring} \\
36 \mathsf{\teststring} & & \mathsf{\teststring} \\
37 \mathsf{\teststring} & & \mathsf{\teststring} \\
38 \end{eqnarray*}
39 %
40 Do the math alphabets match?
41 $
42 \mathnormal{a x \alpha \omega}
43 \mathbf{a x \alpha \omega}
44 \mathsf{\teststring}
45 \quad
46 \mathsf{\teststring}
47 \mathbf{a x \alpha \omega}
48 \mathnormal{T C \Theta \Gamma}
49 $
50
51 \subsection{Vector symbols}
52

```



```

1790 53 Alphabetic symbols for vectors are boldface italic,
1791 54  $\vec{\lambda} = \vec{e}_1 \cdot \vec{a}$ ,
1792 55 while numeric ones (e.g. the zero vector) are bold upright,
1793 56  $\vec{a} + \vec{0} = \vec{a}$ .
1794 57
1795 58 \subsection{Matrix symbols}
1796 59
1797 60 Symbols for matrices are boldface italic, too: %
1798 61 \footnote{However, matrix symbols are usually capital letters whereas
1799 62 vectors
1800 63 are small ones. Exceptions are physical quantities like the force
1801 64 vector  $\vec{F}$  or the electrical field  $\vec{E}$ .%}
1802 65  $\mathtt{\Lambda} = \text{matrixsym}{\Lambda}$ 
1803 66
1804 67
1805 68 \subsection{Tensor symbols}
1806 69
1807 70 Symbols for tensors are sans-serif bold italic,
1808 71
1809 72 \[
1810 73   \text{tensorsym}{\alpha} = \text{tensorsym}{e} \cdot \text{tensorsym}{a}
1811 74   \quad \Longleftarrow \quad
1812 75   \alpha_{ijl} = e_{ijk} \cdot a_{kl}.
1813 76 \]
1814 77
1815 78
1816 79 The permittivity tensor describes the coupling of electric field and
1817 80 displacement: \[
1818 81 \vec{D} = \epsilon_0 \text{tensorsym}{\epsilon}_{\mathrm{r}} \vec{E} \]
1819 82
1820 83
1821 84
1822 85 \newpage
1823 86 \subsection{Bold math version}
1824 87
1825 88 The ‘‘bold’’ math version is selected with the commands
1826 89 \verb+\boldmath+ or \verb+\mathversion{bold}+
1827 90
1828 91 {\boldmath
1829 92   \begin{eqnarray*}
1830 93     \mathnormal{} & & \text{teststring} \\
1831 94     \mathit{} & & \mathit{\text{teststring}} \\
1832 95     \mathrm{} & & \mathrm{\text{teststring}} \\
1833 96     \mathbf{} & & \mathbf{\text{teststring}} \\
1834 97     \mathsf{} & & \mathsf{\text{teststring}} \\
1835 98     \mathtt{} & & \mathtt{\text{teststring}} \\
1836 99   \end{eqnarray*}
1837 100   New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-
1838 101   italic.
1839 102   \begin{eqnarray*}
1840 103     \mathbf{\mathit{}} & & \mathbf{\mathit{\text{teststring}}} \\
1841 104     \mathsf{\mathit{}} & & \mathsf{\mathit{\text{teststring}}} \\
1842 105     \mathtt{\mathit{}} & & \mathtt{\mathit{\text{teststring}}}
1843 106   \end{eqnarray*}
1844 107   %
1845 108
1846 109   Do the math alphabets match?

```



```

1847 108
1848 109   $
1849 110   \mathnormal{a} \mathbf{x} \alpha \omega
1850 111   \mathbf{a} \mathbf{x} \alpha \omega
1851 112   \mathsf{a} \mathbf{x} \alpha \omega
1852 113   \quad
1853 114   \mathsf{T} \mathbf{C} \Theta \Gamma
1854 115   \mathbf{T} \mathbf{C} \Theta \Gamma
1855 116   \mathnormal{T} \mathbf{C} \Theta \Gamma
1856 117   $
1857 118
1858 119   \subsection{Vector symbols}
1859 120
1860 121   Alphabetic symbols for vectors are boldface italic,
1861 122   $ \vec{\lambda} = \vec{e}_1 \cdot \vec{a} $,
1862 123   while numeric ones (e.g. the zero vector) are bold upright,
1863 124   $ \vec{a} + \vec{0} = \vec{a} $.
1864 125
1865 126
1866 127
1867 128
1868 129   \subsection{Matrix symbols}
1869 130
1870 131   Symbols for matrices are boldface italic, too: %
1871 132   \footnote{However, matrix symbols are usually capital letters whereas
1872   vectors
1873   are small ones. Exceptions are physical quantities like the force
1874   vector $ \vec{F} $ or the electrical field $ \vec{E} $. %}
1875 135
1876 136   $ \mathbf{\Lambda} = \mathbf{E} \cdot \mathbf{A} . $
1877 137
1878 138
1879 139   \subsection{Tensor symbols}
1880 140
1881 141   Symbols for tensors are sans-serif bold italic,
1882 142
1883 143   \[
1884 144   \mathbf{\alpha} = \mathbf{e} \cdot \mathbf{a}
1885 145   \quad \Longleftarrow \quad
1886 146   \alpha_{ijl} = e_{ijk} \cdot a_{kl}.
1887 147 \]
1888 148
1889 149   The permittivity tensor describes the coupling of electric field and
1890 150   displacement: \[
1891 151   \vec{D} = \epsilon_0 \mathbf{\epsilon}(\mathbf{r}) \cdot \vec{E} \]
1892 152 \}

```



E3 Abbreviation

This section shows examples of the use of L^AT_EX commands 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 L^AT_EX 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 `\glsp`. 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).



- 1923 • Provide your own link text: style sheet.

1924 The verbatim L^AT_EX code of Sec. E3 is in List. E.3.

Listing E.3: Sample L^AT_EX code for abbreviations usage

```

1 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:
2
3 \begin{itemize}
4
5   \item \acr{xml} and \acr{css}.
6
7   \item Next use: \acr{xml} and \acr{css}.
8
9   \item Full form: \gls{xml} and \gls{css}.
10
11  \item Reset again. \glsresetall{abbreviation}
12
13  \item Start with a capital. \Acr{html}.
14
15  \item Next: \Acr{html}. Full: \Gls{html}.
16
17  \item Prefer capitals? \renewcommand{\acronymfont}[1]{\
      \MakeTextUppercase{#1}} \Acr{xml}. Next: \acr{xml}. Full: \gls{xml} \
    .
18
19  \item Prefer small-caps? \renewcommand{\acronymfont}[1]{\textsc{#1}} \
      \Acr{css}. Next: \acr{css}. Full: \gls{css}.
20
21  \item Resetting all acronyms.\glsresetall{abbreviation}
22
23  \item Here are the acronyms again:
24
25  \item \Acr{html}, \acr{xml} and \acr{css}.
26
27  \item Next use: \Acr{html}, \acr{xml} and \acr{css}.
28
29  \item Full form: \Gls{html}, \gls{xml} and \gls{css}.
30
31  \item Provide your own link text: \glslink{[textbf]css}{style}
32
33 \end{itemize}
```



1925 E4 Glossary

1926 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).

1929 **Please make sure that the entries in `notation.tex` are those that are referenced in the L^AT_EX document files used by this Thesis. Please comment out unused notations and be careful with the commas and brackets in `notation.tex`.**

- 1932 • Matrices are usually denoted by a bold capital letter, such as \mathbf{A} . The matrix’s (i, j) th element is usually denoted a_{ij} . Matrix \mathbf{I} is the identity matrix.
- 1933 • A set, denoted as \mathcal{S} , is a collection of objects.
- 1934 • The universal set, denoted as \mathcal{U} , is the set of everything.
- 1935 • The empty set, denoted as \emptyset , contains no elements.
- 1936 • Functional Analysis is seen as the study of complete normed vector spaces, i.e., Banach spaces.
- 1937 • The cardinality of a set, denoted as $|\mathcal{S}|$, is the number of elements in the set.

1939 The verbatim L^AT_EX code for the part of Sec. E4 is in List. E.4.

Listing E.4: Sample L^AT_EX code for glossary and notations usage

```

1 \begin{itemize}
2
3   \item \Glspl{matrix} are usually denoted by a bold capital letter,
4       such as $\mathbf{A}$. The \gls{matrix}'s $(i,j)$th element is
5       usually denoted $a_{ij}$. \Gls{matrix} $\mathbf{I}$ is the
6       identity \gls{matrix}.
7
8   \item A set, denoted as \gls{not:set}, is a collection of objects.
9
10  \item The universal set, denoted as \gls{not:universalSet}, is the
11      set of everything.
12
13  \item The empty set, denoted as \gls{not:emptySet}, contains no
14      elements.
15
16  \item \Gls{Functional Analysis} is seen as the study of complete
17      normed vector spaces, i.e., Banach spaces.
18
19  \item The cardinality of a set, denoted as \gls{not:cardinality}, is
20      the number of elements in the set.
21
22 \end{itemize}

```



De La Salle University

1941

E5 Figure

1942

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

1943



Fig. E.1 A quadrilateral image example.



1944 Fig. E.1 is a gray box enclosed by a dark border. List. E.5 shows the corresponding
 1945 L^AT_EX code.

Listing E.5: Sample L^AT_EX code for a single figure

```

1 \begin{figure}[!htbp]
2   \centering
3   \includegraphics[width=0.5\textwidth]{example}
4   \caption{A quadrilateral image example.}
5   \label{fig:example}
6 \end{figure}
7 \cleardoublepage
8
9 Fig.~\ref{fig:example} is a gray box enclosed by a dark border. List.~\ref{lst:onefig} shows the corresponding \LaTeX \ code.
10 \end{figure}
```



De La Salle University



(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 L^AT_EX code.

Listing E.6: Sample L^AT_EX code for three figures on top of each other

```
1 \begin{figure} [!htbp]
2   \centering
3   \subbottom[A sub-figure in the top row.]{%
4     \includegraphics [width=0.35\textwidth]{example_gray_box}
5     \label{fig:top}
6   }
7   \vfill
8   \subbottom[A sub-figure in the middle row.]{%
9     \includegraphics [width=0.35\textwidth]{example_gray_box}
10    \label{fig:mid}
11  }
12  \vfill
13  \subbottom[A sub-figure in the bottom row.]{%
14    \includegraphics [width=0.35\textwidth]{example_gray_box}
15    \label{fig:botm}
16  }
17  \caption{Figures on top of each other}
18  \label{fig:tmb}
19 \end{figure}
```



De La Salle University



(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 L^AT_EX code.



Listing E.7: Sample L^AT_EX code for the four figures

```

1 \begin{figure} [!htbp]
2 \centering
3 \subbottom[A sub-figure in the upper-left corner.]{
4 \includegraphics [width=0.45\textwidth]{example_gray_box}
5 \label{fig:upprleft}
6 }
7 \hfill
8 \subbottom[A sub-figure in the upper-right corner.]{
9 \includegraphics [width=0.45\textwidth]{example_gray_box}
10 \label{fig:uppright}
11 }
12 \vfill
13 \subbottom[A sub-figure in the lower-left corner.]{
14 \includegraphics [width=0.45\textwidth]{example_gray_box}
15 \label{fig:lowerleft}
16 }
17 \hfill
18 \subbottom[A sub-figure in the lower-right corner.]{
19 \includegraphics [width=0.45\textwidth]{example_gray_box}
20 \label{fig:lowright}
21 }
22 \caption{Four figures in each corner. See List.\ref{lst:fourfigs} for
the corresponding \LaTeX \ code.}
23 \label{fig:fourfig}
24 \end{figure}

```



1946

E6 Table

1947

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)



1949

List. E.8 shows the corresponding L^AT_EX code.

Listing E.8: Sample L^AT_EX code for making typical table environment

```

1 \begin{center}
2 {\scriptsize
3 \begin{tabularx}{\textwidth}{p{0.1\textwidth}|p{0.2\textwidth}|p{0.5\textwidth}}
4 \caption{Feasible triples for highly variable grid} \label{tab:triple_grid} \\
5 \hline
6 \hline
7 \textbf{Time (s)} &
8 \textbf{Triple chosen} &
9 \textbf{Other feasible triples} \\
10 \hline
11 \endfirsthead
12 \multicolumn{3}{c}{\textit{Continued from previous page}} \\
13 \hline
14 \hline
15 \hline
16 \textbf{Time (s)} &
17 \textbf{Triple chosen} &
18 \textbf{Other feasible triples} \\
19 \hline
20 \endhead
21 \hline
22 \multicolumn{3}{r}{\textit{Continued on next page}} \\
23 \endfoot
24 \hline
25 \endlastfoot
26 \hline
27
28 0 & (1, 11, 13725) & (1, 12, 10980), (1, 13, 8235), (2, 2, 0), (3, 1, 0) \\
29 2745 & (1, 12, 10980) & (1, 13, 8235), (2, 2, 0), (2, 3, 0), (3, 1, 0) \\
30 5490 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
31 8235 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
32 10980 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
33 13725 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
34 16470 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
35 19215 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
36 21960 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
37 24705 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
38 27450 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
39 30195 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
40 32940 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
41 35685 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
42 38430 & (1, 13, 10980) & (2, 2, 2745), (2, 3, 0), (3, 1, 0)

```



De La Salle University

```

2004 43 | 41175 & (1, 12, 13725) & (1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1,
2005   0) \\
2006 44 | 43920 & (1, 13, 10980) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2007 45 | 46665 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
2008 46 | 49410 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
2009 47 | 52155 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1,
2010   0) \\
2011 48 | 54900 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2012 49 | 57645 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2013 50 | 60390 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2014 51 | 63135 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2015 52 | 65880 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2016 53 | 68625 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
2017 54 | 71370 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2018 55 | 74115 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2019 56 | 76860 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2020 57 | 79605 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2021 58 | 82350 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2022 59 | 85095 & (1, 12, 13725) & (1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1,
2023   0) \\
2024 60 | 87840 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2025 61 | 90585 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2026 62 | 93330 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2027 63 | 96075 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2028 64 | 98820 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2029 65 | 101565 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2030 66 | 104310 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2031 67 | 107055 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2032 68 | 109800 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2033 69 | 112545 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3,
2034   1, 0) \\
2035 70 | 115290 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2036 71 | 118035 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2037 72 | 120780 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2038 73 | 123525 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2039 74 | 126270 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3,
2040   1, 0) \\
2041 75 | 129015 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
2042 76 | 131760 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
2043 77 | 134505 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2044 78 | 137250 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2045 79 | 139995 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
2046 80 | 142740 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
2047 81 | 145485 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3,
2048   1, 0) \\
2049 82 | 148230 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
2050 83 | 150975 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2051 84 | 153720 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2052 85 | 156465 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2053 86 | 159210 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2054 87 | 161955 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2055 88 | 164700 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
2056 89 | \end{tabularx}
2057 90 |
2058 91 | \end{center}

```



2060

E7 Algorithm or Pseudocode Listing

2061

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 L^AT_EX code.

2062

2063

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 L^AT_EX code for algorithm or pseudocode listing usage

```

1 \begin{table} [!htbp]
2   \caption{Calculation of $y = x^n$}
3   \label{tab:calcxn}
4   \footnotesize
5   \begin{tabular}{lll}
6     \hline
7     \hline
8     {\bf Input(s):} & & \\
9     $n$ & : & $n$th power; $n \in \mathbb{Z}^{+}$ \\
10    $x$ & : & base value; $x \in \mathbb{R}^{+}$ \\
11    \hline
12    {\bf Output(s):} & & \\
13    $y$ & : & result; $y \in \mathbb{R}^{+}$ \\
14    \hline
15    \hline
16    \\
17  \end{tabular}
18 }
19 \begin{algorithmic}[1]
20 \footnotesize
21   \REQUIRE $n \geq 0 \vee x \neq 0$ \\
22   \ENSURE $y = x^n$ \\
23   \STATE $y \Leftarrow 1$ \\
24   \IF{$n < 0$}
25     \STATE $X \Leftarrow 1 / x$ \\
26     \STATE $N \Leftarrow -n$ \\
27   \ELSE
28     \STATE $X \Leftarrow x$ \\
29     \STATE $N \Leftarrow n$ \\
30   \ENDIF \\
31   \WHILE{$N \neq 0$}
32     \IF{$N$ is even}
33       \STATE $X \Leftarrow X \times X$ \\
34       \STATE $N \Leftarrow N / 2$ \\
35     \ELSE[$N$ is odd]
36       \STATE $y \Leftarrow y \times X$ \\
37       \STATE $N \Leftarrow N - 1$ \\
38     \ENDIF \\
39   \ENDWHILE \\
40 }
41 \end{algorithmic}
42 \end{table}

```



2064

E8 Program/Code Listing

2065

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`)

```

1  /* fibo.c -- It prints out the first N Fibonacci
2   *          numbers.
3   */
4
5  #include <stdio.h>
6
7  int main(void) {
8      int n;           /* Number of fibonacci numbers we will print */
9      int i;           /* Index of fibonacci number to be printed next */
10     int current;    /* Value of the (i)th fibonacci number */
11     int next;        /* Value of the (i+1)th fibonacci number */
12     int twoaway;    /* Value of the (i+2)th fibonacci number */
13
14     printf("How many Fibonacci numbers do you want to compute? ");
15     scanf("%d", &n);
16     if (n<=0)
17         printf("The number should be positive.\n");
18     else {
19         printf("\n\n\tI\tFibonacci(I)\n\t=====\\n");
20         next = current = 1;
21         for (i=1; i<=n; i++) {
22             printf("\t%d\t%d\\n", i, current);
23             twoaway = current+next;
24             current = next;
25             next = twoaway;
26         }
27     }
28 }
29
30 /* The output from a run of this program was:
31
32 How many Fibonacci numbers do you want to compute? 9
33
34 I      Fibonacci(I)
35 =====
36 1      1
37 2      1
38 3      2
39 4      3
40 5      5
41 6      8
42 7      13
43 8      21
44 9      34
45
46 */

```



2067

List. E.11 shows the corresponding L^AT_EX code.

Listing E.11: Sample L^AT_EX code for program listing

1 `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.



2068 E9 Referencing

2069 Referencing chapters: This appendix is in Appendix E, which is about examples in using
 2070 various \LaTeX commands.

2071 Referencing sections: This section is Sec. E9, which shows how to refer to the locations
 2072 of various labels that have been placed in the \LaTeX files. List. E.12 shows the corresponding
 2073 \LaTeX code.

Listing E.12: Sample \LaTeX code for referencing sections

1 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.

2074 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem.
 2075 Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec
 2076 ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus
 2077 placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor.
 2078 Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla
 2079 tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue
 2080 a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris.
 2081 Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit
 2082 amet ipsum. Nunc quis urna dictum turpis accumsan semper.



2083 E9.1 A subsection

2084 Referencing subsections: This section is Sec. E9.1, which shows how to refer to a subsection.
 2085 List. E.13 shows the corresponding L^AT_EX code.

Listing E.13: Sample L^AT_EX code for referencing subsections

```
1 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.
```

2086 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem.
 2087 Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec
 2088 ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus
 2089 placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor.
 2090 Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla
 2091 tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue
 2092 a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris.
 2093 Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit
 2094 amet ipsum. Nunc quis urna dictum turpis accumsan semper.



2095 **E9.1.1 A sub-subsection**

2096 Referencing sub-subsections: This section is Sec. E9.1.1, which shows how to refer to a
 2097 sub-subsection. List. E.14 shows the corresponding L^AT_EX code.

Listing E.14: Sample L^AT_EX code for referencing sub-subsections

1 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.

2098 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem.
 2099 Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec
 2100 ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus
 2101 placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor.
 2102 Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla
 2103 tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue
 2104 a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris.
 2105 Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit
 2106 amet ipsum. Nunc quis urna dictum turpis accumsan semper.



2107 E10 Citing

2108 Citing bibliography content is done using BibTeX. It requires the creation of a BibTeX
2109 file (.bib extension name), and then added in the argument of \bibliography{ } . For
2110 each .bib file, separate them by a comma in the argument of \bibliography{ } without
2111 the extension name. Building your BibTeX file (references.bib) can be done easily with a
2112 tool called JabRef (www.jabref.org).
2113

The following subsections are examples of citations.

2114 E10.1 Books

2115 • [?]

2116 • [?]

2117 • [?]

2118 • [?]

2119 • [?]

2120 • [?]

2121 • [?]

2122 • [?]

2123 • [?]

2124 • [?]

2125 • [?]

2126 • [?]

2127 • [?]

2128 • [?]

2129 • [?]

2130 • [?]

2131 • [?]

2132 • [?]



De La Salle University

- 2133 • [?]
- 2134 • [?]
- 2135 • [?]
- 2136 • [?]
- 2137 • [?]
- 2138 • [?]
- 2139 • [?]
- 2140 • [?]
- 2141 • [?]
- 2142 • [?]
- 2143 • [?]
- 2144 • [?]
- 2145 • [?]
- 2146 • [?]
- 2147 • [?]
- 2148 • [?]
- 2149 • [?]
- 2150 • [?]
- 2151 • [?]
- 2152 • [?]
- 2153 • [?]
- 2154 • [?]
- 2155 • [?]
- 2156 • [?]
- 2157 • [?]
- 2158 • [?]

**E10.2 Booklets**

- [?]

E10.3 Proceedings

- [?]

E10.4 In books

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]



De La Salle University

- 2182 • [?]
- 2183 • [?]
- 2184 • [?]
- 2185 • [?]
- 2186 • [?]
- 2187 • [?]
- 2188 • [?]
- 2189 • [?]

E10.5 In proceedings

- 2191 • [?]
- 2192 • [?]
- 2193 • [?]
- 2194 • [?]
- 2195 • [?]
- 2196 • [?]
- 2197 • [?]

E10.6 Journals

- 2199 • [?]
- 2200 • [?]
- 2201 • [?]
- 2202 • [?]
- 2203 • [?]
- 2204 • [?]



De La Salle University

- 2205 • [?]
- 2206 • [?]
- 2207 • [?]
- 2208 • [?]
- 2209 • [?]
- 2210 • [?]
- 2211 • [?]
- 2212 • [?]
- 2213 • [?]
- 2214 • [?]
- 2215 • [?]
- 2216 • [?]
- 2217 • [?]
- 2218 • [?]
- 2219 • [?]
- 2220 • [?]
- 2221 • [?]
- 2222 • [?]
- 2223 • [?]
- 2224 • [?]
- 2225 • [?]
- 2226 • [?]
- 2227 • [?]
- 2228 • [?]



- 2229 • [?]
- 2230 • [?]
- 2231 • [?]
- 2232 • [?]
- 2233 • [?]

E10.7 Theses/dissertations

- 2235 • [?]
- 2236 • [?]
- 2237 • [?]
- 2238 • [?]
- 2239 • [?]
- 2240 • [?]
- 2241 • [?]

E10.8 Technical Reports and Others

- 2243 • [?]
- 2244 • [?]
- 2245 • [?]
- 2246 • [?]
- 2247 • [?]
- 2248 • [?]
- 2249 • [?]
- 2250 • [?]
- 2251 • [?]



- 2252 • [?]
- 2253 • [?]
- 2254 • [?]
- 2255 • [?]
- 2256 • [?]
- 2257 • [?]

E10.9 Miscellaneous

- 2259 • [?]
- 2260 • [?]
- 2261 • [?]
- 2262 • [?]
- 2263 • [?]
- 2264 • [?]
- 2265 • [?]
- 2266 • [?]
- 2267 • [?]
- 2268 • [?]
- 2269 • [?]
- 2270 • [?]
- 2271 • [?]



2272 E11 Index

2273 For key words or topics that are expected (or the user would like) to appear in the Index, use
 2274 `index{key}` , where `key` is an example keyword to appear in the Index. For example,
 2275 Fredholm integral and Fourier operator of the following paragraph are in the Index.

2276 If we make a very large matrix with complex exponentials in the rows (i.e., cosine real
 2277 parts and sine imaginary parts), and increase the resolution without bound, we approach
 2278 the kernel of the Fredholm integral equation of the 2nd kind, namely the Fourier operator
 2279 that defines the continuous Fourier transform.

2280 List. E.15 is a program listing of the above-mentioned paragraph.

Listing E.15: Sample L^AT_EX 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.
```



2281

E12 Adding Relevant PDF Pages

2282

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.

2283

2284

Listing E.16: Sample L^AT_EX code for including PDF pages

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



2285

XILINX.

UltraScale Architecture and Product Overview

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.



2286

XILINX.

UltraScale Architecture and Product Overview**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						
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.



2287

XILINX.

UltraScale Architecture and Product Overview**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
CIMTs (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*

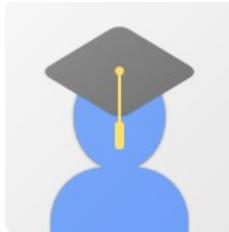
Package ⁽¹⁾⁽²⁾⁽³⁾	Package Dimensions (mm)	VU3P	VU5P	VU7P	VU9P	VU11P	VU13P
		HP, GTY	HP, GTY				
FFVC1517	40x40	520, 40					
FLVF1924	45x45					624, 64	
FLVA2104	47.5x47.5		832, 52	832, 52	832, 52		
FHVA2104	52.5x52.5 ⁽⁴⁾						832, 52
FLVB2104	47.5x47.5		702, 76	702, 76	702, 76	624, 76	
FHVB2104	52.5x52.5 ⁽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:

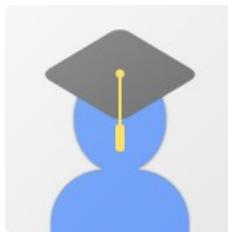
1. Go to [Ordering Information](#) for package designation details.
2. All packages have 1.0mm ball pitch.
3. Packages with the same last letter and number sequence, e.g., A2104, are footprint compatible with all other UltraScale devices with the same sequence. The footprint compatible devices within 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.



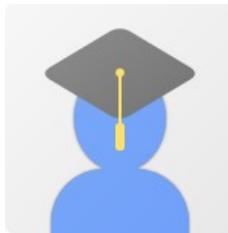
Appendix F VITA



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.

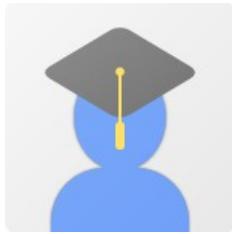


Jiro Renzo D. Tabiolo received the B.Sc., M.Sc., and Ph.D. degrees in chemistry all from the Pamantasan ng Pilipinas, San Juan, Metro Manila, Philippines, in 2020, 2022 and 2025 respectively. He is currently taking up his B.Sc. Computer Engineering studies. He has developed several high-speed packet-switched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



De La Salle University

2308 and node modules. His research interests include high-speed packet-switched networks,
2309 high speed radio interface design, discrete simulation and statistical models for packet
2310 switches.



2311 Ercid Bon B. Valencerina received the B.Sc., M.Sc., and Ph.D.
2312 degrees in chemistry all from the Pamantasan ng Pilipinas, San Juan, Metro Manila,
2313 Philippines, in 2020, 2022 and 2025 respectively. He is currently taking up his B.Sc.
2314 Computer Engineering studies. He has developed several high-speed packet-switched
2315 network systems and node modules. His research interests include high-speed packet-
2316 switched networks, high speed radio interface design, discrete simulation and statistical
2317 models for packet switches.



De La Salle University

2318

Appendix G ARTICLE PAPER(S)

2319

Article/Forum Paper Format

(IEEE LaTeX format)

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

2320

Abstract—The abstract goes here. Lorem ipsum dolor sit amet, consectetur 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, consectetur 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, L^AT_EX, paper, template.

I. INTRODUCTION

THIS demo file is intended to serve as a “starter file” for IEEE article papers produced under L^AT_EX using IEEEtran.cls version 1.8b and later. Lorem ipsum dolor sit amet, consectetur 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, consectetur 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, consectetur 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, consectetur 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.

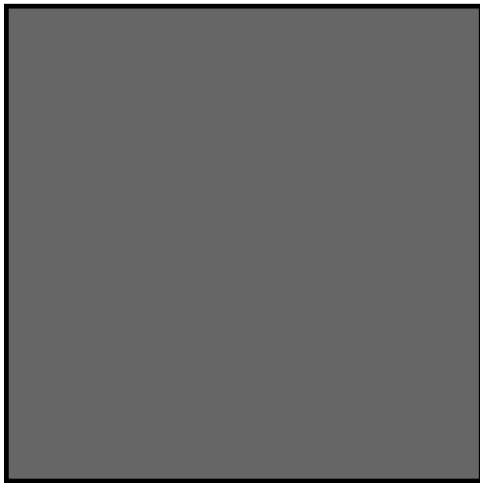
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur 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, consectetur 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,

2321



(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, consectetur 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.

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, consectetur 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].

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, consectetur 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 L^AT_EX 2_& Or L^AT_EX 2_& in 157 minutes.* n.a., 2014.