



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 April, 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

April 1, 2025



38

ABSTRACT

39

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

48

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



TABLE OF CONTENTS

51	Oral Defense Recommendation Sheet	ii
52	Abstract	iii
53	Table of Contents	iv
54	List of Figures	ix
55	List of Tables	xi
56	Abbreviations and Acronyms	xii
57	Notations	xiii
58	Glossary	xiv
59	Chapter 1 INTRODUCTION	1
60	1.1 Background of the Study	2
61	1.2 Prior Studies	3
62	1.3 Problem Statement	7
63	1.4 Objectives and Deliverables	8
64	1.4.1 General Objective (GO)	8
65	1.4.2 Specific Objectives (SOs)	8
66	1.4.3 Expected Deliverables	9
67	1.5 Significance of the Study	11
68	1.5.1 Technical Benefit	11
69	1.5.2 Impact to the Coffee Industry	11
70	1.6 Assumptions, Scope, and Delimitations	12
71	1.6.1 Assumptions	12
72	1.6.2 Scope	12
73	1.6.3 Delimitations	13
74	Chapter 2 LITERATURE REVIEW	14
75	2.1 Existing Work	15
76	2.2 Lacking in the Approaches	23
77	2.3 Summary	25



78	Chapter 3 THEORETICAL CONSIDERATIONS	26
79	3.1 Theoretical Framework	27
80	3.2 Conceptual Framework	27
81	3.3 Quality Assurance Theory	29
82	3.4 Artificial Intelligence Theory	30
83	3.5 Computer Vision Theory	31
84	3.6 Performance Evaluation	32
85	3.7 Existing Technologies and Approaches	33
86	3.8 Density Measurement	33
87	3.9 Summary	34
88	Chapter 4 DESIGN CONSIDERATIONS	35
89	4.1 Mechanical Design	36
90	4.1.1 Screw Feeder	36
91	4.1.2 Rotating Conveyor Table	37
92	4.1.3 Inspection Tray (1st Stage)	38
93	4.1.4 Density Sorter (2nd Stage)	39
94	4.2 Embedded Systems	39
95	4.2.1 Microcontroller	39
96	4.2.2 Sensors	41
97	4.2.3 Motor control	43
98	4.2.4 Operating Voltage	45
99	4.3 Computer Vision System	46
100	4.3.1 Image Processing	46
101	4.3.2 Object Detection and Classification Models	47
102	4.3.3 Object Classification Models	47
103	4.4 Serial Communication	48
104	4.5 Graphical User Interface (GUI)	49
105	4.6 Density Analysis	50
106	4.7 Technical Standards	50
107	4.7.1 Hardware	50
108	4.7.2 Software	51
109	4.7.3 Green Coffee Bean Sorting	52
110	Chapter 5 METHODOLOGY	53
111	5.1 Description of the System	56
112	5.2 Research Design	59
113	5.3 Dataset Collection	59
114	5.3.1 Dataset Collection and Model Training	60
115	5.3.2 Utilization of Open-Source Database	61



116	5.3.3 First Iteration of Dataset Collection	62
117	5.3.4 Second Iteration of Dataset Collection	64
118	5.4 Density Threshold Calibration Using Water Displacement Method	65
119	5.5 Dataset Preparation and Model Training	66
120	5.5.1 Dataset Splitting	66
121	5.5.2 Image Annotation	66
122	5.5.3 Dataset Augmentation Techniques	67
123	5.5.4 Model Evaluation	67
124	5.5.5 Model Benchmarking and Selection	69
125	5.6 Hardware Development	69
126	5.6.1 Screw Feeder	70
127	5.6.2 Rotating Conveyor Table	71
128	5.6.3 Inspection Tray	74
129	5.6.4 Density Sorter	75
130	5.7 Hardware and Software Integration	76
131	5.7.1 Serial Communication	76
132	5.7.2 Recommended Standard 232 (RS-232)	77
133	5.8 Prototype Setup	79
134	5.8.1 Actual Setup	79
135	5.8.2 Lighting Setup for Inspection Tray	81
136	5.8.3 System Operation	85
137	5.9 Prototype Testing	87
138	5.9.1 Sorting Speed	87
139	5.9.2 Defect Sorting Accuracy	88
140	5.9.3 Density Sorting Accuracy	90
141	Chapter 6 RESULTS AND DISCUSSIONS	91
142	6.1 Description of the New Custom Dataset	95
143	6.2 Performance of Classification Models on Custom Dataset	96
144	6.2.1 EfficientNetV2S	97
145	6.2.2 YOLOv8	99
146	6.2.3 YOLOv11-cls	101
147	6.2.4 YOLOv12-cls	103
148	6.3 Actual Performance of Trained Models in the System	104
149	6.4 Sorting Speed	107
150	Chapter 7 CONCLUSIONS, RECOMMENDATIONS, AND FUTURE DI- RECTIVES	109
151	7.1 Concluding Remarks	110
152	7.2 Contributions	110



De La Salle University

154	7.3 Recommendations	111
155	7.4 Future Prospects	111
156	References	112
157	Appendix A STUDENT RESEARCH ETHICS CLEARANCE	115
158	Appendix B ANSWERS TO QUESTIONS TO THIS THESIS	117
159	Appendix C REVISIONS TO THE PROPOSAL	120
160	Appendix D REVISIONS TO THE FINAL	126
161	Appendix E USAGE EXAMPLES	130
162	E1 Equations	131
163	E2 Notations	133
164	E2.1 Math alphabets	133
165	E2.2 Vector symbols	133
166	E2.3 Matrix symbols	133
167	E2.4 Tensor symbols	134
168	E2.5 Bold math version	135
169	E2.5.1 Vector symbols	135
170	E2.5.2 Matrix symbols	135
171	E2.5.3 Tensor symbols	135
172	E3 Abbreviation	139
173	E4 Glossary	141
174	E5 Figure	143
175	E6 Table	149
176	E7 Algorithm or Pseudocode Listing	153
177	E8 Program/Code Listing	155
178	E9 Referencing	157
179	E9.1 A subsection	158
180	E9.1.1 A sub-subsection	159
181	E10 Citing	160
182	E10.1 Books	160
183	E10.2 Booklets	162
184	E10.3 Proceedings	162
185	E10.4 In books	162
186	E10.5 In proceedings	163
187	E10.6 Journals	163



De La Salle University

188	E10.7 Theses/dissertations	165
189	E10.8 Technical Reports and Others	165
190	E10.9 Miscellaneous	166
191	E11 Index	167
192	E12 Adding Relevant PDF Pages	168
193	Appendix F VITA	172
194	Appendix G ARTICLE PAPER(S)	174



195 LIST OF FIGURES

196	3.1 Theoretical Framework	27
197	3.2 Conceptual Framework	28
198	4.1 Screw Feeder Diagram	36
199	4.2 Rotating Conveyor Table 3D Design, 32-inch Rotary Table Accumulator (RTA)	37
200	4.3 Inspector Tray 3D Design	38
201	4.4 Arduino Nano Microcontroller	39
202	4.5 Infrared Sensor	41
203	4.6 TOF10120	42
204	4.7 12V NEMA 17 Stepper Motor	43
205	4.8 6V DC Motor	44
206	4.9 TB6612FNG Motor Driver	44
207	4.10 12V Power Supply	45
208	4.11 MT3608 Step-Up Module	46
209	4.12 C920 Camera	46
210	4.13 Graphical User Interface	49
211	5.1 System Block Diagram	56
212	5.2 Schematic Diagram of the System	57
213	5.3 Design Overview of the System	58
214	5.4 Design and Development Research (DDR) Methodology	59
215	5.5 Manual Sorting Process	60
216	5.6 First Iteration of Data Collection Setup	62
217	5.7 Sample Images from the First Iteration of Dataset Collection	63
218	5.8 Sample Images from the Second Iteration of Dataset Collection	64
219	5.9 Screw Feeder 3D Design	70
220	5.10 Rotating Conveyor Table 3D Design	71
221	5.11 Rotating Conveyor Table with Aluminum Guides	72
222	5.12 Rotating Conveyor Table with IR Sensor	73
223	5.13 Inspection Tray 3D Design	74
224	5.14 Precision Scale	75
225	5.15 Serial Communication Flow for Stage 1 Classification	76
226	5.16 Precision Scale Integration with RS232 for Stage 2 Classification	77
227	5.17 Actual System Setup	79
228	5.18 First Iteration of Lighting Setup	82
229	5.19 Second Iteration of Lighting Setup	83



De La Salle University

230	5.20 Final Iteration of Lighting Setup	84
231	5.21 Top and Bottom View of the Cameras	85
232	6.1 Normalized Confusion Matrix for EfficientNetV2S on Test Dataset	97
233	6.2 Normalized Confusion Matrix for YOLOv8 on Test Dataset	99
234	6.3 Normalized Confusion Matrix for YOLOv11 on Test Dataset	101
235	6.4 Normalized Confusion Matrix for YOLOv12 on Test Dataset	103
236	E.1 A quadrilateral image example.	143
237	E.2 Figures on top of each other. See List. E.6 for the corresponding L ^A T _E X code.	145
238	E.3 Four figures in each corner. See List. E.7 for the corresponding L ^A T _E X code. .	147



239 LIST OF TABLES

240	1.1	Summary of the Literature Review	4
241	1.2	Comparison Table on Existing Studies	6
242	1.3	Expected Deliverables per Objective	10
243	2.1	Review of Related Literature	15
244	2.2	Comparing Proposed Study and Existing Studies	23
245	5.1	Summary of methods for reaching the objectives	54
246	5.2	Sorting Speed Testing Table	87
247	5.3	Good Bean Classification Accuracy Testing Table	88
248	5.4	Specific Defect Classification Accuracy Testing Table	89
249	5.5	Dataset Distribution for Overall Testing	90
250	6.1	Summary of results for achieving the objectives	92
251	6.2	Class Distribution Summary	95
252	6.3	Dataset Split Summary	95
253	6.4	Specific Performance of the Models for Each Defect	104
254	6.5	Model Performance Comparison	107
255	6.6	Sorting Speed Test Conditions	108
256	C.1	Summary of Revisions to the Proposal	121
257	D.1	Summary of Revisions to the Thesis	127
258	E.1	Feasible triples for highly variable grid	149
259	E.2	Calculation of $y = x^n$	153



260 **ABBREVIATIONS**

261	AC	Alternating Current	139
262	CSS	Cascading Style Sheet.....	139
263	HTML	Hyper-text Markup Language	139
264	XML	eXtensible Markup Language	139



265 NOTATION

266	$ \mathcal{S} $	the number of elements in the set \mathcal{S}	141
267	\emptyset	the set with no elements.....	141
268	$h(t)$	impulse response	131
269	\mathcal{S}	a collection of distinct objects	141
270	\mathcal{U}	the set containing everything	141
271	$x(t)$	input signal represented in the time domain	131
272	$y(t)$	output signal represented in the time domain	131

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



279

GLOSSARY

280

Functional Analysis the branch of mathematics concerned with the study of spaces of functions

281

matrix a concise and useful way of uniquely representing and working with linear transformations; a rectangular table of elements



282

LISTINGS

283	E.1 Sample L ^A T _E X code for equations and notations usage	132
284	E.2 Sample L ^A T _E X code for notations usage	136
285	E.3 Sample L ^A T _E X code for abbreviations usage	140
286	E.4 Sample L ^A T _E X code for glossary and notations usage	142
287	E.5 Sample L ^A T _E X code for a single figure	144
288	E.6 Sample L ^A T _E X code for three figures on top of each other	146
289	E.7 Sample L ^A T _E X code for the four figures	148
290	E.8 Sample L ^A T _E X code for making typical table environment	151
291	E.9 Sample L ^A T _E X code for algorithm or pseudocode listing usage	154
292	E.10 Computing Fibonacci numbers	155
293	E.11 Sample L ^A T _E X code for program listing	156
294	E.12 Sample L ^A T _E X code for referencing sections	157
295	E.13 Sample L ^A T _E X code for referencing subsections	158
296	E.14 Sample L ^A T _E X code for referencing sub-subsections	159
297	E.15 Sample L ^A T _E X code for Index usage	167
298	E.16 Sample L ^A T _E X code for including PDF pages	168



De La Salle University

299

Chapter 1

300

INTRODUCTION



301 **1.1 Background of the Study**

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

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



324 among 3 categories: Category 1 includes the most severe issues such as foreign matter
325 and black beans, Category 2 includes less severe defects like broken beans, and Category
326 3 includes minor defects like slight discoloration. Determining the quality of the coffee
327 beans in relation to their defect values is based on quality standards and grading systems
328 such as SCAA protocols guidance or the Philippine National Standard on Green Coffee
329 Bean [Bureau of Agriculture and Fisheries Standards, 2012].

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

343 1.2 Prior Studies

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



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

354

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 EfficientNetV2. Considers classification of 10 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.
--	--	---

355

1.3 Problem Statement

356

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



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

373 **1.4 Objectives and Deliverables**

374 **1.4.1 General Objective (GO)**

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

379 **1.4.2 Specific Objectives (SOs)**

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



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

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

390 **1.4.3 Expected Deliverables**

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



TABLE 1.3 EXPECTED DELIVERABLES PER OBJECTIVE

Objectives	Expected Deliverables
GO: To develop an automated (Arabica) green coffee bean sorter that identifies good, less-dense and defective beans from an unsorted batch of coffee beans. The system will utilize machine vision and density-based analysis for defect detection and classification of the coffee beans, ensuring efficient coffee bean sorting.	A Two-Stage Automated Coffee Bean Sorter System that identifies defective, good beans, and less-dense green coffee bean using machine vision and density-based analysis.
SO1: To gather and create a dataset consisting of 500 high-resolution images of good Arabica green coffee beans and 200 high-resolution images per classification of defective beans (Category 1 & Category 2).	<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



393 **1.5 Significance of the Study**

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

401 **1.5.1 Technical Benefit**

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

410 **1.5.2 Impact to the Coffee Industry**

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



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

417 **1.6 Assumptions, Scope, and Delimitations**

418 **1.6.1 Assumptions**

- 419 1. There would be a defective coffee bean from the green coffee bean test batch;
- 420 2. Identifying the defective coffee beans using the machine vision and density-based
421 analysis would be much more efficient and accurate than manually sorting them;
- 422 3. During testing, test batches will contain 50% good beans and 50% defective beans,
423 60% good beans and 40% defective beans, 70% good beans and 30% defective beans,
424 80% good beans and 20% defective beans, 90% good beans and 10% defective beans,
425 100% good beans;

426 **1.6.2 Scope**

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



431 1.6.3 Delimitations

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



440

Chapter 2

441

LITERATURE REVIEW



442

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.

443

444

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. |
|---|---|



445 **2.3 Summary**

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



462

Chapter 3

463

THEORETICAL CONSIDERATIONS



464

3.1 Theoretical Framework

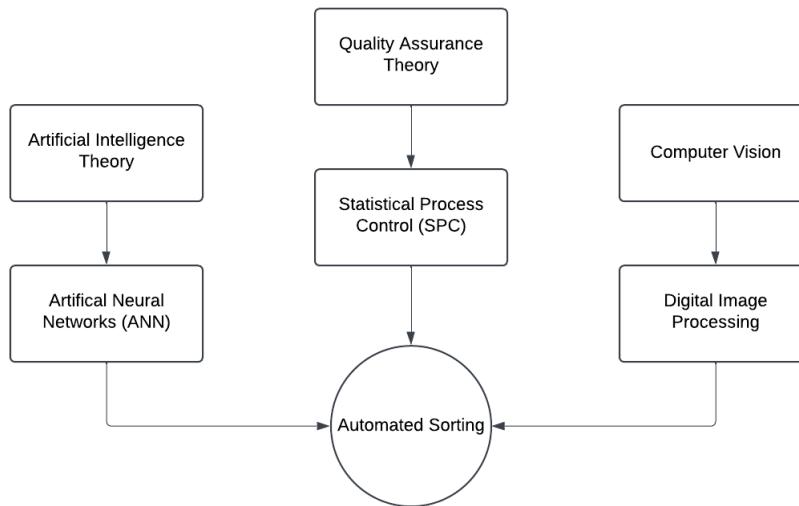


Fig. 3.1 Theoretical Framework

465

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.

469

3.2 Conceptual Framework

470

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

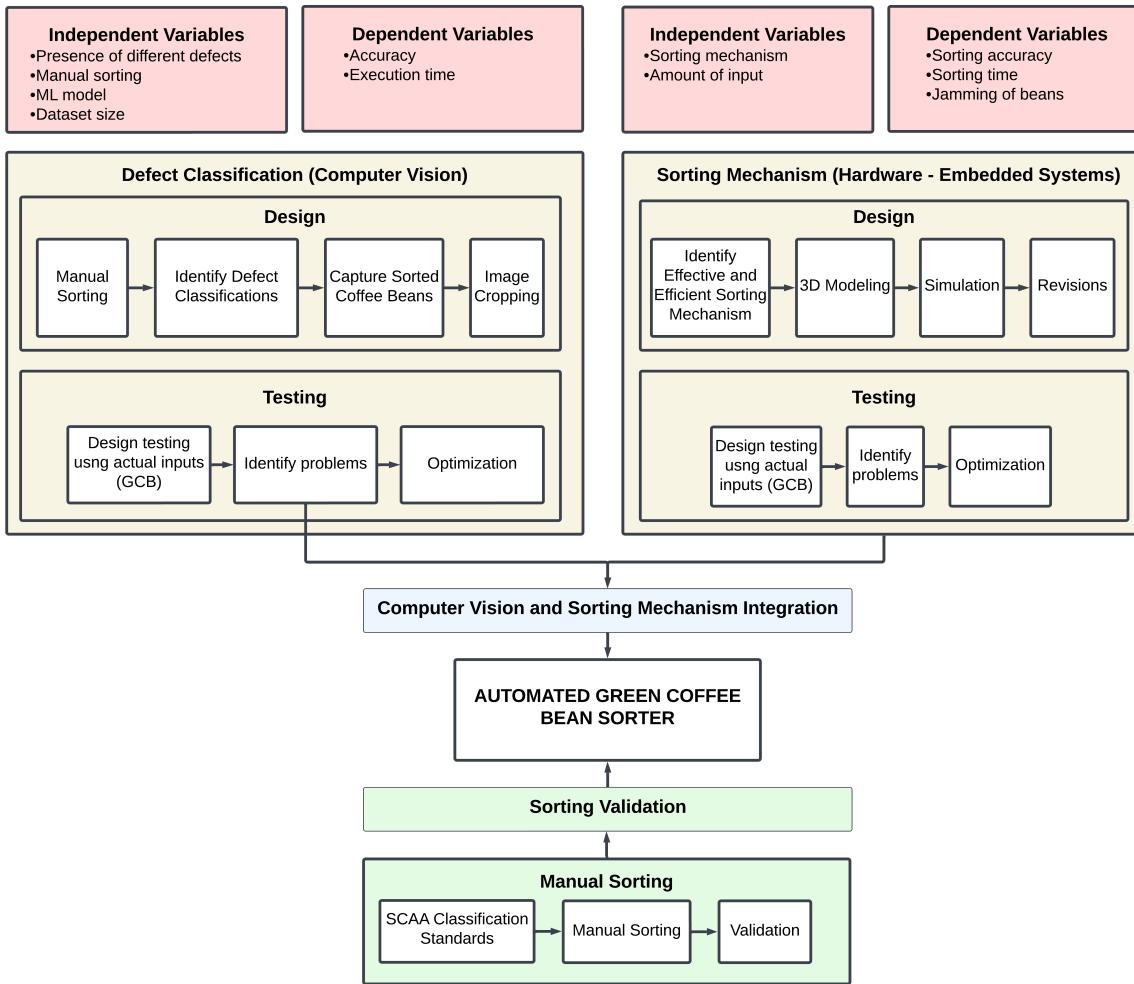


Fig. 3.2 Conceptual Framework

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



477 **3.3 Quality Assurance Theory**

478 Quality assurance theory refers to the set of principles and practices that focuses on estab-
479 lishing a systematic process to ensure that a product or service conforms to a predetermined
480 standard. In the aspect of food and agriculture, there are a number of practices and prin-
481 ciples that ensure the safety and quality of food products. According to [da Cruz et al.,
482 2006], there are a number of practices in place that must be followed, one of which is
483 Good Agricultural Practices, where these procedures are aimed to reduce hazards related to
484 product safety at the farm level. Another one of said practices is the Good manufacturing
485 practice, which were formerly called support programs that provide foundations to the
486 overall food safety management programme. This includes cleaning, maintenance, person-
487 nel training, calibration equipment, quality control, and pest control. Industries that adopt
488 such practices produce the following results, better quality products, greener initiatives
489 and better productivity within a department. Lastly, hazard analysis and critical control
490 points (HACCP), is a science-based system that was created to identify potential hazards
491 and actions to control said hazards. This practice is used to ensure food safety.

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



500 not allowing any primary defects to be present; it should only have less than 5% difference
501 between its sizes. Coffee beans in this grade should also contain a special attribute whether
502 in its body, flavor, aroma, or acidity, and its moisture content should only be in the range
503 of 9-13%. Premium Coffee grade beans should only contain 8 full defects in a sample
504 batch but primary defects are allowed in the sample batch. Similarly to specialty grade
505 coffee beans, its sizes should only contain a 5% difference to one another; it should also
506 contain a special attribute and moisture content should also be similar to its specialty grade
507 counterpart. Exchange coffee grade should contain defects ranging from 9-23 beans in a
508 sample batch, with sizes that can vary up to 50% difference in weight but also only 5% in
509 its sizes. Below standard and off grade coffee beans are classified according to the number
510 of defects present in a sample batch; 24-86 beans for below standard while more than 86
511 beans for off grade. These gradings are used to ensure that quality green coffee beans are
512 produced and ensure that consumers are provided with the best quality available.

513 3.4 Artificial Intelligence Theory

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



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

532 **3.5 Computer Vision Theory**

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

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



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

548 **3.6 Performance Evaluation**

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



563 **3.7 Existing Technologies and Approaches**

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

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

578 **3.8 Density Measurement**

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



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

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

588 3.9 Summary

589 This chapter gives the theoretical and conceptual backgrounds of an automated green coffee
590 bean sorter using Artificial Intelligence (AI), Quality Assurance, and Computer Vision. The
591 theoretical background focuses on key concepts like deep learning models (CNNs, YOLO)
592 used for defect classification, quality assurance principles (GAP, GMP, HACCP) ensuring
593 food safety, and computer vision algorithms (preprocessing, segmentation, and feature
594 extraction) used for image analysis. The conceptual background explains the integration of
595 machine vision for defect detection with embedded systems for sorting, thus conforming to
596 the SCAA coffee grading standards. Performance metrics like accuracy, precision, recall,
597 and F1 score are used for evaluating the performance of the model. Current technologies, for
598 instance, those of [Lualhati et al., 2022] and [Balay et al., 2024], provide insights relevant
599 to image processing and machine learning-based sorting techniques, thus contributing to
600 automated coffee bean classification development.



601

Chapter 4

602

DESIGN CONSIDERATIONS



603 **4.1 Mechanical Design**

604 **4.1.1 Screw Feeder**

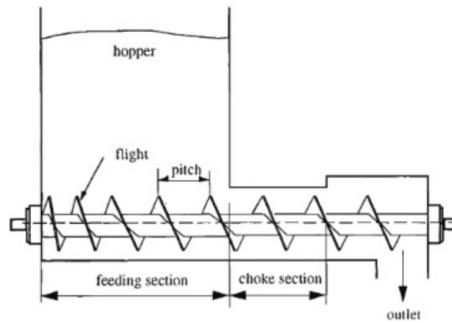


Fig. 4.1 Screw Feeder Diagram

605 Figure 4.1 shows the diagram of a screw feeder. Screw feeders are usually used in
606 industrial fields like agriculture, chemicals, plastics, cements, poultry and food processing.
607 According to [Minglani et al., 2020], screw feeders are specifically used to transport or
608 move granular materials at a controlled rate like corn and wheat. It consists of a rotating
609 screw and small feeding section or the hopper. Despite having big batches of a certain
610 material, screw feeders can control the rate of which these materials are dispensed. With
611 this concept, the group decided to utilize a screw feeder as the input mechanism for the
612 system. This mechanism allows a controlled rate of coffee bean dispensing, which is a
613 significant factor to avoid overcrowding in the rotating conveyor table causing the beans to
614 jam. In addition, batches of coffee beans can be put at once instead of just adding a certain
615 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.

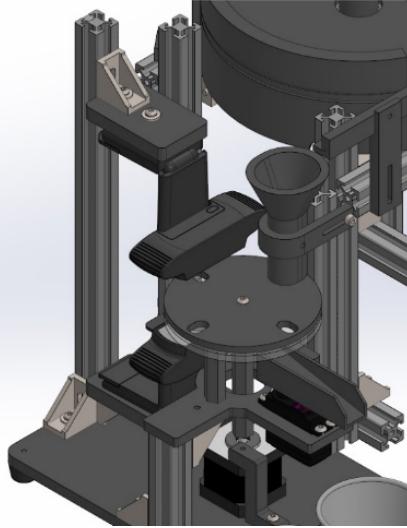


Fig. 4.3 Inspector Tray 3D Design

626 **4.1.3 Inspection Tray (1st Stage)**

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



636 **4.1.4 Density Sorter (2nd Stage)**

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

643 **4.2 Embedded Systems**

644 **4.2.1 Microcontroller**

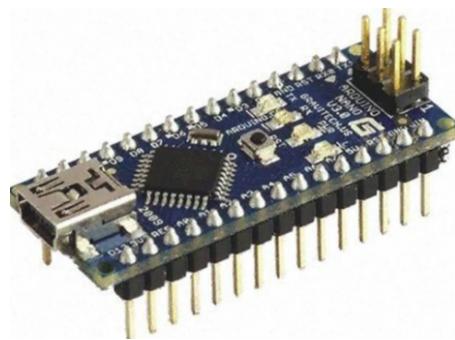


Fig. 4.4 Arduino Nano Microcontroller

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



De La Salle University

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



664

4.2.2 Sensors



Fig. 4.5 Infrared Sensor

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



Fig. 4.6 TOF10120

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



677

4.2.3 Motor control



Fig. 4.7 12V NEMA 17 Stepper Motor

678

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



Fig. 4.8 6V DC Motor

687 For the rotating conveyor table (top feeder), where the beans are initially poured, a
688 6V DC motor is used. The group decided to use this motor due to its high RPM, which
689 is needed for a fast rotation of the rotating conveyor table. The speed of the feeder is
690 regulated to prevent clogging and ensure that the beans are evenly spaced before they
691 enter the inspection tray. The motor speed is fine-tuned through pulse-width modulation
692 (PWM) to synchronize with the stepper motor-driven inspection tray, ensuring a steady
693 input without overwhelming the system.

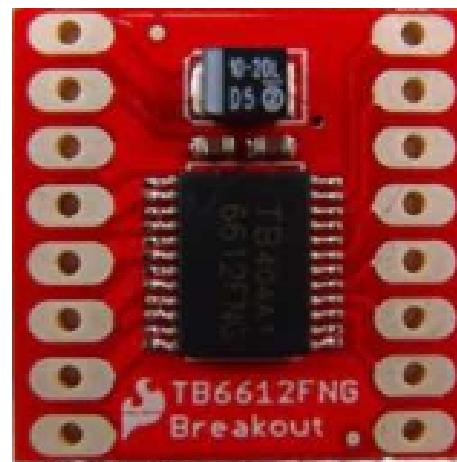


Fig. 4.9 TB6612FNG Motor Driver



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

697 **4.2.4 Operating Voltage**



Fig. 4.10 12V Power Supply

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



Fig. 4.11 MT3608 Step-Up Module

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

710 **4.3 Computer Vision System**

711 **4.3.1 Image Processing**



Fig. 4.12 C920 Camera



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

717 **4.3.2 Object Detection and Classification Models**

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

720 **1. RF-DETR**

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

723 **2. YOLOv11**

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

726 **3. YOLOv12**

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

729 **4.3.3 Object Classification Models**

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



732 **1. EfficientNetV2**

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

735 **2. YOLOv8**

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

738 **3. YOLOv11**

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

741 **4. YOLOv12**

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

744 **4.4 Serial Communication**

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



752

4.5 Graphical User Interface (GUI)



Fig. 4.13 Graphical User Interface

753

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



4.6 Density Analysis

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

4.7 Technical Standards

4.7.1 Hardware

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

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



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

791 Lastly, ISO 30071-1 was standard used to provide sufficient lighting during data
792 collection, and real time bean inspection during sorting process. This standard helps ensure
793 consistent and non-glare lighting conditions, which are essential for the machine vision
794 cameras to accurately capture bean features [International Organization for Standardization,
795 2019]. Uniform illumination improves the reliability of image classification by reducing
796 shadow artifacts and reflections, thereby enhancing overall detection performance.

797 **4.7.2 Software**

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

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



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

812 **4.7.3 Green Coffee Bean Sorting**

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

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



831

Chapter 5

832

METHODOLOGY



TABLE 5.1 SUMMARY OF METHODS FOR REACHING THE OBJECTIVES

Objectives	Methods	Locations
GO: To develop an automated (Arabica) green coffee bean sorter that identifies good, less-dense and defective beans from an unsorted batch of coffee beans. The system will utilize machine vision and density-based analysis for defect detection and classification of the coffee beans, ensuring efficient coffee bean sorting.	<ul style="list-style-type: none"> • DDR Methodology • Description of the System 	Sec. 5.1 on p. 56 Sec. 5.2 on p. 59
SO1: To gather and create a dataset consisting of 500 high-resolution images of good Arabica green coffee beans and 200 high-resolution images per classification of defective beans (Category 1 & Category 2).	<ul style="list-style-type: none"> • Dataset Collection • Manual Sorting 	Sec. 5.3 on p. 59

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. 59 Sec. 5.5 on p. 66
Sec. 5.7.1 on p. 76 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. 5.5 on p. 66
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. 5.4 on p. 65 Sec. 5.6.4 on p. 75



833

5.1 Description of the System

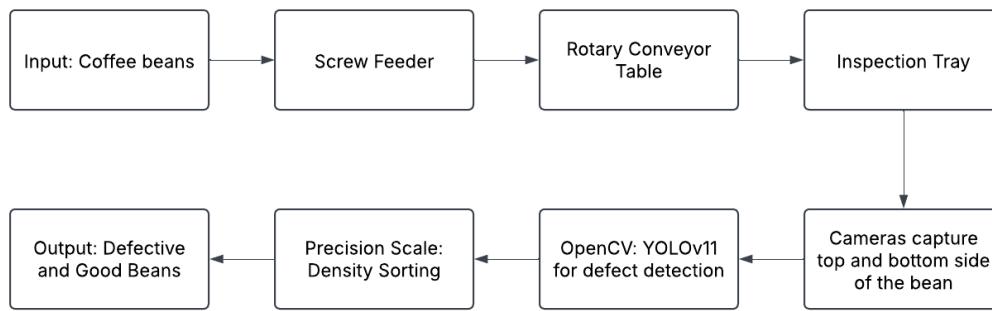


Fig. 5.1 System Block Diagram

834

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

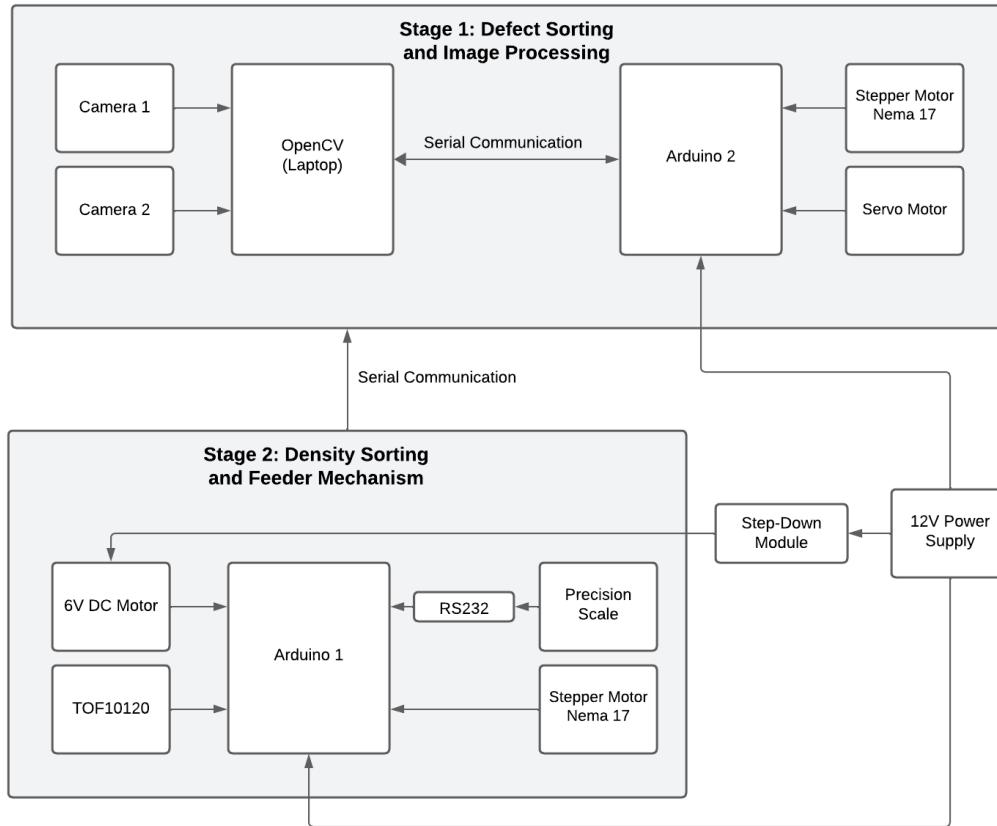


Fig. 5.2 Schematic Diagram of the System

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



853 communication, ensuring smooth coordination.

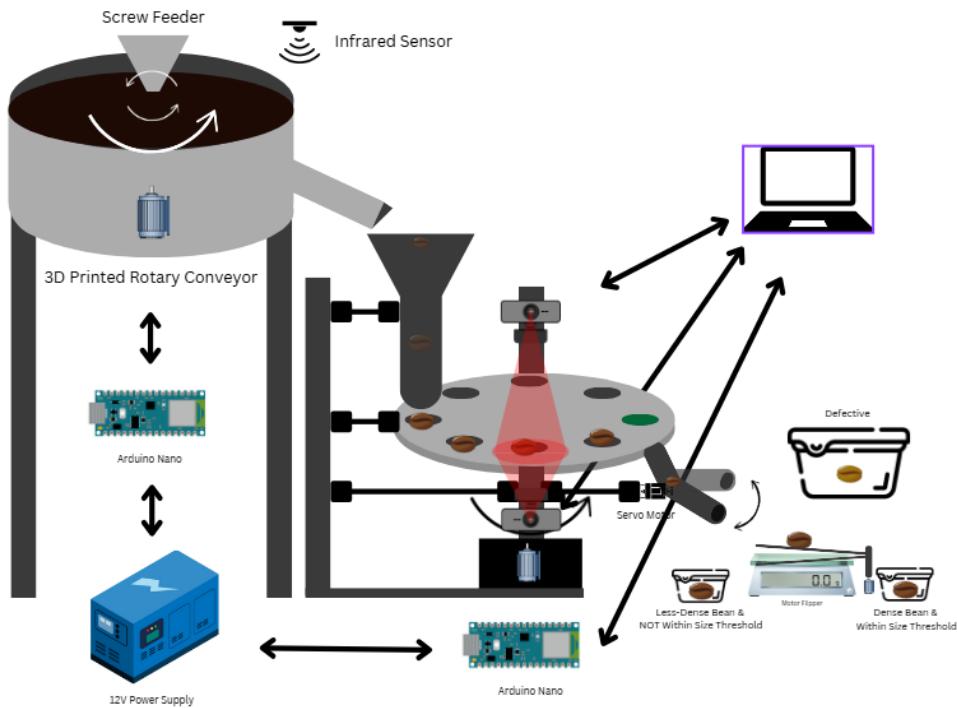


Fig. 5.3 Design Overview of the System

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



861

5.2 Research Design

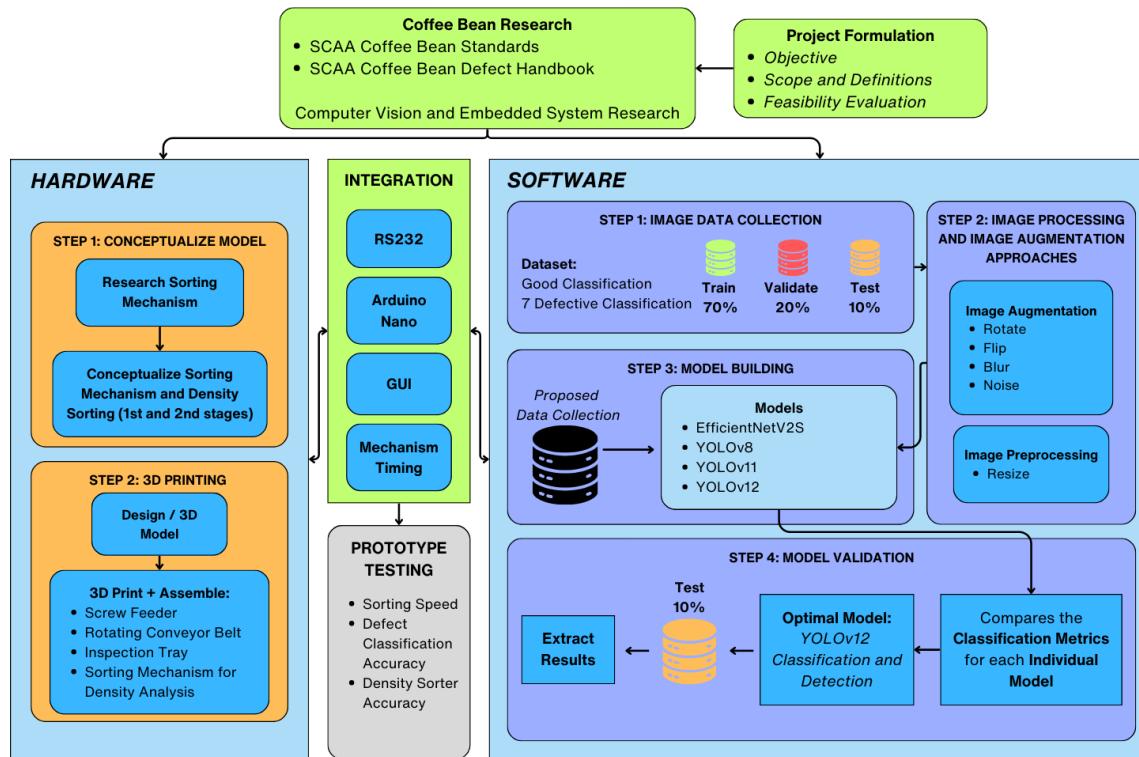


Fig. 5.4 Design and Development Research (DDR) Methodology

862

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

865

5.3 Dataset Collection

866

For dataset collection, Arabica green beans from a farm will be used. Each bean will be captured by a high-resolution camera under sufficient and consistent lighting. Proper lighting is crucial, as it directly affects the visibility of the bean's physical features, minimizing



869 shadows, grain, and other noise that could result from inconsistent illumination. The top
 870 and bottom side pictures of the beans are to be collected. In addition, defective beans of
 871 the same type and origin will be gathered to identify the different classification of defects
 872 (primary and secondary). This study focuses on defects such as Broken, Dried Cherry,
 873 Floater, Full Black, Full Sour, Fungus Damage, and Insect Damage. The dataset will
 874 include at least 500 images of good beans and a minimum of 200 images for each defect
 875 category. To expand the dataset and enhance model training, augmentation techniques such
 876 as scaling, rotation, and mirroring will be applied.

877 5.3.1 Dataset Collection and Model Training

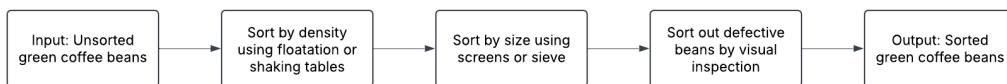


Fig. 5.5 Manual Sorting Process

878 The diagram in Figure 5.5 depicts the representation of the process of manual sorting of
 879 unsorted green coffee beans through a series of steps. First, the beans are sorted by density
 880 using methods such as floatation or shaking tables. This helps in separating the denser
 881 beans, usually pertaining to a more developed and higher quality bean. Then, the beans are
 882 sorted by size using screens and sieves with specific dimensions depending on the variety
 883 of the beans. After this, a thorough visual inspection is performed by the sorters to identify
 884 and remove the defective beans from the batch. To ensure consistency and accuracy, the
 885 group follows the Specialty Coffee Association of America (SCAA) Standards Defect
 886 Handbook, which provide documentation and guidelines for identifying and classifying



887 defective beans. Finally, the process results in the output of sorted green coffee beans,
888 ready for further processing or sale. To ensure the dataset reflects real-world conditions, the
889 group acquired Arabica green coffee beans from Davao. These beans were manually sorted
890 to properly classify defective characteristics before capturing images for dataset creation.
891 This step was crucial for improving the efficiency of batch image capture and ensuring
892 accurate model training, making the system more applicable to Philippine coffee producers.

893 **5.3.2 Utilization of Open-Source Database**

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



901

5.3.3 First Iteration of Dataset Collection



Fig. 5.6 First Iteration of Data Collection Setup



De La Salle University

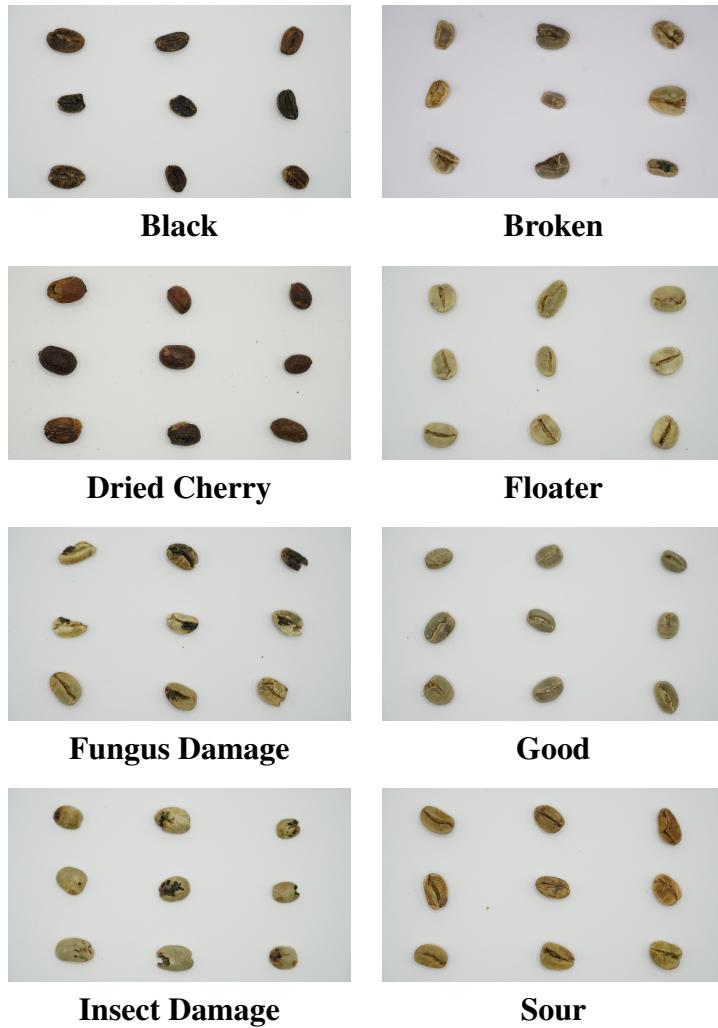


Fig. 5.7 Sample Images from the First Iteration of Dataset Collection

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



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

911 **5.3.4 Second Iteration of Dataset Collection**



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



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

917 **5.4 Density Threshold Calibration Using Water Dis- 918 placement Method**

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

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

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



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

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

938 **5.5 Dataset Preparation and Model Training**

939 **5.5.1 Dataset Splitting**

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

947 **5.5.2 Image Annotation**

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



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

956 **5.5.3 Dataset Augmentation Techniques**

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

962 **5.5.4 Model Evaluation**

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

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



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

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

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

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

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

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

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

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

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

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

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



987 **5.5.5 Model Benchmarking and Selection**

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

993 **5.6 Hardware Development**

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



998

5.6.1 Screw Feeder

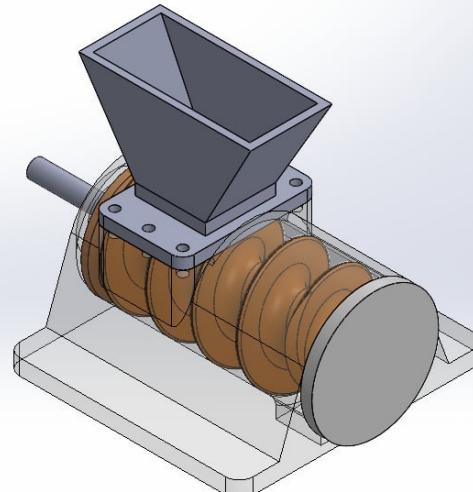


Fig. 5.9 Screw Feeder 3D Design

999

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



1007

5.6.2 Rotating Conveyor Table

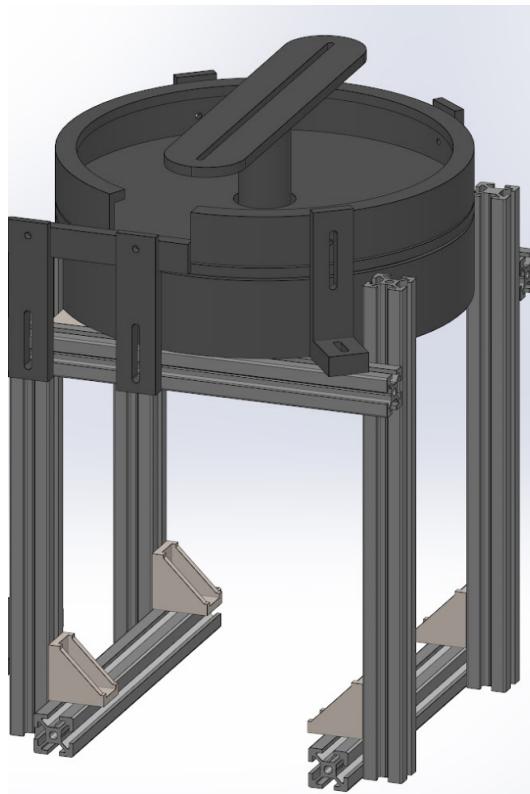


Fig. 5.10 Rotating Conveyor Table 3D Design

1008

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

1009

1010

1011

1012

1013

1014



Fig. 5.11 Rotating Conveyor Table with Aluminum Guides

1015 As shown in Figure 5.11, the installed aluminum guides on the rotating conveyor table
1016 ensures coffee beans to be linearly arranged. This linear arrangement of beans significantly
1017 helped the system to ensure that coffee beans are dropped onto the slide, which connects
1018 the conveyor table to the inspection try, in a one-by-one manner. In addition, the aluminum
1019 guides are also installed to keep the beans from accumulating in one area, which can cause
1020 the jamming of beans. The researchers tested the different motor speeds to observe the
1021 optimal settings that will not cause bean jamming and meet the minimum sorting speed of
1022 the system.

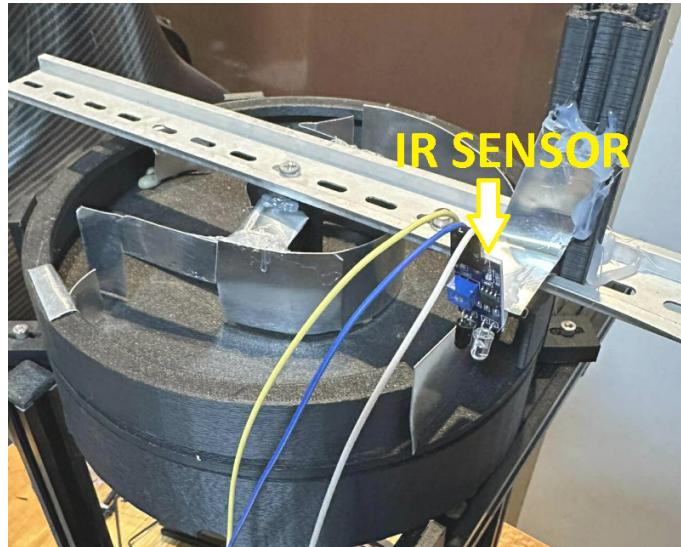


Fig. 5.12 Rotating Conveyor Table with IR Sensor

1023 Initially, the rotating conveyor table is set at a fixed and slow speed to ensure that coffee
1024 beans are dropped into the inspection tray one-by-one. However, at this rate, the time travel
1025 time of the first bean dropped from the center of the table is very long. Thus, the group
1026 decided to add an IR sensor at the edge of the rotating table as seen in Figure 5.12. The
1027 sensor's responsibility is to detect if there is a bean at the edge. If there is no bean detected,
1028 the rotating table is set to a higher speed to expedite the process. On the other hand, if a
1029 bean is detected by the sensor, the rotation of the table is adjusted in such a way that it is
1030 able to drop the beans one-by-one onto the inspection tray. With this sensor integrated into
1031 the system, a higher speed can be set for the rotating table, minimizing the time travel of
1032 the beans from the center to the inspection tray, resulting to a faster sorting time for the
1033 first stage.



1034

5.6.3 Inspection Tray

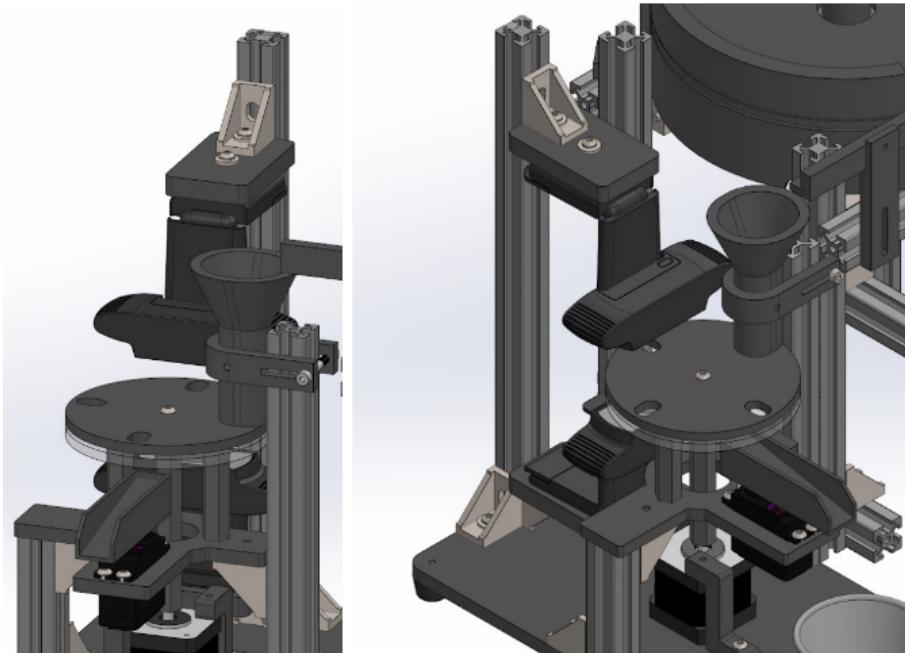


Fig. 5.13 Inspection Tray 3D Design

1035

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



1045 5.6.4 Density Sorter

1046 The density sorter is the second-stage sorting system, tasked with sorting coffee beans
1047 according to their measured density. This is achieved by initially measuring each bean's
1048 mass using a precision weighing scale and volume using the ToF10120 infrared sensor.
1049 After calculating the density, the system triggers a sorting system powered by a NEMA 17
1050 stepper motor, which sorts beans into various collection bins according to their classification.
1051 This sorting operation is such that high-density, specialty-grade beans are kept separate
1052 from low-density, commercial-grade or defective beans. The density sorter's accuracy is
1053 verified by comparing the results of its classification to manual weighing measurements
1054 (ground truth data).



Fig. 5.14 Precision Scale

1055 The U.S. Solid Electronic Precision Balance (0.01g, 1200g capacity, RS232 port,
1056 AC/DC power) was selected for the density sorting mechanism because it is highly accurate,
1057 transmits data in real-time, and is well-calibrated. Its 0.01g precision guarantees accurate
1058 mass readings, which are critical to precise density calculations in sorting coffee beans.
1059 The RS232 port facilitates smooth integration with the microcontroller for automatic data



1060 processing and sorting decisions, minimizing manual errors. Its dual power source (AC
 1061 and battery) also guarantees uninterrupted operation in different environments, making it a
 1062 dependable and efficient part of the coffee bean sorting system.

1063 5.7 Hardware and Software Integration

1064 5.7.1 Serial Communication

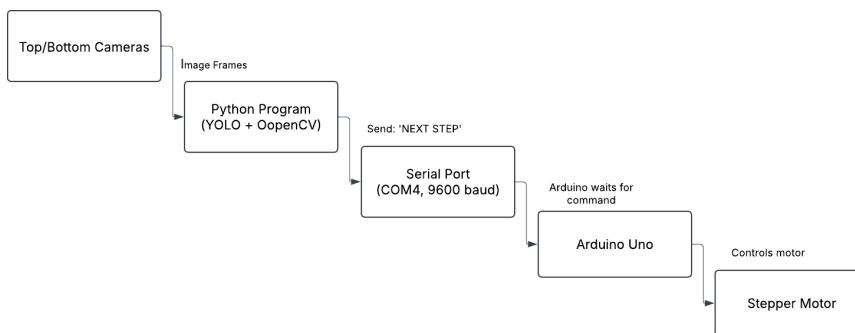


Fig. 5.15 Serial Communication Flow for Stage 1 Classification

1065 The system is generally composed of hardware and software components. Hardware
 1066 components are mainly responsible for collecting data from the coffee beans such as the
 1067 camera and IR sensor, and the sorting mechanisms such as servo motors and stepper motors.
 1068 On the other hand, the software components are the brain of the system which is mainly
 1069 responsible for data processing such as image detection, defect classification of the beans,
 1070 volume and density computation, and control of the mechanisms. Since the system has
 1071 two major components, software and hardware, they should be integrated together for
 1072 the system to be as effective. Thus, serial communication was utilized to integrate the
 1073 hardware and software components of the system. Serial communication is a significant



1074 component in the system as it serves as the communication medium of the hardware and
 1075 software. It enables real-time coordination between the software (YOLO-based image
 1076 detection, classification, and density computation) and the hardware (running in Arduino
 1077 microcontrollers). The said communication is established with the use of a USB serial
 1078 interface using the pyserial library in Python. In addition, this is configured at a baud rate
 1079 of 9600.

1080 5.7.2 Recommended Standard 232 (RS-232)

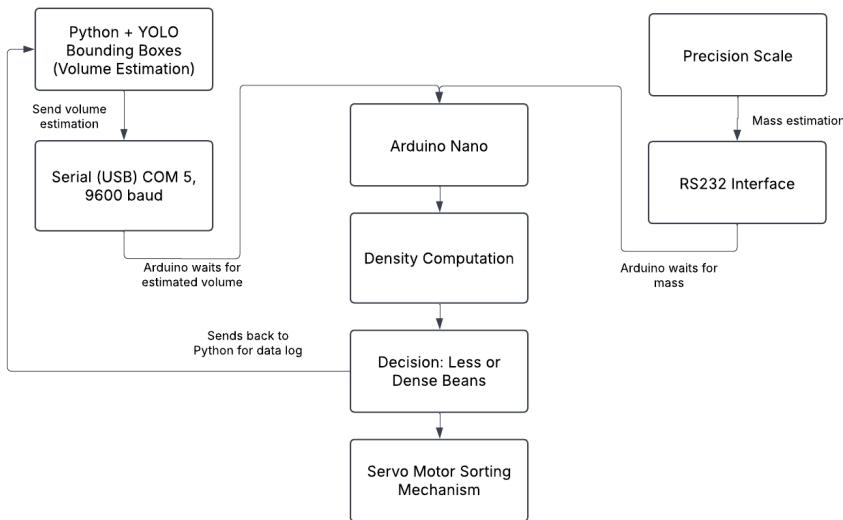


Fig. 5.16 Precision Scale Integration with RS232 for Stage 2 Classification

1081 The stage 2 classification is mainly composed of the sorting mechanism itself, and the
 1082 precision scale to measure the mass of each bean. The bounding boxes from the stage 1
 1083 classification are used to estimate each bean's volume. Additionally, the beans depth is
 1084 also estimated through the IR sensor placed in the rotating conveyor table. With these
 1085 measurements, the volume of each bean, the volume can be calculated using the Tri-axial



De La Salle University

1086 Ellipsoid's volume formula. The system, specifically at the inspection tray mechanism
1087 where the YOLO detection and classification is implemented, has a function move_stepper()
1088 responsible for sending the command from the Python code to the Arduino microcontroller.
1089 When the Arduino receive this command, it executes motor movement that allows the
1090 stepper motor to move at a certain angle that allows the camera to capture the bean.
1091 This function is crucial for the system as this is how each bean in the inspection tray is
1092 fed to the image processing side of the system. This movement rotates the mechanism
1093 holding the coffee beans, positioning the next bean beneath the top and bottom cameras
1094 for inspection. After the motor completes the movement, the Arduino will send back a
1095 message to the program running Python, signalling that the bean is ready for image capture
1096 and further processing. In addition, the Python script is continuously or constantly waiting
1097 for the Arduino's message through the arduino.readline() function, ensuring seamless
1098 communication and faster processing.

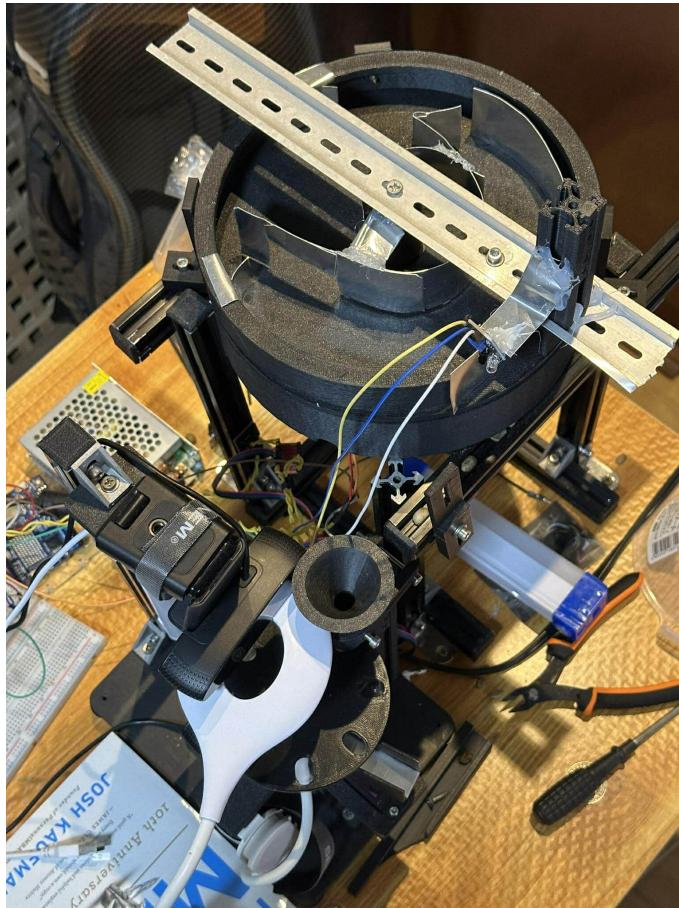
1099 **5.8 Prototype Setup**1100 **5.8.1 Actual Setup**

Fig. 5.17 Actual System Setup

1101 Physical integration of the automatic coffee bean sorter system comprises various integrated
1102 parts with the purpose of enabling effective, accurate, and methodical sorting in terms
1103 of visual defects as well as density categorization. The system involves integration of
1104 mechanical, electronic, and computer vision technologies for optimizing sorting. To



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

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



5.8.2 Lighting Setup for Inspection Tray

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



Fig. 5.18 First Iteration of Lighting Setup

1139 Figure 5.18 shows the initial lighting setup that the researchers implemented on the
1140 system. The initial lighting arrangement was based on a single top-mounted LED lighting.
1141 Although the arrangement was more than bright enough for the top camera, it introduced
1142 random shadows and highlights onto the bottom camera. As a result, only one side of the
1143 bean is accurately inspected. These random elements impacted the model's performance
1144 in detecting bean contours and separating surface flaws, particularly for dark beans or
1145 reflective-surface beans.

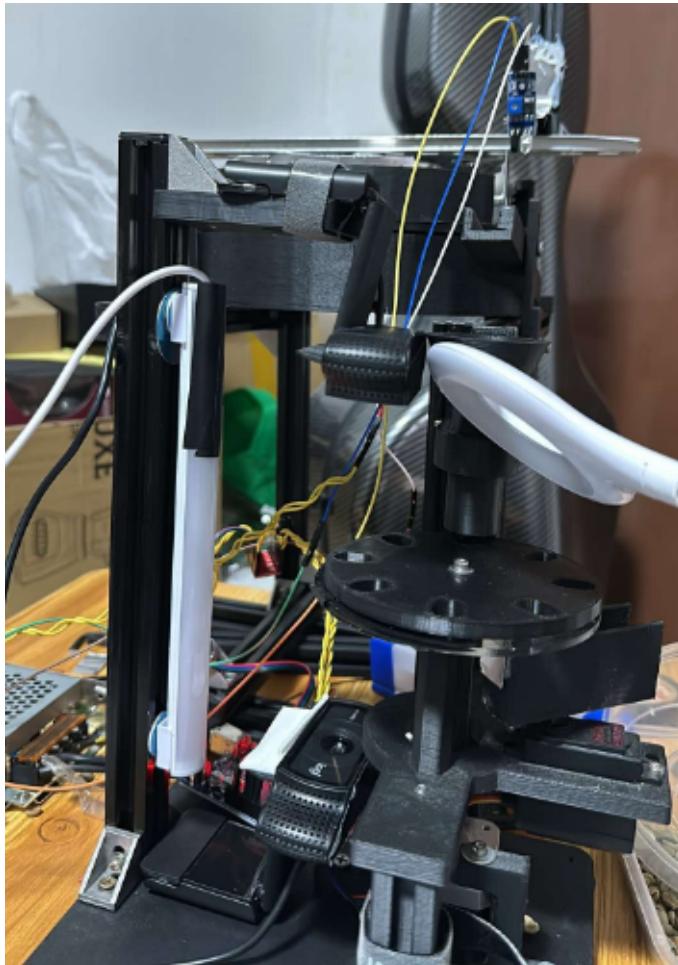


Fig. 5.19 Second Iteration of Lighting Setup

1146 For the second iteration of the lighting setup, the researchers decided to add another
1147 LED strip lighting at the side of the inspection tray, while keeping the LED lighting
1148 mounted at the top. This provided good lighting for both top and bottom cameras. However,
1149 the view of the bottom camera is still a bit dark.

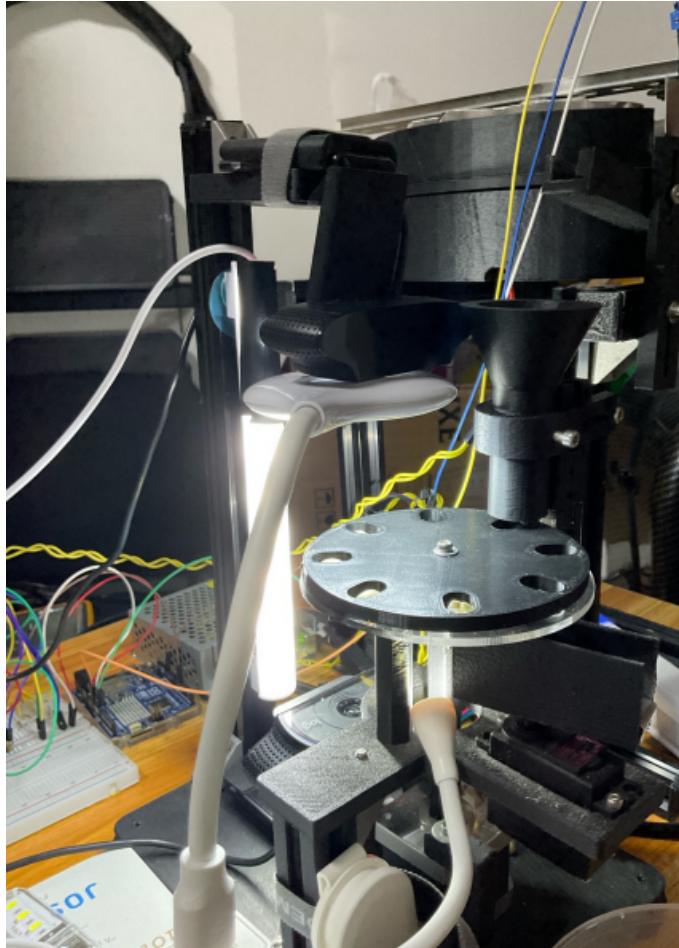


Fig. 5.20 Final Iteration of Lighting Setup

1150 To ensure that both camera views have sufficient lighting and avoid shadows, the
1151 researchers decided to use a total of three LED lights. One is a small ring light placed
1152 exactly above the inspection tray. Another LED light is a stip light placed at the side of the
1153 inspection tray to improve lighting at the side of each bean. Lastly, another small LED light
1154 is placed under the inspection tray to ensure that the bottom camera has enough lighting.

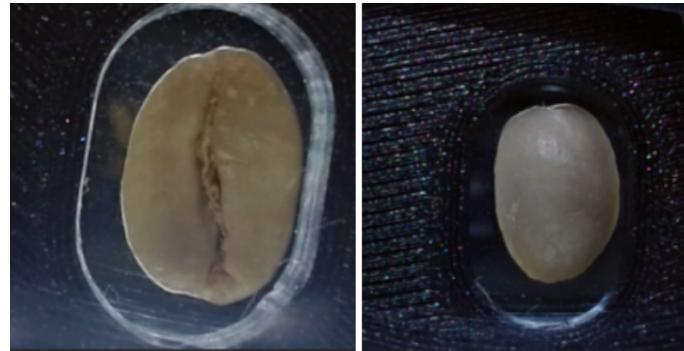


Fig. 5.21 Top and Bottom View of the Cameras

5.8.3 System Operation

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

The process 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



De La Salle University

1170 of top and bottom cameras. The system captures high-resolution images of the bean and
1171 processes the data to determine which classification it belongs. If the bean is identified as
1172 defective, a signal is sent to the servo motor, which redirects the bean into the defective bin
1173 for disposal, if the bean is classified as good, it then proceeds to the second phase of the
1174 system

1175 The second stage involves density-based sorting, where each GCB's weight is measured
1176 using a precision scale, while its volume is determined by the ToF10120 infrared sensor.
1177 The system then calculates the density and classifies the bean accordingly.

1178 The sorting mechanism activates, directing beans into designated collection bins based
1179 on their density. High-density beans, often associated with specialty-grade quality, are
1180 separated from low-density, commercial-grade, or defective beans.



5.9 Prototype Testing

5.9.1 Sorting Speed

TABLE 5.2 SORTING SPEED TESTING TABLE

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

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



1191

5.9.2 Defect Sorting Accuracy

TABLE 5.3 GOOD BEAN CLASSIFICATION ACCURACY TESTING TABLE

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

1192

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

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



TABLE 5.4 SPECIFIC DEFECT CLASSIFICATION ACCURACY TESTING TABLE

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

1198 For further accuracy testing of the computer vision model in actual implementation, the
 1199 researchers also included testing trials for each defect type. Table 5.4 shows how each trial
 1200 is conducted. For example, the defect type chosen for the test is the Sour defect type. The
 1201 first trial contains 100 sour beans. For the second trial, 80 sour beans and 20 randomly
 1202 selected beans, excluding the chosen defect type which is sour. Thus, the random beans are
 1203 always the other classes except the chosen defect type to be tested. In this test, correctly
 1204 classified beans and misclassified beans are also considered to compute for the accuracy of
 1205 the system. By testing the system under different defect distributions, the robustness of the
 1206 machine vision model can be assessed.



TABLE 5.5 DATASET DISTRIBUTION FOR OVERALL TESTING

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

1207 Lastly, to assess the overall accuracy and reliability of the first stage, machine vision-
 1208 based defect classification, a trial consisting of a predefined dataset of 160 coffee beans
 1209 was conducted. Each category consists of 20 beans as shown in Table 5.5, including good
 1210 beans and the other defect types such as black, dried cherry, fungus, insect damage, sour,
 1211 floater, and broken beans.

1212 **5.9.3 Density Sorting Accuracy**

1213 To assess the accuracy of the mechanism, it will rely on measuring the accuracy and the
 1214 reliability of the density sorting mechanism in sorting out the dense beans to the less dense
 1215 beans. To successfully determine the accuracy of the system, the basis will be the scale,
 1216 where the system should be able to sort the dense beans to the less dense bean in relation
 1217 to the detected weight in the scale. A successful system should be able to sort with an
 1218 accuracy of 85



1219

Chapter 6

1220

RESULTS AND DISCUSSIONS

6. Results and Discussions



De La Salle University

TABLE 6.1 SUMMARY OF RESULTS FOR ACHIEVING THE OBJECTIVES

Objectives	Results	Locations
GO: To develop an automated (Arabica) green coffee bean sorter that identifies good, less-dense and defective beans from an unsorted batch of coffee beans. The system will utilize machine vision and density-based analysis for defect detection and classification of the coffee beans, ensuring efficient coffee bean sorting.	<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. 95
SO1: To gather and create a dataset consisting of 500 high-resolution images of good Arabica green coffee beans and 200 high-resolution images per classification of defective beans (Category 1 & Category 2).	<ul style="list-style-type: none"> • Acquired 257 images of Black coffee beans • Gathered 301 images of Broken coffee beans • Gathered 305 images of Dried Cherry coffee beans • Acquired 288 images of Floater coffee beans • Acquired 301 images of Fungus Damage coffee beans • Gathered 1565 images of Good coffee beans • Acquired 345 images of Insect Damage coffee beans • Gathered 320 images of Sour coffee beans 	Sec. 6.1 on p. 95

Continued on next page



Continued from previous page

Objectives	Results	Locations
SO2: To improve the synchronization between the machine vision system and the embedded sorting mechanism, ensuring defect sorting of at least 20 beans per minute for stage one, solving issues such as non-synchronization of the system.	<ul style="list-style-type: none"> • Achieved 22 beans per minute for stage one of the system 	Sec. 6.4 on p. 107

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. 6.3 on p. 104
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	205
Broken	203
Dried Cherry	206
Floater	202
Fungus Damage	207
Good	604
Insect Damage	201
Sour	202
Total	2030

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

TABLE 6.3 DATASET SPLIT SUMMARY

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

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



1229 images) and 4% (82 images) respectively, with the training images later augmented 3× per
1230 image.

1231 **6.2 Performance of Classification Models on Cus- 1232 tom Dataset**

1233 Four different classification models, such as EfficientNet, YOLOv8, YOLOv11 and
1234 YOLOv12, were benchmarked to determine the most optimal model to be used for the sys-
1235 tem. Each model was trained using a custom dataset manually gathered by the researchers.
1236 In addition, augmentations such as rotation, flip, blur and noise, were applied.



1237

6.2.1 EfficientNetV2S

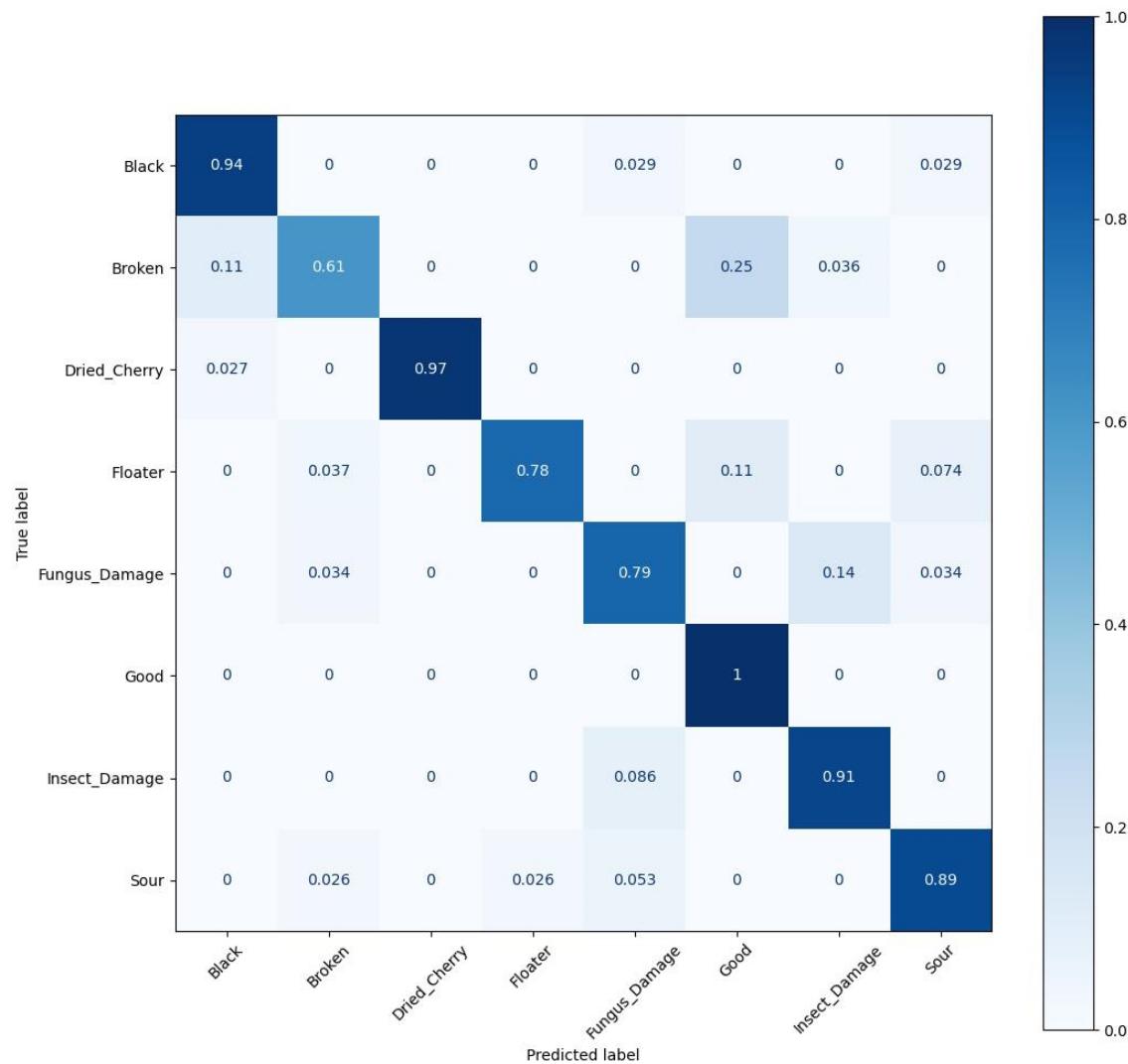


Fig. 6.1 Normalized Confusion Matrix for EfficientNetV2S on Test Dataset

1238

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

1239

1240



De La Salle University

1241 though EfficientNet was able to classify the Good bean class perfectly (1.00) and accurately
1242 for classes like Dried Cherry (0.97) and Black (0.94), its classification was poor for many
1243 defect classes. In particular, the model exhibited significant misclassification in the Broken
1244 bean class, with just 61% correctly classified, while a significant 25% were misclassified as
1245 Good. Likewise, for Floater and Fungus Damage, EfficientNetV2 had true positive rates
1246 of only 0.78 and 0.79, respectively, with some floaters being mistaken as Fungus Damage
1247 (11%) and Sour (7.4%). This trend indicates that EfficientNet found it difficult to identify
1248 subtle visual variations between defect types, particularly when texture or color change
1249 overlapped among classes.



1250

6.2.2 YOLOv8



Fig. 6.2 Normalized Confusion Matrix for YOLOv8 on Test Dataset

1251

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



1258 Insect Damage and Sour beans, at 2% each, which would suggest some overlap in texture
1259 or color patterns that the model found difficulty in distinguishing. However, there was a
1260 lesser, but still significant confusion between Sour and Fungus Damage, where Sour beans
1261 were misclassified at 0.10 within other classes. The Insect Damage class performed well at
1262 0.94, though there was some confusion (6%) with Fungus Damage. Broken beans reached
1263 0.91, with small misclassifications into Dried Cherry and others. Most importantly, there
1264 was no confusion with the Background class, indicating YOLOv8's excellent capability
1265 of isolating and detecting bean contours well. In general, YOLOv8 provides balanced
1266 performance, with satisfactory overall accuracy across different classes.



1267

6.2.3 YOLOv11-cls



Fig. 6.3 Normalized Confusion Matrix for YOLOv11 on Test Dataset

1268

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



1275 Damage, though misclassification did occur into Sour beans (7%) and Insect Damage (2%),
1276 which points to some confusion between defects that have comparable texture degradation.
1277 The Insect Damage class achieved a strong 0.92, but was at times confused with Black
1278 and Fungus Damage, both by 3%. The performance of the model slightly declined in the
1279 Sour bean class, which exhibited the lowest true positive rate of 0.89, with significant
1280 misclassifications to Fungus Damage (7%), indicative of visual discoloration or wrinkling
1281 overlap. In general, YOLOv11 shows a good balance in performance, being excellent in
1282 clean categories and keeping stable results for complicated defect types. Its high precision
1283 with low false positives on most classes indicate its potential in real-time defect detection
1284 applications, with room for improvement through additional dataset augmentation for
1285 biologically deteriorated beans.



1286

6.2.4 YOLOv12-cls



Fig. 6.4 Normalized Confusion Matrix for YOLOv12 on Test Dataset

1287

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



1294 over other classes like Dried Cherry, Floater, and Insect Damage, each contributing 1–2%
 1295 to the confusion. Likewise, Fungus Damage was classified correctly 88% of the time, but
 1296 exhibited confusion primarily with Insect Damage (5%) and Good beans (2%), meaning
 1297 overlap of surface stain or odd texture. The Floater class was highly accurate at 0.97 and
 1298 had little confusion. Insect Damage, despite maintaining a consistency of 0.92, had some
 1299 misclassifications as Fungus Damage (10%). Overall, YOLOv12 is a well-balanced and
 1300 high-performing model, with leading accuracy in classes that have clear visual differences
 1301 and moderate misclassification in Fungus and Insect-damaged beans, which are still visually
 1302 complex. The performance of the model shows an enhanced capability to generalize
 1303 between defect types.

1304 6.3 Actual Performance of Trained Models in the 1305 System

TABLE 6.4 SPECIFIC PERFORMANCE OF THE MODELS FOR EACH DEFECT

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

Continued on next page



De La Salle University

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

Continued on next page



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

1306

1307

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

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321



1322 model, where YOLO-based models in general perform better than EfficientNetV2 when
 1323 it comes to precision and recall, particularly on real-time classification tasks. There are
 1324 still trade-offs in terms of performance noticed in defect types with shared visual features,
 1325 showing that more comprehensive image preprocessing or feature enhancement may be
 1326 needed for future versions.

TABLE 6.5 MODEL PERFORMANCE COMPARISON

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

1327 Table 6.5 summarizes the overall performance of each classification model by presenting
 1328 average Precision, Recall, F1-Score, and Accuracy for all defect types. This general
 1329 overview enables comparison of each model's overall performance regardless of particular
 1330 defect classes. We can see that YOLOv12 performs the best among all the models with the
 1331 best average accuracy of 90.0%, and well-balanced precision and recall. This confirms its
 1332 good detection consistency and minimal false positives across the trials during the actual
 1333 testing. YOLOv11 and YOLOv8 are close second and third, with average accuracies of
 1334 88.8% and 85.6%, respectively, showing consistent performance but with slightly higher
 1335 misclassification rates. EfficientNetV2, although effective in detecting significant defects,
 1336 had the poorest performance at 81.2% accuracy.

1337 6.4 Sorting Speed



TABLE 6.6 SORTING SPEED TEST CONDITIONS

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

1338 Table 6.6 presents the prototype system's sorting speed performance under different
 1339 test conditions. The conveyor table speed and inspection tray motor speed is constant at
 1340 175 RPM and 343 RPM, respectively, to ensure consistency in all trials. The sorting speed,
 1341 expressed in beans per minute, indicates the system's capacity to recognize and process
 1342 coffee beans. The outcomes indicate that the system maintained a steady average sorting
 1343 rate of 22 beans per minute in most conditions, such as 100



1344 **Chapter 7**

1345 **CONCLUSIONS, RECOMMENDATIONS, AND**
1346 **FUTURE DIRECTIVES**



1347 7.1 Concluding Remarks

1348 The study was able to present the design, development, and actual implementation of a two-
1349 staged automated green coffee bean sorting system, utilizing computer vision and embedded
1350 systems. The design is composed of a rotating conveyor table, a dual-camera inspection
1351 tray, defect sorting mechanism, and density-based sorting mechanism. In addition, four
1352 deep learning-based classification models such as EfficientNetV2, YOLOv8, YOLOv11,
1353 and YOLOv12 were benchmarked. These models were deployed and tested into the actual
1354 defect sorting system with a test dataset of 20 beans per classification, where the YOLOv12
1355 achieved the highest accuracy of 90.0%. In terms of the sorting speed, the system was
1356 tested in 5 trials, where it achieved an average sorting speed of 22.2 beans per minute. The
1357 system was tested under varying quality distributions and maintained consistent sorting
1358 speeds, thereby confirming its practical viability. Overall, the results indicate that the
1359 integration of deep learning and embedded automation offers a robust and scalable solution
1360 for post-harvest coffee bean quality assessment.

1361 7.2 Contributions

1362 This study contributed to the coffee industry in the Philippines by introducing a two-
1363 stage automated coffee bean sorter that enhances coffee quality assessment by segregating
1364 defective beans and sorting dense and less-dense beans. This system integrates machine
1365 vision and density-based sorting, ensuring that high-quality, dense beans and potential
1366 specialty-grade coffee are selected for further processing. This system can support the
1367 Philippine coffee industry's efforts to enhance product quality and meet global specialty
1368 coffee standards to improve market competitiveness.



1369 **7.3 Recommendations**

1370 The following are the recommendations for further study of this design:

- 1371 • Optimize the density-based sorting mechanism
- 1372 • Improvement of system portability by reducing the overall size and weight of the
- 1373 system

1374 **7.4 Future Prospects**

1375 This study offers a building block for future innovation in intelligent post-harvest coffee
1376 processing. A potential extension is combining cloud-based data storage and analytics for
1377 traceability at the batch level and remote monitoring. Another would be the deployment of
1378 light inference models on microcontroller units (MCUs) to facilitate real-time, on-device
1379 computation, thus minimizing system latency and increasing portability. Additional re-
1380 search might also investigate the use of unsupervised or semi-supervised learning methods
1381 to identify new or infrequent defects without depending solely on labeled data. Commer-
1382 cially, the system can be scaled to process greater volumes using modular conveyor lines
1383 and parallel sorting stations. These developments would greatly benefit coffee producers
1384 by providing consistent, efficient, and objective bean quality assessment.



REFERENCES

- 1386 [Akbar et al., 2021] Akbar, M., Rachmawati, E., and Sthevanie, F. (2021). Visual feature and
1387 machine learning approach for arabica green coffee beans grade determination. In *Proceedings*
1388 of the 6th International Conference on Communication and Information Processing, ICCIP '20,
1389 page 97–104, New York, NY, USA. Association for Computing Machinery.
- 1390 [Amadea et al., 2024] Amadea, V., Rachmawati, E., Ferdian, E., and Akbar, M. N. S. (2024).
1391 Defect detection in arabica green coffee beans based on grade quality. In *Proceedings of the*
1392 2024 10th International Conference on Computing and Artificial Intelligence, ICCAI '24, pages
1393 103–110, New York, NY, USA. Association for Computing Machinery.
- 1394 [Arboleda et al., 2020] Arboleda, E. R., Fajardo, A. C., and Medina, R. P. (2020). Green coffee
1395 beans feature extractor using image processing. *TELKOMNIKA (Telecommunication Computing*
1396 *Electronics and Control)*, 18(4):2027–2034.
- 1397 [Balay et al., 2024] Balay, D., Cabrera, R., Jensen, J., and Mayuga, K. (2024). *Automatic Sorting*
1398 *of Defective Coffee Beans through Computer Vision*. PhD thesis, De La Salle University.
- 1399 [Balbin et al., 2020] Balbin, J. R., Del Valle, C. D., Lopez, V. J. L. G., and Quiambao, R. F. (2020).
1400 Grading and profiling of coffee beans for international standards using integrated image pro-
1401 cessing algorithms and back-propagation neural network. In *2020 IEEE 12th International*
1402 *Conference on Humanoid, Nanotechnology, Information Technology, Communication and Con-*
1403 *trol, Environment, and Management (HNICEM)*, pages 1–6.
- 1404 [Bali and Tyagi, 2020] Bali, S. and Tyagi, S. S. (2020). Evaluation of transfer learning techniques
1405 for classifying small surgical dataset. In *2020 10th International Conference on Cloud Computing,*
1406 *Data Science & Engineering (Confluence)*, pages 744–750.
- 1407 [Barbosa et al., 2019] Barbosa, M. d. S. G., Scholz, M. B. d. S., Kitzberger, C. S. G., and Benassi,
1408 M. d. T. (2019). Correlation between the composition of green arabica coffee beans and the
1409 sensory quality of coffee brews. *Food Chemistry*, 292:275–280.
- 1410 [Bureau of Agriculture and Fisheries Standards, 2012] Bureau of Agriculture and Fisheries Stan-
1411 dards (2012). Green coffee beans – specifications.
- 1412 [Bureau of Agriculture and Fisheries Standards, 2022] Bureau of Agriculture and Fisheries Stan-
1413 dards (2022). Green coffee bean sorter — specifications.
- 1414 [Córdoba et al., 2021] Córdoba, N., Moreno, F. L., Osorio, C., Velásquez, S., Fernandez-Alduenda,
1415 M., and Ruiz-Pardo, Y. (2021). Specialty and regular coffee bean quality for cold and hot
1416 brewing: Evaluation of sensory profile and physicochemical characteristics. *LWT*, 145:111363.
- 1417 [da Cruz et al., 2006] da Cruz, A. G., Cenci, S. A., and Maia, M. C. A. (2006). Quality assurance
1418 requirements in produce processing. *Trends in Food Science & Technology*, 17(8):406–411.
- 1419 [Dabek et al., 2022] Dabek, P., Krot, P., Wodecki, J., Zimroz, P., Szrek, J., and Zimroz, R. (2022).



De La Salle University

- 1420 Measurement of idlers rotation speed in belt conveyors based on image data analysis for diagnostic
1421 purposes. *Measurement*, 202:111869.
- 1422 [Das et al., 2019] Das, S., Hollander, C. D., and Suliman, S. (2019). Automating visual inspection
1423 with convolutional neural networks. *Annual Conference of the PHM Society*, 11(1).
- 1424 [Datov and Lin, 2019] Datov, A. and Lin, Y.-C. (2019). Classification and grading of green coffee
1425 beans in asia.
- 1426 [de Oliveira et al., 2016] de Oliveira, E. M., Leme, D. S., Barbosa, B. H. G., Rodarte, M. P., and
1427 Pereira, R. G. F. A. (2016). A computer vision system for coffee beans classification based on
1428 computational intelligence techniques. *Journal of Food Engineering*, 171:22–27.
- 1429 [Deepti and Prabadevi, 2024] Deepti, R. and Prabadevi, B. (2024). Yolotransformer-transdetect:
1430 a hybrid model for steel tube defect detection using yolo and transformer architectures —
1431 international journal on interactive design and manufacturing (ijidem).
- 1432 [García et al., 2019] García, M., Candeló-Becerra, J. E., and Hoyos, F. E. (2019). Quality and
1433 defect inspection of green coffee beans using a computer vision system. *Applied Sciences*,
1434 9(19):4195.
- 1435 [González et al., 2019] González, A. L., Lopez, A. M., Taboada Gaytán, O., and Ramos, V. M.
1436 (2019). Cup quality attributes of catimors as affected by size and shape of coffee bean (coffeea
1437 arabica l.). *International Journal of Food Properties*, 22(1):758–767.
- 1438 [Huang et al., 2019] Huang, N.-F., Chou, D.-L., and Lee, C.-A. (2019). Real-time classification
1439 of green coffee beans by using a convolutional neural network. In *2019 3rd International
1440 Conference on Imaging, Signal Processing and Communication (ICISPC)*, pages 107–111.
- 1441 [International Coffee Association, 2023] International Coffee Association (2023). Summary coffee
1442 report & outlook december 2023.
- 1443 [International Organization for Standardization, 1995] International Organization for Standardization
1444 (1995). Green and roasted coffee — determination of free-flow bulk density of whole beans
1445 (routine method).
- 1446 [International Organization for Standardization, 2007] International Organization for Standardization
1447 (2007). Safety of machinery – risk assessment – part 2: Practical guidance and examples of
1448 methods.
- 1449 [International Organization for Standardization, 2010] International Organization for Standardization
1450 (2010). Safety of machinery – general principles for design – risk assessment and risk
1451 reduction.
- 1452 [International Organization for Standardization, 2015] International Organization for Standardization
1453 (2015). Systems and software engineering – systems and software quality requirements and
1454 evaluation (square) – measurement of data quality.
- 1455 [International Organization for Standardization, 2019] International Organization for Standardization
1456 (2019). Information technology — development of user interface accessibility part 1: Code



De La Salle University

- 1457 of practice for creating accessible ict products and services.
- 1458 [International Organization for Standardization, 2022] International Organization for Standardiza-
- 1459 tion (2022). Framework for artificial intelligence (ai) systems using machine learning (ml).
- 1460 [Lee and Tai, 2020] Lee, W.-C. and Tai, P.-L. (2020). Defect detection in striped images using a
- 1461 one-dimensional median filter. *Applied Sciences*, 10(3):1012.
- 1462 [Lualhati et al., 2022] Lualhati, A. J. N., Mariano, J. B., Torres, A. E. L., and Fenol, S. D. (2022).
- 1463 Development and testing of green coffee bean quality sorter using image processing and artificial
- 1464 neural network. *Mindanao Journal of Science and Technology*, 20(1).
- 1465 [Luis et al., 2022] Luis, V. A. M., Quinones, M. V. T., and Yumang, A. N. (2022). Classification of
- 1466 defects in robusta green coffee beans using yolo. In *ResearchGate*.
- 1467 [Minglani et al., 2020] Minglani, D., Sharma, A., Pandey, H., Dayal, R., Joshi, J. B., and Subrama-
- 1468 niam, S. (2020). A review of granular flow in screw feeders and conveyors. *Powder Technology*,
- 1469 366:369–381.
- 1470 [Pragathi and Jacob, 2024] Pragathi, S. P. and Jacob, L. (2024). Arabica coffee bean grading into
- 1471 specialty and commodity type based on quality using visual inspection. *ResearchGate*.
- 1472 [Santos and Baltazar, 2022] Santos, D. T. and Baltazar, M. D. (2022). *The Philippine Coffee*
- 1473 *Industry Roadmap, 2021-2025*. Department of Agriculture, Bureau of Agricultural Research.
- 1474 Google-Books-ID: QkBT0AEACAAJ.
- 1475 [Santos et al., 2020] Santos, F., Rosas, J., Martins, R., Araújo, G., Viana, L., and Gonçalves, J.
- 1476 (2020). Quality assessment of coffee beans through computer vision and machine learning
- 1477 algorithms. *Coffee Science - ISSN 1984-3909*, 15:e151752–e151752.
- 1478 [Srisang et al., 2019] Srisang, N., Chanpaka, W., and Chungcharoen, T. (2019). The performance
- 1479 of size grading machine of robusta green coffee bean using oscillating sieve with swing along
- 1480 width direction. *IOP Conference Series: Earth and Environmental Science*, 301(1):012037.
- 1481 [Susanibar et al., 2024] Susanibar, G., Ramirez, J., Sanchez, J., and Ramirez, R. (2024). Develop-
- 1482 ment of an automated machine for green coffee beans classification by size and defects —
- 1483 request pdf. *ResearchGate*.
- 1484 [Tampon, 2023] Tampon, V. (2023). 63 coffee statistics you need to know for 2024 and beyond.
- 1485 [Wu et al., 2024] Wu, L., Hao, H.-Y., and Song, Y. (2024). A review of metal surface defect
- 1486 detection based on computer vision. *Acta Automatica Sinica*.

1487

Produced: April 1, 2025, 22:22



De La Salle University

1488

Appendix A STUDENT RESEARCH ETHICS CLEARANCE

1489



De La Salle University

1490

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

1491

Appendix B ANSWERS TO QUESTIONS TO THIS THESIS

1492



De La Salle University



- 1493 **B1 How important is the problem to practice?**
- 1494 **B2 How will you know if the solution/s that you will**
 1495 **achieve would be better than existing ones?**
- 1496 **B2.1 How will you measure the improvement/s?**
- 1497 B2.1.1 What is/are your basis/bases for the improvement/s?
- 1498 B2.1.2 Why did you choose that/those basis/bases?
- 1499 B2.1.3 How significant are your measure/s of the improvement/s?
- 1500 **B3 What is the difference of the solution/s from ex-**
 1501 **isting ones?**
- 1502 B3.1 How is it different from previous and existing ones?
- 1503 **B4 What are the assumptions made (that are behind**
 1504 **for your proposed solution to work)?**
- 1505 B4.1 Will your proposed solution/s be sensitive to these as-
 1506 sumptions?
- 1507 B4.2 Can your proposed solution/s be applied to more general
 1508 cases when some assumptions are eliminated? If so, how?
- 1509 **B5 What is the necessity of your approach / pro-**
 1510 **posed solution/s?**
- 1511 B5.1 What will be the limits of applicability of your proposed so-
 1512 lution/s?
- 1513 B5.2 What will be the message of the proposed solution to
 1514 technical people? How about to non-technical managers and
 1515 business people?
- 1516 **B6 How will you know if your proposed solution/s**
 1517 **is/are correct?**
- 1518 B6.1 Will your results warrant the level of mathematics used
 1519 (i.e., will the end justify the means)?



De La Salle University

1532

Appendix C REVISIONS TO THE PROPOSAL

1533



De La Salle University

- 1534 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.
- 1535
- 1536 1. Examiner
- 1537 2. Comment
- 1538 3. Summary of how the comment was addressed
- 1539 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, 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.</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. 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>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>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

C. Revisions to the Proposal



De La Salle University

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

C. Revisions to the Proposal



De La Salle University

Continued from previous page



De La Salle University

1540

Appendix D REVISIONS TO THE FINAL

1541



- 1542 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.
- 1543
- 1544 1. Examiner
- 1545 2. Comment
- 1546 3. Summary of how the comment has been addressed
- 1547 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

1548

Appendix E USAGE EXAMPLES

1549



1550 The user is expected to have a working knowledge of L^AT_EX. A good introduction is
 1551 in [?]. Its latest version can be accessed at <http://www.ctan.org/tex-archive/info/lshort>.

1552 E1 Equations

1553 The following examples show how to typeset equations in L^AT_EX. This section also shows
 1554 examples of the use of \gls{ } commands in conjunction with the items that are in
 1555 the notation.tex file. **Please make sure that the entries in notation.tex are**
 1556 **those that are referenced in the L^AT_EX document files used by this Thesis. Please**
 1557 **comment out unused notations and be careful with the commas and brackets in**
 1558 **notation.tex .**

1559 In (E.1), the output signal $y(t)$ is the result of the convolution of the input signal $x(t)$
 1560 and the impulse response $h(t)$.

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

1561 Other example equations are as follows.

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

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

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



1562

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
17 \label{eq:conv}
18 \end{eqnarray}
19 Other example equations are as follows.
20
21 \begin{eqnarray}
22     \left[ \frac{V_1}{I_1} \right] = \begin{bmatrix} A & B \\ C & D \end{bmatrix}
23
24 \label{eq:ABCD}
25 \end{eqnarray}
26
27 \begin{eqnarray}
28 \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
29 \end{eqnarray}
30
31 \begin{eqnarray}
32 \left| \zeta(x)^3 \zeta(x + iy)^4 \zeta(x + 2iy) \right| = \exp \sum_{n,p} \frac{3 + 4 \cos(ny \log p) + \cos(2ny \log p)}{np^{nx}} \geq 1
33 \end{eqnarray}

```



	E2 Notations												
1563													
1564	In order to use the standardized notation, the user is highly suggested to see the ISO 80000-2 standard [?].												
1565													
1566	See https://en.wikipedia.org/wiki/Help:Displaying_a_formula and https://en.wikipedia.org/wiki/List_of_mathematical_symbols for L ^A T _E X maths and other notations, respectively.												
1567													
1568	The following were taken from <code>isomath-test.tex</code> .												
1569	E2.1 Math alphabets												
1570	If there are other symbols in place of Greek letters in a math alphabet, it uses T1 or OT1 font encoding instead of OML.												
1571													
	<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>$\mathbf{A}, \mathbf{B}, \mathbf{\Gamma}, \mathbf{\Delta}, \mathbf{\Theta}, \mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Pi}, \mathbf{\Sigma}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Omega}, ff, fi, \mathbf{\beta}, ^!, \mathbf{v}, \mathbf{w}, 0, 1, 9$</td> </tr> <tr> <td>mathsf</td> <td>$\mathsf{A}, \mathsf{B}, \mathsf{\Gamma}, \mathsf{\Delta}, \mathsf{\Theta}, \mathsf{\Lambda}, \mathsf{\Xi}, \mathsf{\Pi}, \mathsf{\Sigma}, \mathsf{\Phi}, \mathsf{\Psi}, \mathsf{\Omega}, ff, fi, \mathsf{\beta}, ^!, \mathsf{v}, \mathsf{w}, 0, 1, 9$</td> </tr> <tr> <td>mathtt</td> <td>$\mathtt{A}, \mathtt{B}, \mathtt{\Gamma}, \mathtt{\Delta}, \mathtt{\Theta}, \mathtt{\Lambda}, \mathtt{\Xi}, \mathtt{\Pi}, \mathtt{\Sigma}, \mathtt{\Phi}, \mathtt{\Psi}, \mathtt{\Omega}, \mathtt{ff}, \mathtt{fi}, \mathtt{\beta}, ^!, \mathtt{v}, \mathtt{w}, 0, 1, 9$</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	$\mathbf{A}, \mathbf{B}, \mathbf{\Gamma}, \mathbf{\Delta}, \mathbf{\Theta}, \mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Pi}, \mathbf{\Sigma}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Omega}, ff, fi, \mathbf{\beta}, ^!, \mathbf{v}, \mathbf{w}, 0, 1, 9$	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	$\mathtt{A}, \mathtt{B}, \mathtt{\Gamma}, \mathtt{\Delta}, \mathtt{\Theta}, \mathtt{\Lambda}, \mathtt{\Xi}, \mathtt{\Pi}, \mathtt{\Sigma}, \mathtt{\Phi}, \mathtt{\Psi}, \mathtt{\Omega}, \mathtt{ff}, \mathtt{fi}, \mathtt{\beta}, ^!, \mathtt{v}, \mathtt{w}, 0, 1, 9$
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	$\mathtt{A}, \mathtt{B}, \mathtt{\Gamma}, \mathtt{\Delta}, \mathtt{\Theta}, \mathtt{\Lambda}, \mathtt{\Xi}, \mathtt{\Pi}, \mathtt{\Sigma}, \mathtt{\Phi}, \mathtt{\Psi}, \mathtt{\Omega}, \mathtt{ff}, \mathtt{fi}, \mathtt{\beta}, ^!, \mathtt{v}, \mathtt{w}, 0, 1, 9$												
1572	New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-italic.												
	<table> <tr> <td>mathbfit</td> <td>$\mathbf{A}, \mathbf{B}, \mathbf{\Gamma}, \mathbf{\Delta}, \mathbf{\Theta}, \mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Pi}, \mathbf{\Sigma}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Omega}, \mathbf{\alpha}, \mathbf{\beta}, \mathbf{\pi}, \mathbf{\nu}, \mathbf{\omega}, \mathbf{v}, \mathbf{w}, \mathbf{o}, \mathbf{1}, \mathbf{9}$</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>mathsfbf</td> <td>$\mathsf{A}, \mathsf{B}, \mathsf{\Gamma}, \mathsf{\Delta}, \mathsf{\Theta}, \mathsf{\Lambda}, \mathsf{\Xi}, \mathsf{\Pi}, \mathsf{\Sigma}, \mathsf{\Phi}, \mathsf{\Psi}, \mathsf{\Omega}, \mathsf{\alpha}, \mathsf{\beta}, \mathsf{\pi}, \mathsf{\nu}, \mathsf{\omega}, \mathsf{v}, \mathsf{w}, \mathsf{o}, \mathsf{1}, \mathsf{9}$</td> </tr> </table>	mathbfit	$\mathbf{A}, \mathbf{B}, \mathbf{\Gamma}, \mathbf{\Delta}, \mathbf{\Theta}, \mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Pi}, \mathbf{\Sigma}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Omega}, \mathbf{\alpha}, \mathbf{\beta}, \mathbf{\pi}, \mathbf{\nu}, \mathbf{\omega}, \mathbf{v}, \mathbf{w}, \mathbf{o}, \mathbf{1}, \mathbf{9}$	mathsfit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$	mathsfbf	$\mathsf{A}, \mathsf{B}, \mathsf{\Gamma}, \mathsf{\Delta}, \mathsf{\Theta}, \mathsf{\Lambda}, \mathsf{\Xi}, \mathsf{\Pi}, \mathsf{\Sigma}, \mathsf{\Phi}, \mathsf{\Psi}, \mathsf{\Omega}, \mathsf{\alpha}, \mathsf{\beta}, \mathsf{\pi}, \mathsf{\nu}, \mathsf{\omega}, \mathsf{v}, \mathsf{w}, \mathsf{o}, \mathsf{1}, \mathsf{9}$						
mathbfit	$\mathbf{A}, \mathbf{B}, \mathbf{\Gamma}, \mathbf{\Delta}, \mathbf{\Theta}, \mathbf{\Lambda}, \mathbf{\Xi}, \mathbf{\Pi}, \mathbf{\Sigma}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Omega}, \mathbf{\alpha}, \mathbf{\beta}, \mathbf{\pi}, \mathbf{\nu}, \mathbf{\omega}, \mathbf{v}, \mathbf{w}, \mathbf{o}, \mathbf{1}, \mathbf{9}$												
mathsfit	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$												
mathsfbf	$\mathsf{A}, \mathsf{B}, \mathsf{\Gamma}, \mathsf{\Delta}, \mathsf{\Theta}, \mathsf{\Lambda}, \mathsf{\Xi}, \mathsf{\Pi}, \mathsf{\Sigma}, \mathsf{\Phi}, \mathsf{\Psi}, \mathsf{\Omega}, \mathsf{\alpha}, \mathsf{\beta}, \mathsf{\pi}, \mathsf{\nu}, \mathsf{\omega}, \mathsf{v}, \mathsf{w}, \mathsf{o}, \mathsf{1}, \mathsf{9}$												
1573	Do the math alphabets match?												
1574	$ax\alpha\omega ax\alpha\omega ax\alpha\omega \quad TC\Theta\Gamma TC\Theta\Gamma TC\Theta\Gamma$												
1575	E2.2 Vector symbols												
1576	Alphabetic symbols for vectors are boldface italic, $\lambda = e_1 \cdot a$, while numeric ones (e.g.												
1577	the zero vector) are bold upright, $a + 0 = a$.												
1578	E2.3 Matrix symbols												
1579	Symbols for matrices are boldface italic, too: ¹ $A = E \cdot A$.												

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 .

**E2.4 Tensor symbols**

1580 Symbols for tensors are sans-serif bold italic,

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

1581 The permittivity tensor describes the coupling of electric field and displacement:

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



	E2.5 Bold math version												
1583													
1584	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	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, ff, fi, \beta, ^!, v, w, 0, 1, 9$												
1585	New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-italic.												
	<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	$A, B, \Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Phi, \Psi, \Omega, \alpha, \beta, \pi, \nu, \omega, v, w, 0, 1, 9$												
1586	Do the math alphabets match?												
1587	$a x \alpha \omega a x \alpha \omega a x \alpha \omega \quad T C \Theta \Gamma T C \Theta \Gamma T C \Theta \Gamma$												
1588	E2.5.1 Vector symbols												
1589	Alphabetic symbols for vectors are boldface italic, $\lambda = e_1 \cdot a$, while numeric ones (e.g.												
1590	the zero vector) are bold upright, $a + 0 = a$.												
1591	E2.5.2 Matrix symbols												
1592	Symbols for matrices are boldface italic, too: ² $\Lambda = E \cdot A$.												
1593	E2.5.3 Tensor symbols												
1594	Symbols for tensors are sans-serif bold italic,												
	$\alpha = e \cdot a \iff \alpha_{ijl} = e_{ijk} \cdot a_{kl}.$												
1595	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 .



1596 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{\rmfamily} & & \mathsf{\teststring} \\
28 \mbox{\bfseries\rmfamily} & & \mathbf{\teststring} \\
29 \mbox{\rmfamily} & & \mathsf{\teststring} \\
30 \mbox{\rmfamily} & & \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 Do the math alphabets match?
40
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

```



De La Salle University

```

1651 53 Alphabetic symbols for vectors are boldface italic,
1652 54  $\vec{\lambda} = \vec{e}_1 \cdot \vec{a}$ ,
1653 55 while numeric ones (e.g. the zero vector) are bold upright,
1654 56  $\vec{a} + \vec{0} = \vec{a}$ .
1655 57
1656 58 \subsection{Matrix symbols}
1657 59
1658 60 Symbols for matrices are boldface italic, too: %
1659 61 \footnote{However, matrix symbols are usually capital letters whereas
1660 62 vectors
1661 62 are small ones. Exceptions are physical quantities like the force
1662 63 vector  $\vec{F}$  or the electrical field  $\vec{E}$ .%}
1663 64 }
1664 65  $\mathbf{\Lambda} = \mathbf{E} \cdot \mathbf{A}$ .
1665 66
1666 67
1667 68 \subsection{Tensor symbols}
1668 69
1669 70 Symbols for tensors are sans-serif bold italic,
1670 71
1671 72 \[
1672 73   \alpha = e \cdot \alpha
1673 74   \quad \Longleftarrow \quad
1674 75   \alpha_{ijl} = e_{ijk} \cdot a_{kl}.
1675 76 \]
1676 77
1677 78
1678 79 The permittivity tensor describes the coupling of electric field and
1679 80 displacement: \[
1680 81 \vec{D} = \epsilon_0 \cdot \epsilon_r \cdot \vec{E} \]
1681 82
1682 83
1683 84
1684 85 \newpage
1685 86 \subsection{Bold math version}
1686 87
1687 88 The ‘‘bold’’ math version is selected with the commands
1688 89 \verb+\boldmath+ or \verb+\mathversion{bold}+
1689 90
1690 91 {\boldmath
1691 92   \begin{eqnarray*}
1692 93     \mathnormal & & \text{teststring} \\
1693 94     \mathit & & \mathit{\text{teststring}} \\
1694 95     \mathrm & & \mathrm{\text{teststring}} \\
1695 96     \mathbf & & \mathbf{\text{teststring}} \\
1696 97     \mathsf & & \mathsf{\text{teststring}} \\
1697 98     \mathtt & & \mathtt{\text{teststring}} \\
1698 99   \end{eqnarray*}
1699 100   New alphabets bold-italic, sans-serif-italic, and sans-serif-bold-
1700 101   italic.
1701 101 {\begin{eqnarray*}
1702 102   \mathbf & & \mathbf{\text{teststring}} \\
1703 103   \mathsf & & \mathsf{\text{teststring}} \\
1704 104   \mathsfbf & & \mathsfbf{\text{teststring}}
1705 105 \end{eqnarray*}
1706 106 %
1707 107 Do the math alphabets match?

```



```

1708 108      $
1709 109      \mathnormal {a x \alpha \omega}
1710 110      \mathbf{f} {a x \alpha \omega}
1711 111      \mathsf{fbf}{a x \alpha \omega}
1712 112      \quad
1713 113      \mathsf{fbf}{T C \Theta \Gamma}
1714 114      \mathbf{f} {T C \Theta \Gamma}
1715 115      \mathnormal {T C \Theta \Gamma}
1716 116      \mathnormal {T C \Theta \Gamma}
1717 117      $
1718 118
1719 119      \subsection{Vector symbols}
1720 120
1721 121      Alphabetic symbols for vectors are boldface italic,
1722 122      $ \vec{\lambda} = \vec{e}_1 \cdot \vec{a} $,
1723 123      while numeric ones (e.g. the zero vector) are bold upright,
1724 124      $ \vec{a} + \vec{0} = \vec{a} $.
1725 125
1726 126
1727 127
1728 128
1729 129      \subsection{Matrix symbols}
1730 130
1731 131      Symbols for matrices are boldface italic, too: %
1732 132      \footnote{However, matrix symbols are usually capital letters whereas
1733 133      vectors
1734 134      are small ones. Exceptions are physical quantities like the force
1735 135      vector $ \vec{F} $ or the electrical field $ \vec{E} $. %}
1736 136      $ \mathbf{sym}{\Lambda} = \mathbf{sym}{E} \cdot \mathbf{sym}{A} . $
1737 137
1738 138
1739 139      \subsection{Tensor symbols}
1740 140
1741 141      Symbols for tensors are sans-serif bold italic,
1742 142
1743 143      \[
1744 144      \mathbf{sym}{\alpha} = \mathbf{sym}{e} \cdot \mathbf{sym}{a}
1745 145      \quad \Longleftarrow \quad
1746 146      \alpha_{ijl} = e_{ijk} \cdot a_{kl}.
1747 147
1748 148
1749 149      The permittivity tensor describes the coupling of electric field and
1750 150      displacement: \[
1751 151      \vec{D} = \epsilon_0 \mathbf{sym}{\epsilon} \cdot \mathbf{sym}{r} \cdot \vec{E} \]
1752 152
1753 153
1754 154

```



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).



- 1784 • Provide your own link text: style sheet.

1785 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}
```



1786 E4 Glossary

1787 This section shows examples of the use of `\gls{ }` commands in conjunction with the
 1788 items that are in the `glossary.tex` and `notation.tex` files. Note that entries in
 1789 `notation.tex` are prefixed with “`not:`” label (see List. E.4).

1790 **Please make sure that the entries in `notation.tex` are those that are referenced
 1791 in the L^AT_EX document files used by this Thesis. Please comment out unused notations
 1792 and be careful with the commas and brackets in `notation.tex`.**

- 1793 • Matrices are usually denoted by a bold capital letter, such as \mathbf{A} . The matrix’s (i, j) th
 1794 element is usually denoted a_{ij} . Matrix \mathbf{I} is the identity matrix.
- 1795 • A set, denoted as \mathcal{S} , is a collection of objects.
- 1796 • The universal set, denoted as \mathcal{U} , is the set of everything.
- 1797 • The empty set, denoted as \emptyset , contains no elements.
- 1798 • Functional Analysis is seen as the study of complete normed vector spaces, i.e.,
 1799 Banach spaces.
- 1800 • The cardinality of a set, denoted as $|\mathcal{S}|$, is the number of elements in the set.

1801 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

1802

E5 Figure

1803

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

1804



Fig. E.1 A quadrilateral image example.



1805 Fig. E.1 is a gray box enclosed by a dark border. List. E.5 shows the corresponding
1806 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}

```



1807

E6 Table

1808

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)



1810 List. E.8 shows the corresponding L^AT_EX code.

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

```

1811 1 \begin{center}
1812 2 {\scriptsize
1813 3 \begin{tabularx}{\textwidth}{p{0.1\textwidth}|p{0.2\textwidth}|p{0.5\textwidth}}
1814 4 \caption{Feasible triples for highly variable grid} \label{tab:triple_
1815 5 \hline
1816 6 \hline
1817 7 \textbf{Time (s)} &
1818 8 \textbf{Triple chosen} &
1819 9 \textbf{Other feasible triples} \\
1820 10 \hline
1821 11 \endfirsthead
1822 12 \multicolumn{3}{c}{\textit{Continued from previous page}}} \\
1823 13 \hline
1824 14 \hline
1825 15 \hline
1826 16 \textbf{Time (s)} &
1827 17 \textbf{Triple chosen} &
1828 18 \textbf{Other feasible triples} \\
1829 19 \hline
1830 20 \endhead
1831 21 \hline
1832 22 \multicolumn{3}{r}{\textit{Continued on next page}}} \\
1833 23 \endfoot
1834 24 \hline
1835 25 \endlastfoot
1836 26 \hline
1837 27
1838 28 0 & (1, 11, 13725) & (1, 12, 10980), (1, 13, 8235), (2, 2, 0), (3, 1, 0)
1839 29 \\
1840 30 2745 & (1, 12, 10980) & (1, 13, 8235), (2, 2, 0), (2, 3, 0), (3, 1, 0)
1841 31 \\
1842 32 5490 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1843 33 8235 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1,
1844 34 0) \\
1845 35 10980 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1,
1846 36 0) \\
1847 37 13725 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1,
1848 38 0) \\
1849 39 16470 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1850 40 19215 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1,
1851 41 0) \\
1852 42 21960 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1,
1853 43 0) \\
1854 44 24705 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1,
1855 45 0) \\
1856 46 27450 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1,
1857 47 0) \\
1858 48 30195 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
1859 49 32940 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1860 50 35685 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1861 51 38430 & (1, 13, 10980) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1862 52
1863 53
1864 54

```



De La Salle University

```

1865 43 | 41175 & (1, 12, 13725) & (1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1,
1866 0) \\
1867 44 | 43920 & (1, 13, 10980) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1868 45 | 46665 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
1869 46 | 49410 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
1870 47 | 52155 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1,
1871 0) \\
1872 48 | 54900 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1873 49 | 57645 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1874 50 | 60390 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1875 51 | 63135 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1876 52 | 65880 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1877 53 | 68625 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
1878 54 | 71370 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1879 55 | 74115 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1880 56 | 76860 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1881 57 | 79605 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1882 58 | 82350 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1883 59 | 85095 & (1, 12, 13725) & (1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1,
1884 0) \\
1885 60 | 87840 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1886 61 | 90585 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1887 62 | 93330 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1888 63 | 96075 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1889 64 | 98820 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1890 65 | 101565 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1891 66 | 104310 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1892 67 | 107055 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1893 68 | 109800 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1894 69 | 112545 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3,
1895 1, 0) \\
1896 70 | 115290 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1897 71 | 118035 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1898 72 | 120780 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1899 73 | 123525 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1900 74 | 126270 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3,
1901 1, 0) \\
1902 75 | 129015 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
1903 76 | 131760 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
1904 77 | 134505 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1905 78 | 137250 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1906 79 | 139995 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
1907 80 | 142740 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
1908 81 | 145485 & (1, 12, 16470) & (1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3,
1909 1, 0) \\
1910 82 | 148230 & (2, 2, 2745) & (2, 3, 0), (3, 1, 0) \\
1911 83 | 150975 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1912 84 | 153720 & (1, 12, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1913 85 | 156465 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1914 86 | 159210 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1915 87 | 161955 & (1, 13, 16470) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1916 88 | 164700 & (1, 13, 13725) & (2, 2, 2745), (2, 3, 0), (3, 1, 0) \\
1917 89 | \end{tabularx} \\
1918 90 | } \\
1919 91 | \end{center}

```



1921

E7 Algorithm or Pseudocode Listing

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

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

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

Require: $n \geq 0 \vee x \neq 0$

Ensure: $y = x^n$

```

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

```

Listing E.9: Sample 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     {\bfseries Input(s):} & & \\
9     $n$ & : & $n$th power; $n \in \mathbb{Z}^{+}$ \\
10    $x$ & : & base value; $x \in \mathbb{R}^{+}$ \\
11    \hline
12    {\bfseries 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}

```



1925

E8 Program/Code Listing

1926
1927

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 */

```



1928

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.



1929

E9 Referencing

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

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

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

```
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.
```

1935

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.



1944

E9.1 A subsection

1945

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

1946

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.
```

1947

1948 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.

1949

1950

1951

1952

1953

1954

1955



De La Salle University

1956

E9.1.1 A sub-subsection

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

1959

1960 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
 1961 ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus
 1962 placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor.
 1963 Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla
 1964 tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue
 1965 a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris.
 1966 Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit
 1967 amet ipsum. Nunc quis urna dictum turpis accumsan semper.



1968 E10 Citing

1969
 1970
 1971
 1972
 1973
 1974 Citing bibliography content is done using BibTeX. It requires the creation of a BibTeX file (.bib extension name), and then added in the argument of `\bibliography{ }` . For each .bib file, separate them by a comma in the argument of `\bibliography{ }` without the extension name. Building your BibTeX file (references.bib) can be done easily with a tool called JabRef (www.jabref.org).

The following subsections are examples of citations.

1975 E10.1 Books

- 1976 • [?]
- 1977 • [?]
- 1978 • [?]
- 1979 • [?]
- 1980 • [?]
- 1981 • [?]
- 1982 • [?]
- 1983 • [?]
- 1984 • [?]
- 1985 • [?]
- 1986 • [?]
- 1987 • [?]
- 1988 • [?]
- 1989 • [?]
- 1990 • [?]
- 1991 • [?]
- 1992 • [?]
- 1993 • [?]



De La Salle University

1994	• [?]
1995	• [?]
1996	• [?]
1997	• [?]
1998	• [?]
1999	• [?]
2000	• [?]
2001	• [?]
2002	• [?]
2003	• [?]
2004	• [?]
2005	• [?]
2006	• [?]
2007	• [?]
2008	• [?]
2009	• [?]
2010	• [?]
2011	• [?]
2012	• [?]
2013	• [?]
2014	• [?]
2015	• [?]
2016	• [?]
2017	• [?]
2018	• [?]
2019	• [?]



2020 **E10.2 Booklets**

- [?]

2022 **E10.3 Proceedings**

- [?]

2024 **E10.4 In books**

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]

- [?]



- 2043 • [?]
- 2044 • [?]
- 2045 • [?]
- 2046 • [?]
- 2047 • [?]
- 2048 • [?]
- 2049 • [?]
- 2050 • [?]

2051 **E10.5 In proceedings**

- 2052 • [?]
- 2053 • [?]
- 2054 • [?]
- 2055 • [?]
- 2056 • [?]
- 2057 • [?]
- 2058 • [?]

2059 **E10.6 Journals**

- 2060 • [?]
- 2061 • [?]
- 2062 • [?]
- 2063 • [?]
- 2064 • [?]
- 2065 • [?]



De La Salle University

- 2066 • [?]
- 2067 • [?]
- 2068 • [?]
- 2069 • [?]
- 2070 • [?]
- 2071 • [?]
- 2072 • [?]
- 2073 • [?]
- 2074 • [?]
- 2075 • [?]
- 2076 • [?]
- 2077 • [?]
- 2078 • [?]
- 2079 • [?]
- 2080 • [?]
- 2081 • [?]
- 2082 • [?]
- 2083 • [?]
- 2084 • [?]
- 2085 • [?]
- 2086 • [?]
- 2087 • [?]
- 2088 • [?]
- 2089 • [?]



- 2090 • [?]
- 2091 • [?]
- 2092 • [?]
- 2093 • [?]
- 2094 • [?]

2095 **E10.7 Theses/dissertations**

- 2096 • [?]
- 2097 • [?]
- 2098 • [?]
- 2099 • [?]
- 2100 • [?]
- 2101 • [?]
- 2102 • [?]

2103 **E10.8 Technical Reports and Others**

- 2104 • [?]
- 2105 • [?]
- 2106 • [?]
- 2107 • [?]
- 2108 • [?]
- 2109 • [?]
- 2110 • [?]
- 2111 • [?]
- 2112 • [?]



- 2113 • [?]
- 2114 • [?]
- 2115 • [?]
- 2116 • [?]
- 2117 • [?]
- 2118 • [?]

E10.9 Miscellaneous

- 2120 • [?]
- 2121 • [?]
- 2122 • [?]
- 2123 • [?]
- 2124 • [?]
- 2125 • [?]
- 2126 • [?]
- 2127 • [?]
- 2128 • [?]
- 2129 • [?]
- 2130 • [?]
- 2131 • [?]
- 2132 • [?]



2133 E11 Index

2134 For key words or topics that are expected (or the user would like) to appear in the Index, use
 2135 `\index{key}`, where `key` is an example keyword to appear in the Index. For example,
 2136 Fredholm integral and Fourier operator of the following paragraph are in the Index.

2137 If we make a very large matrix with complex exponentials in the rows (i.e., cosine real
 2138 parts and sine imaginary parts), and increase the resolution without bound, we approach
 2139 the kernel of the Fredholm integral equation of the 2nd kind, namely the Fourier operator
 2140 that defines the continuous Fourier transform.

2141 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.
```



E12 Adding Relevant PDF Pages

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

Listing E.16: Sample 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}
```



2146

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.



2147

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.



2148

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*

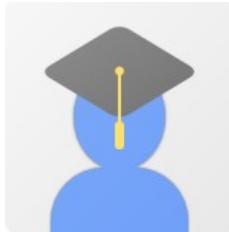
Package ⁽¹⁾⁽²⁾⁽³⁾	Package Dimensions (mm)	VU3P	VU5P	VU7P	VU9P	VU11P	VU13P
		HP, GTY	HP, GTY				
FFVC1517	40x40	520, 40					
FLVF1924	45x45					624, 64	
FLVA2104	47.5x47.5		832, 52	832, 52	832, 52		
FHVA2104	52.5x52.5 ⁽⁴⁾						832, 52
FLVB2104	47.5x47.5		702, 76	702, 76	702, 76	624, 76	
FHVB2104	52.5x52.5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

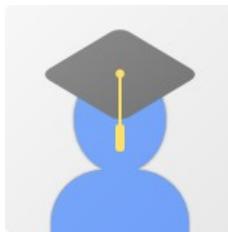
1. Go to [Ordering Information](#) for package designation details.
2. All packages have 1.0mm ball pitch.
3. Packages with the same last letter and number sequence, e.g., A2104, are footprint compatible with all other UltraScale devices with the same sequence. The footprint compatible devices within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



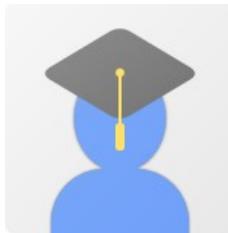
Appendix F VITA



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.

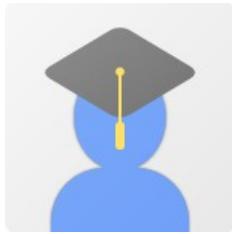


Jiro Renzo D. Tabiolo received the B.Sc., M.Sc., and Ph.D. degrees in chemistry all from the Pamantasan ng Pilipinas, San Juan, Metro Manila, Philippines, in 2020, 2022 and 2025 respectively. He is currently taking up his B.Sc. Computer Engineering studies. He has developed several high-speed packet-switched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



De La Salle University

2169 and node modules. His research interests include high-speed packet-switched networks,
2170 high speed radio interface design, discrete simulation and statistical models for packet
2171 switches.



2172 Ercid Bon B. Valencerina received the B.Sc., M.Sc., and Ph.D.
2173 degrees in chemistry all from the Pamantasan ng Pilipinas, San Juan, Metro Manila,
2174 Philippines, in 2020, 2022 and 2025 respectively. He is currently taking up his B.Sc.
2175 Computer Engineering studies. He has developed several high-speed packet-switched
2176 network systems and node modules. His research interests include high-speed packet-
2177 switched networks, high speed radio interface design, discrete simulation and statistical
2178 models for packet switches.



De La Salle University

2179

Appendix G ARTICLE PAPER(S)

2180

Article/Forum Paper Format

(IEEE LaTeX format)

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

2181

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,

2182



(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.