



**HUTECH**  
Đại học Công nghệ Tp.HCM

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH**

## **ĐỒ ÁN TỐT NGHIỆP**

**MỘT HỆ THỐNG PHÂN LOẠI ĐỘ CHÍN QUẢ CÀ CHUA  
DỰA TRÊN KỸ THUẬT THỊ GIÁC MÁY TÍNH**

**NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA**

**GIẢNG VIÊN HƯỚNG DẪN: TS PHẠM QUỐC THIỆN**

**Sinh viên thực hiện:**

Bùi Công Thịnh

**MSSV:**

1811051670

**Lớp:**

18DTDA3

Tp. Hồ Chí Minh, tháng 8 năm 2022

## VIỆN KỸ THUẬT HUTECH

## PHIẾU ĐĂNG KÝ ĐỀ TÀI ĐỒ ÁN/KHÓA LUÂN TỐT NGHIỆP

*Hệ: Chính Quy*

1. Họ và tên sinh viên/ nhóm sinh viên đăng ký đề tài (số só trong nhóm: 01):

(1) Bùi Công Thịnh MSSV: 1811051670 Lớp: 18DTDA3

Điện thoại: 0364464165 Email: buicongthinh132@gmail.com

## Ngành : Kỹ Thuật Điều Khiển Và Tự Động Hóa

- ## 2. Tên đề tài đăng ký :

*Một hệ thống phân loại độ chín quả cà chua dựa trên kỹ thuật thị giác máy tính*

Sinh viên đã hiểu rõ yêu cầu của đề tài và cam kết thực hiện đề tài theo tiến độ và hoàn thành đúng thời hạn.

TP. HCM, ngày 01 tháng 04 năm 2022.

## Sinh viên đăng ký *Ký và ghi rõ họ tên)*

## **Giảng viên hướng dẫn**

TS. Pham Quốc Thiện

Bùi Công Thịnh

**Viện Kỹ thuật Hutech**

**PHIẾU GIAO NHIỆM VỤ  
THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP**

(*Mỗi sinh viên một phiếu, GVHD ghi rõ tên đề tài và nhiệm vụ của từng sinh viên. GVHD chuyển cho SV để nộp về VP Viện.*)

**1. Sinh viên thực hiện đề tài**

Họ tên : Bùi Công Thịnh      MSSV :1811051670      Lớp: 18DTDA3  
Điện thoại : 0364464165      Email: buicongthinh132@gmail.com  
Ngành : Kỹ Thuật Điều Khiển và Tự Động Hóa

**2. Tên đề tài:**

*Một hệ thống phân loại độ chín quả cà chua dựa trên kỹ thuật thị giác máy tính*

**3. Nhiệm vụ thực hiện đề tài:**

Chương 1: Giới thiệu

1. Thực trạng của phân loại độ chín cà chua
2. Tổng quan về các hệ thống tự động hoặc giải pháp liên quan đến việc phân loại độ chín cà chua
3. Xác định mục tiêu và giới hạn của đề tài
4. Mô tả khái quát hệ thống dự kiến để đạt được mục tiêu (bao gồm cấu trúc hệ thống, các thông số kỹ thuật yêu cầu để thiết kế, các yêu cầu về an toàn, )
  - + Bản vẽ tổng thể của hệ thống
  - + Các thông số kỹ thuật yêu cầu để thiết kế
  - + Các yêu cầu về an toàn
  - + Sơ đồ khối của hệ thống
  - + Mô tả chức năng của hệ thống
  - + Các phần mềm và công cụ tính toán

Chương 2: Thiết kế

1. Thiết kế Cơ khí cho hệ thống
  - Phân tích và lựa chọn kết cấu phù hợp với yêu cầu thiết kế
  - Bản vẽ mô tả hệ thống trên CAD
2. Tính toán chọn thiết bị truyền động (khí nén, thủy lực, điện, ...)
3. Lựa chọn thiết bị điều khiển, cảm biến, khí cụ điện, thiết bị bảo vệ, ...

4. Thiết lập bản vẽ điện.

+ Bản vẽ mạch động lực

+ Bản vẽ mạch điều khiển

#### Chương 3: Thiết kế phần mềm

1. Tìm hiểu các bước triển khai Mask-RCNN dùng Python và đánh giá hiệu quả của model
2. Tìm hiểu các bước triển khai Labview và kết nối với Mask-RCNN từ Python
3. Xác định bảng địa chỉ vào ra của hệ thống
4. Mô tả hoạt động của hệ thống
5. Thiết kế lưu đồ giải thuật
6. Lập trình điều khiển hệ thống

#### Chương 4: Tổng hợp hệ thống

1. Ché tạo phần cứng
2. Thực nghiệm hoặc mô phỏng hệ thống
3. Phân tích và đánh giá kết quả

#### Chương 5: Kết luận và hướng phát triển

Tài liệu tham khảo

Phụ lục

- Chương trình điều khiển
- Bản vẽ các loại (theo tiêu chuẩn TCVN)

TP. HCM, ngày 19 tháng 4 năm 2022

**Giảng viên hướng dẫn**

(Ký và ghi rõ họ tên)

**Sinh viên thực hiện**

(Ký và ghi rõ họ tên)

**Bùi Công Thịnh**

**TS. Phạm Quốc Thiện**

## **LỜI CAM ĐOAN**

Tôi cam đoan đây là công trình nghiên cứu của tôi.

Đề tài MỘT HỆ THỐNG PHÂN LOẠI CÀ CHUA DỰA TRÊN KỸ THUẬT THỊ GIÁC MÁY TÍNH là thành quả của quá trình học tập, nghiên cứu và phát triển của tôi dưới sự định hướng, hướng dẫn tận tình của TS. Phạm Quốc Thiện.

Những nhận định, dẫn chứng, mã nguồn và tài liệu tham khảo từ các nghiên cứu, bài báo khoa học của đề tài được trích dẫn trung thực từ những nguồn mở, đã được công bố rộng rãi. Các số liệu, kết quả đạt được của đề tài nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tp. Hồ Chí Minh, ngày 24 tháng 7 năm 2022  
(Ký tên và ghi rõ họ tên)

Bùi Công Thịnh

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến thầy Phạm Quốc Thiện. Người đã cùng đồng hành, hướng dẫn, chỉ bảo tận tình để em có thể hoàn thành đồ án tốt nghiệp. Em cảm thấy thật sự may mắn khi được Thầy hướng dẫn đồ án tốt nghiệp. Đó không chỉ là những kiến thức, hướng dẫn mà còn là những sự động viên, quan tâm của Thầy.

Em cũng xin cảm ơn tới Quý Thầy, Cô Viện Kỹ Thuật HUTECH đã tận tình chỉ dạy kiến thức, có những chia sẻ, định hướng trong suốt những năm học tập tại trường. Đó là những hành trang quý báu để em bước vào quá trình lập thân, lập nghiệp sau này.

Những kỷ niệm, tình cảm, kí ức về HUTECH, Quý Thầy, Cô sẽ không thể phai mờ, nó sẽ theo em suốt những năm tháng sau này.

Em xin chân thành cảm ơn!

Bùi Công Thịnh

## TÓM TẮT

Cà chua là một rau quan trọng, là nguồn cung cấp dinh dưỡng tuyệt vời. Việc phân loại độ chín quả cà chua là rất quan trọng trong quá trình xử lý sau thu hoạch để cung cấp cà chua chất lượng tốt và đồng đều cho người tiêu dùng và chế biến sâu.

Trong đồ án tốt nghiệp này, một hệ thống phân loại cà chua dựa trên kỹ thuật thị giác máy tính đã được nghiên cứu, phát triển. Mô hình này được xây dựng nhằm có thể ứng dụng các hệ thống tự động vào quá trình phân loại cà chua.

Mô hình với các thành phần điển hình của một hệ thống thị giác máy tính phân loại trái cây [1], gồm: bộ phận vận chuyển, nguồn sáng tích hợp, máy ảnh và máy tính. Mô hình Mask R - CNN, được phát triển trong môi trường Python được dùng làm lõi của bộ phận xử lý để phân loại từng mức độ chín quả cà chua. Chương trình LabVIEW với giao diện người dùng (GUI) được phát triển với mô hình Mask R - CNN được tích hợp bằng việc sử dụng thư viện Python Node để thực hiện phân loại trực tuyến. Chương trình LabVIEW trên máy tính được kết nối với các thiết bị như PLC, Arduino thông qua giao thức truyền thông công nghiệp Modbus TCP/IP.

Đề tài đã xây dựng thành công mô hình phân loại trái cà chua dựa trên kỹ thuật thị giác máy tính. Hệ thống đã phân loại được các mức độ chín cà chua dựa trên mô hình Mask RCNN, mạng truyền thông Modbus TCP/IP hoạt động ổn định. Tuy nhiên, vẫn còn hạn chế về tốc độ xử lý, phân loại cà chua.

Hướng nghiên cứu trong tương lai là cải thiện tốc độ phân loại của mô hình, thông qua việc nâng cấp thiết bị điều khiển, cải thiện độ chính xác của mô hình Mask R - CNN. Phát triển mô hình có thể phân loại theo thời gian thực. Đồng thời, có thể sử đổi và sử dụng để phân loại các loại trái cây có kích thước tương tự.

## ABSTRACT

Tomatoes are an important vegetable that is an excellent source of nutrients. The grading of tomato fruit maturity is very important during post-harvest handling to provide uniform and good quality tomatoes for the consumer and for intensive processing.

In this graduation project, a tomato classification system based on computer vision techniques has been researched and developed. This model is built to be able to apply automatic systems to the tomato sorting process.

Model with typical components of a fruit sorting computer vision system [1], including: transporter, integrated light source, camera and computer. The Mask R - CNN model, developed in the Python environment, is used as the core of the processor to classify each tomato ripeness. LabVIEW program with user interface (GUI) developed with Mask R - CNN model integrated using Python Node library to perform online classification. The LabVIEW program on the computer is connected to devices such as PLCs, Arduinos through the Modbus TCP/IP industrial communication protocol.

The project has successfully built a model to classify tomatoes based on computer vision techniques. The system has classified the levels of tomato ripening based on the Mask RCNN model, the Modbus TCP/IP communication network operates stably. However, there are still limitations on the processing speed and classification of tomatoes.

Future research direction is to improve the classification speed of the model, through the upgrade of the control device, to improve the accuracy of the Mask R - CNN model. Real-time classifiable model development. At the same time, it can be modified and used to classify fruits of similar size.

# MỤC LỤC

LỜI MỞ ĐẦU.....	1
Chương 1: GIỚI THIỆU .....	2
1.1 Thực trạng của phân loại độ chín quả cà chua .....	2
1.1.1 Trên thế giới.....	2
1.1.2 Tại Việt Nam .....	5
1.2 Mục tiêu và giới hạn đề tài .....	6
1.2.1 Mục tiêu .....	6
1.2.2 Giới hạn đề tài.....	7
Chương 2: TỔNG QUAN GIẢI PHÁP .....	8
2.1 Các giải pháp kỹ thuật đã được áp dụng trên thực tế.....	8
2.1.1 Giải pháp về bộ phận vận chuyển .....	8
2.1.2 Giải pháp kỹ thuật về phân loại .....	9
2.2 Giải pháp đề tài đề xuất. ....	11
2.2.1 Bộ phận vận chuyển.....	11
2.2.2 Giải pháp phần mềm .....	12
2.2.3 Giải pháp thiết bị điều khiển.....	13
2.2.4 Giải pháp truyền thông. ....	13
2.3 Mô tả khái quát hệ thống dự kiến .....	16
Chương 3: THIẾT KẾ MÔ HÌNH .....	24
3.1 Tính toán chọn thiết bị truyền động.....	24
3.1.1 Phác thảo 3D mô hình băng tải .....	24
3.1.2 Tính toán động cơ băng tải .....	24
3.2 Thiết bị điều khiển, truyền thông và cảm biến. ....	27
3.2.1 PLC Siemens.....	28
3.2.2 Arduino Uno R3 và shield ethernet .....	29
3.2.3 TP – Link Desktop Switch.....	31
3.2.4 ELP - USB8MP02G – SFV .....	32
3.2.5 Cảm biến quang Optex ZR – 350N .....	34
3.3 Nút nhấn, bộ chuyển đổi nguồn và thiết bị bảo vệ .....	35
3.3.1 Nút nhấn.....	35
3.3.2 Nút nhấn dừng khẩn cấp .....	36
3.3.3 Bộ chuyển đổi nguồn 24 VDC .....	36
3.3.4 Relay trung gian OMRON MY4N .....	37
3.3.5 Động cơ DC Servo MG996R.....	37

3.3.6 Tính toán, lựa thiết bị bảo vệ .....	38
3.4 Thiết lập bản vẽ điện.....	39
3.4.1 Bản vẽ mạch động lực.....	39
3.4.2 Bản vẽ mạch điều khiển.....	40
Chương 4: THIẾT KẾ PHẦN MỀM .....	42
4.1 Các bước triển khai Mask R-CNN dùng Python và đánh giá hiệu quả model .....	42
4.1.1 Tổng quan về mạng neural tích chập và RCNNs.....	42
4.1.2 Mask R – CNN .....	48
4.1.3 Các bước triển khai mô hình Mask R – CNN.....	49
4.1.4 Đánh giá mô hình Mask R – CNN.....	58
4.2 Thiết kế chương trình điều khiển và giao diện người dùng .....	62
4.2.1 Lưu đồ giải thuật của chương trình.....	62
4.2.2 Thư viện và các kỹ thuật sử dụng .....	66
Chương 5: THI CÔNG MÔ HÌNH .....	75
5.1 Hình ảnh mô hình thi công .....	75
5.2 Giao diện điều khiển trên máy tính.....	79
Chương 6: KẾT QUẢ, HƯỚNG PHÁT TRIỂN .....	80
6.1 Kết quả đạt được .....	80
6.2 Hạn chế .....	80
6.3 Hướng phát triển .....	80
TÀI LIỆU THAM KHẢO .....	82
CHƯƠNG TRÌNH ĐIỀU KHIỂN .....	85
Chương trình Python: .....	85
Chương trình Arduino Uno:.....	86
Chương trình PLC S7 – 1200: .....	88
Chương trình LabVIEW .....	91
TÀI LIỆU KỸ THUẬT MỘT SỐ THIẾT BỊ.....	93
Động cơ giảm tốc 5840 – 31ZY .....	93
Động cơ servo MG - 996R.....	93
MCB Fuji .....	94
Cảm biến Optex ZR – 350N .....	95

## DANH SÁCH CÁC BẢNG

<b>Bảng 3. 1</b> Thông số kỹ thuật chính của Arduino Uno .....	30
<b>Bảng 4. 1</b> Confusion matrix của mô hình Mask R - CNN .....	59
<b>Bảng 4. 2</b> Bảng các tham số của Confusion matrix.....	59
<b>Bảng 4. 3</b> Bảng địa chỉ kết nối chương trình PLC.....	73
<b>Bảng 4. 4</b> Bảng địa chỉ kết nối chương trình Arduino.....	74
<b>Bảng 4. 5</b> Bảng địa chỉ kết nối mạng truyền thông Modbus TCP/IP .....	74

## **DANH MỤC CÁC TỪ VIẾT TẮT**

<b>STT</b>	<b>Từ viết tắt</b>	<b>Từ gốc</b>
1	RFID	Radio Frequency Identification
2	GUI	Graphical User Interface
3	LabVIEW	Laboratory Virtual Instrumentation Engineering Workbench
4	R - CNN	Region with CNN
5	PLC	Programmable Logic Control
6	ML	Machine Learning
7	USB	Universal Serial Bus
8	MCB	Miniature Circuit Breaker
9	COCO	Common Object in Context
10	BIC	Border/Interiorpixel Classification
11	CNN	Convolutional neural network
12	ROI	Region of Interest
13	IDE	Integrated Development Environment
14	GIMP	GNU Image Manipulation Program

# DANH SÁCH CÁC HÌNH

<b>Hình 1. 1</b> Hình máy phân loại GeoSortIII [4] .....	3
<b>Hình 1. 2</b> Máy phân loại Tomato Tek II [6] .....	4
<b>Hình 1. 3</b> Máy phân loại của anh Nguyễn Hồng Chương [8] .....	5
<b>Hình 2. 1</b> Con lăn hình diabolo [9].....	8
<b>Hình 2. 2</b> Buồng tối của hệ thống phân loại Greefa [9] .....	9
<b>Hình 2. 3</b> Hình ảnh Greefa – flaps [9] .....	10
<b>Hình 2. 4</b> Mô tả cơ cấu phân loại của dòng máy Tomato Tek .....	10
<b>Hình 2. 5</b> Thông kê về tỉ lệ các mạng truyền thông trong công nghiệp. [16] .....	14
<b>Hình 2. 6</b> Giao thức Modbus và các kỹ thuật truyền dẫn [17] .....	15
<b>Hình 2. 7</b> Bản vẽ 3D mô hình hệ thống dự kiến.....	16
<b>Hình 2. 8</b> Sơ đồ khối hệ thống.....	18
<b>Hình 2. 9</b> Biểu tượng Spyder và giao diện chính. ....	19
<b>Hình 2. 10</b> Giao diện chính của phần mềm LabVIEW (ảnh chụp màn hình cá nhân) .....	20
<b>Hình 2. 11</b> Biểu tượng phần mềm TIA Portal V17 (ảnh chụp màn hình cá nhân) ..	21
<b>Hình 2. 12</b> Biểu tượng phần mềm Arduino (ảnh chụp màn hình).....	22
<b>Hình 2. 13</b> Biểu tượng phần mềm SolidWorks (ảnh chụp màn hình cá nhân) .....	22
<b>Hình 3. 1</b> Mô hình 3D băng tải và cách bố trí động cơ .....	24
<b>Hình 3. 2</b> Mô tả con lăn và vector lực .....	25
<b>Hình 3. 3</b> Một hệ thống dẫn động băng tải điển hình [20]. .....	26
<b>Hình 3. 4</b> Động cơ 5840 – 31ZY (ảnh chụp thiết bị của mô hình).....	27
<b>Hình 3. 5</b> Hình ảnh PLC S7 – 1200 (ảnh chụp thiết bị của mô hình) .....	28
<b>Hình 3. 6</b> Arduino Uno R3 [18].....	30
<b>Hình 3. 7</b> Shield Arduino Ethernet W5100 [18] .....	31
<b>Hình 3. 8</b> Switch TP – Link (ảnh chụp thiết bị mô hình) .....	32
<b>Hình 3. 9</b> Camera ELP (ảnh cắt từ catalogue thiết bị) .....	33
<b>Hình 3. 10</b> Cảm biến ZR – 350N và gương (ảnh cắt catalogue thiết bị).....	34
<b>Hình 3. 11</b> Sơ đồ đấu nối (ảnh cắt từ catalogue thiết bị) .....	34
<b>Hình 3. 12</b> Nút nhấn nhả không đèn xanh – đỏ CHINT – NP2 .....	35
<b>Hình 3. 13</b> Nút dừng khẩn cấp CHINT NP2 - ES .....	36
<b>Hình 3. 14</b> Nguồn tổ ong 24 V .....	37
<b>Hình 3. 15</b> MCB Fuji.....	38
<b>Hình 3. 16</b> Phần bản vẽ thiết bị động lực. .....	39
<b>Hình 3. 17</b> Hình ảnh bản vẽ điều khiển PLC.....	40
<b>Hình 3. 18</b> Bản vẽ đấu nối Arduino.....	41

<b>Hình 4. 1</b> Các bước trong R – CNN [23].....	43
<b>Hình 4. 2</b> Đánh giá mô hình Fast R - CNN .....	45
<b>Hình 4. 3</b> Đánh giá tốc độ Faster R – CNN và các mô hình khác .....	47
<b>Hình 4. 4</b> 3 hình ảnh mẫu cà chua thua thập lần 1. ....	50
<b>Hình 4. 5</b> Hình ảnh một mẫu cà chua lần thu thập 2 .....	50
<b>Hình 4. 6</b> Giao diện chính của GIMP .....	51
<b>Hình 4. 7</b> Chuẩn bị vẽ biên dạng. ....	51
<b>Hình 4. 8</b> Cà chua sau khi vẽ biên dạng .....	52
<b>Hình 4. 9</b> Lớp layer màu xanh trên hình dạng cà chua.....	52
<b>Hình 4. 10</b> Hình ảnh sau xử lý.....	53
<b>Hình 4. 11</b> Cấu trúc tệp mask_definitions.json của đề tài .....	55
<b>Hình 4. 12</b> Tệp dataset_info.json của đề tài. ....	56
<b>Hình 4. 13</b> Cú pháp lệnh tạo tệp coco_annotations.json .....	56
<b>Hình 4. 14</b> Giao diện chính Colab (ảnh chụp màn hình).....	58
<b>Hình 4. 15</b> Kết quả kiểm tra mô hình .....	61
<b>Hình 4. 16</b> Lưu đồ giải thuật chính.....	62
<b>Hình 4. 17</b> Lưu đồ giải thuật chương trình PLC .....	64
<b>Hình 4. 18</b> Lưu đồ giải thuật chương trình Arduino .....	65
<b>Hình 4. 19</b> Hàm Node Python (ảnh chụp trong phần mềm LabVIEW).....	67
<b>Hình 4. 20</b> Hàm Close Python Session (ảnh chụp trong phần mềm LabVIEW) .....	67
<b>Hình 4. 21</b> Hàm Open Python Session (ảnh chụp trong phần mềm LabVIEW).....	68
<b>Hình 4. 22</b> Biểu tượng thư viện Modbus API trong LabVIEW .....	68
<b>Hình 4. 23</b> Vị trí thư viện Modbus trong LabVIEW .....	68
<b>Hình 4. 24</b> Các hàm trong chức năng Master API .....	69
<b>Hình 4. 25</b> Cấu trúc thiết lập truyền thông thư viện MgsModbus của đề tài .....	71
<b>Hình 4. 26</b> Khối MB_SERVER sử dụng trong đề tài .....	72
<b>Hình 4. 27</b> Khối MB_CONFIG và các thiết lập của đề tài .....	72

## LỜI MỞ ĐẦU

Tự động hóa là một ngành kỹ thuật quan trọng trong một nền công nghiệp hiện đại. Tự động hóa đang được ứng dụng rộng rãi trong rất nhiều lĩnh vực khác nhau: công nghiệp sản xuất, gia công cơ khí, dây chuyền lắp ráp tự động, ứng dụng kiểm soát chất lượng... Trong nền sản xuất hiện đại, tự động hóa giúp cải thiện năng suất lao động, cắt giảm chi phí nhân công, nâng cao chất lượng sản phẩm, giảm chi phí nguyên liệu.

Với sự phát triển của công nghệ Máy học (Machine Learning), thị giác máy tính đã đạt được những tiến bộ rất lớn. Ngày càng có nhiều ứng dụng đưa thị giác máy tính vào các dây chuyền phân loại, xử lý sản phẩm nhằm thực hiện quá trình tự động hóa.

Là sinh viên theo học ngành “Kỹ thuật điều khiển và tự động hóa”, em hiểu được một phần sự quan trọng của độ chính xác, đồng đều, liên tục của một hệ thống phân loại sản phẩm. Với những kiến thức đã được học, em muốn được nghiên cứu và tìm hiểu, thiết kế chế tạo HỆ THỐNG PHÂN LOẠI ĐỘ CHÍN QUẢ CÀ CHUA DỰA TRÊN KỸ THUẬT THỊ GIÁC MÁY TÍNH. Mặc dù đã cố gắng hết sức mình để hoàn thiện hệ thống nhưng do còn nhiều khó khăn về kiến thức nên không tránh khỏi những thiếu sót.

# **Chương 1: GIỚI THIỆU**

## **1.1 Thực trạng của phân loại độ chín quả cà chua**

Cà chua (*Solanum lycopersicum*) là cây rau quan trọng thứ hai trên thế giới sau khoai tây. Sản lượng hiện tại trên toàn thế giới là khoảng 170,75 triệu tấn quả tươi được sản xuất trên 5,02 triệu ha ở hơn 150 quốc gia [2]. Cà chua là nguồn cung cấp tuyệt vời với nhiều chất dinh dưỡng và các chất chuyển hóa thứ cấp quan trọng đối với sức khỏe con người: chất khoáng, vitamin C và E, B-carotene, lycopene, flavonoid, axit hữu cơ, phenol và chất diệp lục.

Cà chua là loại quả hô hấp đột biến, có thể thu hoạch ở các giai đoạn chín khác nhau và tiếp tục chín trong quá trình vận chuyển, xử lý, bảo quản, tiêu thụ. Do vậy, độ chín thu hoạch có mối liên quan chặt chẽ với chất lượng quả khi chín [3]. Việc phân loại độ chín cà chua là rất quan trọng trong quá trình xử lý sau thu hoạch để cung cấp cà chua chất lượng tốt và đồng đều cho tiêu dùng và chế biến sâu.

### **1.1.1 Trên thế giới**

Với sự phát triển của khoa học công nghệ, các hãng chế tạo trên thế giới đã đưa ra nhiều hệ thống, máy phân loại cà chua với độ chính xác cao, hoạt động đáng tin cậy đáp ứng được nhu cầu của thị trường. Có nhiều công nghệ khác nhau đã được sử dụng để tạo nên những máy, hệ thống phân loại này. Hai xu hướng công nghệ chính là: sử dụng các cảm biến màu sắc; sử dụng các camera tích hợp thị giác máy tính.

- Máy phân loại trái cây và rau củ GeoSortIII.

GeoSortIII do công ty N.M. Bartlett Inc nghiên cứu và phát triển [4]. Được thành lập năm 1912, là công ty cung cấp các sản phẩm phục vụ trong ngành nông nghiệp của Canada [5]. GeoSortIII có thể phân loại tám quả mỗi giây. Có thể cùng lúc độ chín, kích thước và khối lượng trái.



**Hình 1. 1** Hình máy phân loại GeoSortIII [4]

Máy có 3 module [4] để thực hiện các chức năng:

- Module kích thước: Kích thước trái cây được đo bằng camera hồng ngoại CCD độ phân giải cao (đen trắng). Kích thước của mỗi quả được đo nhiều lần bằng sự kết hợp đặc biệt giữa phần cứng và phần mềm. Điều này đảm bảo phép đo cực kỳ chính xác.
- Module màu: Màu sắc của trái cây được xác định bằng cách sử dụng máy ảnh màu CCD đặc biệt. Trái cây được phân loại bằng cách sử dụng các máy ảnh này theo giá trị trung bình đối với má hồng, màu sắc hoặc màu nền. Hơn nữa, việc sử dụng các bộ lọc đặc biệt giúp loại bỏ ảnh hưởng của bất kỳ sự phản chiếu nào.
- Module trọng lượng: Trái cây được cân điện tử bằng hệ thống cân 3 điểm tiên tiến. Hệ thống này được tìm thấy trong đơn vị chuyển giao. Điều này đảm bảo độ chính xác cực cao và giảm thiểu nguy cơ làm nứt bẩn các phần tử cân.

- TomatoTek II Field Sorter [6]



**Hình 1.2** Máy phân loại Tomato Tek II [6]

Sản phẩm do công ty WECO (Hoa Kỳ) phát triển. TomatoTek II xác định và loại bỏ các sản phẩm cà chua không màu, thối, cháy nắng và các sản phẩm lỗi khác, cũng như tất cả các loại MOT (vật liệu không phải là cà chua) bao gồm bụi bẩn, cuống và đá. Thiết bị có thể phân loại khối lượng cà chua trên diện tích 6 ha/ngày [6]. Các tính năng chính bao gồm:

- Nhận dạng tần số vô tuyến (RFID)
- Hệ thống định vị toàn cầu (GPS)
- Điều khiển từ xa không dây
- Bảng điều khiển của trình điều khiển. Ghi và truyền dữ liệu
- TomatoTek II chỉ yêu cầu một màn hình điều khiển duy nhất và đáp ứng được tất cả các kiểu máy gặt và chiều rộng.

Máy có thể phân loại 2 mức cà chua là xanh và chín. Việc đặt trên khung gầm xe cơ giới lớn phù hợp với các trang trại, cánh đồng mẫu lớn có mức độ cơ giới hóa cao.

Sẽ là thách thức lớn khi hoạt động với các diện tích canh tác nhỏ lẻ, chia cắt với nền đất yếu tại Việt Nam.

### 1.1.2 Tại Việt Nam

Theo đánh giá của Bộ Công Thương Việt Nam, phương pháp sản xuất rau quả, ché biến, phân loại bảo quản sau thu hoạch còn tiến hành thủ công, lạc hậu [7]. Những năm gần đây, đã có một số máy phân loại cà chua do người nông dân sáng chế, giúp nâng cao năng suất lao động. Tiêu biểu là Máy rửa, phân loại, đánh bóng cà chua của anh Nguyễn Hồng Chương (Lâm Đồng).

- Máy rửa, phân loại và đánh bóng cà chua (Nguyễn Hồng Chương) [8]

Máy rửa, phân loại và đánh bóng trái cà chua của anh Nguyễn Hồng Chương hoạt động mỗi ngày 8 tiếng, đạt năng suất 20 tấn, bằng khoảng 20 người dùng phương pháp thủ công để làm việc.



**Hình 1. 3** Máy phân loại của anh Nguyễn Hồng Chương [8]

Máy rửa, phân loại, hong sấy khô nước và đánh bóng trên trái cà chua có hình chữ L, bao gồm các bộ phận cơ bản như thùng đựng cà chua thô. Từ đây, một bộ băng truyền hoạt động tương tự như cầu thang máy sẽ tự động lấy cà chua chuyển tới bộ phận sàng lọc các loại chất thải còn dính trên quả như rác, lá, cuống, cà chua

hư hỏng, thối... tách bỏ phần chất thải này ra khỏi quả. Cà chua tiếp tục được chuyển tới bộ phận đựng nước rửa.

Khi đã được rửa sạch, băng chuyên chuyên tới vùng sấy khô nước trên cà chua, sau đó đưa tới vùng phân chia thành các loại cà chua lớn nhỏ khách nhau và đánh bóng trái cà chua. Khi đã hoàn tất các công đoạn trên, trái cà chua được đưa ra ngoài bằng các máng trượt, mỗi máng là một loại kích cỡ khác nhau. “Nay với chiếc máy này, vẫn lượng cà chua đó chỉ làm có một buổi thôi, lại chỉ cần có 4 người ngồi, thay vì trước đây phải cần 20 [8].

Sản phẩm này đã giải quyết cơ bản các vấn đề của phân loại cà chua, giá thành hợp lý so với các sản phẩm của nước ngoài. Tuy vậy, ở khâu phân loại cà chua theo màu sắc vẫn phải sử dụng lao động thủ công.

## 1.2 Mục tiêu và giới hạn đề tài

### 1.2.1 Mục tiêu

- Mục tiêu của đề tài là xây dựng thành công một mô hình hệ thống phân loại độ chín trái cà chua dựa trên kỹ thuật thị giác máy tính.
- Một giao diện người dùng thân thiện được thực hiện trong môi trường LabVIEW để vận hành hệ thống.
- Mô hình Mask R-CNN được phát triển trong môi trường Python để phát hiện cho hệ thống phân loại.
- Các thành phần của mô hình được kết nối qua mạng truyền thông công nghiệp Modbus TCP/IP. Mạng hoạt động ổn định, phục vụ tốt việc giao tiếp, điều khiển, giám sát các thiết bị.
- Thiết kế phần mềm, phần cứng linh hoạt, có thể dễ dàng sửa đổi, áp dụng để xây dựng các mô hình phân loại trái cây khác có kích thước tương đương cà chua.

### **1.2.2 Giới hạn đè tài**

Do kiến thức, khả năng thi công, chi phí thiết bị và thời gian có giới hạn, đè tài dừng lại ở mức độ tìm hiểu, thiết kế và thi công mô hình có thể phân loại theo 3 mức màu sắc, độ chính xác ở mức trên 90%.

Từ những kết quả đạt được của đè tài là cơ sở chỉnh sửa, bổ sung, cải thiện nhằm phát triển một hệ thống phân loại có độ chính xác cao, tốc độ nhanh, hoạt động ổn định có thể áp dụng vào thực tế phục vụ sản xuất.

## **Chương 2: TỔNG QUAN GIẢI PHÁP**

### **2.1 Các giải pháp kỹ thuật đã được áp dụng trên thực tế.**

Tìm hiểu thực tế trên các máy phân loại thương mại hóa của Greefa GeoSort, WECO tác giả nhận thấy có một số giải pháp hiệu quả cho các hệ thống phân loại trái cây nói chung, cà chua nói riêng.

#### **2.1.1 Giải pháp về bộ phận vận chuyển**

Phần băng tải vận chuyển: một băng tải được ghép từ các con lăn được thiết kế đặc biệt. Các con lăn hình diabolo được gắn các bàn chải (diabolo-shaped brush). Thiết kế này giúp cho khi đi qua buồng tối, nơi đặt các camera trái cây sẽ xoay nhiều góc độ khác nhau [9]. Camera sẽ chụp được nhiều ảnh hơn, từ đó kết quả đánh giá sẽ chính xác hơn.



**Hình 2. 1 Con lăn hình diabolo [9]**

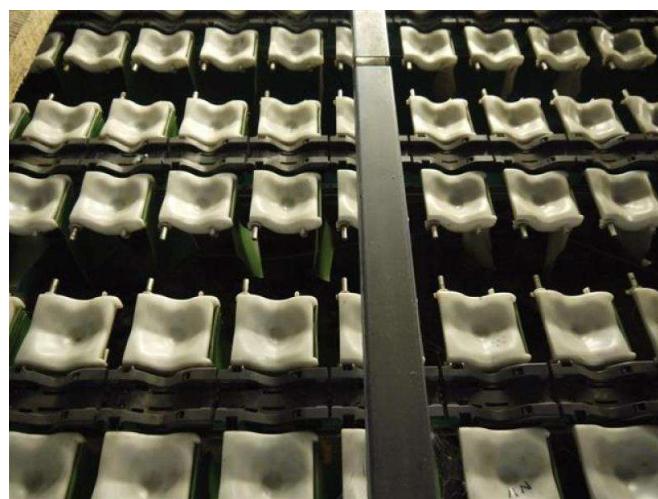
### 2.1.2 Giải pháp kỹ thuật về phân loại

Cà chua, hay các loại trái cây được phân loại sẽ được đưa qua buồng tối. Camera, đèn trợ sáng sẽ được đặt trên giá treo trong buồng tối này. Buồng tối này đã loại bỏ đi ánh sáng từ môi trường bên ngoài, giúp cho chất lượng ảnh chụp luôn được đảm bảo. Tránh cho việc phải liên tục thay đổi các tham số liên quan tới độ sáng của camera.



**Hình 2. 2** Buồng tối của hệ thống phân loại Greefa [9]

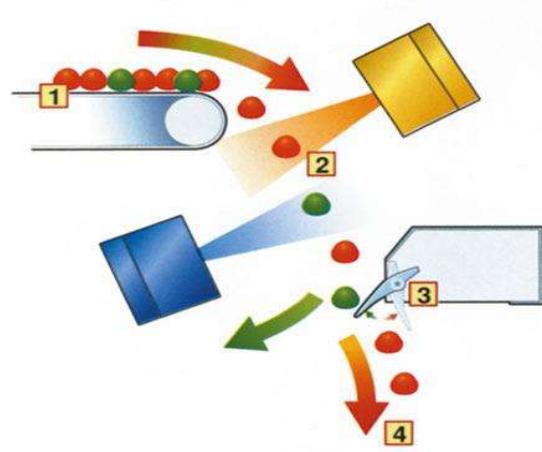
Trái cây sẽ được vận chuyển trên 1 băng truyền với cơ cấu được gọi là Greefa – flaps. Là sáng chế độc quyền của Greefa, nắp được thiết kế giúp cho việc phân loại trở nên hiệu quả, liên tục.



**Hình 2. 3** Hình ảnh Greefa – flaps [9]

Những nắp này vừa đóng vai trò như một bộ phận vận chuyển, vừa là một thiết bị phân loại. Khi đi qua vị trí lưu giữ đúng với nhãn quả (do hệ thống xử lý phân tích đưa ra) đang đặt trên nắp, nắp sẽ nghiêng hoặc mở ra (với nắp kép) quả sau phân loại sẽ tới đúng vị trí chứa được người vận hành cài đặt.

Máy phân loại của công ty WECO sử dụng các thanh đánh, loại bỏ trực tiếp các quả hư hỏng.



**Hình 2. 4** Mô tả cơ cấu phân loại của dòng máy Tomato Tek

Hệ thống phân loại của WECO sử dụng thiết bị quang học phân tích màu trên quả cà chua, và loại bỏ những quả không đạt yêu cầu bằng cần gạt. Giải pháp này đem lại một tốc độ phân loại nhanh, độ tin cậy cao. Tuy nhiên, máy chỉ có 2 mức phân loại là xanh và chín, gây lãng phí 1 lượng cà chua không đạt yêu cầu chín.

## 2.2 Giải pháp để tài để xuất.

Do có những hạn chế lớn về công nghệ, kỹ thuật gia công, cũng như về mặt tài chính để tài đã lựa chọn một giải pháp đơn giản hơn, những vẫn đáp ứng đủ các thành phần của một hệ thống phân loại dựa trên kỹ thuật thị giác máy tính. Gồm:

- Bộ phận vận chuyển: băng tải dây đai
- Buồng tối, camera và đèn.
- Cơ cấu phân loại sử dụng động cơ servo điều khiển góc quay.
- Máy tính.
- Thiết bị điều khiển: PLC, Arduino.
- Truyền thông Modbus TCP/IP qua Ethernet và Switch

### 2.2.1 Bộ phận vận chuyển

Mô hình sử dụng băng tải phẳng. Băng tải phẳng đã giải quyết cơ bản vấn đề bộ phận vận chuyển của đè tài:

- Giúp vận chuyển cà chua một cách thuận tiện và nhanh chóng.
- Cấu tạo đơn giản, dễ thiết kế chế tạo, thi công mô hình.
- Vật liệu để chế tạo có nguồn cung cấp lớn, sẵn có trên thị trường.

Tuy nhiên, đi kèm với các thuận lợi kể trên, băng tải phẳng cũng có một số hạn chế với mô hình:

- Chỉ thể hiện được một mặt của cà chua. Điều này hạn chế lớn đối với độ chính xác của mô hình.

- Do là một mặt phẳng, cà chua có thể bị xô lệch, lăn trong quá trình vận chuyển gây khó khăn cho camera thu hình.

### 2.2.2 Giải pháp phần mềm

Trong thập kỷ qua, các nhà nghiên cứu đã cố gắng phát triển các hệ thống tự động dựa trên thị giác máy tính để giải quyết việc phân loại hoa quả thủ công tồn tại nhiều thời gian. Nhìn chung, một hệ thống thị giác máy tính để phân loại trái cây bao gồm các thành phần như bộ phận vận chuyển, nguồn sáng tích hợp, máy ảnh và máy tính [1]. Hệ thống phân loại tự động và kiểm tra chất lượng táo, được phát triển bởi Sofu và các cộng sự [10] đã sử dụng hai máy ảnh để thu thập hình ảnh. Hệ thống được đề xuất có thể phân loại trực tuyến 15 quả táo mỗi giây với độ chính xác 73-96%. Abdul Wajid và các cộng sự [11] đã trích xuất không gian màu RGB và các giá trị xám dựa trên BIC (Border/Interior pixel Classification) để phân biệt tình trạng cam (chín, chưa chín và đóng vảy hoặc thối) với kết quả đạt được về độ chính xác là 93,13%. Phương pháp phát hiện và phân loại là vô cùng quan trọng trong hệ thống phân loại cà chua tự động

Với sự phát triển của công nghệ máy học (Machine Learning) gần đây, thị giác máy tính trở nên mạnh mẽ hơn và hữu ích hơn cho việc kiểm soát chất lượng trái cây. RCNNs (Region-based Convolutional Neural Network) là một tập hợp con chuyên biệt trong ML đã trở thành kỹ thuật phổ biến trong việc phát hiện và phân đoạn hình ảnh trái cây trong những năm gần đây.

RCNN và các phiên bản cải tiến (Fast R - CNN, Faster R - CNN, Mask R - CNN) trích xuất nhiều đề xuất từ hình ảnh đầu vào, sau đó CNN được sử dụng để dự đoán lớp và hộp giới hạn của vùng đề xuất [12]. Faster R - CNN tạo ra kết quả với nhãn lớp (class label) và một hộp giới hạn (bounding box) với tốc độ nhanh hơn Fast R - CNN nhờ tích hợp thuật toán đề xuất các region. Mask R - CNN là phiên bản mở rộng của Faster R - CNN, có thêm một nhánh để dự đoán mask của đối tượng. Yang Yu và cộng sự tại China Agricultural University đã xây dựng mô hình phát hiện dâu tây cho robot thu hoạch dựa trên mô hình Mask R-CNN với tỉ lệ phát hiện trung bình 95.78% [13].

Thị giác máy tính kết hợp với LabVIEW đã được sử dụng rộng rãi trong việc kiểm tra chất lượng của trái cây [14, 15].

Qua những tìm hiểu về các kỹ thuật phân loại trái cây và sự định hướng của Thầy GVHD, giải pháp kỹ thuật của đề tài đưa ra là kết hợp phần mềm LabVIEW với mô hình Mask R - CNN. Đây là kỹ thuật, định hướng kiến thức cho giải pháp về bài toán phân loại cà chua của đề tài.

### **2.2.3 Giải pháp thiết bị điều khiển**

Với mục tiêu đặt ra của đề tài là phát triển một mô hình được định hướng gần sát với thực tế sản xuất, PLC là thiết bị điều khiển chính các cơ cấu trong mô hình hệ thống.

Bộ điều khiển khả trình - PLC (Programmable Logic Control) là loại thiết bị cho phép thực hiện linh hoạt các thuật toán điều khiển logic thông qua 1 ngôn ngữ lập trình thay cho việc phải thực hiện các thuật toán đó bằng các mạch số. Do có chương trình điều khiển bên trong nên PLC trở thành bộ điều khiển số nhỏ gọn, dễ dàng thay đổi thuật toán và đặc biệt là dễ dàng trao đổi thông tin với môi trường xung quanh.

Với mô hình của đề tài, việc sử dụng PLC có thể coi là dư thừa và tốn kém. Tuy vậy, việc sử dụng một thiết bị được ứng dụng vào thực tế sản xuất cao như PLC kéo gần hơn tính khả thi của mô hình khi hướng tới mục đích cuối cùng của mọi đề tài là thực tế sản xuất.

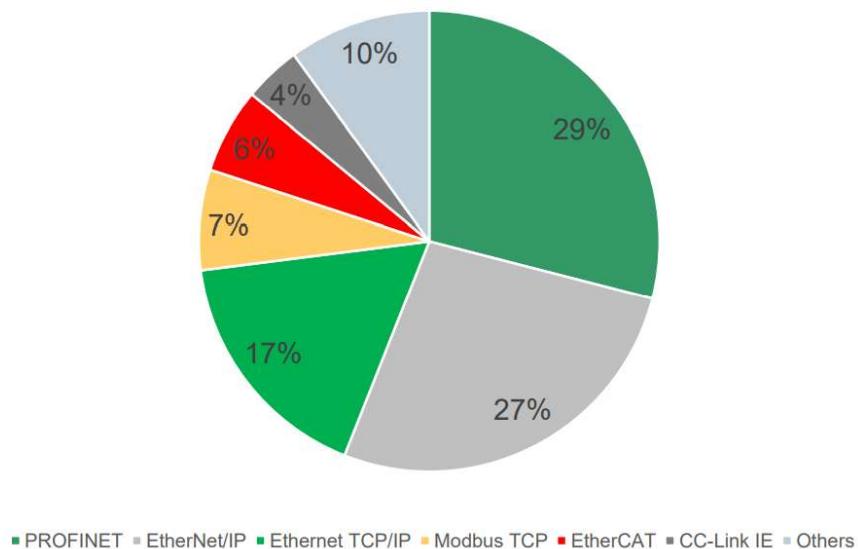
### **2.2.4 Giải pháp truyền thông.**

#### **2.2.4.1 Giao thức truyền thông**

Trong giao tiếp tự động hóa, một cuộc đối thoại/hội thoại giữa các thiết bị diễn ra một cách có hệ thống, được thực hiện thông qua các loại mạng giao tiếp khác nhau và bằng các ngôn ngữ khác nhau. Tùy thuộc vào nhà sản xuất, một ngôn ngữ cụ thể hoặc độc quyền được sử dụng hoặc có thể là ngôn ngữ mở (Open language) cho công nghiệp.

Giao thức mở (Open protocol) có nghĩa là các thông số kỹ thuật được phát hành và có thể được sử dụng bởi bất kỳ ai một cách tự do hoặc theo giấy phép.

Điều này cung cấp cho người dùng nhiều sự lựa chọn hơn về thiết bị hoặc hệ thống có thể được sử dụng để.



**Hình 2. 5** Thống kê về tỉ lệ các mạng truyền thông trong công nghiệp. [16]

Việc áp dụng mạng truyền thông công nghiệp với đè tài sẽ kéo gần khoảng cách giữa mô hình hệ thống đè tài xây dựng và ứng dụng sản xuất thực tế và giải quyết một số vấn đề sau:

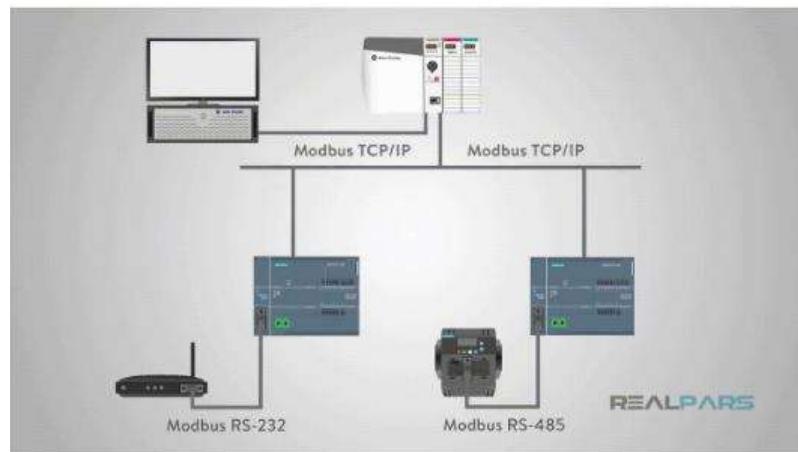
- Các hệ thống phân loại sử dụng thị giác máy tính trong mô hình hay sản xuất đều cần có máy tính. Máy tính có thể là máy tính nhúng, máy tính công nghiệp... Tuy vậy trong môi trường sản xuất công nghiệp, việc đè máy tính cạnh các thiết bị, máy móc với môi trường có nhiều hóa chất, bụi, nhiệt... sẽ làm ảnh hưởng tới độ bền, tin cậy của máy tính và hệ thống phân loại.
- Việc sử dụng truyền thông là tiền đề, cơ sở cho việc nghiên cứu, sửa đổi, nâng cấp, các thiết bị phù hợp đè ứng dụng vào sản xuất. Việc thay thế một thiết bị

vào hệ thống dùng chuẩn giao tiếp công nghiệp sẽ tiện lợi hơn việc phải sửa đổi lại chương trình điều khiển để phù hợp với thiết bị.

#### 2.2.4.2 Giải pháp truyền thông dung giao thức truyền thông Modbus TCP/IP

Modbus là giao thức truyền thông lâu đời nhất và phổ biến nhất trong lĩnh vực tự động hóa quá trình và SCADA [17]. Một số giao thức Modbus tồn tại cho cổng nối tiếp và Ethernet và phổ biến nhất là:

- Modbus RTU (nối tiếp).
- Modbus ASCII (nối tiếp).
- Modbus TCP (Ethernet).
- Modbus Plus (nối tiếp).



**Hình 2. 6** Giao thức Modbus và các kỹ thuật truyền dẫn [17]

Các thiết bị Modbus giao tiếp bằng kỹ thuật Master-Slave:

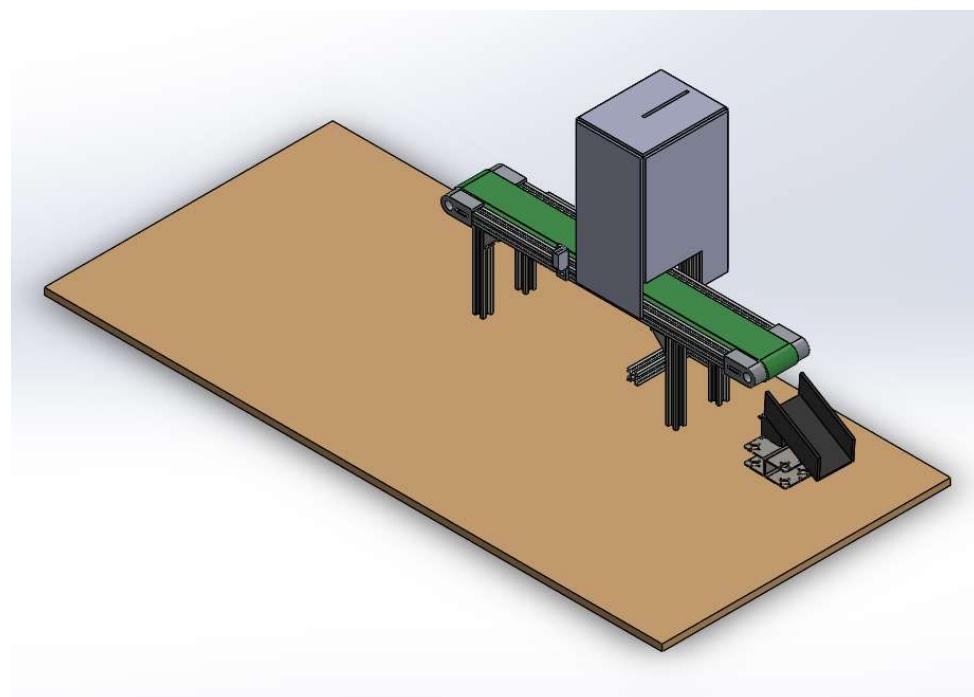
- Một thiết bị đóng vai trò là Master → bắt đầu quá trình giao tiếp (gửi yêu cầu/lệnh).
- Các thiết bị khác (Slave) phản hồi bằng cách cung cấp dữ liệu được yêu cầu cho Master hoặc thực thi hành động được yêu cầu.

Slave có thể là bất kỳ thiết bị ngoại vi nào như: bộ chuyển đổi I/O, van. Driver động cơ và các loại thiết bị đo lường khác → xử lý thông tin và gửi dữ liệu phản hồi đến master. Modbus và các giao thức khác có thể đồng thời tồn tại trên cùng một giao diện vật lý [17].

Để tài sử dụng nhiều thiết bị không đồng nhất: Máy tính, PLC, Arduino nên việc lựa chọn sử dụng hệ thống mạng TCP dụng Ethernet sẽ linh hoạt hơn.

## 2.3 Mô tả khái quát hệ thống dự kiến

### 2.3.1 Bản vẽ mô hình 3D dự kiến hệ thống



**Hình 2. 7** Bản vẽ 3D mô hình hệ thống dự kiến

Hệ thống này bao gồm:

- Camera để thu ảnh cho hệ thống xử lý.
- Đèn cung cấp ánh sáng cho camera.
- Cảm biến quang phát hiện quả cà chua.

- Cơ cấu vận chuyển: gồm động cơ và băng tải. Vận chuyển băng tải qua hệ thống camera tới cơ cấu phân loại
- Cơ cấu phân loại: gồm động cơ servo, khung và máng xoay.

Trên cơ sở đó, tiến hành thiết kế lần lượt các cơ cấu cơ khí của hệ thống.

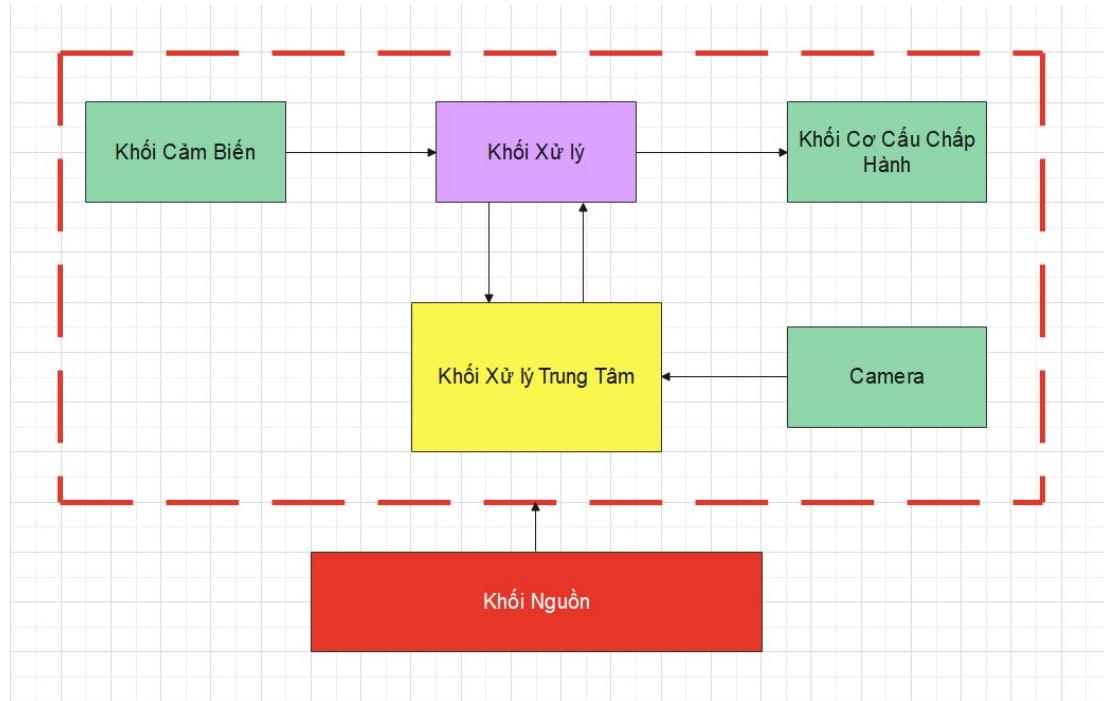
### **2.3.2 Các thông số kỹ thuật yêu cầu để thiết kế**

- Điện áp sử dụng: 1 pha 220V, 50Hz. Dòng tải: tùy theo công suất động cơ, đèn sử dụng và các thiết bị ngoại vi khác, thông thường không nhỏ hơn 5A.
- Khung sườn thiết bị: được thiết kế bằng nhôm định hình, chắc chắn, dễ tháo lắp, thay đổi và nâng cấp.
- Mặt phẳng đặt thiết bị: gỗ thông phẳng, chắc chắn.
- Có băng tải vận chuyển cà chua, khối lượng vận chuyển: 200 g, chiều dài 60 – 70 cm, hoạt động ổn định.
- Camera có cảm biến hình ảnh, độ nét cao, có thể điều chỉnh tiêu cự. Ngõ ra USB 2.0, 3.0 tương thích với Laptop.
- Có đèn, buồng chụp được thiết kế hợp lý với tỉ lệ mô hình, gần với thực tế.
- Mô hình có thể áp dụng một trong các giao thức truyền thông công nghiệp. Gần với thực tiễn cuộc sống.
- Cảm biến: sử dụng cảm biến quang có độ nhạy cao, tin cậy.
- Cơ cấu phân loại đơn giản, hiệu quả, có thời gian đáp ứng nhanh.
- Giao diện người dùng (GUI) thân thiện, dễ sử dụng. Phần mềm được xây dựng theo module, có thể tùy biến, nâng cấp dễ dàng.

### **2.3.3 Các yêu cầu về an toàn**

- Các biện pháp an toàn điện: được trang bị MCB chống giật trên toàn hệ thống và các thiết bị điện khác đạt chuẩn CE.
- Nút Emergency (dừng khẩn cấp) khi có sự cố xảy ra.

#### 2.3.4 Sơ đồ khái của hệ thống



Hình 2. 8 Sơ đồ khái hệ thống.

#### 2.3.5 Mô tả chức năng của hệ thống

- Khối xử lý trung tâm: Máy tính. Nhận dữ liệu từ PLC, Arduino, xử lý hình ảnh thu về từ camera, chạy mô hình Mask R – CNN trên LabVIEW để phân loại cà chua.
- Khối cảm biến: xác định vị trí của cà chua khi đi tới buồng tối.
- Khối xử lý: bao gồm PLC và Arduino. Đọc tín hiệu cảm biến, điều khiển các động cơ, đèn và giao tiếp với máy tính qua mạng truyền thông công nghiệp
- Khối cơ cầu chấp hành: gồm băng tải vận chuyển và máng xoay cà chua.
- Camera: Thu thập hình ảnh

### 2.3.6 Các phần mềm và công cụ tính toán

#### 2.3.6.1 Spyder IDE

Spyder là một môi trường phát triển tích hợp đa nền tảng mã nguồn mở để lập trình khoa học bằng ngôn ngữ Python.



Hình 2.9 Biểu tượng Spyder và giao diện chính.

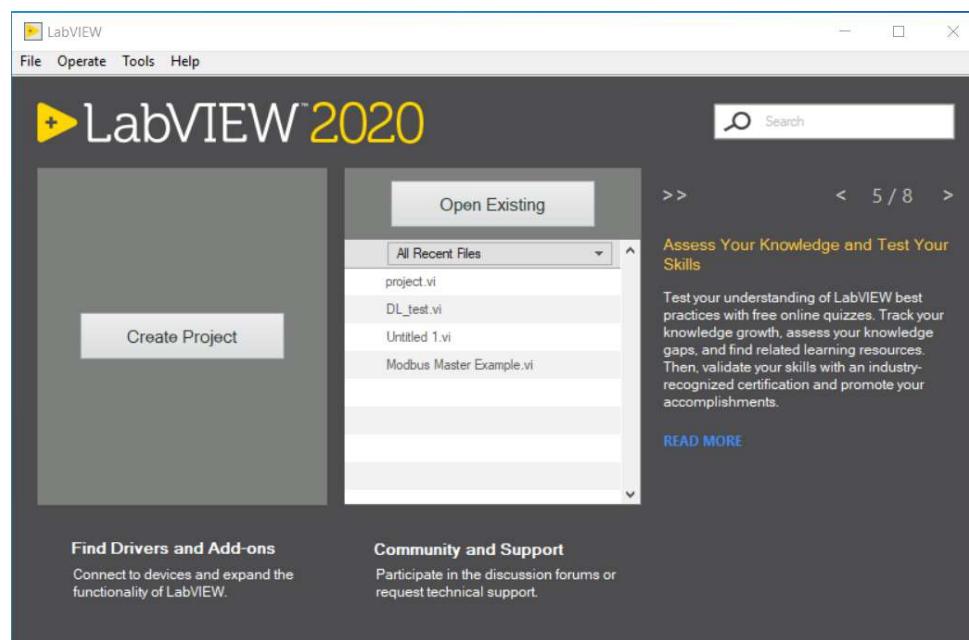
Một số tính năng chính của Spyder:

- Đánh dấu cú pháp có thể tùy chỉnh
- Tính khả dụng của các điểm ngắt
- Có thể xóa biến tự động
- Hỗ trợ tất cả các lệnh IPython
- Hiển thị nội tuyến cho đồ họa được tạo bằng Matplotlib
- Thực thi tương tác cho phép chạy dòng, tệp...

Với đê tài, Spyder được sử dụng là môi trường phát triển để xây dựng bộ dữ liệu COCO, mô hình Mask R-CNN. Bên cạnh Spyder còn cần sử dụng các thư

viện được viết bằng ngôn ngữ Python trong quá trình xây dựng tập dữ liệu, nội dung này sẽ được trình bày ở phần sau của cuốn báo cáo.

### 2.3.6.2 LabVIEW 2020 SP1



**Hình 2. 10** Giao diện chính của phần mềm LabVIEW (ảnh chụp màn hình cá nhân)

LabVIEW là một môi trường lập trình đồ họa có thể dễ dàng phát triển các giao diện người dùng cho các ứng dụng kỹ thuật.

LabVIEW 2020 được sử dụng trong đề tài để chạy mô hình Mask R-CNN, giao tiếp Modbus TCP/IP với PLC và Arduino, thu thập hình ảnh từ camera và xây dựng giao diện người dùng để điều khiển, giám sát toàn bộ mô hình. Để làm được các yêu cầu trên, ngoài các khối lệnh cơ bản, các thư viện Node Python, Modbus API Library, NImaxdx của LabVIEW đã được sử dụng.

### 2.3.6.3 TIA Portal V17



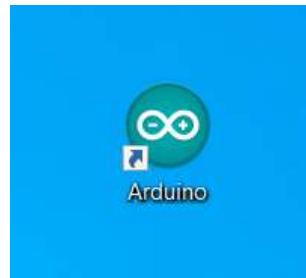
**Hình 2. 11** Biểu tượng phần mềm TIA Portal V17 (ảnh chụp màn hình cá nhân)

TIA Portal (Totally Intergrated Automation Portal) là một phần mềm tổng hợp của nhiều phần mềm điều hành quản lý tự động hóa, vận hành điện của hệ thống do Siemens AG phát triển. Một số phần mềm được sử dụng nhiều của TIA Portal:

- STEP 7 Professional. Là công cụ để cấu hình, lập trình cho các bộ điều khiển khả năng (PLC) SIMATIC. Bao gồm các dòng CPU S7 – 1200, S7 – 1500, S7 – 300, S7 – 400.
- WinCC Comfort/Advanced/Professional
- STEP 7 Startdrive
- PLC SIM
- SIMATIC OPC UA

TIA Portal V17 là phiên bản nâng cấp mới nhất. Với đòn bẩy tài chính, TIA Portal được sử dụng để lập trình cho PLC S7 – 1200

#### 2.3.6.4 Arduino IDE



**Hình 2. 12** Biểu tượng phần mềm Arduino (ảnh chụp màn hình)

“Phần mềm Arduino mã nguồn mở (IDE) giúp bạn dễ dàng viết mã và tải nó lên các board. Phần mềm này có thể được sử dụng với bất kỳ board Arduino nào.” [18]

Đó là những gì mà các kỹ sư phát triển phần mềm từ Arduino.cc đã miêu tả về Arduino IDE.

Để tài đã sử dụng các thư viện được hỗ trợ bởi Arduino IDE (gồm: servo.h, mgsModbus) để điều khiển động cơ servo cho cơ cấu phân loại cũng như truyền thông Modbus TCP/IP trong hệ thống.

#### 2.3.6.5 SolidWorks



**Hình 2. 13** Biểu tượng phần mềm SolidWorks (ảnh chụp màn hình cá nhân)

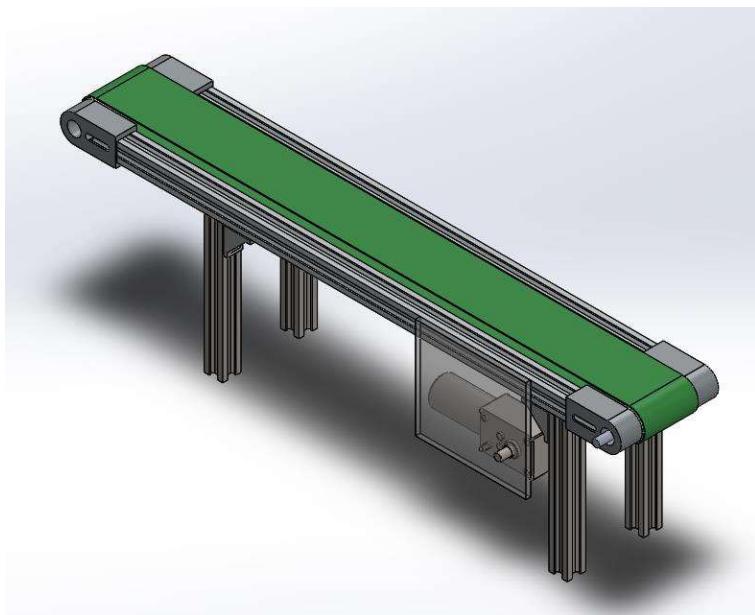
SolidWorks là phần mềm thiết kế 3D chạy trên hệ điều hành Windows và có mặt từ năm 1997, được tạo bởi công ty Dassault Systèmes SolidWorks Corp.

SolidWorks đã giúp ích rất lớn trong việc thiết kế mô hình phần cứng hệ thống phân loại cà chua. Việc thiết kế các chi tiết trở nên dễ dàng nhờ tính trực quan, phương pháp xây dựng mô hình 3D tham số, nhanh chóng, dễ dàng và tiện lợi. SolidWorks giúp dễ dàng lắp ráp các chi tiết tạo thành một mô hình hoàn chỉnh.

## Chương 3: THIẾT KẾ MÔ HÌNH

### 3.1 Tính toán chọn thiết bị truyền động

#### 3.1.1 Phác thảo 3D mô hình băng tải



**Hình 3. 1** Mô hình 3D băng tải và cách bố trí động cơ

- Đối tượng vận chuyển của băng tải: quả cà chua
- Kích cỡ được xác định bằng đường kính mặt cắt lớn nhất của quả. Để tài sử dụng các quả cà chua có đường kính từ 30 – 56 mm, ứng với mã kích cỡ 1 – 4 theo TCVN 4845:2007 [them endnote]
- Các thông số của băng tải:
  - Chiều cao: 150 mm
  - Chiều dài: 600 mm
  - Chiều rộng (lòng băng tải): 60 mm. Đáp ứng được yêu cầu vận chuyển
  - Đường kính ruller: 30 mm

#### 3.1.2 Tính toán động cơ băng tải

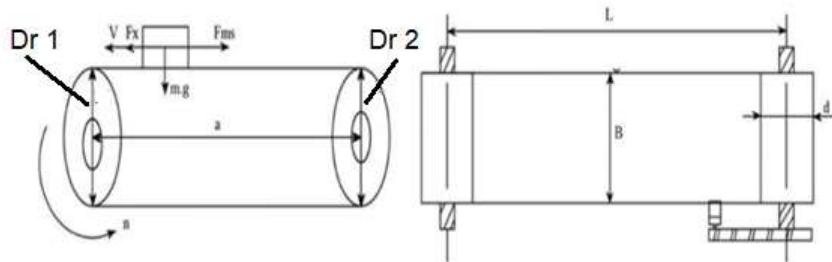
Do đây là thiết kế mô hình băng tải phẳng, nên các thông số về kích thước được cân đối cho phù hợp với kích thước chung của mô hình và có tính thẩm mỹ. Một

số chi tiết như con lăn đỡ băng, cơ cấu tăng đơ... được lược bỏ, phù hợp với tính chất của mô hình. Thiết kế cũng định hướng việc sử dụng động cơ gắn sẵn hộp giảm tốc để tiện cho việc tính toán.

Đè tài cũng bỏ qua một số tính toán như lực căng dây, khối lượng phân bố dây băng tải... Tập trung vào việc tính công suất dẫn động, tốc độ quay.

- Các thông số cho trước băng tải:

- Đường kính con lăn dẫn động (roller):  $D_R = D_{R1} = D_{R2} = 30 \text{ mm}$
- Đường kính pulley 1:  $D_1 = 15 \text{ mm}$
- Đường kính pulley 2:  $D_2 = 15 \text{ mm}$
- Tốc độ vận chuyển của băng tải:  $0.02 \text{ m/s}$
- Khối lượng trung bình cà chua có trên băng tải:  $m = 0.2 \text{ Kg}$



**Hình 3. 2** Mô tả con lăn và vector lực

- Để vật cân bằng trên băng tải khi chuyển động, ta có tổng hợp lực:

$$\vec{P} + \vec{N} + \vec{F}_{ms} + \vec{F}_k = 0$$

- Trọng lượng P:

$$P = m.g = 0.2 \times 9.8 = 1,96 \text{ N}$$

- Phản lực N của băng tải với sản phẩm:  $N = P = 1,96 \text{ N}$
- Lực ma sát của bộ truyền:

$$F_{ms} = k.N = 0.3 \times 1,96 = 0,588 \text{ N}$$

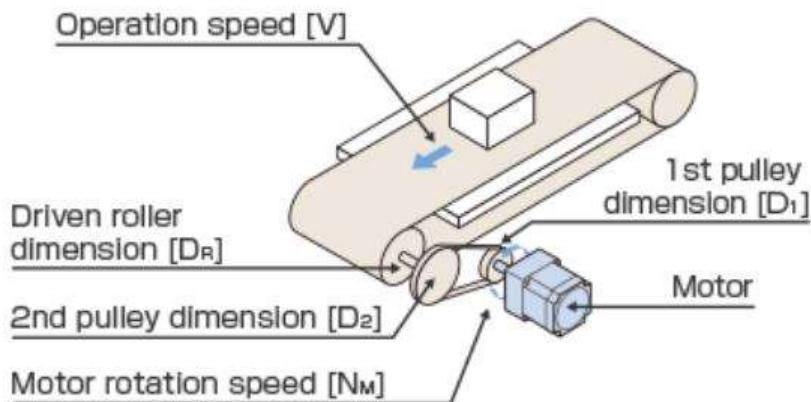
k là hệ số ma sát giữa cà chua và băng tải. Chọn  $k = 0.3$ , là hệ số ma sát giữa cà chua và nhựa [19].

- Lực kéo:

$$F_k = ma + F_{ms} = 0,2 \times \frac{0,02^2}{2 \times 0,4} + 0,588 = 0,5881 \text{ N}$$

- Momen:

$$M = F_k \times (D_R : 2) = 0,588 \times 0,015 = 0,00882 \text{ N.m}$$



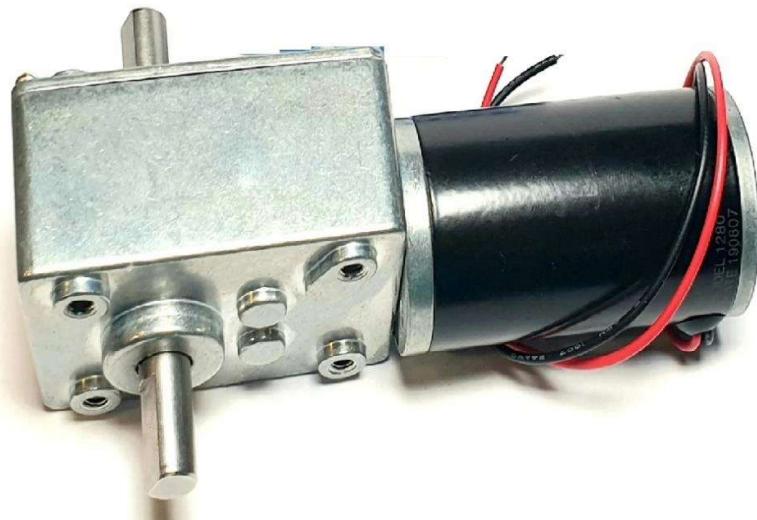
**Hình 3. 3** Một hệ thống dẫn động băng tải điển hình [20].

- Tốc độ quay của động cơ:

$$N_M = 1000 \times \frac{V}{\pi D_R} + \frac{D_2}{D_1} \times 60 \quad [20]$$

$$N_M = 1000 \times \frac{0.02}{\pi 30} + \frac{15}{15} \times 60 = 12,73 \text{ r/min}$$

- Động cơ lựa chọn: Động cơ giảm tốc 5840 – 31ZY



**Hình 3.4** Động cơ 5840 – 31ZY (ảnh chụp thiết bị của mô hình)

- Thông số kỹ thuật chính:
  - Điện áp: 12 VDC
  - Tỉ số truyền (reduction ratio): 31:1
  - Dòng không tải (no load current): 350 mA
  - Dòng có tải (load current): <1.5 A
  - Tốc độ không tải (no load rpm): 260 rpm
  - Tốc độ có tải (load rpm): 200 rpm
  - Lực kéo Moment định mức (load torque): 5 Kg.cm
  - Lực leo Moment tối đa (max torque): 8 Kg.cm

### **3.2 Thiết bị điều khiển, truyền thông và cảm biến.**

Để đáp ứng các yêu cầu đặt ra của đề tài các thiết bị phù hợp đã được lựa chọn:

- PLC Siemens SIMATIC S7 – 1200
- Arduino Uno R3 và shield ethernet
- Switch ethernet
- Camera ELP
- Cảm biến quang Optex ZR-350N

### 3.2.1 PLC Siemens

Đề tài sử dụng bộ SIMATIC S7 – 1200 (CPU 6ES7211 – 1AE40 – 0XB0).



**Hình 3. 5** Hình ảnh PLC S7 – 1200 (ảnh chụp thiết bị của mô hình)

- Thông số kỹ thuật chính:
  - Loại sản phẩm: CPU 1211 C
  - Mã sản phẩm: 6ES7211 – 1AE40 – 0XB0
  - Phiên bản phần mềm: V4,5
  - Phần mềm lập trình: STEP 7 V17
  - Ngõ vào số: 6 DI, 24VDC
  - Ngõ vào tương tự: 2 AI, 0 - 10 VDC
  - Ngõ ra số: 4 DO, 24 VDC
  - Bộ nhớ chương trình/dữ liệu: 50 KB
  - Điện áp nguồn cấp: 20,4 – 28,8 VDC

- Dòng điện: 300 mA (rated value) – 900 mA (max)
- Kích thước: 10,10 x 10,60 x 8,70 (cm)
- Khối lượng: 0,303 Kg

SIMATIC S7 – 1200 được trang bị nhiều cơ chế giao tiếp truyền thông:

- Tích hợp giao diện điều khiển I/O PROFINET.
- Module truyền thông Master/Slave PROFIBUS DP.
- Module GPRS để kết nối với mạng di động GSM/G.
- Module LTE để liên lạc trong các mạng di động thế hệ thứ 4 (sự phát triển lâu dài).
- Bộ xử lý truyền thông để kết nối với phần mềm TeleControl Server Basic điều khiển phần mềm trung tâm thông qua Ethernet, cũng như để giao tiếp an toàn qua mạng dựa trên IP.
- Bộ xử lý truyền thông để kết nối trung tâm điều khiển cho các ứng dụng điều khiển từ xa.
- RF120C để kết nối với hệ thống nhận dạng SIMATIC.
- Module SM1278 để kết nối cảm biến IO-Link và thiết bị truyền động.
- Kết nối point-to-point thông qua các module truyền thông hoặc bo mạch truyền thông.

### **3.2.2 Arduino Uno R3 và shield ethernet**

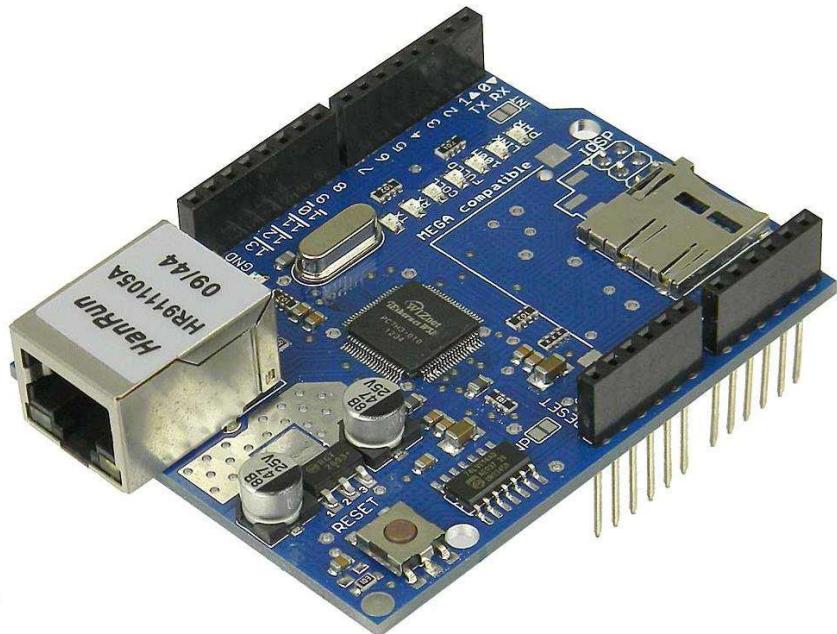
Arduino là một hệ thống nhúng khép kín với các thiết bị ngoại vi, bộ xử lý và bộ nhớ.



**Hình 3. 6** Arduino Uno R3 [18]

STT	Tiêu chí	Vị điều khiển ATmega328
1	Điện áp hoạt động	5V
2	Điện áp đầu vào	7 – 12V
3	Điện áp đầu vào (giới hạn)	6 – 20V
4	Chân Digital I/O	14 (Với 6 chân PWM Output)
5	Chân PWM Digital I/O	6
6	Chân đầu vào Analog	6
7	Dòng sử dụng I/O Pin	20mA
8	Dòng sử dụng 3.3V Pin	50mA
9	Bộ nhớ Flash	32 kB (Atmega328)
10	SRAM	2 kB (Atmega328)
11	EEPROM	1 kB (Atmega328)
12	Clock Speed	16Mhz
13	LED_BULTIN	13
14	Chiều dài	68.6 mm
15	Chiều rộng	53.4 mm
16	Trọng lượng	25g

**Bảng 3. 1** Thông số kỹ thuật chính của Arduino Uno



**Hình 3. 7** Shield Arduino Ethernet W5100 [18]

- Shield Arduino Ethernet cho phép Arduino Uno kết nối với mạng. Cổng RJ 45 trên shield này giúp Uno có thể kết nối với mạng cùng với việc sử dụng một vài thư viện hỗ trợ. Thiết bị chính là một chip ethernet W5100, cổng RJ 45, khe cắm SD card (không bắt buộc sử dụng). Có thể tìm hiểu chi tiết về chip W5100 tại đây [21]

### 3.2.3 TP – Link Desktop Switch

Để thể xây dựng mạng Modbus TCP/IP, cần có những thiết bị như switch (bộ chia mạng), hub để kết nối các thiết bị: laptop, PLC, Arduino. TP – Link Desktop Switch TL – SF1005D là thiết bị phù hợp được lựa chọn.



**Hình 3. 8** Switch TP – Link (ảnh chụp thiết bị mô hình)

- Thông số kỹ thuật chính:
  - Số lượng cổng RJ45: 5
  - Tốc độ truyền: Ethernet: 10, Fast Ethernet 100 Mbps
  - Tiêu chuẩn kỹ thuật: IEEE 802.3, IEEE 802.3i, IEEE 802.3u, IEEE 802.3x
  - Kích thước: 103,5 x 70 x 22 mm
  - Điện áp: 5 VDC
  - Dòng điện: 20 mA

### 3.2.4 ELP - USB8MP02G – SFV

Camera là thiết bị quan trọng, thành phần không thể thiếu của một hệ thống thị giác máy tính. Nhằm đáp ứng nhu cầu đặt ra của đề tài với camera là thiết bị có thể thu thập hình ảnh có độ nét cao, hoạt động ổn định camera ELP 8MP đã được lựa chọn.



**Hình 3.9** Camera ELP (ảnh cắt từ catalogue thiết bị)

- Thông số kỹ thuật chính:
  - Resolution: 8 Megapixel 3264 x 2248
  - Công nghệ cảm biến: CMOS
  - Cảm biến hình ảnh: Sony IMX 179
  - Ống kính: CS 5 – 50 mm (có thể điều chỉnh)
  - Kích thước ống kính: 1/3,2 inch
  - Định dạng hình ảnh: MJPEG, YUY2
  - Điện áp: 5 VDC (cổng USB)

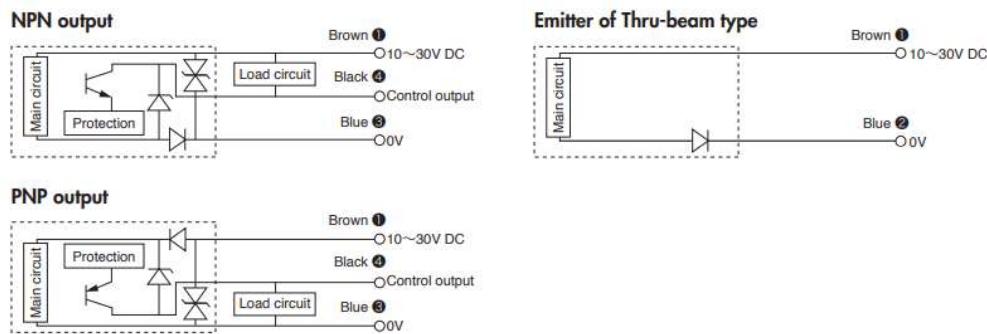
### 3.2.5 Cảm biến quang Optex ZR – 350N



**Hình 3. 10** Cảm biến ZR – 350N và gương (ảnh cắt catalogue thiết bị)

- Thông số kỹ thuật chính:
  - Loại: Phản chiếu ngược phân cực
  - Khoảng cách phát hiện: 3,5 m
  - Thời gian đáp ứng: 0,5 msec
  - Ngõ ra: NPN
  - Điện áp cung cấp: 10 – 30 VDC,
  - Dòng tiêu thụ: 20 mA (max)
- Cách đấu nối với thiết bị điều khiển:

Circuit diagram



**Hình 3. 11** Sơ đồ đấu nối (ảnh cắt từ catalogue thiết bị)

### 3.3 Nút nhấn, bộ chuyển đổi nguồn và thiết bị bảo vệ

#### 3.3.1 Nút nhấn



**Hình 3. 12** Nút nhấn nhả không đèn xanh – đỏ CHINT – NP2

- Thông số kỹ thuật:
  - Điện áp đèn: 24 VDC/20mA
  - Điện áp hoạt động: AC - 220V – 4,5 A, DC - 110V - 0,6A
  - Tiếp điểm: NO, NC, 1 NO + 1 NC
  - Màu sắc: Đỏ, Xanh, Vàng
  - Tiêu chuẩn: IEC/EN60947-5-1

### 3.3.2 Nút nhấn dừng khẩn cấp



**Hình 3. 13** Nút dừng khẩn cấp CHINT NP2 - ES

- Thông số kỹ thuật:
  - Điện áp hoạt động: AC – 220 V – 4,5 A, DC - 110V – 0,6 A
  - Tiếp điểm: NC
  - Tiêu chuẩn: IEC/EN60947-5-1

### 3.3.3 Bộ chuyển đổi nguồn 24 VDC

Do các thiết bị CPU PLC, các cảm biến quang, các cuộn hút relay sử dụng nguồn 24VDC nên sử dụng bộ chuyển đổi nguồn điện lưới 1 pha 220V về nguồn 24V để cấp cho các thiết bị



**Hình 3. 14 Nguồn tần số 24 V**

- Thông số của Nguồn tần số 24V 5A:
  - Công suất: 120W.
  - Điện áp đầu ra: 24 VDC.
  - Bảo vệ: Quá tải, quá áp, ngắn mạch.
  - Đầu vào: 110 VAC - 220 VAC
  - Đầu ra: 2 cặp.
  - Kích thước: 110 x 220 x 49mm.

#### **3.3.4 Relay trung gian OMRON MY4N**

- Thông số kỹ thuật:
  - Điện áp: 24 VDC
  - Dòng điện: 37,7 mA
  - Giới hạn tải: 250VAC/5A, 30VDC/5A

#### **3.3.5 Động cơ DC Servo MG996R.**

- Thông số kỹ thuật chính:
  - Khối lượng: 55 g
  - Kích thước: 40,7 x 19,7 x 42,9 mm
  - Momen xoắn: 9,4 kg.cm
  - Running Current: 500 mA – 900 mA

- Dòng tối đa (Stall Current): 2,5 A

### 3.3.6 Tính toán, lựa thiết bị bảo vệ

- Các thiết bị sử dụng nguồn 12 VDC:
  - Động cơ giảm tốc:  $I_n = 1,6 \text{ A}$
  - Arduino Uno + Shiled Ethernet:  $I_n = 0,2 \text{ A}$
  - Động cơ servo:  $I_n = 0,15 \text{ A}$ , Stall current: 2,5 A
  - Đèn trợ sáng:  $I_n = 0,2 \text{ A}$
- Các thiết bị sử dụng nguồn 24 VDC
  - PLC S7 – 1200 1211 DC/DC/DC:  $I_{max} = 0,75 \text{ A}$
  - Relay trung gian:  $I_n = 0,037 \times 2 = 0,074 \text{ A}$
  - Cảm biến ZR – 350N: 0,02 A
  - Nút nhấn (nút nhấn start, stop, emergency):  $I_n = 0,02 \times 3 = 0,06 \text{ A}$

Do dòng điện sử dụng cho nghiên cứu là 220V tần số 50Hz. Để cung cấp điện áp phù hợp cho các thiết bị, mô hình sử dụng: 1 bộ chuyển đổi nguồn 24 VDC 5A, 1 bộ chuyển đổi nguồn 12 VDC 5A.

Để bảo vệ toàn bộ hệ thống thiết bị, đè tài sử dụng MCB Fuji 10A

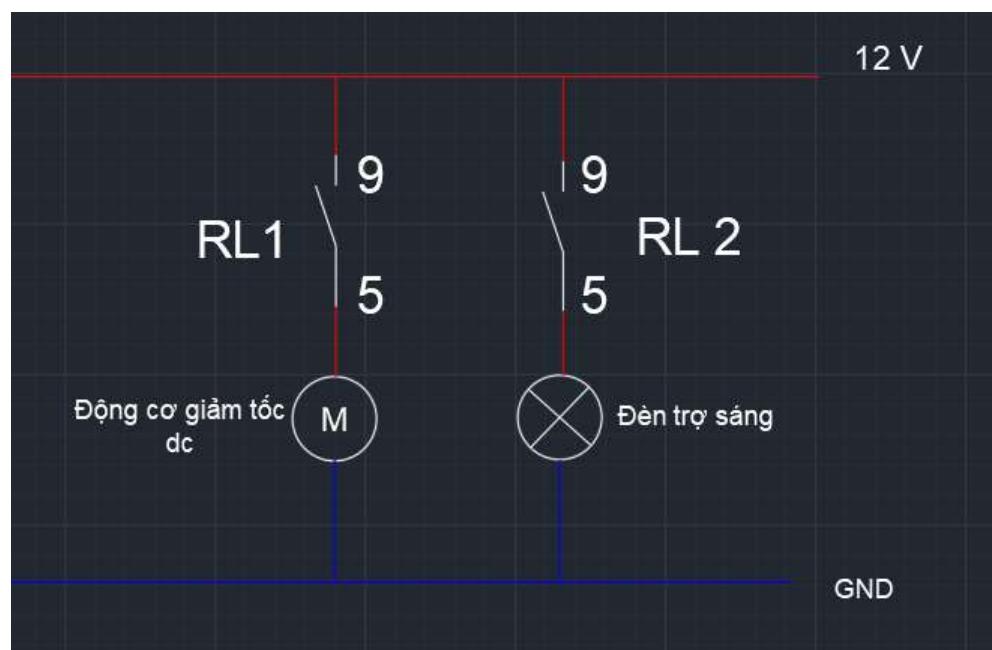


**Hình 3. 15** MCB Fuji

- Thông số chính của MCB Fuji BC63E1CG
  - Dòng danh định:  $I_{n} = 10 \text{ A}$
  - Dòng cắt 6 kA
  - Số cực: 2P
  - Số pha: 1

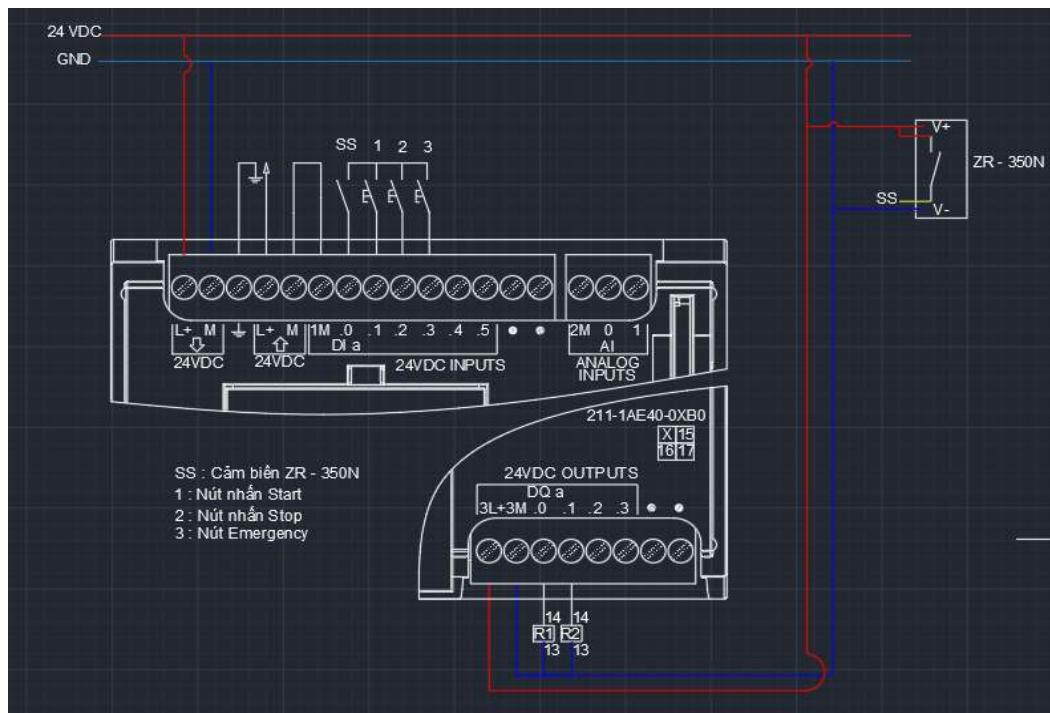
### 3.4 Thiết lập bản vẽ điện

#### 3.4.1 Bản vẽ mạch động lực

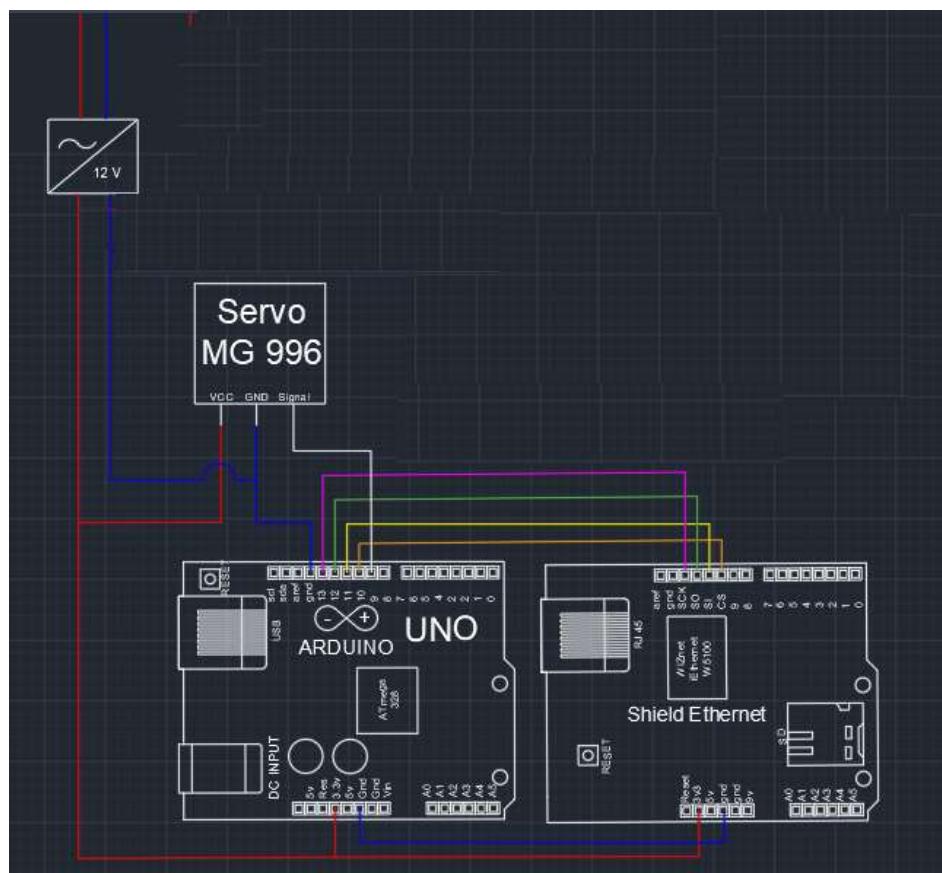


**Hình 3. 16** Phần bản vẽ thiết bị động lực.

### 3.4.2 Bản vẽ mạch điều khiển



Hình 3.17 Hình ảnh bản vẽ điều khiển PLC



Hình 3. 18 Bản vẽ đấu nối Arduino

## Chương 4: THIẾT KẾ PHẦN MỀM

### 4.1 Các bước triển khai Mask R-CNN dùng Python và đánh giá hiệu quả model

#### 4.1.1 Tổng quan về mạng neural tích chập và RCNNs

##### 4.1.1.1 Convolutional neural network (mạng neural tích chập)

Neural network là một hệ thống tính toán lấy cảm hứng từ sự hoạt động của các nơ-ron trong hệ thần kinh. Mô hình mạng neural tích chập (CNN) là một trong những mô hình nhận dạng và phân loại hình ảnh trong mạng neural [12].

CNN phân loại hình ảnh bằng cách lấy hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định. Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy  $H \times W \times D$  ( $H$ : Chiều cao,  $W$ : Chiều roojjing,  $D$ : Độ dày). Ví dụ: hình ảnh là mảng ma trận RGB  $6 \times 6 \times 3$  (3 ở đây là giá trị RGB)

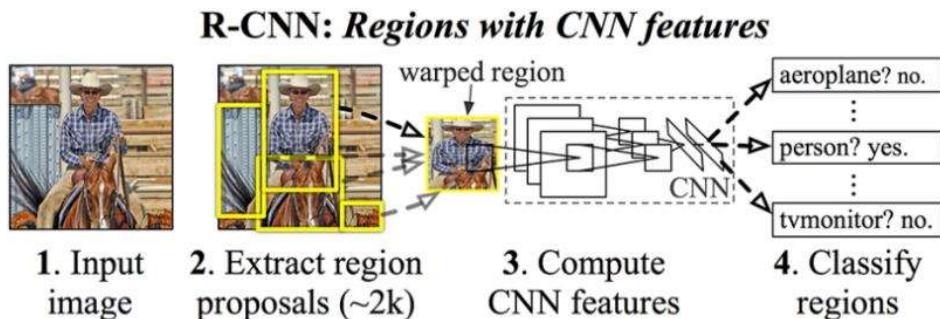
- Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernals), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1. Hình dưới đây là toàn bộ lưỡng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị.

##### 4.1.1.2 R – CNN (Region with CNN feature)

R – CNN được giới thiệu lần đầu vào 2014 bởi Ross Girshick và các cộng sự ở UC Berkely, nó là một trong những ứng dụng nền móng đầu tiên của mạng nơ-ron tích chập đối với vấn đề định vị, phát hiện và phân đoạn đối tượng [22].

- Ý tưởng thuật toán R-CNN:

- Bước 1: Dùng Selective Search algorithm để lấy ra khoảng 2000 bounding box trong input mà có khả năng chưa đối tượng.
- Bước 2: Với mỗi bounding box ta xác định xem nó là đối tượng nào (người, ô tô, xe đạp...)



**Hình 4. 1** Các bước trong R – CNN [23]

- Selective search algorithm

Input của thuật toán là ảnh màu, output là khoảng 2000 region proposal (bounding box) mà có khả năng chứa các đối tượng.

Đầu tiên ảnh được segment qua thuật toán Graph Based Image Segmentation. Tiếp theo, các vùng màu được nhóm với nhau dựa trên độ tương đồng về màu sắc, hướng gradient, kích thước... Cuối cùng các region proposal được xác định dựa trên các nhóm vùng màu.

- Phân loại region proposal

Bài toán trở thành phân loại ảnh cho các region proposal. Do thuật toán selective search cho tới 2000 region proposal nên có rất nhiều region proposal không chứa đối tượng nào. Vậy nên ta cần thêm 1 lớp background (không chứa đối tượng nào). Ví dụ như hình dưới ta có 4 region proposal, ta sẽ phân loại mỗi bounding box là người, ngựa hay background

Sau đó các region proposal được resize lại về cùng kích thước và thực hiện transfer learning với feature extractor, sau đó các extracted feature được cho vào thuật toán SVM để phân loại ảnh. Bên cạnh đó thì extracted feature cũng được dùng để dự đoán 4 offset values cho mỗi cạnh. Ví dụ như khi region proposal chứa người nhưng chỉ có phần thân và nửa mặt, nửa mặt còn lại không có trong

region proposal đó thì offset value có thể giúp mở rộng region proposal để lấy được toàn bộ người.

- Vấn đề với R-CNN

Hồi mới xuất hiện thì thuật toán hoạt động khá tốt cho với các thuật toán về computer vision trước đó nhờ vào CNN, tuy nhiên nó vẫn có khá nhiều hạn chế:

- Vì với mỗi ảnh ta cần phân loại các class cho 2000 region proposal nên thời gian train rất lâu.
- Không thể áp dụng cho real-time vì mỗi ảnh trong test set mất tới 47s để xử lý.

#### 4.1.1.3 Fast R – CNN

Fast R-CNN được giới thiệu bởi cùng tác giả của R - CNN, nó giải quyết được một số hạn chế của R - CNN để cải thiện tốc độ.

Tương tự như R - CNN thì Fast R - CNN vẫn dùng selective search để lấy ra các region proposal. Tuy nhiên là nó không tách 2000 region proposal ra khỏi ảnh và thực hiện bài toán image classification cho mỗi ảnh. Fast R-CNN cho cả bức ảnh vào ConvNet (một vài convolutional layer + max pooling layer) để tạo ra convolutional feature map. Sau đó các vùng region proposal được lấy ra tương ứng từ convolutional feature map. Tiếp đó được Flatten và thêm 2 Fully connected layer (FCs) để dự đoán lớp của region proposal và giá trị offset values của bounding box.

Tuy nhiên là kích thước của các region proposal khác nhau nên khi Flatten sẽ ra các vector có kích thước khác nhau nên không thể áp dụng neural network được. Thủ nhỉn lại xem ở trên R-CNN đã xử lý như thế nào? Nó đã resize các region proposal về cùng kích thước trước khi dùng transfer learning. Tuy nhiên ở feature map ta không thể resize được, nên ta phải có cách gì đó để chuyển các region proposal trong feature map về cùng kích thước => Region of Interest (ROI) pooling ra đời

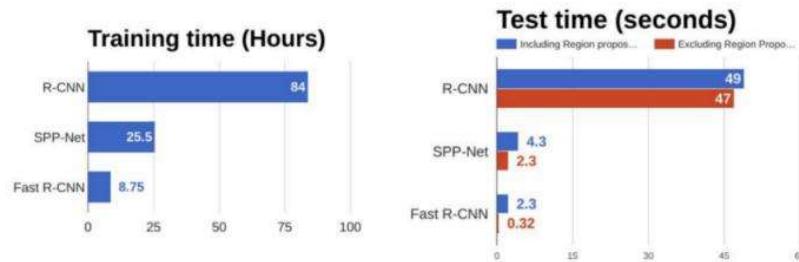
- Region of Interest (ROI) pooling

ROI pooling là một dạng của pooling layer. Điểm khác so với max pooling hay average pooling là bất kể kích thước của tensor input, ROI pooling luôn cho ra output có kích thước cố định được định nghĩa trước. Ta kí hiệu  $a/b$  là phần nguyên của  $a$  khi chia cho  $b$  và  $a\%b$  là phần dư của  $a$  khi chia cho  $b$ .

Ví dụ:  $10/3 = 3$  và  $10\%3 = 1$ . Gọi input của ROI pooling kích thước  $m*n$  và output có kích thước  $h*k$  (thông thường  $h, k$  nhỏ ví dụ  $7*7$ ).

- Ta chia chiều rộng thành  $h$  phần, ( $h-1$ ) phần có kích thước  $m/h$ , phần cuối có kích thước  $m/h + m\%h$ .
- Tương tự ta chia chiều dài thành  $k$  phần, ( $k-1$ ) phần có kích thước  $n/k$ , phần cuối có kích thước  $n/k + n\%k$ .

**Đánh giá Fast R-CNN**



**Hình 4. 2** Đánh giá mô hình Fast R - CNN

Fast R - CNN khác với R - CNN là nó thực hiện feature map với cả ảnh sau đó với lấy các region proposal ra từ feature map, còn R - CNN thực hiện tách các region proposal ra rồi mới thực hiện CNN trên từng region proposal. Do đó Fast R - CNN nhanh hơn đáng kể nhờ tối ưu việc tính toán bằng Vectorization.

Tuy nhiên nhìn hình trên ở phần test time với mục Fast R - CNN thì thời gian tính region proposal rất lâu và làm chậm thuật toán => Cần thay thế thuật toán selective search. Giờ người ta nghĩ đến việc dùng deep learning để tạo ra region proposal => Faster R - CNN ra đời

#### 4.1.1.4 Faster R – CNN

Faster R - CNN không dùng thuật toán selective search để lấy ra các region proposal, mà nó thêm một mạng CNN mới gọi là Region Proposal Network (RPN) để tìm các region proposal.

Đầu tiên cả bức ảnh được cho qua pre-trained model để lấy feature map. Sau đó feature map được dùng cho Region Proposal Network để lấy được các region proposal. Sau khi lấy được vị trí các region proposal thì thực hiện tương tự Fast R - CNN.

- *Region Proposal Network (RPN)*

Input của RPN là feature map và output là các region proposal. Ta thấy các region proposal là hình chữ nhật. Mà một hình chữ nhật được xác định bằng 2 điểm ở 2 góc, ví dụ A ( $x_{min}, y_{min}$ ) và B ( $x_{max}, y_{max}$ ). Nhận xét:

- Khi RPN dự đoán ta phải ràng buộc  $x_{min} < x_{max}$  và  $y_{min} < y_{max}$ .
- Hơn nữa các giá trị x, y khi dự đoán có thể ra ngoài khơi bức ảnh => Cần một kĩ thuật mới để biểu diễn region proposals => Anchor ra đời.

Ý tưởng là thay vì dự đoán 2 góc ta sẽ dự đoán điểm trung tâm ( $x_{center}, y_{center}$ ) và width, height của hình chữ nhật. Như vậy mỗi anchor được xác định bằng 4 tham số ( $x_{center}, y_{center}, width, height$ ).

Vì không sử dụng Selective search nên RPN ban đầu cần xác định các anchor box có thể là region proposal, sau đó qua RPN thì chỉ output những anchor box chắc chắn chứa đối tượng.

Việc của RPN là lấy ra các region proposal giống như selective search thôi chứ không phải là phân loại ảnh.

Mô hình RPN khá đơn giản, feature map được cho qua Conv layer  $3*3$ , 512 kernels. Sau đó với mỗi anchor lấy được ở trên, RPN thực hiện 2 bước:

1. Dự đoán xem anchor đây là foreground (chứa object) hay background (không chứa object)
2. Dự đoán 4 offset value cho  $x_{center}, y_{center}, width, height$  cho các anchor.

Nhận xét: có rất nhiều anchor bị chồng lên nhau nên non-maxima suppression được dùng để loại bỏ các anchor chồng lên nhau. Sau cùng dựa vào phần trăm dự đoán background RPN sẽ lấy N anchor (N có thể 2000, 1000, thậm chí 100 vẫn chạy tốt) để làm region proposal.

- Intersection over Union (IoU)

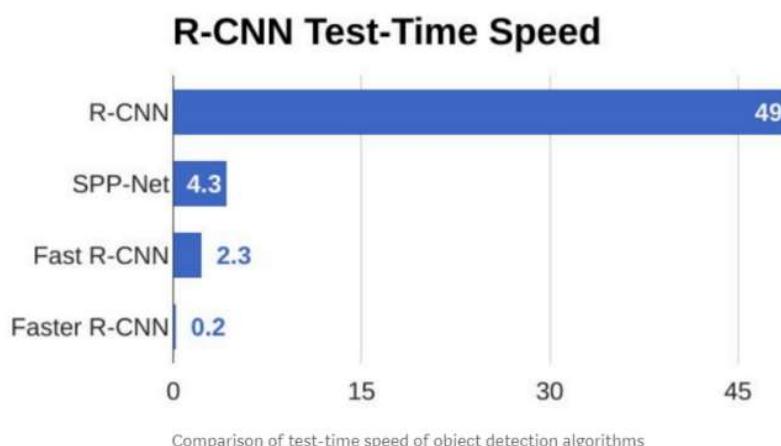
IoU được sử dụng trong bài toán object detection, để đánh giá xem bounding box dự đoán đối tượng khớp với ground truth thật của đối tượng

- Non-maxima suppression

Ở trong Region Proposal Network đầu tiên ta có khoảng 9000 anchor box (tập Input) tuy nhiên ta chỉ muốn giữ lại 100 anchor (tập Ouput) làm region proposal.

Ta sẽ làm như sau:

- Chọn ra anchor box (A) có xác suất là foreground lớn nhất trong tập Input.
- Thêm A vào tập Ouput.
- Loại bỏ A và các anchor box trong tập Input mà có hệ số IoU với A lớn hơn 0.5 ra khỏi tập Input.
- Kiểm tra nếu tập Input rỗng hoặc tập Output đủ 100 anchor thì dừng lại, nếu không quay lại bước 1.
- Kết quả của Faster R – CNN.



**Hình 4.3** Đánh giá tốc độ Faster R – CNN và các mô hình khác

#### 4.1.2 Mask R – CNN

Mask R - CNN về cơ bản là phần mở rộng của Faster R - CNN. Faster R - CNN được sử dụng rất nhiều trong các bài toán phát hiện đối tượng. Khi chúng ta cho một hình ảnh vào Faster R - CNN sẽ trả ra nhãn và bounding box của từng đối tượng cụ thể trong một hình ảnh.

- Backbone Model

Tương tự như ConvNet trong Faster R - CNN thì trong Mask R - CNN được tác giả sử dụng kiến trúc Resnet101 để trích xuất thông tin từ hình ảnh đầu vào.

- Region Proposal Network (RPN)

Trong bước này model sử dụng tính năng được trích xuất áp dụng vào mạng RPN để dự đoán đối tượng có trong khi vực đó hay không. Sau bước này chúng ta sẽ thu được bounding box tại các vùng có thể có đối tượng từ mô hình dự đoán.

- Region of Interest (RoI)

Các bounding box từ các khu vực phát hiện đối tượng sẽ có những kích thước khác nhau nên qua bước này sẽ gộp toàn bộ các bounding box đó về 1 kích thước nhất định tại 1 đối tượng. Tiếp theo, các vùng này được chuyển qua một fully connected layer để dự đoán nhãn lớp và hộp giới hạn. Như mình đã nói 1 đối tượng sẽ có rất nhiều bounding box với các kích thước khác nhau sau đó nó sẽ được loại bỏ dần qua việc tính toán IOU như sau:

$$\text{IoU} = \text{Area of the intersection} / \text{Area of the union}$$

IoU hay là vùng giao nhau giữa bounding box dự đoán và bounding box thực tế chia cho bounding box thực tế. Nếu IOU lớn hơn hoặc bằng 0,5 thì sẽ quan tâm còn nhỏ hơn thì sẽ loại bỏ.

- Segmentation Mask

Chúng ta có ROI dựa trên những giá trị IOU qua việc tính toán nên tác giả đã thêm một nhánh mask vào trong kiến trúc hiện tại. Ở đây, mô hình đã phân đoạn

tất cả các đối tượng trong hình ảnh. Đây là bước cuối cùng trong Mask R - CNN nơi chúng tôi dự đoán mask cho tất cả các đối tượng trong ảnh

#### 4.1.3 Các bước triển khai mô hình Mask R – CNN

Để phát triển một mô hình Mask R – CNN hay bất cứ một mô hình ML nào, đều cần chuẩn bị các thứ sau:

- Chuẩn bị tập dữ liệu đầu vào. Ở đề tài này đó là ảnh cà chua chín ở 3 mức màu sắc khác nhau.
- Chuẩn bị, thiết lập môi trường lập trình phù hợp với kỹ thuật sử dụng.

Với mô hình Mask R – CNN này, đề tài sẽ sử dụng bộ dữ liệu COCO. COCO (Common Objects in Context) là một tập dữ liệu phát hiện, phân đoạn và phụ đề đối tượng quy mô lớn [24]. COCO có một số tính năng:

- Phân đoạn đối tượng
- Công nhận trong ngữ cảnh
- Phân đoạn nội dung siêu điểm ảnh
- 330 K hình ảnh (>200 K được dán nhãn)
- 1,5 triệu phiên bản đối tượng
- 80 danh mục đối tượng
- 91 danh mục nội dung
- 5 chú thích cho mỗi hình ảnh
- 250.000 người có điểm mẫu chố

Có thể tìm hiểu sâu hơn về tập dữ liệu COCO tại đây [24].

##### 4.1.3.1 Chuẩn bị nguồn ảnh

Tổng cộng đã có gần 300 bức ảnh cà chua chín đã được thu thập sau 2 lần lấy mẫu. Cụ thể:

- Lần 1: Khoảng 200 bức ảnh được lấy mẫu tại thành phố Uông Bí, tỉnh Quảng Ninh vào ngày 18/02/2022. Cà chua tươi được thu hoạch thủ công, và lấy mẫu ngay sau đó. Nhiệt độ khi thu hoạch: 13 °C. Nhiệt độ khi lấy mẫu: 10 °C



**Hình 4.4** 3 hình ảnh mẫu cà chua thưa thập lần 1.

- Lần 2: Khoảng 100 bức ảnh được lấy mẫu tại thành phố Thủ Đức, thành phố Hồ Chí Minh vào ngày 5/03/2022. Cà chua được mua tại chợ Thủ Đức. Nhiệt độ khi lấy mẫu: 30 °C



**Hình 4.5** Hình ảnh một mẫu cà chua lần thu thập 2

#### 4.1.3.2 Xử lý mẫu

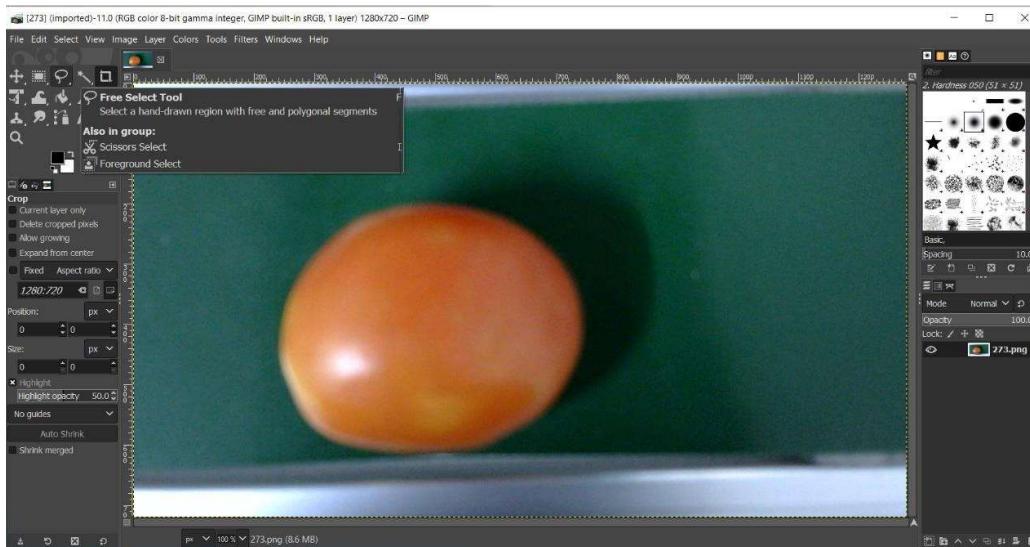
Từng ảnh cà chua sau khi thu thập được xử lý bằng phần mềm GIMP. GIMP là một phần mềm chỉnh sửa hình ảnh sẽ được sử dụng để chú thích các hình ảnh trong tập dữ liệu của đề tài.

Bằng cách sử dụng GIMP để tạo mặt nạ hình ảnh của tập dữ liệu.



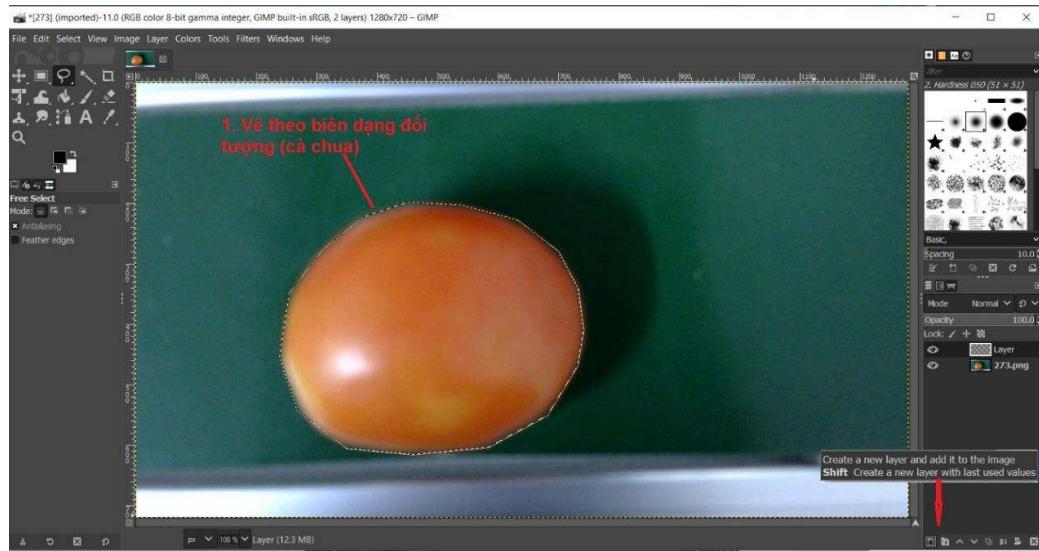
**Hình 4. 6** Giao diện chính của GIMP

- Các bước để tạo mask cho hình ảnh:
  - Sử dụng công cụ Free Select Tool, như hình 4.7



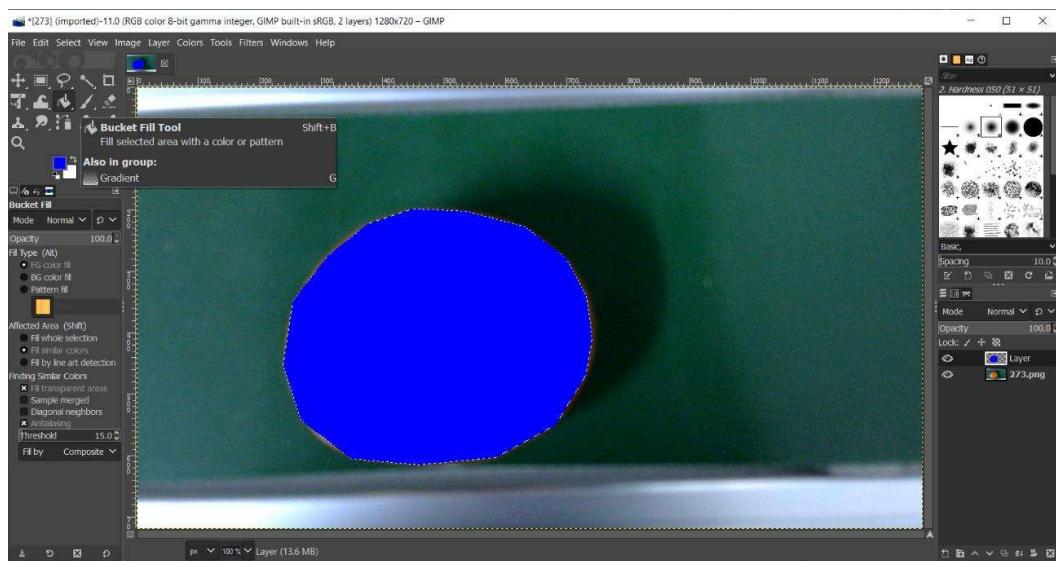
**Hình 4. 7** Chuẩn bị vẽ biên dạng.

- Vẽ theo biên dạng cà chua



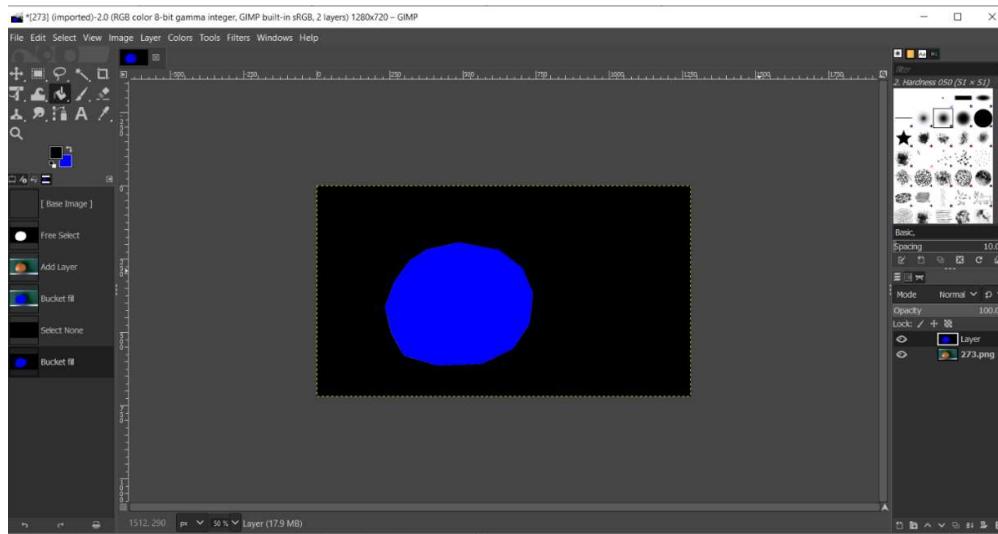
**Hình 4. 8** Cà chua sau khi vẽ biên dạng

- Tạo một layer mới bằng cách nhấn vào biểu tượng dấu mũi tên chỉ như vị trí ở hình 4.8
- Tô màu cho layer.



**Hình 4. 9** Lớp layer màu xanh trên hình dạng cà chua

- Hoàn tất bằng việc tô màu đen cho tất cả các phần còn lại của ảnh.



**Hình 4. 10** Hình ảnh sau xử lý

- Lưu lại, hoàn thành việc tạo mask cho một mẫu.

Đó là các bước để tiến hành tạo mặt nạ thủ công. Cũng có thể sử dụng các phần mềm khác để làm việc này một cách dễ dàng hơn: GIMP – ML, Make Sense...

#### 4.1.3.3 Thiết lập môi trường Anaconda

Từ thực tế quá trình thực hiện đề tài, và những tìm hiểu từ cộng đồng thì việc thiết lập một môi trường ảo cho Python là vô cùng quan trọng.

Môi trường ảo là một công cụ để duy trì không gian riêng biệt cho một dự án với các phụ thuộc và các thư viện của nó ở một nơi. Môi trường này thì riêng biệt cho một dự án cụ thể và không ảnh hưởng đến các phụ thuộc của các dự án khác.

Có nhiều cách tạo môi trường ảo Python từ môi trường sẵn của máy tính, từ Anaconda...

Các hướng dẫn tạo môi trường có thể được tìm thấy tại đây [25]

#### 4.1.3.4 Tạo một tập dữ liệu COCO

COCO lưu trữ các dữ liệu annotation trong 1 file có định dạng JSON. Với 1 file dữ liệu JSON của COCO sẽ gồm những block chính như sau:

- info: Trường dữ liệu này chứa thông tin về tập dataset.

- licenses: Trường dữ liệu này chứa thông tin về giấy phép của tập dataset
- categories: Trường dữ liệu này chứa một danh sách tên các đối tượng
- images: Trường dữ liệu này lưu trữ thông tin của các hình ảnh như đường dẫn đến ảnh, tên hình ảnh, id của ảnh (mỗi ảnh có 1 id duy nhất).
- annotations: Trường dữ liệu này chứa thông tin về các bounding boxes của các đối tượng trong 1 ảnh đối với bài toán object detection, thông tin về segmentation cho bài toán segment hoặc thông tin về caption cho bài toán image captioning.

Mã nguồn để tạo một tập dữ liệu COCO tùy chỉnh được cung cấp tại [26]

#### 4.1.3.4.1 Tạo định nghĩa mặt nạ cho tệp dữ liệu JSON

Để tạo nên tập dữ liệu COCO, cần tạo tệp JSON định nghĩa mặt nạ. Các tệp JSON được tạo thành từ các cặp khóa – giá trị ở một định dạng cụ thể.

- Có thể tạo tệp JSON bằng nhiều trình soạn thảo văn bản khác nhau: Notepad, Visual Studio Code, Sublime Text...
- Tệp mới được lưu dưới dạng mask\_definitions.json
- Tệp dữ liệu mới sẽ có cấu trúc như hình 4.11

```

1   {
2     "masks":
3     {
4       "images/1.png":
5       {
6         "mask": "masks/1.png",
7         "color_categories":
8         {
9           "(0, 255, 0)": {"category": "grade_1", "super_category": "fruit"}
10        }
11      },
12      "images/2.png":
13      {
14        "mask": "masks/2.png",
15        "color_categories":
16        {
17          "(255, 0, 0)": {"category": "grade_2", "super_category": "fruit"}
18        }
19      },
20      "images/3.png":
21      {
22        "mask": "masks/3.png",
23        "color_categories":
24        {
25          "(0, 0, 255)": {"category": "grade_3", "super_category": "fruit"}
26        }
27      },
28      "super_categories":
29      {
30        "fruit": ["grade_1", "grade_2", "grade_3"]
31      }
32    }
33  }

```

**Hình 4. 11** Cấu trúc tệp mask\_definitions.json của đề tài

#### 4.1.3.4.2 Tạo tệp JSON thông tin dữ liệu

Bước tiếp theo, cần tạo tệp chú thích COCO là tệp dataset\_info.json. Tệp này được yêu cầu để tạo chú thích tập dữ liệu COCO.

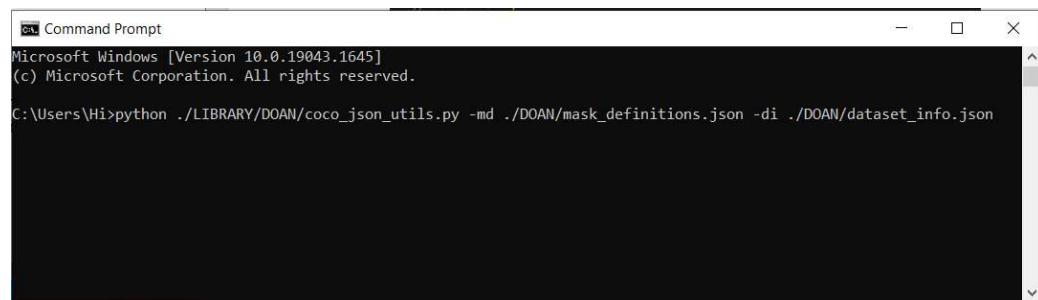
```
{ dataset_info.json > ...
1  [
2    "info": {
3      "description": "DATN dataset",
4      "version": "1",
5      "url": "no-url/datasets.com",
6      "year": 2022,
7      "contributor": "BuiCongThinh",
8      "date_created": "14/03/2022"
9    },
10   "license": {
11     "url": "no-url/licenses.com",
12     "id": 0,
13     "name": "testing"
14   }
15 ]
16 }
```

**Hình 4. 12** Tệp dataset\_info.json của đè tài.

#### 4.1.3.4.3 Tạo tệp coco\_annotations.json

Sử dụng file json\_utils.py sẽ tạo tệp coco\_annotations.json.

Sử dụng Command Prompt hoặc Anaconda Prompt điều hướng đến thư mục chứa bộ dữ liệu tùy chỉnh. Chạy tập lệnh coco\_json\_utils.py với cú pháp như hình 4.13



**Hình 4. 13** Cú pháp lệnh tạo tệp coco\_annotations.json

#### 4.1.3.5 Huấn luyện mô hình Mask R – CNN

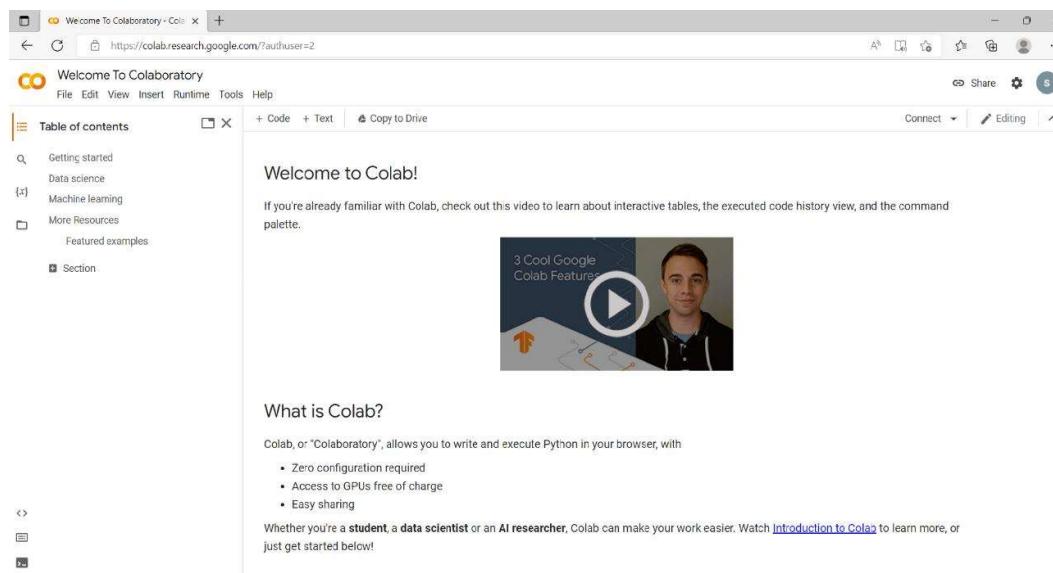
Cách triển khai Mask R – CNN trên Python 3, Keras và TensorFlow được các tác giả công bố tại [1]. Kho bao gồm:

- Mã nguồn của Mask R - CNN được xây dựng trên FPN và ResNet101.
- Mã đào tạo cho MS COCO
- Các trọng số được đào tạo trước cho máy tính xách tay MS COCO
- Jupyter để trực quan hóa đường ống phát hiện ở mỗi bước
- Lớp ParallelModel để đào tạo đa GPU
- Đánh giá về số liệu MS COCO (AP)
- Ví dụ về đào tạo trên tập dữ liệu tùy chỉnh.

Từ cơ sở mã nguồn gốc, khi thực hiện đã có một số chỉnh sửa để phù hợp tập dữ liệu và cấu hình thiết bị:

- Thiết lập đường dẫn, mẫu được huấn luyện trước
- Thiết lập lại kích thước hình ảnh huấn luyện phù hợp với tập dữ liệu.
- Thiết lập lại số lớp, kiến trúc mạng ResNet, Epoch (số lần mà training set được duyệt qua một lần và weights được cập nhật) ...

Quá trình huấn luyện sẽ phụ thuộc rất lớn vào cấu hình thiết bị. Với máy tính có cấu hình mạnh mẽ, có tích hợp GPU thời gian huấn luyện sẽ nhanh hơn. Để tài sử dụng máy tính Dell Inspiron (I5 – 1135G7, 2.42 GHz) với gần 300 bức ảnh, thời gian huấn luyện với 8 Epoch là 12 h. Đó là một hạn chế lớn của máy tính cấu hình yếu. Giải pháp cho vấn đề này là sử dụng Google Colab.



**Hình 4. 14** Giao diện chính Colab (ảnh chụp màn hình)

Colab (hay còn gọi là "Colaboratory") cho phép viết và thực thi Python trong trình duyệt với các lợi ích sau:

- Không yêu cầu cấu hình
- Sử dụng miễn phí GPU
- Chia sẻ dễ dàng

Colab có thể được sử dụng dễ dàng, ngay cả với những người bắt đầu. Hướng dẫn sử dụng Colab tại [27]. Việc huấn luyện mô hình phân loại của đế tài trở nên nhanh chóng hơn rất nhiều. Các bước chuẩn bị cho tập dữ liệu vẫn tiến hành như cũ, nhưng sẽ huấn luyện trực tuyến mô hình trên Colab thay vì máy tính. Với 20 Epoch, thời gian huấn luyện chỉ mất chưa tới 130'.

#### 4.1.4 Đánh giá mô hình Mask R – CNN

Để đánh giá hiệu quả của mô hình Mask R – CNN, 40 hình ảnh cà chua của tập dữ liệu (13 trái loại 1, 12 trái loại 2, 15 trái loại 3) được sử dụng để kiểm tra độ chính xác của mô hình. Chi tiết trong bảng 4.

Dự đoán \ Thực tế	Loại 1	Loại 2	Loại 3
Loại 1	12	1	0
Loại 2	1	10	0
Loại 3	0	1	15

Bảng 4. 1 Confusion matrix của mô hình Mask R - CNN

Confusion Matrix ma trận nhầm lẫn hay ma trận lỗi là một bộ cục bảng cụ thể cho phép hình dung hiệu suất của một thuật toán. Ma trận nhầm lẫn là một trong những kỹ thuật đo lường hiệu suất phổ biến nhất và được sử dụng rộng rãi cho các mô hình phân loại [28].

Dự đoán \ Thực tế	Loại 1	Loại 2	Loại 3
Loại 1	TP 1	FN 1.1	FN 1.2
Loại 2	FN 2.1	TP 2	FN 2.2
Loại 3	FN 3.1	FN 3.2	T 3

Bảng 4. 2 Bảng các tham số của Confusion matrix

Một số thuật ngữ cơ bản sau:

- True positive (TP): Số các ca dự đoán dương tính đúng hay dương tính thật:
  - o  $TP 1 = 12$
  - o  $TP 2 = 10$
  - o  $TP 3 = 15$
- True negative (TN): Số các ca dự đoán âm tính đúng hay âm tính thật:
  - o  $TN 1 = 10 + 0 + 11 + 15 = 36$
  - o  $TN 2 = 12 + 0 + 0 + 15 = 27$
  - o  $TN 3 = 12 + 1 + 1 + 10 = 24$
- False positive (FP): Số các ca dự đoán dương tính sai hay dương tính giả.
  - o  $FP 1 = 1 + 0 = 1$

- FP 2 = 1 + 1 = 2
- FP 3 = 0 + 0 = 0
- False negative (FN): Số các ca dự đoán âm tính sai hay âm tính giả.
  - FN 1 = 1 + 0 = 1
  - FN 2 = 1 + 0 = 1
  - FN 3 = 0 + 1 = 1

Các chỉ số đánh giá mô hình Mask R – CNN của mô hình:

- Độ chính xác – Accuracy

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}$$

- Độ nhạy - Tỷ lệ dương tính thực – Recal (TPR)

$$TPR = \frac{TP}{TP + FN}$$

- Precision Or Positive Predictive Value (PPV): Tỉ lệ dương tính đoán đúng

$$PPV = \frac{TP}{TP + FP}$$

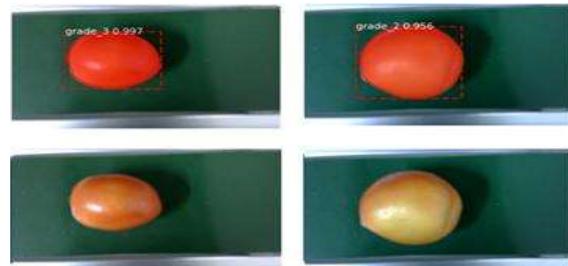
- F1 score - Điểm F1: Điểm F1 là một trung bình hài hòa Precision và Recall.

$$F1 = \frac{2 * PPV * TPR}{PPV + TPR}$$

Kết quả thực nghiệm với quả loại 1, accuracy, precision, recall và F1 score lần lượt là 95%, 92.3%, 92.3%, 92.3%. Với quả loại 2, accuracy, precision, recall và F1 score lần lượt là 92.5%, 90%, 83.3%, 86.5%. Với quả loại 3, accuracy, precision, recall và F1 score lần lượt là 97.5%, 93.7%, 100%, 96.7%

Từ kết quả thực nghiệm, có một tỉ lệ nhầm lẫn giữa loại 1 với loại 2 và loại 2 với loại 3. Nguyên nhân xuất phát từ số lượng mẫu sử dụng trong huấn luyện còn ít và chưa đa dạng. Màu sắc của loại 2 có sự phân bố giữa loại 1 và loại 3 có thể dẫn đến sự nhầm lẫn này. Sự nhầm lẫn này có thể được cải thiện qua việc bổ sung thêm dữ liệu hình ảnh về các trạng thái chín của trái cà chua và chia nhỏ độ chín thành nhiều giai đoạn hơn.

Nhìn chung, mô hình Mask R-CNN có thể đạt kết quả tốt về phân loại quả và có thể áp dụng cho một hệ thống phân loại tự động, như kết quả hình 4.15

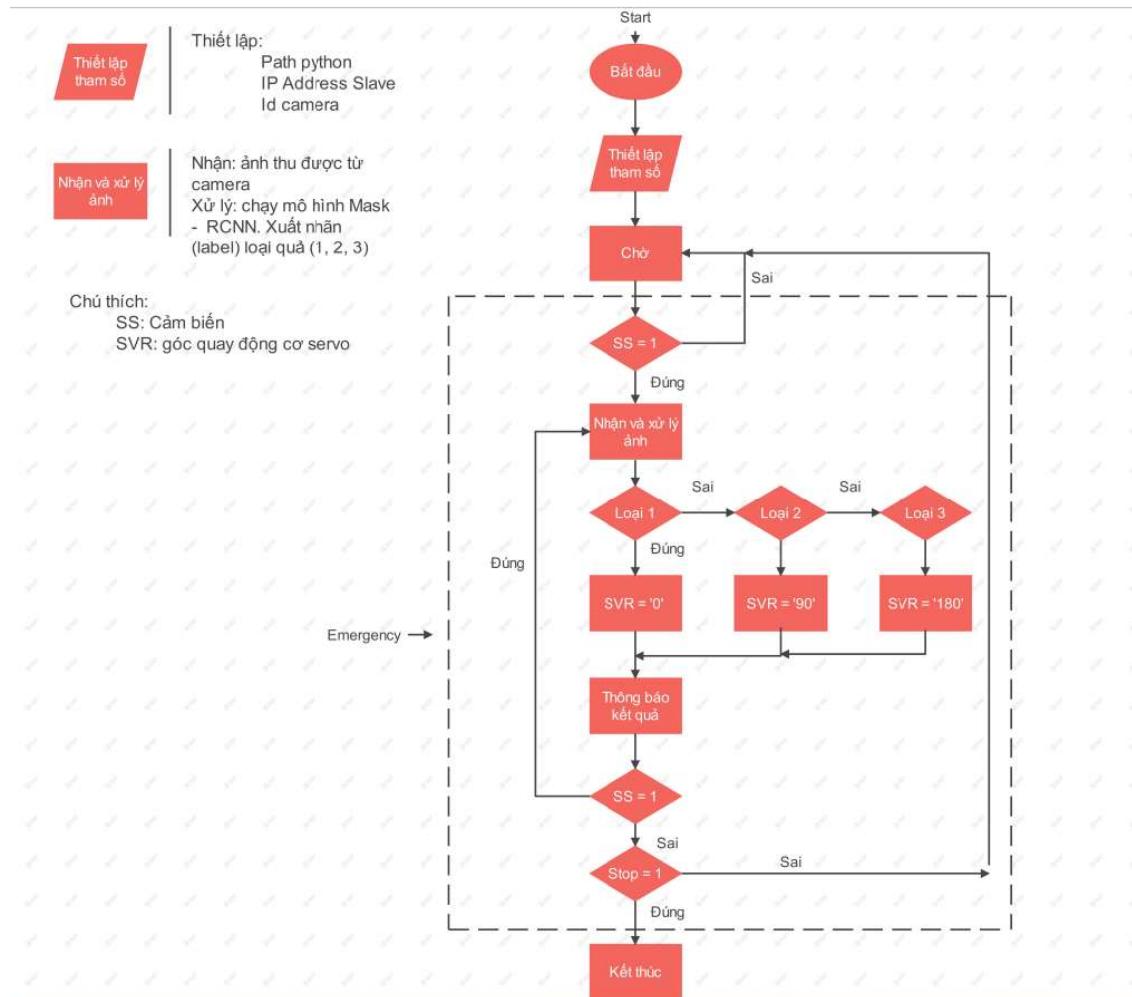


**Hình 4. 15** Kết quả kiểm tra mô hình

## 4.2 Thiết kế chương trình điều khiển và giao diện người dùng

### 4.2.1 Lưu đồ giải thuật của chương trình.

#### 4.2.1.1 Lưu đồ giải thuật



**Hình 4. 16** Lưu đồ giải thuật chính

- Giải thích lưu đồ giải thuật chính:
  - **Bắt đầu** chạy chương trình
  - **Thiết lập tham số:** chọn đường dẫn tới vị trí chứa file Python, phiên bản Python. Lựa chọn kết nối camera. Nhập địa chỉ IP PLC, Arduino sẽ kết nối.
  - **Chờ:** Lúc này các kết nối trong mạng Modbus TCP/IP đã kết nối thành công. Chờ tín hiệu gửi về từ server hoặc điều khiển từ người dùng.

- SS = 1: Cảm biến phát hiện cà chua được đưa tới vị trí buồng tối. PLC đọc và gửi tín hiệu về LabVIEW. Nếu đúng thì chương trình hoạt động quá trình tiếp theo. Nếu sai, chương trình tiếp tục chờ.
- Nhận và xử lý ảnh: Chương trình LabVIEW nhận ảnh thu về từ camera. Phân tích thang màu RGB và gửi tới mô hình Mask R - CNN chạy trong LabVIEW. Mô hình Mask R – CNN chạy, và đưa ra kết quả là nhãn (label).
- Dựa vào kết quả nhãn là loại nào, chương trình chính gửi giữ liệu điều khiển vị trí góc quay servo, thông báo kết quả nên giao diện người dùng.
- SS = 1, Đúng - chương trình lặp lại. Sai – chương trình chuyển xuống điều kiện Stop.
- Stop = 1, Đúng – chương trình kết thúc. Sai – chương trình về trạng thái chờ.

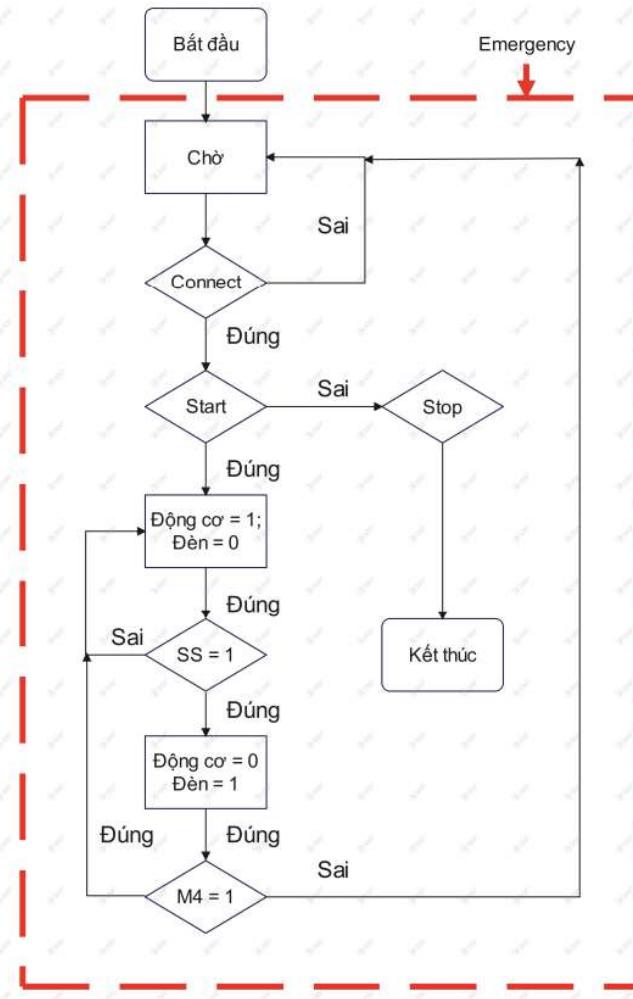
Chú thích:

Connect: Kết nối giữa PLC với LabVIEW qua mạng Modbus được thiết lập

Động cơ = 1: động cơ hoạt động;  
= 0: động cơ tắt

Đèn = 1: đèn hoạt động; = 0: đèn tắt

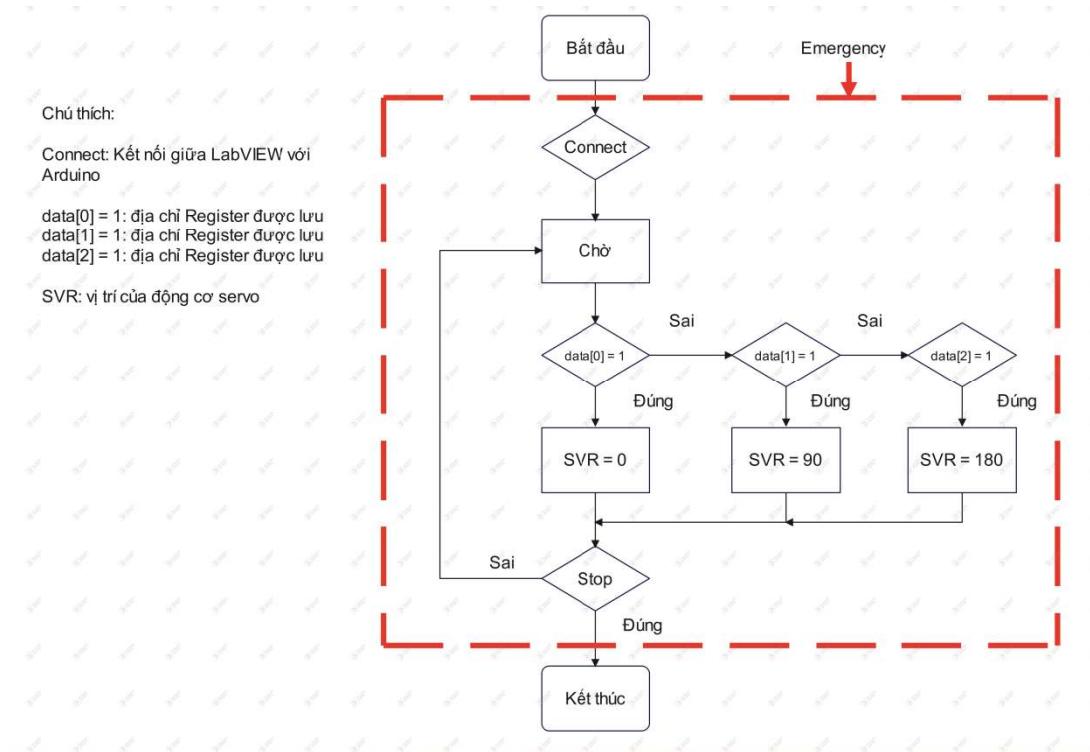
Start: Nút nhấn Start  
Stop: Nút nhấn Stop



Hình 4.17 Lưu đồ giải thuật chương trình PLC

- Giải thích lưu đồ giải thuật PLC:
  - Bắt đầu chương trình
  - Chờ: PLC khởi động, chờ kết nối từ LabVIEW hoặc các lệnh điều khiển từ người vận hành.
  - Connect: Kết nối mạng truyền thông Modbus TCP/IP. Đúng – chuyển trạng thái sau. Sai – ở trạng thái chờ.
  - Start: Nút nhấn. Đúng – khởi động băng tải vận chuyển cà chua.

- Động cơ hoạt động, vận chuyển cà chua tới vị trí buồng tối. Động cơ dừng lại, PLC đọc và gửi tín hiệu về LabVIEW.
- M4 = 1: M4 là biến tạm lưu nhận tín hiệu gửi từ chương trình Labview. Biến này được gửi khi chương trình LabView đã xác định thành công loại cà chua. Nếu đúng, động cơ hoạt động vận chuyển quả đã nhận biết tới vị trí phân loại, quá trình hoạt động lặp lại. Nếu sai, về trạng thái chờ.
- Stop: Đúng, kết thúc chương trình.



**Hình 4. 18** Lưu đồ giải thuật chương trình Arduino

- Giải thích lưu đồ giải thuật PLC:
  - Bắt đầu chương trình
  - Connect: Arduino khởi động, chờ kết nối từ kết nối mạng truyền thông Modbus TCP/IP.
  - Chờ: Arduino chờ nhận dữ liệu từ LabVIEW qua mạng truyền thông

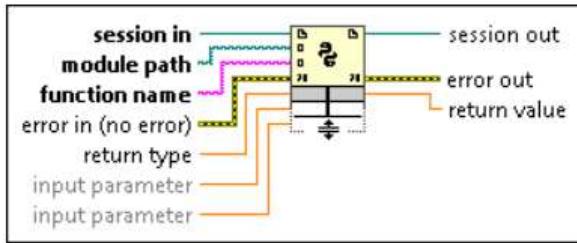
- Để tài sử dụng 3 địa chỉ Register. Với mỗi kết quả phân loại từ mô hình Mask R – CNN trên LabVIEW, một bit sẽ được ghi vào địa chỉ tương ứng với kết quả phân loại.
  - Dựa theo địa chỉ bit được ghi, Arduino điều khiển vị trí góc quay động cơ servo tương ứng.
  - Stop: Nếu nhận được yêu cầu dừng, chương trình kết thúc. Sai – quay lại trạng thái chờ lượt phân loại tiếp theo.
- 4.2.1.2 Phân tích lưu đồ giải thuật

#### **4.2.2 Thư viện và các kỹ thuật sử dụng.**

- Để mô hình Mask R – CNN có thể chạy trong môi trường LabVIEW cần sử dụng hàm Python Node.
- Thư viện NI-IMAQdx (LabVIEW) hỗ trợ thiết lập các thông số và thu thập hình ảnh của máy ảnh.
- Để kết nối PLC, Arduino với LabVIEW, những thư viện, công cụ hỗ trợ cần được sử dụng:
  - Thư viện LabVIEW Modbus API
  - Thư viện MgsModbus hỗ trợ Arduino
  - MB\_SLAVE với PLC

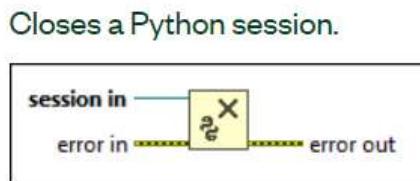
##### **4.2.2.1 Thư viện sử dụng cho LabVIEW**

- Python Node
  - Một lưu ý quan trọng trước khi sử dụng các hàm Python của LabVIEW là việc phiên bản LabVIEW 2020 trở về trước hỗ trợ phiên bản Python 3.6. Nếu sử dụng các phiên bản Python 3.7, 3.8, 3.9 cần sử dụng phiên bản LabVIEW 2021 trở lên để được hỗ trợ [29].



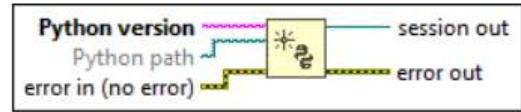
**Hình 4. 19** Hàm Node Python (ảnh chụp trong phần mềm LabVIEW)

- Đầu vào, ra của một Node Python:
  - Session in: nhập phiên bản Python.
  - Module path: đường dẫn tới vị trí tệp Python sẽ sử dụng.
  - Function name: hàm cụ thể trong tệp Python.
  - Return type: chỉ định kiểu dữ liệu của giá trị trả về.
  - Input parameter: tham số đầu vào của hàm Python. Có thể thay đổi để tăng thêm số lượng đầu vào của hàm Python.
  - Error in: mô tả các điều kiện lỗi xảy ra trước khi chạy Node Python.
  - Session out: trả về một tham chiếu đến phiên Python
  - Return value: giá trị trả về của hàm Python.
  - Error out: chứa thông tin lỗi (nếu có).
- Hàm Closes Python Session: Đóng một phiên Python. Đầu vào (session in) chỉ định một tham chiếu đến phiên Python.



**Hình 4. 20** Hàm Close Python Session (ảnh chụp trong phần mềm LabVIEW)

- Hàm Open Python Session: Mở một phiên Python với một phiên bản Python cụ thể. Đầu ra (session out) trả về một tham chiếu đến phiên Python



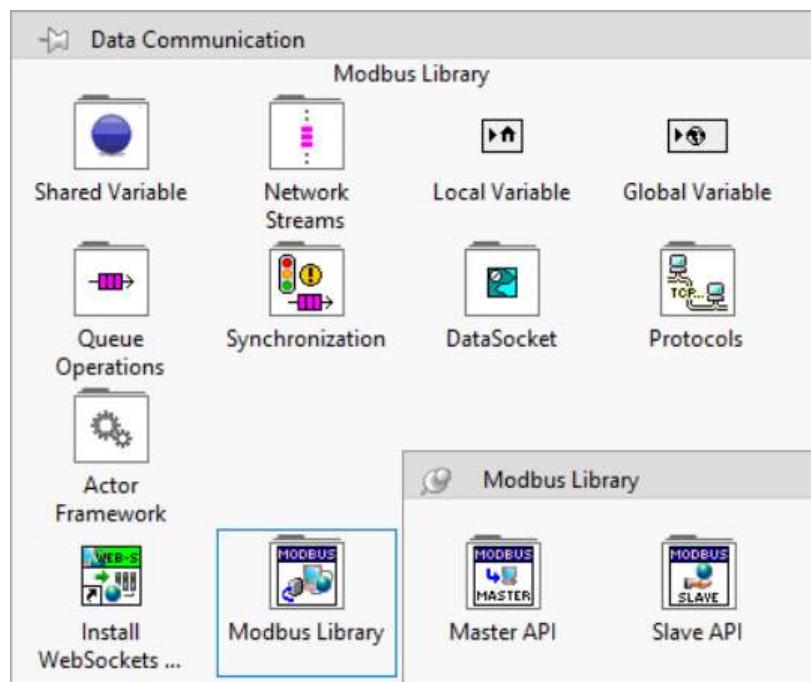
**Hình 4. 21** Hàm Open Python Session (ảnh chụp trong phần mềm LabVIEW)

- Thư viện Modbus API



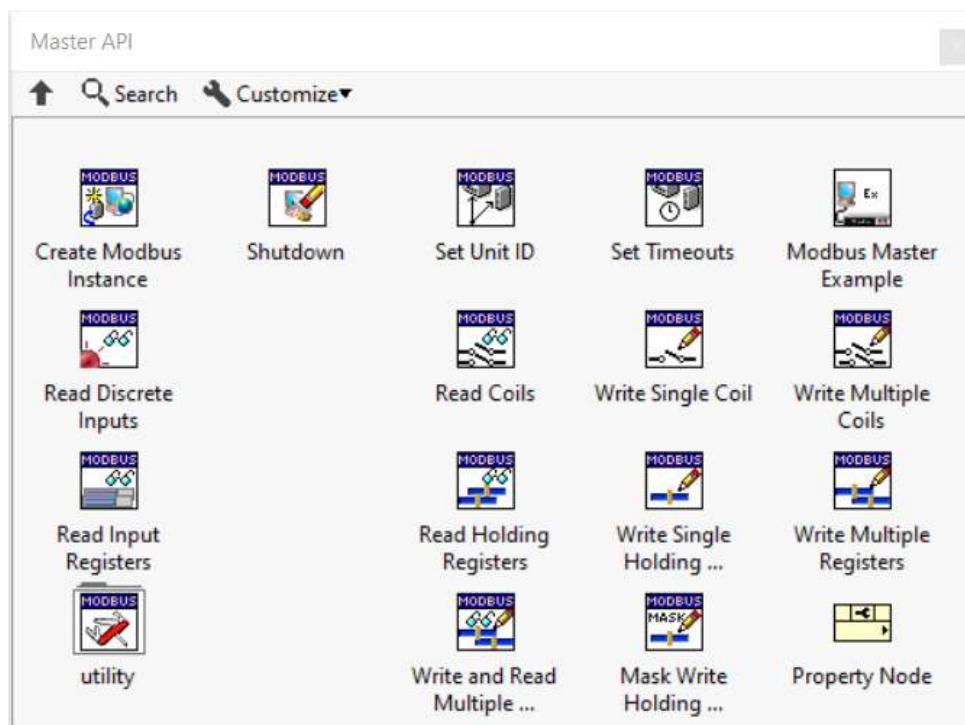
**Hình 4. 22** Biểu tượng thư viện Modbus API trong LabVIEW

- Thư viện này không có sẵn trên LabVIEW mà được phân phối dưới dạng gói VI. Cần sử VI Package Manager JKI để có thể cài đặt.
- Sau khi cài đặt thành công, Modbus API sẽ có sẵn trên bảng Data Communication.



**Hình 4. 23** Vị trí thư viện Modbus trong LabVIEW

- Thư viện Modbus API bao gồm giao diện cấp thấp và cung cấp quyền kiểm soát đối với Modbus không được cung cấp bởi các máy chủ I / O với chi phí quản lý kết nối và tập lệnh yêu cầu do API cấp cao hơn cung cấp [30].
- Tất cả các hàm trong thư viện này được bao bọc bởi một lớp API bao gồm việc triển khai chức năng master và slave. API này là những gì được hiển thị trên bảng màu. Bản chất hướng đối tượng của mã giúp bạn dễ dàng sửa đổi hành vi API hoặc thậm chí là hành vi chức năng cốt lõi khi cần thiết [30].
- Với yêu cầu của đè tài, chức năng Master API của thư viện được sử dụng.



**Hình 4. 24** Các hàm trong chức năng Master API

- Hướng dẫn về chức năng, cách sử dụng các hàm có thể xem bằng cách: Open LabVIEW → Help → Modbus Library
- Hoặc có thể tải về mã nguồn của thư viện được cung cấp tại đây [31].

#### 4.2.2.2 Thư viện sử dụng cho Arduino

- Để Arduino có thể thực hiện truyền thông Modbus, bên cạnh việc phải sử dụng shiled Ethernet còn cần một thư viện hỗ trợ chức năng truyền thông.
- Thư viện được lựa chọn là MgsModbus, có thể tải xuống miễn phí. Thư viện này được bắt đầu như một phụ bản của thư viện Modbus.h từ Dee Wykoff năm 2011 [32].
- Thư viện cung cấp cho chương trình khả năng trở thành một chủ, một phụ hoặc cả hai trên mạng TCP.
- Các chức năng Modbus 1, 2, 3, 4, 5, 6 15 và 16 được thực hiện. Cổng 502 được sử dụng làm tiêu chuẩn [32].
- Thư viện sử dụng một không gian dữ liệu duy nhất cho tất cả dữ liệu Modbus. Điều này có nghĩa là tất cả các không gian dữ liệu Modbus đều chồng lên nhau và tất cả đều bắt đầu bằng 0 (không). Điều này có nghĩa là một số chức năng Modbus có thể truy cập cùng một dữ liệu. Người dùng phải gán địa chỉ Modbus chính xác cho dữ liệu.
- Với Arduino đóng vai trò slave trong mạng truyền thông, cần chú ý các thiết lập sau:

```
#include "MgsModbus.h"

MgsModbus Mb;

// Ethernet settings (depending on MAC and Local network)
byte mac[] = {0x90, 0xA2, 0xDA, 0x0E, 0x94, 0xB5};
IPAddress ip(192, 168, 0, 2);
IPAddress gateway(192, 168, 0, 10);
 IPAddress subnet(255, 255, 255, 0);

void setup()
{
    // initialize the ethernet device
    Ethernet.begin(mac, ip, gateway, subnet);

    Mb.MbData[0] = 0; //40001
    Mb.MbData[1] = 0; //40002
    Mb.MbData[2] = 0; //40003
    svr.attach(9);
}
```

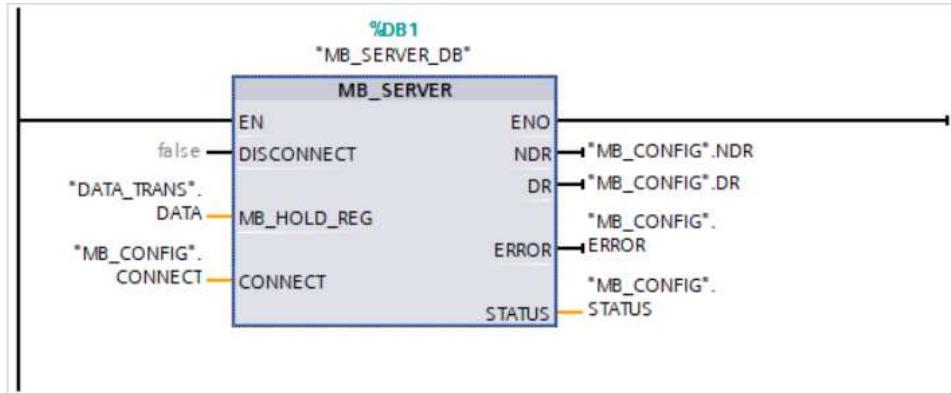
**Hình 4. 25** Cấu trúc thiết lập truyền thông thư viện MgsModbus của đề tài

Cấu trúc khởi tạo truyền thông từ thư viện MgsModbus

- Khai báo việc sử dụng thư viện: MgsModbus
- Thiết lập địa chỉ IP, Gateway cho Arduino
- Khởi tạo thiết bị ethernet
- Khai báo các địa chỉ thanh ghi sử dụng trong chương trình

#### 4.2.2.3 Khối lệnh sử dụng cho PLC

Trong mạng truyền thông này, PLC S7 – 1200 đóng vai trò Server của mạng, khối MB\_SERVER được sử dụng.



**Hình 4. 26** Khối MB\_SERVER sử dụng trong đề tài

- Có thể lấy khối trong phần mềm TIA Portal theo thứ tự sau: Instructions → Communication → Comunication processor → MODBUS → MB\_SLAVE.
- Để thuận lợi cho việc kết nối, tạo một khối DB, với tên gọi MB\_CONFIG.

MB_CONFIG			
	Name	Data type	Start value
1	Static		
2	CONNECT	TCON_IP_v4	
3	InterfaclId	HW_ANY	64
4	ID	CONN_OUC	1
5	ConnectionType	Byte	11
6	ActiveEstablished	Bool	false
7	RemoteAddress	IP_V4	
8	ADDR	Array[1..4] of Byte	
9	ADDR[1]	Byte	192
10	ADDR[2]	Byte	168
11	ADDR[3]	Byte	0
12	ADDR[4]	Byte	3
13	RemotePort	UInt	0
14	LocalPort	UInt	502
15	NDR	Bool	false
16	DR	Bool	false
17	ERROR	Bool	false
18	STATUS	Word	16#0

**Hình 4. 27** Khối MB\_CONFIG và các thiết lập của đề tài

- Những thiết lập cần lưu ý khi tạo khối DB:
  - Kết nối mạng truyền thông sử dụng theo chuẩn: TCON\_IP\_v4

- Mục InterfaceId, hardware id (HW\_ANY) là: 64. Tham số này có thể thay đổi theo loại PLC, có thể kiểm tra dễ dàng trên phần mềm TIA Portal
- Mục ConnectionType, với Start value: 11. Có định, không thể sử dụng các loại kết nối khác, là yêu cầu của PLC để hỗ trợ truyền thông TCP. Có thể tra rõ hơn trong mục CONNECT parameter
- Ở mục LocalPort với Start value: 502. Đây là giá trị mặc định, hoàn toàn có thể thay đổi. PLC S7 – 1200 hỗ trợ số cổng kết nối cục bộ trong phạm vi 1 – 49151.
- Mục RemoteAddress, mảng ADDR [1 – 4] kiểu dữ liệu Byte ứng với địa chỉ IP của Client. Ví dụ: Địa chỉ IP Client là 192.168.0.3 thì ADDR [1] = 192; ADDR [2] = 168; ADDR [3] = 0; ADDR [4] = 3.

#### 4.2.2.4 Địa chỉ kết nối

Bảng địa chỉ kết nối của PLC				
STT	Address	Input	Output	Tag name
1	I0.0	x		Signal sensor
2	I0.1	x		Nút nhấn Start
3	I0.2	x		Nút nhấn Stop
4	I0.3	x		Nút Emergency
5	Q0.0		x	Đèn trợ sáng
6	Q0.1		x	Động cơ

Bảng 4.3 Bảng địa chỉ kết nối chương trình PLC

Bảng địa chỉ kết nối của Arduino			
STT	Address Arduino	Address Shiled Ethernet	MG – 996R
1	PIN 3V3	VCC	
2	GND	GND	GND
3	PIN 10 – SS	CS	
4	PIN 11 – MISO	SI	
5	PIN 12 – SCK	SO	
6	PIN 13 – SCK	SCK	

7	PIN 5V5		VCC
8	PIN 9		Signal wire

**Bảng 4. 4** Bảng địa chỉ kết nối chương trình Arduino

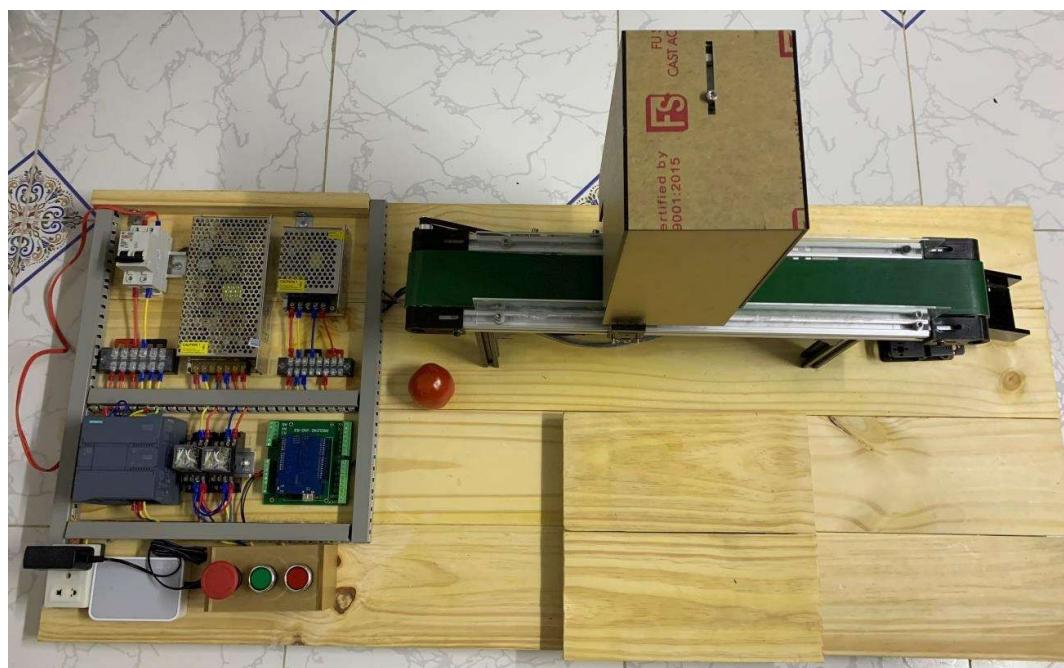
Bảng địa chỉ kết nối truyền thông Modbus TCP/IP				
STT	Tên	IP Address	Subnetmask	Default gateway
1	Laptop	192.168.0.100	255.255.255.0	192.168.0.10
2	PLC S7 – 1200 1211	192.168.0.1	255.255.255.0	
3	Arduino Uno R3	192.168.0.2	255.255.255.0	

**Bảng 4. 5** Bảng địa chỉ kết nối mạng truyền thông Modbus TCP/IP

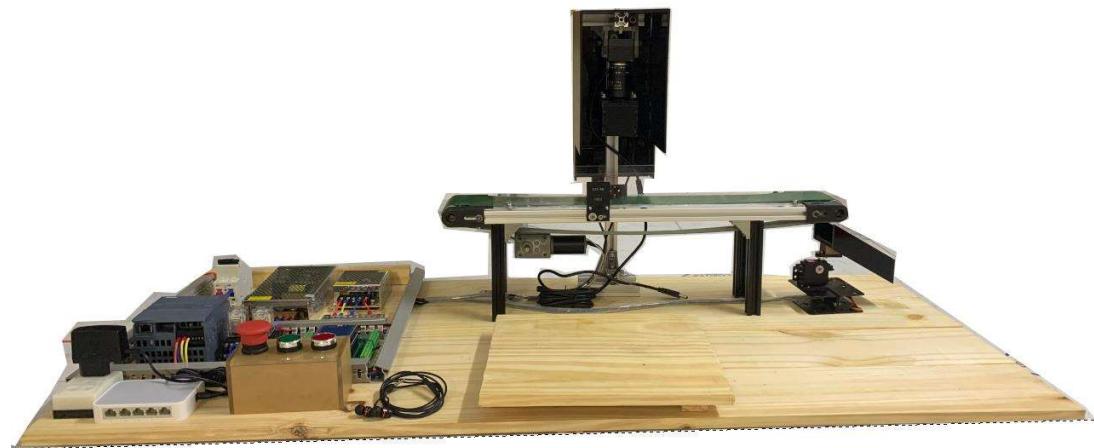
## Chương 5: THI CÔNG MÔ HÌNH

### 5.1 Hình ảnh mô hình thi công

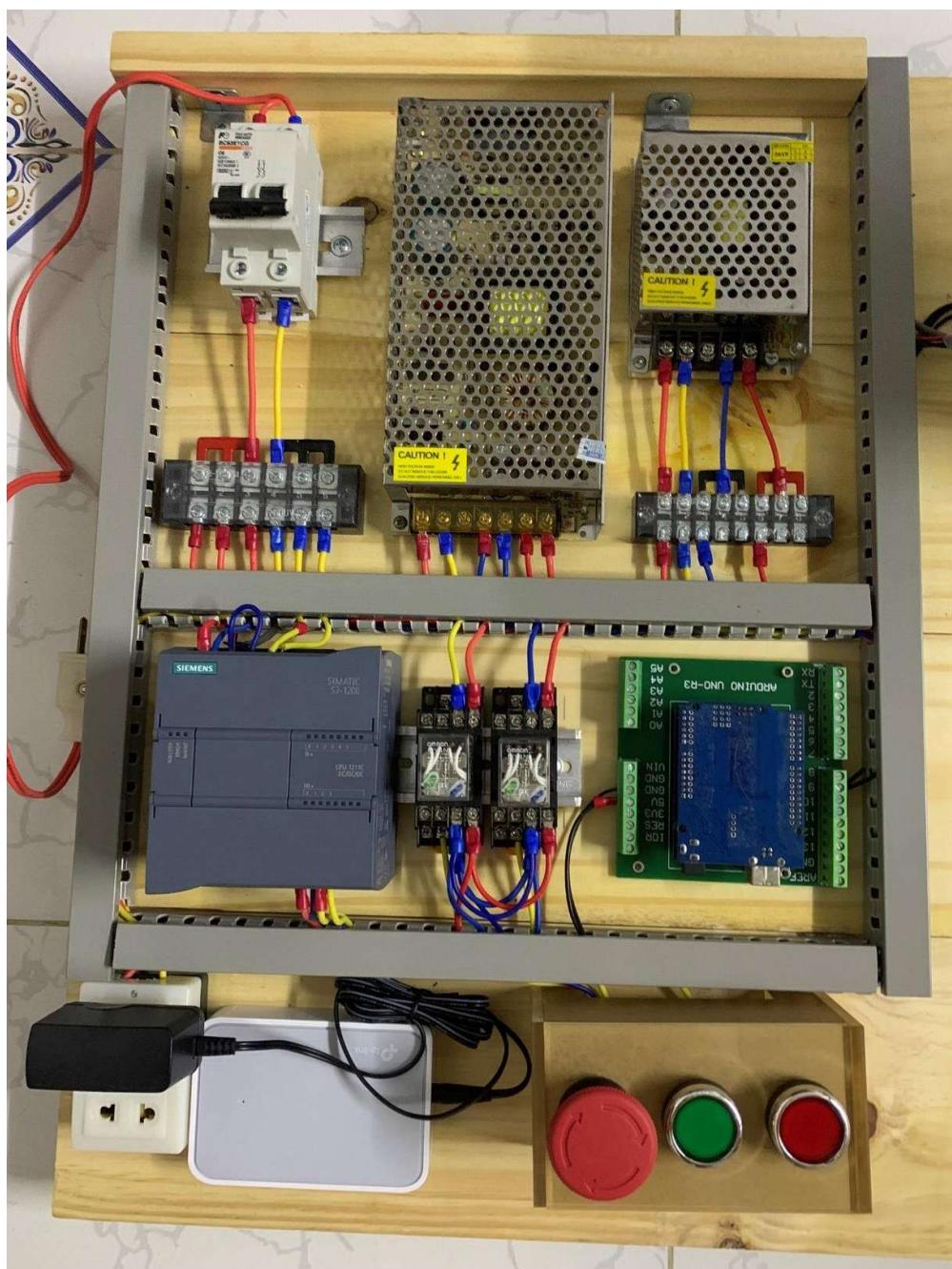
- Mô hình gồm 2 phần:
  - Phần thiết kế cơ khí: băng tải vận chuyển, camera, bàn xoay, đèn trợ sáng
  - Phần bảng điện điều khiển: MCB, PLC, Relay, Arduino & Shiled Ethernet, Switch, nút nhấn điều khiển.
- Các vật liệu được lựa chọn sử dụng:
  - Đế mô hình: gỗ thông tám, kích thước: 1200 x 140 x 13 mm, số lượng: 4.
  - Khung thân băng tải: nhôm định hình 20 x 20 mm
  - Gá gắn động cơ, máng xoay, buồng tối: nhựa Mica 5 mm
  - Dây băng tải: dây nhựa PVC
  - Vít lục giác M5 10 mm, vít lục giác M4 10 mm, vít tạo ren...
- Công cụ, thiết bị hỗ trợ việc thi công:
  - Máy cắt laze (thuê dịch vụ)
  - Máy khoan, kìm, keo dán...
- Các thiết bị điều khiển sử dụng đã trình bày ở Chương 3 của báo cáo.
- Một số hình ảnh thi công.



**Hình 5. 1** Tổng thể mô hình nhìn từ phía trên.



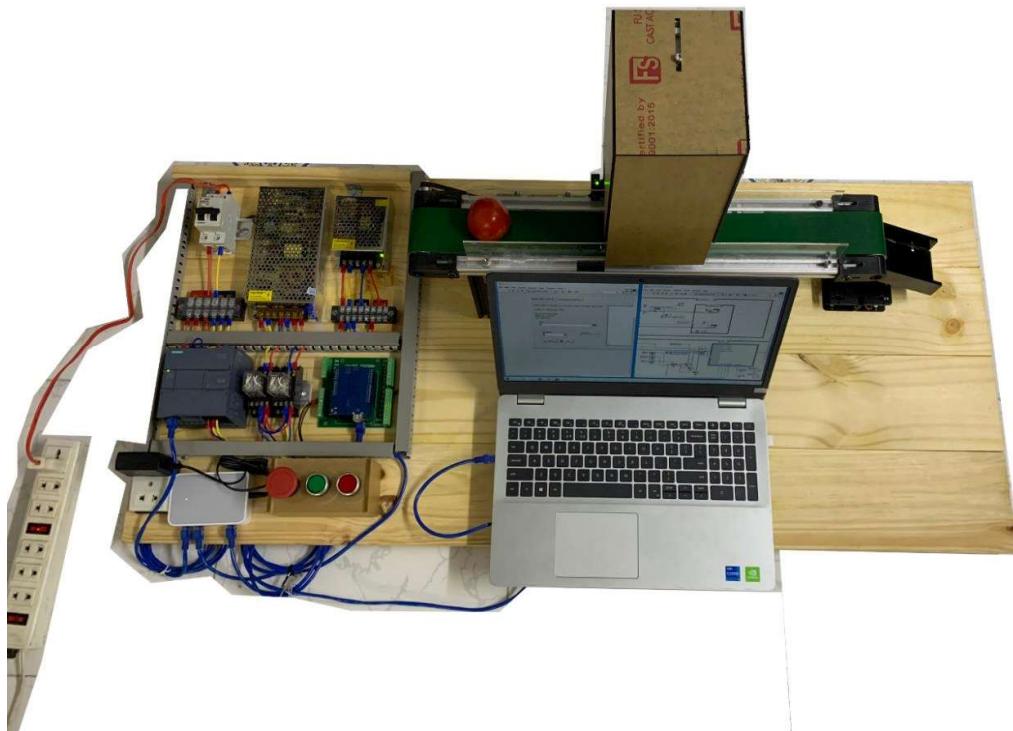
**Hình 5. 2** Tổng thể mô hình (mặt trước)



**Hình 5. 3** Hình ảnh lắp đặt thiết bị



**Hình 5. 4** Hình ảnh trong buồng tối. Gồm giá treo, camera và đèn trợ sáng.



**Hình 5. 5** Hình ảnh mô hình khi chạy thử nghiệm

## 5.2 Giao diện điều khiển trên máy tính



Hình 5. 6 Giao diện người dùng chính.



Hình 5. 7 Giao diện kết nối truyền thông Modbus TCP/IP

# **Chương 6: KẾT QUẢ, HƯỚNG PHÁT TRIỂN**

## **6.1 Kết quả đạt được**

Đề tài đã đạt được một số kết quả cụ thể với đề tài này:

- Đã thiết kế thành công mô hình phân loại cà chua hoàn chỉnh, gồm: phần mềm điều khiển; cơ cấu vận chuyển; cơ cấu phân loại.
- Xây dựng thành công mô hình Mask R – CNN. Mô hình cho thời gian ứng nhanh, độ chính xác tương đối cao.
- Kết hợp thành công mô hình Mask R – CNN vào môi trường đồ họa của LabVIEW. Sự kết hợp tốt giữa một giải pháp phân loại hình ảnh hiện đại với một môi trường lập trình giao diện đồ họa mạnh mẽ.
- Xây dựng thành công mạng truyền thông Modbus TCP/IP giữa LabVIEW với PLC và Arduino hoạt động ổn định.

## **6.2 Hạn chế**

- Tốc độ xử lý của mô hình còn chậm, còn một khoảng cách lớn với các hệ thống hoạt động thương mại.
- Giải pháp phân loại dùng máng xoay còn đơn giản, chưa được tối ưu hóa.
- Thiết kế, lựa chọn băng tải vận chuyển cà chua chưa hợp lý, cà chua còn bị lăn.

## **6.3 Hướng phát triển**

- Bổ sung thêm dữ liệu hình ảnh, chia nhỏ các thang màu giúp đa dạng thêm tập dữ liệu nhằm cải thiện độ chính xác, tính tin cậy của mô hình R – CNN.
- Nghiên cứu phát triển mô hình Mask R – CNN trực tiếp trong môi trường lập trình LabVIEW bằng việc sử dụng NI Vision Development Module của LabVIEW. Việc mô hình Mask R – CNN trực tiếp trên LabVIEW thay vì phải trung gian qua Node Python như hiện tại sẽ cải thiện đáng kể tốc độ.
- Nâng cấp, lựa chọn thiết bị phần cứng cho mô hình. Cấu hình máy tính đặc biệt quan trọng, ảnh hưởng trực tiếp tới tốc độ xử lý của mô hình. Đây là hạn

chế lớn nhất mà đê tài đã mắc phải trong quá trình xây dựng mô hình phân loại này.

- Về cơ cấu vận chuyển và phân loại, để cải thiện giải pháp cần kết hợp thêm các kỹ sư cơ khí, cơ điện tử nhằm có một giải pháp tốt, hiệu quả hơn.

## TÀI LIỆU THAM KHẢO

1. Mohammadi Baneh, N., et al., *Mechatronic components in apple sorting machines with computer vision*. 2018. **12**(2): p. 1135-1155.
2. Khan, I.A., *Foreword*, in *Sustainable Management of Arthropod Pests of Tomato*, W. Wakil, G.E. Brust, and T.M. Perring, Editors. 2017, Academic Press: San Diego. p. xv.
3. Anju-kumari, B.M., S. Guleria, and A.J.H.J. Kumari, *Influence of stage of harvest on shelf life and quality of tomato*. 1993. **6**(2): p. 89-92.
4. Inc, N.M.B., *Fruit and Vegetable Grading and Sorting*. URL: <https://www.agriculture-xprt.com/products/fruit-and-vegetable-grading-and-sorting-328999>.
5. Inc, N.M.B., "The Canadian Fruit & Vegetable Specialists". URL: <https://www.bartlett.ca/>.
6. WECO, *WECO's TomatoTek tomato sorter*. 2020.
7. Anh, T.T., *Tăng cường giải pháp phát triển sản xuất, tiêu thụ rau quả, trái cây theo hướng bền vững*. [ngày đăng: 14/05/2015], URL: [http://arit.gov.vn/tin-tuc/tang-cuong-giai-phap-phat-trien-san-xuat-tieu-thu-rau-qua-trai-cay-theo-huong-ben-vung-d8ea229c\\_3542/](http://arit.gov.vn/tin-tuc/tang-cuong-giai-phap-phat-trien-san-xuat-tieu-thu-rau-qua-trai-cay-theo-huong-ben-vung-d8ea229c_3542/). 2015.
8. Ngân, K., *Sững sót với những sáng chế cực "độc" của anh nông dân Lâm Đồng*. [ngày đăng 12/05/2016]. URL: <https://cand.com.vn/Cong-nghe/Ngac-nhien-voi-may-rua-phan-loai-danh-bong-ca-chua-cua-nong-dan-Lam-dong-i390057/>.
9. duijndam-machines, *Greefa GeoSort 6 line sorting machine*. URL: <https://www.duijndam-machines.com/en/machines/1696/greefa-geosort-6-line-sorting-machine/>.
10. Sofu, M.M., et al., *Design of an automatic apple sorting system using machine vision*. 2016. **127**: p. 395-405.
11. Wajid, A., et al. *Recognition of ripe, unripe and scaled condition of orange citrus based on decision tree classification*. in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. 2018. IEEE.
12. Tuân, N.T., *Deep Learning Cơ Bản*. 2020: p. 123 - 126.

13. Yu, Y., et al., *Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN*. 2019. **163**: p. 104846.
14. Kanade, A. and A. Shaligram. *Development of machine vision based system for classification of Guava fruits on the basis of CIE1931 chromaticity coordinates*. in *2015 2nd international symposium on physics and technology of sensors (ISPTS)*. 2015. IEEE.
15. Chauhan, A.P.S. and A.P.J.I.J.o.C.A. Singh, *Development of an intelligent virtual grader for estimation of fruit quality*. 2013. **62**(17).
16. Vietnam, S., *Giải pháp Mạng truyền thông Modbus với SIMATIC PLC S7*. 2020: p. 5.
17. Thiện, T.P.Q., *Bài giảng Mạng Truyền Thông Công Nghiệp*. Đại học Công Nghệ TP.HCM. 2021: p. 15.
18. Arduino.cc, *Getting Started with Arduino products*. URL: <https://www.arduino.cc/en/Guide>.
19. Albaloushi, N., M. Azam, and A. Amer Eissa, *Mechanical Properties of Tomato Fruits Under Storage Conditions 1*. Journal of Applied Sciences Research, 2012. **8**: p. 3053-3064.
20. Orientalmotor, *Speed Conversion*. URL: [https://www.orientalmotor.co.th/om/support\\_tool\\_sg/speed/?fbclid=IwAR12scZ9YJnLT8F\\_kEpwC3uS3wu8ZMs5rTgQ-Hg9etcqIeVt4QRXZDNKhRA](https://www.orientalmotor.co.th/om/support_tool_sg/speed/?fbclid=IwAR12scZ9YJnLT8F_kEpwC3uS3wu8ZMs5rTgQ-Hg9etcqIeVt4QRXZDNKhRA).
21. WIZnet Co., I., *W5100 Datasheet*. URL: [https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100\\_Datasheet\\_v1\\_1\\_6.pdf](https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100_Datasheet_v1_1_6.pdf). 2008.
22. Tuân, N.T., *Deep Learning Cơ Bản*. 2020: p. 194 - 196.
23. Girshick, R., et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
24. Lin, T.-Y., et al. *Microsoft coco: Common objects in context*. in *European conference on computer vision*. 2014. Springer.
25. Thắng, N.C., *Chi tiết các cài đặt và thiết lập môi trường ảo trong lập trình Python*. [ngày đăng: 15/07/2020]. URL: <https://miae.vn/2020/07/15/virutal-env-moi-truong-ao-la-gi/>.

26. Kelly, A., *COCO Synth provides tools for creating synthetic COCO datasets*. [cited 2021 29 July]; Available from: <https://github.com/akTwelve/cocosynth>. 2021.
27. Google, *Getting Started CoLab*. URL: <https://colab.research.google.com/>.
28. Ngoc, S.H., *Confusion Matrix/ Ma trận nhầm lẫn / Ma trận lỗi*. [ngày đăng: 19/08/2021]; URL: <https://viblo.asia/p/confusion-matrix-ma-tran-nham-lan-ma-tran-loi-V3m5WQB7Z07>. 2021.
29. NI Systems Engineering, *Installing Python for Calling Python Code*. URL: <https://www.ni.com/en-vn/support/documentation/supplemental/18/installing-python-for-calling-python-code.html>. 2020.
30. Smithd, T., *NI LabVIEW Modbus API*. [ngày đăng: 22/04/2022], URL: <https://forums.ni.com/t5/Reference-Design-Content/LabVIEW-Modbus-API/ta-p/3524019>. 2022.
31. NI Systems Engineering, *LabVIEW-Modbus-API*. URL: <https://github.com/NISystemsEngineering/LabVIEW-Modbus-API>. 2019.
32. Gerritse, M., *my ModbusTCP library for the Arduino*. URL: <https://myarduinoprojects.com/modbus.html>. 2013.

# PHỤ LỤC

## CHƯƠNG TRÌNH ĐIỀU KHIỂN

Chương trình Python:

```
import os
import numpy as np
#-----
# Duong dan toi mo hinh da duoc dao tao
MODEL_DIR = 'D:/File_doan/Mask_RCNN/project/mask_rcnn_tomato_0020.h5'
# Import mrcnn libraries
import mrcnn.model as modellib
from tomato import NCKH_Config
from mrcnn import visualize

#-----
# Thiet lap chay suy luan
class InferenceConfig(NCKH_Config):
    GPU_COUNT = 1
    IMAGES_PER_GPU = 1
    IMAGE_MIN_DIM = 720
    IMAGE_MAX_DIM = 1280
    DETECTION_MIN_CONFIDENCE = 0.3
    inference_config = InferenceConfig()
    # Recreate the model in inference mode
    model = modellib.MaskRCNN(mode="inference",
                               config=inference_config,
                               model_dir=MODEL_DIR)
    # Get path to saved weights
    # Duong dan toi trained weights
    model_path = os.path.join(MODEL_DIR)
```

```

model.load_weights(model_path, by_name=True)

#-----
class_names = ['grade_1','grade_2','grade_3']
def readimage(red,green,blue):
    # rois: [N, (y1, x1, y2, x2)] detection bounding boxes
    # class_ids: [N] int class IDs
    red = np.array(red)
    green = np.array(green)
    blue = np.array(blue)
    row = red.shape[0]
    col = red.shape[1]
    rgb = np.zeros((row, col, 3), dtype='uint8')
    rgb[:, :, 0] = red
    rgb[:, :, 1] = green
    rgb[:, :, 2] = blue
    results = model.detect([rgb], verbose=1)
    r = results[0]
    r1 = np.array(r['rois'][0])
    r2 = np.array(r['class_ids'][0])
    r1 = np.uint16(r1)
    r2 = np.uint16(r2)
    r1 = np.insert(r1, 4, r2)
    return r2
#-----

```

Chương trình Arduino Uno:

```

// Khai bao cac thu vien su dung
#include <SPI.h>
#include <Ethernet.h>
#include <Servo.h>
#include "MgsModbus.h"

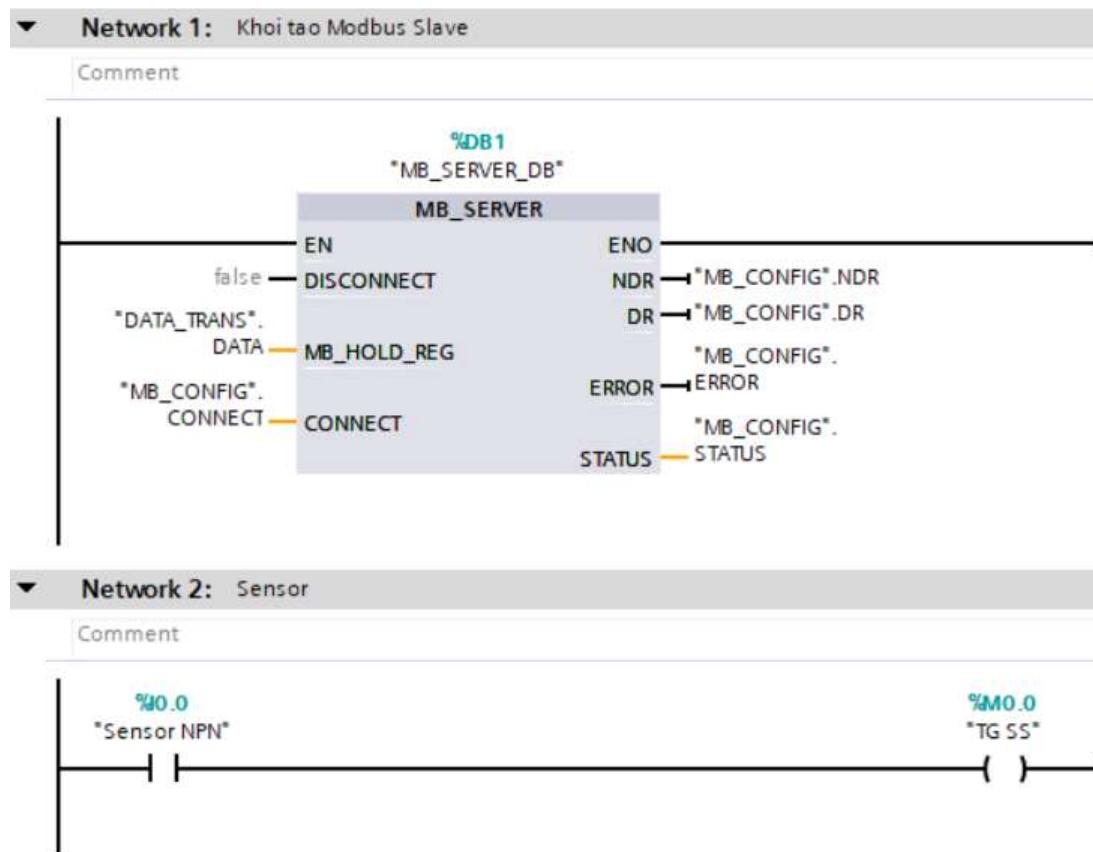
```

```

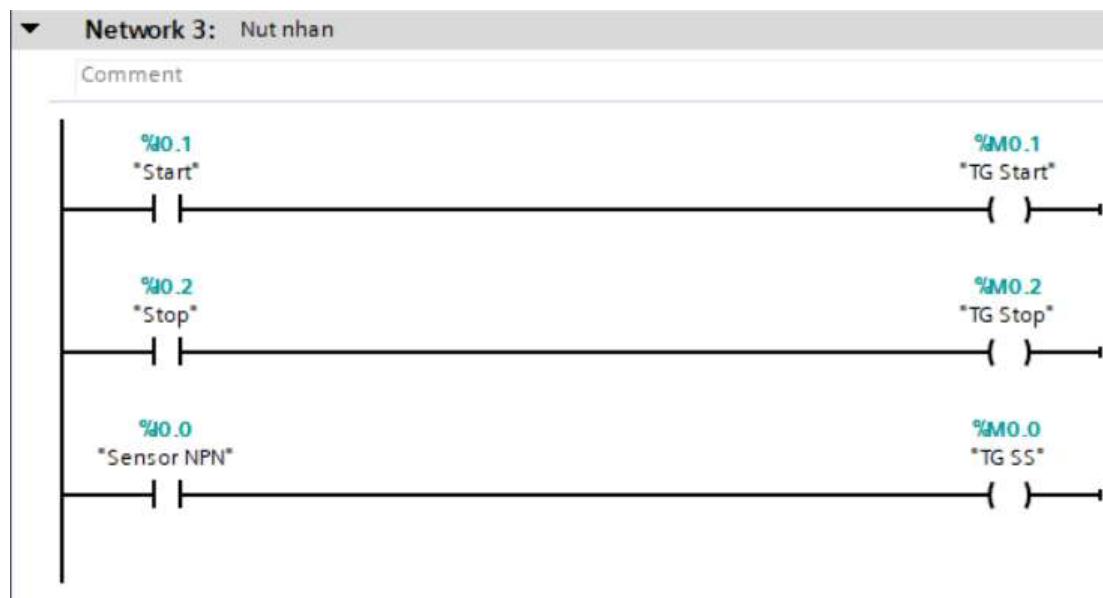
Servo svr; // khai tao bien dieu khien
int val; // khai bao kieu du lieu bien val
MgsModbus Mb;
// Cai dat cac thong so dia chi mang cua Arduino
byte mac[] = {0x90, 0xA2, 0xDA, 0x0E, 0x94, 0xB5 };
IPAddress ip(192, 168, 0, 2);
IPAddress gateway(192, 168, 0, 10);
IPAddress subnet(255, 255, 255, 0);
void setup()
{
    // khai tao thiet bi
    Ethernet.begin(mac, ip, gateway, subnet);
    // Thiet lap gia tri ban dau cua MbData
    Mb.MbData[0] = 0; //40001 // dia chi nhan bit khi qua loai 1
    Mb.MbData[1] = 0; //40002 // dia chi nhan bit khi qua loai 2
    Mb.MbData[2] = 0; //40003 // dia chi nhan bit khi qua loai 3
    svr.attach(9); // khai bao pin dieu khien servo
}
void loop()
{
    int amx0 = Mb.MbData[0];
    int amx1 = Mb.MbData[1];
    int amx2 = Mb.MbData[2];
    if (bitRead(amx0,0)){ svr.write(0);}
    else if (bitRead(amx1,0)){ svr.write(90);}
    else if (bitRead(amx2,0)){ svr.write(180);}
    delay(500);
    // Chay chuong trinh Modbus Slave tu thu vien MgsModbus
    Mb.MbsRun();
}

```

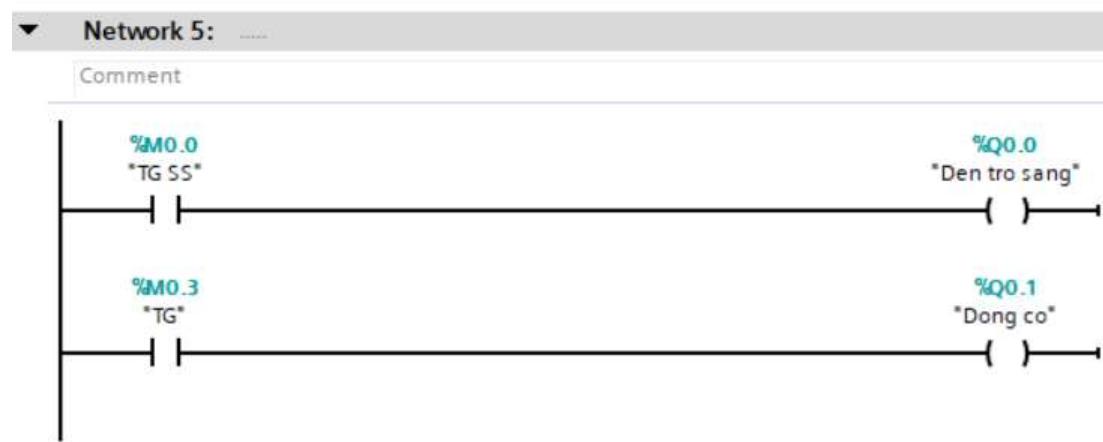
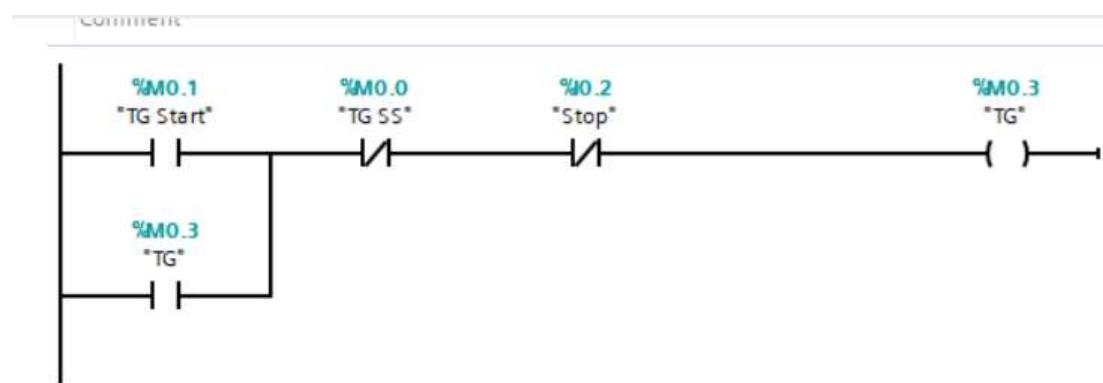
## Chương trình PLC S7 – 1200:



**Phụ Lục 1. 1** Khởi tạo MB\_Server và thiết lập biến trung gian cho đầu vào cảm biến



**Phụ Lục 1.2** Tạo các biến trung gian cho nút nhấn



**Phụ Lục 1.3** Cách dòng lệnh điều khiển

MB_CONFIG									
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	CONNECT	TCON_IP_v4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	InterfaceId	HW_ANY	64		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		HW-identifier of I-E-interface submodule
4	ID	CONN_OUC	1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		connection reference / identifier
5	ConnectionType	Byte	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		type of connection: 11=TCP/IP, 19=UDP (17=TCP
6	ActiveEstablished	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		active/pассивное подключение установления
7	RemoteAddress	IP_V4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote IP address (IPv4)
8	ADDR	Array[1..4] of Byte			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
9	ADDR[1]	Byte	192		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
10	ADDR[2]	Byte	168		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
11	ADDR[3]	Byte	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
12	ADDR[4]	Byte	3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
13	RemotePort	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote UDP/TCP port number
14	LocalPort	UInt	502		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		local UDP/TCP port number
15	NDR	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
16	DR	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
17	ERROR	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
18	STATUS	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

**Phụ Lục 1. 4 Khối MB\_CONFIG**

DATA_TRANS								
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static							
2	DATA	Array[0..10] ...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	DATA[0]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	DATA[1]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	DATA[2]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	DATA[3]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	DATA[4]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	DATA[5]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	DATA[6]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	DATA[7]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	DATA[8]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	DATA[9]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	DATA[10]	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

**Phụ Lục 1. 5 Khối DATA\_TRANS**

Ethernet addresses \_\_\_\_\_

Interface networked with \_\_\_\_\_

Subnet: PN/IE\_1  
Add new subnet

Internet protocol version 4 (IPv4)

Set IP address in the project  
 IP Address: 192 . 168 . 0 . 1  
 Subnet mask: 255 . 255 . 255 . 0

Use router  
 Router address: 0 . 0 . 0 . 0

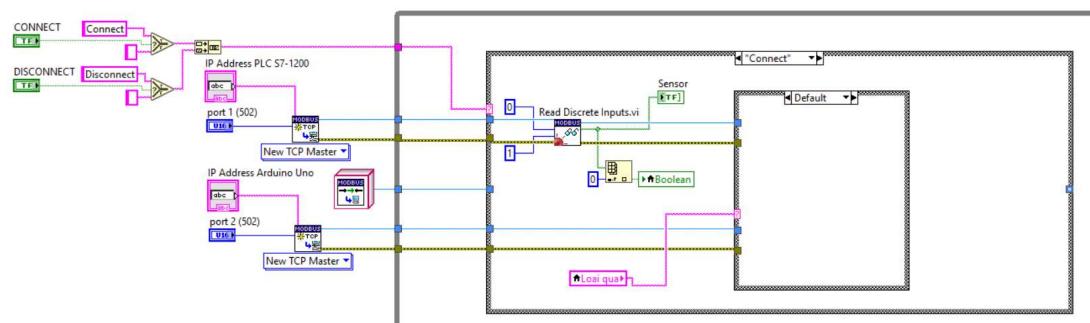
IP address is set directly at the device

**Phụ Lục 1. 6 Cài đặt địa chỉ IP và kết nối cho PLC S7 – 1200**

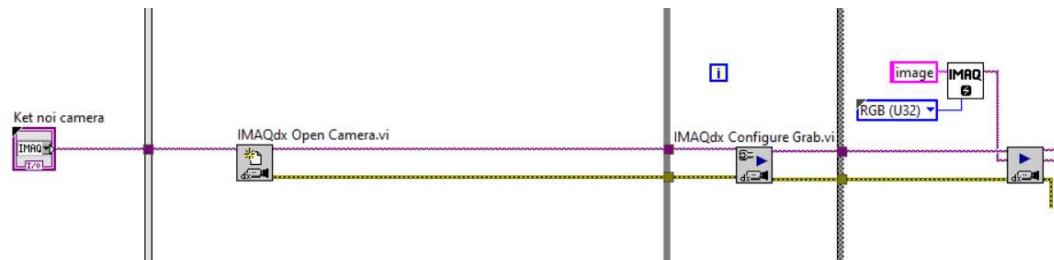
Default tag table							
	Name	Data type	Address	Retain	Access...	Write...	Visible...
1	Sensor NPN	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	Start	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Stop	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	TG SS	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	TG Start	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	TG Stop	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	TG	Bool	%M0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Den tro sang	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Dong co	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Phụ Lục 1. 7 Tag name và địa chỉ I/O của chương trình**

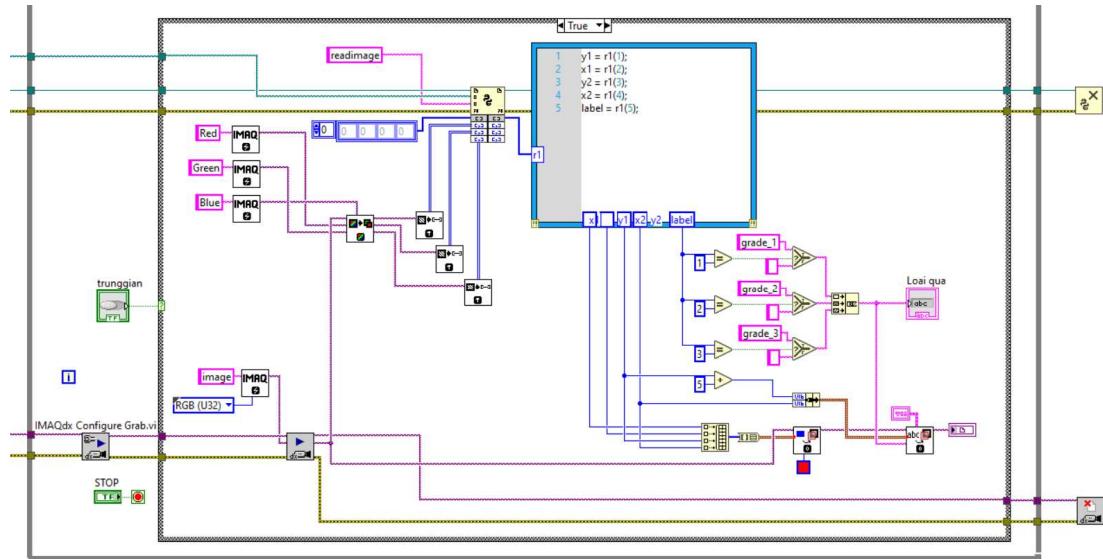
### Chương trình LabVIEW



**Phụ Lục 1. 8 Phản kết nối truyền thông Modbus**



**Phụ Lục 1. 9 Khởi tạo và thu thập hình ảnh bằng camera**



**Phụ Lục 1. 10** Chương trình chính với Node Python

Bên cạnh các chương trình chính, ở các chương trình LabVIEW, Python, Arduino còn cần sử dụng thêm các thư viện. Việc thể hiện các thư viện này trong cuốn báo cáo không thể hiện được đầy đủ cấu trúc, cũng như chức năng của từng thư viện. Vì vậy, toàn bộ chương trình, thư viện và mô hình Mask R – CNN đã đào tạo của đề tài đã được tải lên kho lưu trữ github tại địa chỉ:  
[“https://github.com/thinhbuicong/doantotnghiep”](https://github.com/thinhbuicong/doantotnghiep)

# TÀI LIỆU KỸ THUẬT MỘT SỐ THIẾT BỊ

## Động cơ giảm tốc 5840 – 31ZY

DC Worm Gear Motor 5840-31ZY

DC12V1280 Parameter:

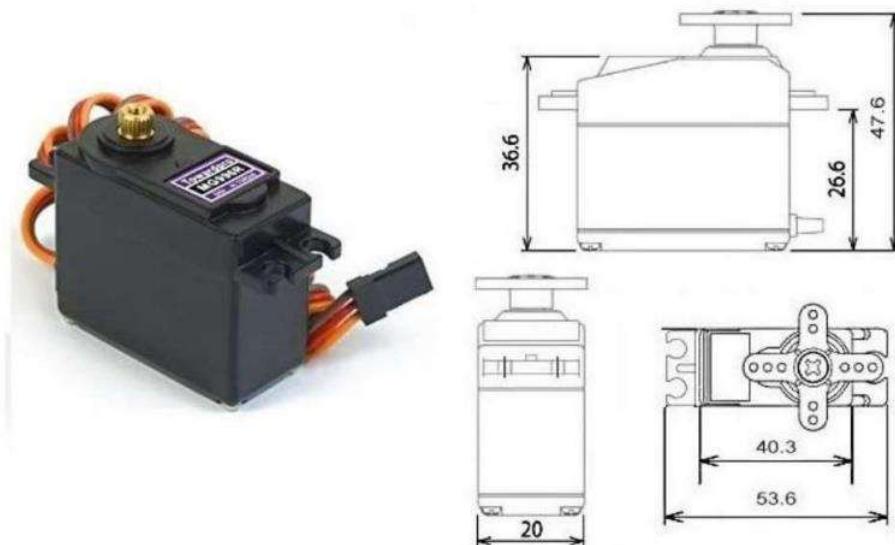
Reduction ratio	17	31	50	100	200	290	505	670
No load Current MA	≤350	≤350	≤350	≤350	≤350	≤350	≤350	≤350
No load RPM	470	260	160	80	40	27	16	12
Load torque Kg.cm	2.7	5	8.0	16.0	32.0	46.0	70.0	70.0
Load RPM	380	200	130	60	30	20	12	9
Load Current A	≤1.6	≤1.6	≤1.6	≤1.6	≤1.6	≤1.6	≤1.6	≤1.6
Max Torque Kg.cm	4.4	8	13.0	25.0	51.0	70.0	70.0	70.0
Stall Current A	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5

DC24V2480 Parameter:

Reduction ratio	17	31	50	100	200	290	505	670
No load Current MA	≤200	≤200	≤200	≤200	≤200	≤200	≤200	≤200
No load RPM	470	260	160	80	40	27	16	12
Load torque Kg.cm	2.7	5	8.0	16.0	32.0	46.0	70.0	70.0
Load RPM	380	200	130	60	30	20	12	9
Load Current A	≤1.6	≤1.6	≤1.6	≤1.6	≤1.6	≤1.6	≤1.6	≤1.6
Max Torque Kg.cm	4.4	8	13.0	25.0	51.0	70.0	70.0	70.0
Stall Current A	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4

## Động cơ servo MG - 996R

### MG996R High Torque Metal Gear Dual Ball Bearing Servo



## Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V ), 11 kgf·cm (6 V )
- Operating speed: 0.17 s/60° (4.8 V ), 0.14 s/60° (6 V )
- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA – 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5 µs
- Stable and shock proof double ball bearing design
- Temperature range: 0 °C – 55 °C

## MCB Fuji

### ● Specifications

- Rated voltage: AC230/400V, 50/60Hz
- Rated current: page 5-6
- Mechanical life: 10000 times
- Tripping characteristic: C: 5~10In  
D: 10-14In
- Breaking capacity

Tripping characteristic	Rated current (A)	Rated operational voltage (V)	Rated breaking capacity (kA)
Curve C	1~40	230/400	6
	50, 63	230/400	4.5
Curve D	1~40	230/400	4.5

### ● Wiring Capacity

Rated current (A)	Wire size (mm <sup>2</sup> )
In≤63	25

### ● Standards and Certificates

- IEC 60898-1, GB 10963.1
- CE, CCC

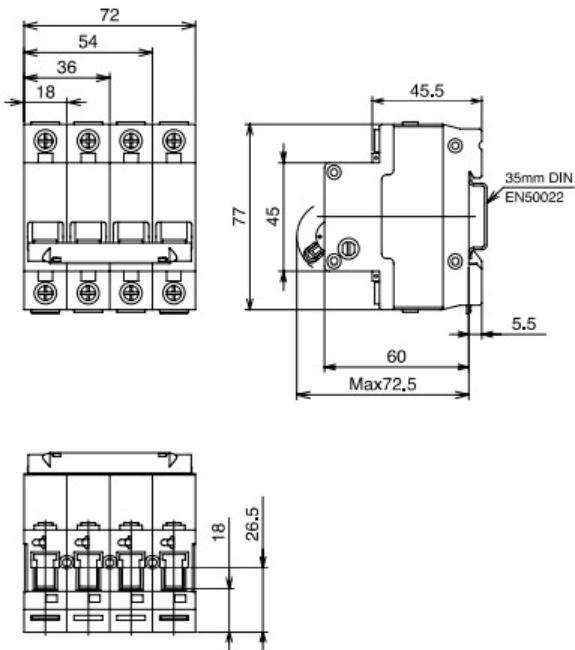
### ● Product Features

- With short circuit current limiting structure-high breaking capacity of lated short circuit.
- Short-circuit and overload protection
- Screw clamp, shock-proof wiring terminals
- Numerous accessories available
- Modularization-random combination, series mating
- Standard TH35 mm IEC rail mounting

### ● Working Condition

- Ambient temperature: -35°C to +70°C
- Altitude: ≤2000m
- Air humidity: ≤95%
- Pollution degree: II
- A place where there should not be significant shock or vibration

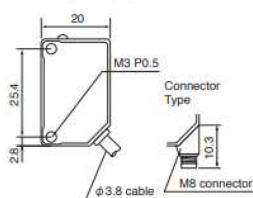
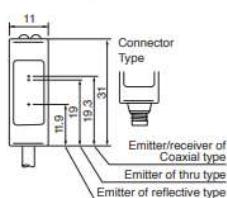
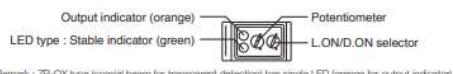
● Dimensions, mm



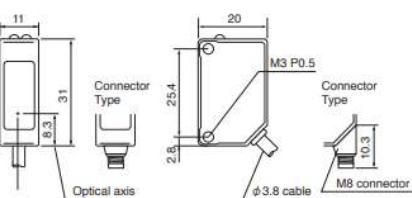
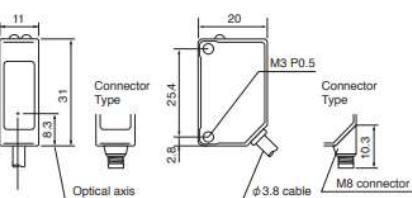
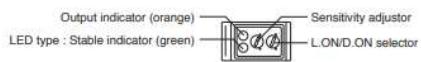
Cảm biến Optex ZR – 350N

## Dimensions

### General Use Type except BG5



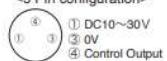
### BG5 Type



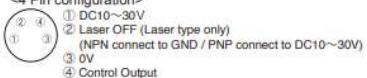
#### <Cable>

Brown : DC10~30V  
Blue : GND  
Black : Control Output  
Gray : Laser OFF (Laser type only)  
(NPN connect to GND / PNP connect to DC10~30V)

#### <3 Pin configuration>

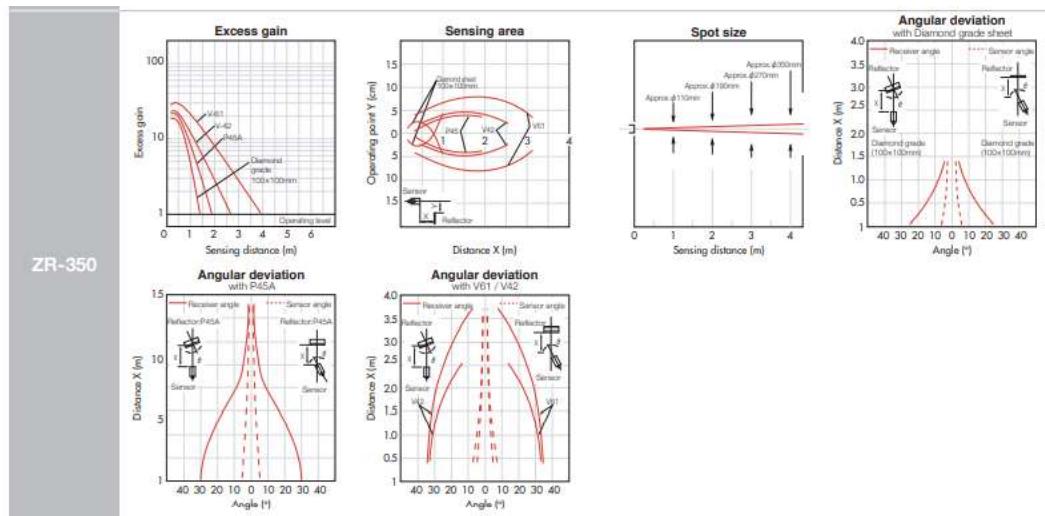


#### <4 Pin configuration>

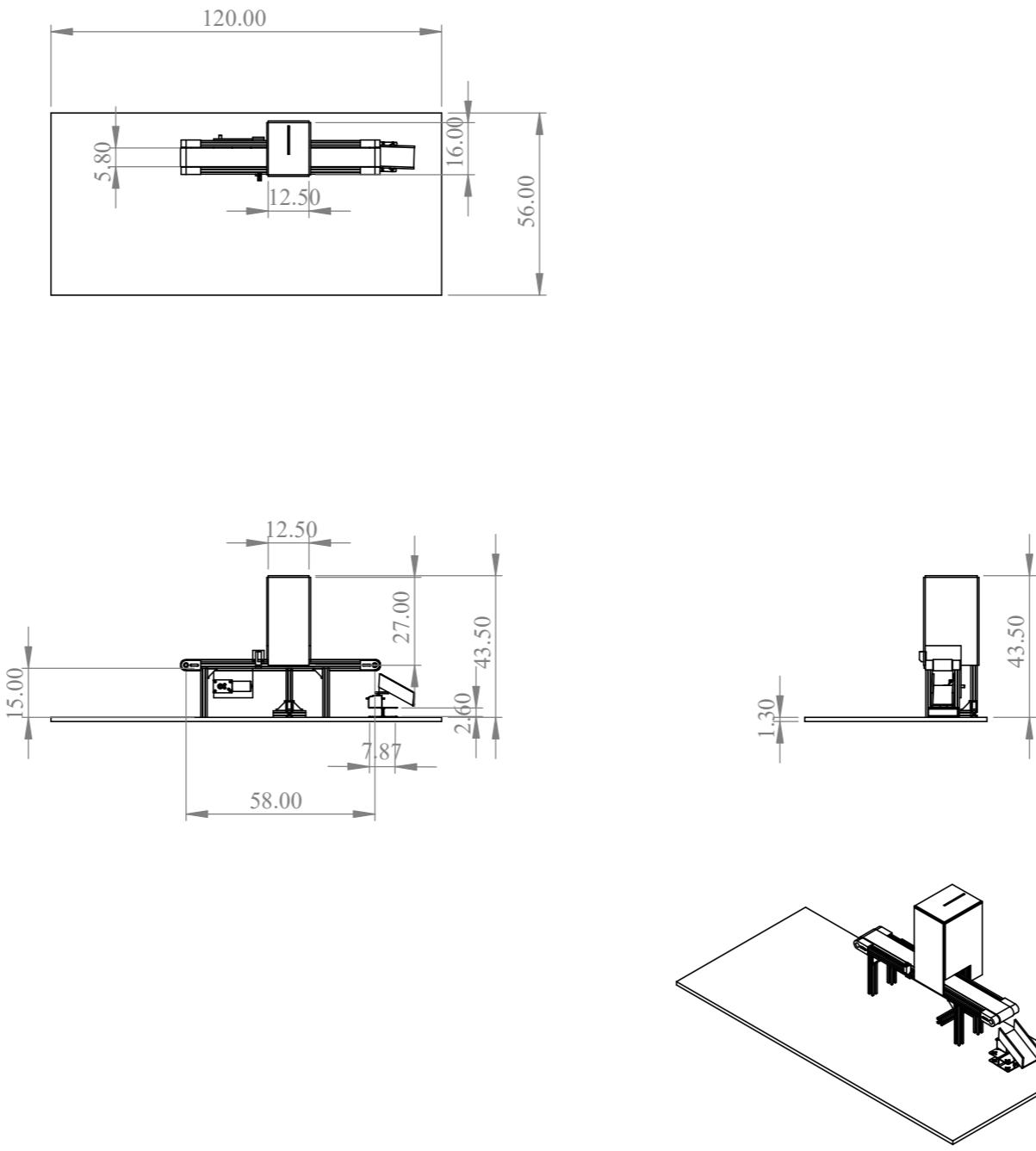


## Specifications

Model	Thru-beam	Polarized Retro-reflective	Diffuse-reflective	Convergent	Divergent	Glass detection
Cable type	NPN ZT-1200N	ZR-350N	ZD-70N	ZD-L09N	ZD-W20N	ZR-Q200N
	PNP ZT-1200P	ZR-350P	ZD-70P	ZD-L09P	ZD-W20P	ZR-Q200P
M8 Connector type	NPN ZT-1200CN	ZR-350CN	ZD-70CN	ZD-L09CN	ZD-W20CN	ZR-Q200CN
	PNP ZT-1200CP	ZR-350CP	ZD-70CP	ZD-L09CP	ZD-W20CP	ZR-Q200CP
Sensing distance	12 m	3.5m (V-61 reflector)	70cm	1-9 cm	1-200 mm	2 m (V-61 reflector)
Response time	0.5 msec					
Spot size	1.5m @ 12 meter	350mm @ 3.5 meter	100mm @ 700mm	12mm @ 90mm	30mm @ 50mm	300mm @ 2 meter
Operating mode	Light On/Dark On selectable					
Hysteresis	NIL		20% Max	10% Max	20% Max	NIL
Light source	Red LED					
Sensitivity adjustment	1-turn trimmer (240 degree)					
Control output	NPN or PNP Open collector, 100mA Max / 30V DC					
Supply voltage	DC 10 - 30V incl. 10% ripple					
Power consumption	20mA Max					
LED Indicator	Dual Indicator (Output : orange, Stable : green)					
Protection category	IP67					
Shock resistance	100G (1,000m/S <sup>2</sup> )					
Environmental illuminance	Sunlight : 10,000 lux Max., Incandescent lamp : 3,000 lux max.					
Operating temp/humidity	-25 to 55°C / 35 to 85%RH					
Materials	ABS with glass fiber (Case), PMMA (Lens)					







## ĐỒ ÁN TỐT NGHIỆP

MỘT HỆ THỐNG PHÂN LOẠI CÀ CHUA DỰA TRÊN KỸ THUẬT THỊ GIÁC MÁY TÍNH

Tr.nhiệm	Họ và tên	Ký	Ngày	Bản vẽ tổng thể hệ thống	Mã đè tài	77
Thiết kế	Bùi Công Thịnh				Tỉ lệ	1 : 5
Hướng dẫn	TS. Phạm Quốc Thiện				Đơn vị	cm
Duyệt					TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM	

