

ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA ĐIỆN

**ĐỒ ÁN TỐT NGHIỆP**  
**CAPSTONE PROJECT**

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

**ĐỀ TÀI:**

**NGHIÊN CỨU VÀ PHÁT TRIỂN HỆ THỐNG THỊ  
GIÁC MÁY TÍNH TÍCH HỢP ROBOT ĐỂ LOẠI  
BỎ TẠP CHẤT TRONG SẢN XUẤT SỢI**

Người hướng dẫn: **TS. NGUYỄN THỊ KIM TRÚC**  
**KS. NGUYỄN XUÂN BÌNH**

Sinh viên thực hiện:

- 1. NGUYỄN VĂN MẾN – MSSV: 105200337 – LỚP: 20TDH2**
- 2. VĂN NGỌC PHƯƠNG – MSSV: 105200340 – LỚP: 20TDH2**
- 3. LÊ VĂN TUẤN – MSSV: 105200349 – LỚP: 20TDH2**

**Đà Nẵng, 6/2025**

## TÓM TẮT

### Tên đề tài: **Nghiên Cứu Và Phát Triển Hệ Thống Thị Giác Máy Tính Tích Hợp Robot Để Loại Bỏ Tạp Chất Trong Sản Xuất Sợi**

Sinh viên thực hiện:

Nguyễn Văn Mến

Số thẻ SV: 105200337

Lê Văn Tuấn

Số thẻ SV: 105200349

Văn Ngọc Phương

Số thẻ SV: 105200340

Lớp: 20TDH2

Trong sản xuất sợi dệt may, tạp chất như mảnh gỗ, xơ cây, sợi nylon, lông vũ hay bụi bẩn sau quy trình tách sơ ngoại lai vẫn còn sót lại, buộc công nhân phải kiểm tra và loại bỏ thủ công, gây gián đoạn dây chuyền và tiềm ẩn sai sót trong chất lượng. Đề tài này hướng đến việc thiết lập một hệ thống tự động hoàn toàn, kết hợp camera công nghiệp và mô hình phát hiện vật thể YOLOv11 để nhận diện tạp chất theo đặc trưng hình thái và màu sắc trong thời gian thực.

Kết quả nhận diện được sử dụng trong mô-đun động học nghịch – thuận để tính toán chính xác vị trí gấp trên cơ cấu robot Delta, đồng thời Matlab được ứng dụng mô phỏng không gian làm việc, đảm bảo robot vận hành trong giới hạn an toàn. Toàn bộ thuật toán và giao tiếp phần cứng (Arduino Mega, driver TB6600, servo) được tích hợp trên Visual Studio Code, thực hiện chu trình “chụp ảnh – nhận diện – tính tọa độ – gấp thả” liên tục mà không cần can thiệp thủ công. Thí nghiệm trên mô hình thu nhỏ chứng minh hệ thống hoạt động ổn định, nhanh chóng, chính xác và có khả năng nâng cao năng suất, đồng nhất chất lượng sợi, đồng thời mở ra hướng ứng dụng AI sâu rộng trong tự động hóa sản xuất.

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ và Tên sinh viên	Số thẻ SV	Lớp	Chuyên ngành
Nguyễn Văn Mến	105200337	20TDH2	Kỹ thuật điều khiển & Tự động hóa
Lê Văn Tuấn	105200349	20TDH2	Kỹ thuật điều khiển & Tự động hóa
Văn Ngọc Phương	105200340	20TDH2	Kỹ thuật điều khiển & Tự động hóa

1. Tên đề tài đồ án:

### **Nghiên Cứu Và Phát Triển Hệ Thống Thị Giác Máy Tính Tích Hợp Robot Để Loại Bỏ Tạp Chất Trong Sản Xuất Sợi**

2. Đề tài thuộc diện:  Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

Khảo sát thực tế và lấy vật mẫu sản phẩm sợi bông lỗi tại công ty Cổ phần Dệt May hòa Thọ

4. Nội dung các phần thuyết minh và tính toán:

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Nguyễn Văn Mến	Tìm hiểu tổng quan về Robot Delta Tìm hiểu ứng dụng của robot Delta trong quy trình phân loại tạp chất
2	Văn Ngọc Phương	Nghiên cứu động học của robot
3	Lê Văn Tuấn	Tìm giải pháp truyền thông dữ liệu cho hệ thống điều khiển Viết và hoàn thiện báo cáo thuyết minh đồ án

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Nguyễn Văn Mến	Lập trình chương trình chính cho toàn hệ thống Lập trình xử lý ảnh YOLOv11 Lập trình điều khiển Robot Delta Mô phỏng vùng làm việc của robot Delta trên Matlab/Simulink

		Nghiên cứu cách truyền thông dữ liệu Thiết kế giao diện điều khiển
2	Văn Ngọc Phương	Lắp đặt mô hình cơ khí Robot Delta và tay gắp Tính chọn thiết bị điện và cơ khí, tay gắp Thiết kế sơ đồ đấu nối điện cho hệ thống Chụp ảnh, đánh nhãn cho quá trình train xử lý ảnh Lập trình điều khiển Robot Delta
3	Lê Văn Tuấn	Chụp ảnh, đánh nhãn cho quá trình train xử lý ảnh Tính chọn thiết bị điện và cơ khí, tay gắp Hỗ trợ lập trình chương trình chính Thi công đấu nối điện cho toàn bộ hệ thống Lập trình xử lý ảnh YOLOv11

5. Ngày giao nhiệm vụ đồ án: 17/2/2025

6. Ngày hoàn thành đồ án: 1/6//2025

Đà Nẵng, ngày tháng năm 2025

**Trưởng Bộ môn Tự động hóa**

**Người hướng dẫn**

TS. Giáp Quang Huy

Nguyễn Thị Kim Trúc

**PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP**  
(Phiếu dành cho người hướng dẫn/sinh viên)

Họ và Tên sinh viên	Số thẻ SV	Lớp	Chuyên ngành
Nguyễn Văn Mến	105200337	20TDH2	Kỹ thuật điều khiển & Tự động hóa
Lê Văn Tuấn	105200349	20TDH2	Kỹ thuật điều khiển & Tự động hóa
Văn Ngọc Phương	105200340	20TDH2	Kỹ thuật điều khiển & Tự động hóa

Tên đề tài ĐATNCP: **Nghiên Cứu Và Phát Triển Hệ Thống Thị Giác Máy Tính Tích Hợp Robot Để Loại Bỏ Tạp Chất Trong Sản Xuất Sợi**

Họ tên người HD: TS. Nguyễn Thị Kim Trúc

Đơn vị: Đại học Bách Khoa ĐN

Tuần	Ngày	Khối lượng		GVHD ký tên
		đã thực hiện (%)	tiếp tục thực hiện (%)	
1	17/02/2025	Nhận đề tài	Gặp người hướng dẫn định hướng đồ án	
2	24/02/2025	Nghiên cứu các ứng dụng của robot Delta trong công nghiệp	Phân tích và xác định phần cứng của hệ thống	
3	03/03/2025	Xác định yêu cầu và bài toán đặt ra		
4	10/03/2025	Duyệt lần 1: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
5	17/03/2025	Nghiên cứu cấu tạo, cấu trúc robot Delta và tay gắp	Xác định và tính chọn các thành phần cơ khí của robot Delta	
6	24/03/2025	Nghiên cứu cơ sở lý thuyết của mô hình AI	Tìm kiếm nguồn dữ liệu và train mô hình YOLO	

7	31/03/2025	Lên phương án thiết kế hệ thống, quy trình công nghệ	Thiết kế bản vẽ thi công lắp đặt thiết bị, bản vẽ thi công điện, sơ đồ kết nối hệ thống	
8	07/04/2025	Duyệt lần 2: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
9	14/04/2025	Nghiên cứu phương trình động học robot Delta, lên phương án thiết kế mô hình, tính chọn thiết bị	Mô hình hóa, kiểm chứng phương trình động học, tính toán không gian làm việc trên phần mềm Matlab Simulink	
10	21/04/2025	Tinh chỉnh bản thiết kế mô hình (Cơ khí, điện) Thi công lắp đặt mô hình	Thi công lắp đặt mô hình hệ thống robot Delta Thi công đấu nối tủ điện	
11	28/04/2025	Lập trình chương trình điều khiển giám sát	Tiến hành chạy kiểm nghiệm và tiến hành đo đạc xác định các thông số thực tế của cơ cấu	
12	05/05/2025	Duyệt lần 3: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
13	12/05/2025	Thiết kế giao diện màn hình điều khiển giám sát	Tiến hành chạy mô phỏng màn hình điều khiển giám sát trên pycharm	
14	19/05/2025	Chạy thử với chương trình điều khiển, chỉnh sửa mô hình	Tiến hành đo đạc và kiểm nghiệm hiệu suất làm việc của mô hình	
15	26/05/2025	Hoàn thiện mô hình, tổng hợp tài liệu, hoàn thiện các phần trong báo cáo		

## LỜI NÓI ĐẦU VÀ CẢM ƠN

Chúng em xin gửi lời cảm ơn sâu sắc tới người hướng dẫn – cô Nguyễn Thị Kim Trúc, người đã tận tâm theo dõi tiến độ công việc và luôn đóng góp những ý kiến quý báu, giúp nhóm hoàn thành đề án tốt nghiệp một cách xuất sắc. Sự tận tụy và sự chỉ bảo của cô đã truyền cảm hứng và mở ra những góc nhìn mới, góp phần hoàn thiện năng lực chuyên môn của mỗi thành viên trong nhóm.

Chúng em cũng xin bày tỏ lòng biết ơn chân thành tới giám đốc, anh Nguyễn Xuân Bình và toàn thể cán bộ của Tổng Công ty Cổ phần Dệt May Hòa Thọ, những người đã tạo điều kiện thuận lợi và hỗ trợ nhiệt tình cho nhóm trong suốt quá trình thực hiện đề tài. Sự hợp tác và tâm huyết của quý công ty đã góp phần không nhỏ vào thành công của dự án.

Mặc dù thời gian thực hiện đề án có hạn và năng lực còn nhiều hạn chế, chúng em nhận thức được những thiếu sót tồn tại. Vì vậy, rất mong nhận được những góp ý, chỉ bảo từ quý thầy cô để chúng em có thêm động lực và cơ hội hoàn thiện kiến thức, từ đó phục vụ tốt hơn trong công việc sau này.

Một lần nữa, chúng em xin chân thành cảm ơn tất cả mọi người đã đồng hành và hỗ trợ trong quá trình thực hiện đề án này.

## LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT

Chúng em xin cam đoan đề tài: “**Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi**” là sự nghiên cứu của nhóm em (Nguyễn Văn Mến, Lê Văn Tuấn, Văn Ngọc Phương) dưới sự hướng dẫn của TS. Nguyễn Thị Kim Trúc (Khoa Điện – Trường Đại học Bách Khoa – Đại học Đà Nẵng); Giám Đốc Nguyễn Xuân Bình, ngoài ra không có bất cứ sự sao chép của người khác.

Đề tài, nội dung báo cáo là sản phẩm mà nhóm em đã nghiên cứu trong quá trình học tập tại trường cũng như thời gian tham gia thực tập và thực hiện đề tài tại Công ty.

Các số liệu, kết quả trong báo cáo này là hoàn toàn trung thực, em xin chịu hoàn toàn trách nhiệm, kỷ luật của bộ môn và nhà trường nếu như có vấn đề xảy ra.

*Đà Nẵng, ngày 16 tháng 6 năm 2025*

**Nhóm đồng tác giả thực hiện**

**Sinh viên 1**

**Sinh viên 2**

**Sinh viên 3**

**Nguyễn Văn Mến**

**Lê Văn Tuấn**

**Văn Ngọc Phương**

# MỤC LỤC

TÓM TẮT.....	i
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP .....	ii
PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP.....	iv
LỜI NÓI ĐẦU VÀ CẢM ƠN .....	vi
LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT .....	vii
MỤC LỤC .....	viii
DANH MỤC HÌNH ẢNH.....	xi
DANH MỤC BẢNG BIỂU.....	xiii
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT.....	xiv
MỞ ĐẦU .....	1
1. Mục đích của đề tài.....	1
2. Mục tiêu của đề tài .....	1
3. Phạm vi và đối tượng nghiên cứu.....	2
4. Phương pháp nghiên cứu .....	2
Chương 1:TỔNG QUAN ĐỀ TÀI .....	4
1.1, Giới thiệu chung .....	4
1.1.1 Tính cấp thiết của đề tài.....	4
1.1.2. Những vấn đề tồn tại trong kiểm tra và loại bỏ tạp chất lỗi .....	5
Kết luận chương 1 .....	9
Chương 2: THIẾT KẾ HỆ THỐNG, LỰA CHỌN THIẾT BỊ VÀ THI CÔNG PHẦN CỨNG.....	10
2.1 Tổng quan về hệ thống .....	10
2.2. Giới thiệu Robot .....	10
2.2.1. Robot nối tiếp và robot song song.....	10
2.2.2. Giới thiệu Robot Delta .....	11
2.2.3. Phân loại .....	11
2.3. Giới thiệu chung về các thiết bị.....	13
2.3.1. Giới thiệu chung về Arduino .....	13
2.3.2. Giới thiệu chung về động cơ bước và bộ điều khiển động cơ.....	14
2.4 Yêu cầu thiết kế phần cứng .....	15
2.5 Thiết kế mô hình trên phần mềm Solidworks .....	15
2.5.1 Đế cố định Robot.....	15
2.5.2 Đế di động Robot.....	16

2.5.3 Khớp trên Robot .....	16
2.5.4 Khớp hình bình hành Robot .....	17
2.5.4 Khung Robot .....	17
2.6. Một số chi tiết khác .....	18
2.7 Lựa chọn thiết bị .....	19
2.7.1 Động cơ .....	19
2.7.2 Danh sách các thiết bị.....	19
2.7.3 Camera.....	22
2.8 Thi công phần cứng.....	22
2.8.1 Mô hình thực tế sau khi thi công.....	22
2.8.2 Sơ đồ hệ thống.....	22
2.8.3 Thi công tủ điện.....	24
Kết luận chương 2: .....	24
Chương 3: CƠ SỞ LÝ THUYẾT VỀ ROBOT DELTA .....	25
3.1 Các thông số của Robot Delta.....	25
3.2. Tính toán động học cho Robot Delta .....	25
3.2.1. Động học nghịch.....	25
3.2.2. <i>Động học thuận</i> .....	31
3.3. Không gian làm việc của Robot Delta.....	34
Kết luận chương 3 .....	35
Chương 4: XÂY DỰNG VÀ TRIỂN KHAI MÔ HÌNH THỊ GIÁC MÁY TÍNH....	37
4.1. Tổng quan về thị giác máy tính, xử lý ảnh và các phần mềm .....	37
4.1.1 Giới thiệu về thị giác máy tính .....	37
4.1.2 Giới thiệu về xử lý ảnh.....	37
4.1.3 Giới thiệu phần mềm sử dụng trong quá trình huấn luyện.....	38
4.2 Giới thiệu bài toán Object Detection và mô hình Yolo .....	41
4.2.1 Bài toán Object Detection .....	41
4.2.2 Mô hình YOLO ( YOU ONLY LOOK ONCE ).....	42
4.2.3 Những cải tiến của YOLOv11 so với các phiên bản trước .....	48
4.3 Phương pháp Homography .....	48
4.3.1 Giới thiệu.....	48
4.4 Quy trình huấn luyện YOLO .....	50
4.5 Chuẩn bị tập dữ liệu .....	51
4.5.1 Thu thập dữ liệu.....	51
4.5.2 Dán nhãn dữ liệu .....	51
4.5.3 Phân chia tập dữ liệu .....	52

4.5.4	Chuẩn hoá và làm giàu dữ liệu.....	53
4.6	Quá trình huấn luyện.....	55
4.6.1	Quá trình huấn luyện trên phần mềm Kaggle .....	55
4.6.2	Kết quả huấn luyện.....	56
4.6.3	Đánh giá huấn luyện.....	58
4.7	Tính toán và xác định vị trí đối tượng.....	58
	Kết luận chương 4: .....	67
	<b>CHƯƠNG 5: TÍCH HỢP HỆ THỐNG VÀ KIỂM THỬ THỰC NGHIỆM</b> .....	68
5.1.	Mô tả quy trình hoạt động và lưu đồ giải thuật của hệ thống.....	68
5.2.	Lập trình, thiết kế phần mềm điều khiển và giao diện người dùng .....	70
	Kết luận chương 5 .....	71
	<b>Chương 6: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ</b> .....	72
6.1.	Kiểm chứng động học nghịch trên checkerboard .....	72
6.1.1.	Đánh giá kết quả toàn hệ thống.....	74
	Tổng kết chương 6.....	79
	<b>KẾT LUẬN</b> .....	80
	Hướng phát triển.....	80
	<b>PHỤ LỤC 1</b> .....	1
	<b>PHỤ LỤC 2</b> .....	1

## DANH MỤC HÌNH ẢNH

Hình 1.1 Hình ảnh sơ ngoại lai được thu gom từ máy SP-FPU .....	4
Hình 1.2 Các tạp chất thường xuất hiện trên sợi bông .....	5
Hình 1.3 Quy trình sản xuất sợi tại nhà máy Hoà Thọ .....	6
Hình 1.4 Quá trình tách thủ công của công nhân để thu hồi bông có thể sử dụng.....	8
Hình 1.5 Quy trình phát hiện và loại bỏ tạp chất bằng AI và Robot Delta .....	8
Hình 2.1 Sơ đồ tổng quan về hệ thống .....	10
Hình 2.2 Một số mô hình robot delta .....	11
Hình 2.3 Robot Delta kiểu ba khớp xoay .....	12
Hình 2.4 Robot Delta kiểu ba khớp trượt.....	12
Hình 2.5 Arduino Mega 2560.....	13
Hình 2.6 Sơ đồ chân Arduino Mega 2560.....	13
Hình 2.7 Một số loại động cơ bước .....	14
Hình 2.8 Driver TB 6600.....	15
Hình 2.9 Đế cố định robot .....	16
Hình 2.10 Đế di động robot.....	16
Hình 2.11 Khớp trên robot .....	17
Hình 2.12 Khớp hình bình hành robot.....	17
Hình 2.13 Khung robot.....	17
Hình 2.14 Phần cứng sau thi công và tay gấp .....	22
Hình 2.15 Module MLS2596 .....	23
Hình 2.16 Sơ đồ đi dây tín hiệu điều khiển.....	23
Hình 2.17 Hệ thống điện sau khi hoàn tất đấu nối .....	24
Hình 3.1 Mô hình động học robot delta .....	26
Hình 3.2 Giao tuyến của hình cầu và mặt phẳng YZ .....	26
Hình 3.3 Mặt phẳng YZ.....	27
Hình 3.4 Hệ toạ độ mới ( $x'0, y'0, z'0$ ) .....	30
Hình 3.5 Mô hình động học robot delta .....	31
Hình 3.6 Tọa độ các điểm $J1', J2', J3'$ trên mặt phẳng Oxy .....	32
Hình 3.7 Không gian làm việc của Robot Delta trong. a) Không gian 3 chiều xyz; Không gian 2 chiều b) xy; c) xz; d) yz .....	35

Hình 4.1 Phần mềm roboflow .....	38
Hình 4.2 Phần mềm Kaggle.....	38
Hình 4.3 Xxe tự lái cũng phụ thuộc nhiều vào computer vision.....	42
Hình 4.4 Cách xác định anchor box cho vật thể .....	43
Hình 4.5 Cách xác định vật thể của YOLO .....	43
Hình 4.6 So sánh 5 mô hình YOLOv11 .....	44
Hình 4.7 Cấu trúc của YOLOv11 .....	44
Hình 4.8 Chỉ số đánh giá độ chính xác IoU .....	46
Hình 4.9 Biểu đồ Precision và Recall .....	47
Hình 4.10 Mối quan hệ giữa hệ tọa độ ảnh và mặt phẳng làm việc trong phép biến đổi Homography .....	49
Hình 4.11 Quy trình chuyển đổi hệ tọa độ .....	50
Hình 4.12 Quy trình phát hiện và xác định vị trí đối tượng bằng mô hình YOLO .....	51
Hình 4.13 Quy trình huấn luyện YOLO.....	51
Hình 4.14 Dán nhãn dữ liệu .....	52
Hình 4.15 Tỷ lệ chia tập train/validation/test trong hệ thống.....	52
Hình 4.16 Cấu trúc thư mục dữ liệu cho hệ thống .....	53
Hình 4.17 Cấu hình chuẩn hóa dữ liệu trong Roboflow.....	53
Hình 4.18 Các hình ảnh mới được tạo sau khi đã làm giàu .....	54
Hình 4.19 Nhiều lần chương trình tự dừng khi quá trình huấn luyện sắp hoàn thành .	55
Hình 4.20 Quá trình huấn luyện mô hình trên phần mềm Kaggle .....	56
Hình 4.21 Kết quả huấn luyện thời gian thực với 2052 ảnh đầu vào.....	57
Hình 4.22 Kết quả huấn luyện thời gian thực với 8554 ảnh đầu vào.....	57
Hình 4.23 Kết quả huấn luyện với thời gian thực .....	58
Hình 4.24 Kết quả sau khi huấn luyện với yolov11n.....	59
Hình 4.25 Kết quả sau khi huấn luyện với yolov11m ( cấu hình cơ bản).....	59
Hình 4.26 Kết quả sau khi huấn luyện với yolov11m ( mở rộng tham số).....	59
Hình 4.27 Ma trận Confusion.....	60
Hình 4.28 Đồ thị F1-Confidence Curve. ....	61
Hình 4.29 Đồ thị độ phủ Recall - Confidence. ....	62
Hình 4.30 Đồ thị độ chính xác Precision - Confidence.....	63

Hình 4.31 Precision-Recall Curve:.....	64
Hình 4.32 Gán tọa độ Oxy với ảnh chứa tâm pixel.....	65
Hình 4.33 Lấy tọa độ 5 điểm trên thực tế.....	65
Hình 5.1 Các lưu đồ thuật toán của hệ thống .....	69
Hình 5.2 Hệ thống thị giác máy tính tích hợp robot Delta trong quá trình loại bỏ tạp chất sợi .....	70
Hình 5.3 Hệ tọa độ Robot so với checkerboard .....	72
Hình 6.1 Kiểm chứng động học nghịch.....	73

## **DANH MỤC BẢNG BIỂU**

Bảng 2.1 Các chi tiết khác khi thi công robot .....	18
Bảng 2.2 Động cơ bước 57HS56.....	19
Bảng 2.3 Danh sách các thiết bị trong hệ thống.....	20
Bảng 2.4: Web cam máy tính .....	22
Bảng 3.1 Thông số các chi tiết Robot.....	25
Bảng 4.1: Tọa độ thực các điểm đã chọn .....	66
Bảng 6.1: Kiểm chứng động học nghịch .....	73
Bảng 6.2: Quy trình giai đoạn hoạt động của hệ thống.....	74
Bảng 6.3: Đánh giá độ nhận diện hệ thống .....	78

## **DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT**

UART: Universal Asynchronous Receiver/Transmitter

YOLO: You Only Look Once

AI: Artificial Intelligence

mAP: mean Average Precision

IoU: Intersection over Union

GPS: Global Positioning System

CNC: Computer Numerical Control

NMS: Network Management System

## **GIẢI THÍCH THUẬT NGỮ TIẾNG ANH**

Precision Độ chính xác

Recall: Độ bao phủ

F1-score trung bình điều hòa



## MỞ ĐẦU

### 1. Mục đích của đề tài

Trong quá trình sản xuất sợi tại các doanh nghiệp dệt may, mặc dù đã ứng dụng các thiết bị tách sợi ngoại lai (ví dụ: SP-FPU) để loại bỏ tạp chất, vẫn tồn tại một lượng đáng kể (gỗ vụn, xơ cây, sợi nilon, lông vũ, bụi...) sót lại. Để đảm bảo chất lượng đầu vào trước khi đưa sợi vào công đoạn dệt, các nhà máy thường bố trí công nhân kiểm tra và tách loại thủ công ở khâu cuối cùng. Tuy nhiên, quy trình này mang tính thủ công cao, tốn nhiều thời gian, dễ phát sinh sai sót và không đảm bảo độ ổn định về chất lượng.

Đề tài này nhằm phát triển một hệ thống tự động tích hợp công nghệ thị giác máy tính và robot Delta để thay thế hoàn toàn khâu loại bỏ tạp chất thủ công. Hệ thống sẽ tự động phát hiện và phân loại chính xác các tạp chất còn sót lại trên bề mặt sợi, đồng thời sử dụng robot Delta để loại bỏ ngay lập tức, giúp nâng cao năng suất, đồng đều chất lượng sản phẩm và giảm sự phụ thuộc vào lao động thủ công. Giải pháp này không chỉ đáp ứng nhu cầu tự động hóa và chuẩn hóa trong ngành dệt may hiện đại mà còn góp phần tối ưu hóa chi phí và tăng cường khả năng cạnh tranh cho doanh nghiệp.

### 2. Mục tiêu của đề tài

Để hiện thực hóa mục tiêu tự động hóa khâu loại bỏ tạp chất cuối cùng trong sản xuất sợi, đề tài đặt ra các mục tiêu cụ thể sau. Về thị giác máy tính, trước hết cần xây dựng một tập dữ liệu hình ảnh thực tế chứa các loại tạp chất phổ biến (nilon, xơ cây, gỗ vụn, lông vũ...), đồng thời thực hiện gán nhãn chính xác cho từng đối tượng. Trên cơ sở đó, mô hình phát hiện đối tượng YOLOv11 sẽ được huấn luyện và tối ưu hóa để nhận diện tạp chất trên nền bông với cả độ chính xác và tốc độ đáp ứng yêu cầu thời gian thực. Hiệu năng của mô hình sẽ được đánh giá thông qua các chỉ số Precision, Recall, F1-score và mAP, nhằm kiểm chứng tính khả thi khi triển khai trong môi trường sản xuất.

Về thiết kế cơ khí và điều khiển robot, nhóm sẽ thiết kế và chế tạo mô hình robot Delta phù hợp với không gian làm việc quy định, sử dụng phần mềm SolidWorks kết hợp in 3D cho các chi tiết cơ khí. Trên nền tảng đó, bài toán động học thuận và nghịch của robot Delta được giải quyết để xây dựng hàm chuyển đổi tọa độ giữa ảnh đầu vào và vị trí thực tế, phục vụ cho việc lập trình điều khiển chính xác. Điều khiển robot sẽ thực hiện qua Arduino và driver TB6600, đảm bảo quá trình gắp và loại bỏ tạp chất diễn ra ổn định, khớp với tọa độ phát hiện từ mô hình AI.

Về tích hợp hệ thống, một giao diện người dùng (UI) trực quan sẽ được phát triển để hiển thị kết quả phát hiện và hỗ trợ vận hành bán tự động khi cần thiết. Quy trình đồng bộ giữa khâu chụp ảnh, nhận diện, tính toán vị trí và điều khiển robot gắp – loại bỏ sẽ được thiết kế khép kín, từ đó cho phép chạy thử nghiệm trọn vẹn trên mô hình tại xưởng sản xuất.

Cuối cùng, hệ thống sẽ được thử nghiệm trong môi trường mô hình để đánh giá độ chính xác của robot khi xử lý các loại tạp chất khác nhau. Kết quả về thời gian xử lý, tỉ lệ loại bỏ chính xác và tính ổn định sẽ được so sánh trực tiếp với phương pháp thủ công hiện hành, giúp xác định ưu thế cũng như những hạn chế—ví dụ như hiện tượng rung động của robot hoặc sai lệch trong phân lớp tạp chất—từ đó đề xuất các cải tiến cho giai đoạn phát triển tiếp theo

### **3. Phạm vi và đối tượng nghiên cứu**

**Đối tượng nghiên cứu:** Chia thành ba phần chính: (1) các tạp chất trong nguyên liệu sợi (gỗ vụn, nilon, xơ cây, lông vũ...) với mục tiêu khảo sát đặc tính để phục vụ nhận diện tự động; (2) robot Delta dạng song song, tập trung đánh giá động học, độ chính xác tọa độ và tốc độ phản hồi khi gắp – loại bỏ; (3) hệ thống thị giác máy tính sử dụng YOLOv11, nhằm phát hiện và phân loại tạp chất trên nền bông theo thời gian thực với cả độ chính xác và hiệu suất cao.

**Phạm vi thực hiện:** Trong khuôn khổ nghiên cứu, mô hình robot Delta được thiết kế và chế tạo trên quy mô thu nhỏ thay vì triển khai trên dây chuyền công nghiệp thực tế. Mô hình YOLOv11 chỉ được huấn luyện với một lớp duy nhất là “tạp chất”, thay vì phân loại chi tiết từng loại chất bẩn, nhằm đơn giản hóa bài toán và tập trung vào hiệu năng phát hiện chung. Dữ liệu hình ảnh được thu thập trực tiếp từ mô hình thí nghiệm dưới điều kiện chiếu sáng được kiểm soát chặt chẽ, đảm bảo tính đồng nhất và nhất quán của tập huấn luyện. Toàn bộ quá trình phát triển và vận hành hệ thống sử dụng kết hợp Arduino cho điều khiển robot, SolidWorks cho thiết kế cơ khí, Kaggle cho huấn luyện mô hình, Roboflow cho quản lý và gán nhãn dữ liệu, cùng các thư viện xử lý ảnh hiện đại để xử lý và phân tích hình ảnh.

### **4. Phương pháp nghiên cứu**

Trong quá trình thực hiện đề tài, nhóm đã áp dụng kết hợp nhiều phương pháp nghiên cứu để từng bước xây dựng và hoàn thiện hệ thống. Cụ thể như sau:

- **Phương pháp khảo sát thực tế:** Nhóm đã đến khảo sát tại Công ty Dệt May Hòa Thọ để tìm hiểu quy trình hiện tại, đặc biệt là khâu loại bỏ tạp chất bằng tay. Qua đó, nhóm xác định rõ nhu cầu tự động hóa và các vấn đề còn tồn tại trong thực tế sản xuất.

*Capstone Project: “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”*

- **Phương pháp thiết kế và mô phỏng:** Nhóm sử dụng phần mềm SolidWorks để thiết kế các bộ phận cơ khí của robot Delta, như khung, tay gấp, đế cố định... Sau đó mô phỏng vùng làm việc và tính toán động học trên Matlab/Simulink để kiểm tra khả năng hoạt động của robot.
- **Phương pháp huấn luyện mô hình nhận diện:** Nhóm tiến hành thu thập hình ảnh thực tế, gán nhãn dữ liệu bằng phần mềm Roboflow và huấn luyện mô hình nhận diện tạp chất bằng thuật toán YOLOv11 trên Kaggle với sự hỗ trợ của GPU để tăng tốc độ xử lý.
- **Phương pháp tích hợp hệ thống:** Các thành phần như camera, mô hình AI, robot Delta, vi điều khiển Arduino Mega và driver điều khiển động cơ được kết nối và lập trình hoạt động đồng bộ. Việc tích hợp giúp hệ thống có thể tự động phát hiện và loại bỏ tạp chất theo thời gian thực.
- **Phương pháp kiểm tra và đánh giá:** Nhóm kiểm tra hệ thống qua nhiều lần chạy thử. Mô hình AI được

## **5. Cấu trúc đề án.**

Đề tài tốt nghiệp được trình bày với các nội dung chính sau:

Chương 1: Tổng quan đề tài.

Chương 2: Thiết kế hệ thống, lựa chọn thiết bị và thi công phần cứng.

Chương 3: Cơ sở lý thuyết về robot Delta.

Chương 4: Xây dựng và triển khai mô hình thị giác máy tính.

Chương 5: Tích hợp hệ thống và kiểm thử thực nghiệm.

Chương 6: Thực nghiệm, đánh giá kết quả.

## Chương 1: TỔNG QUAN ĐỀ TÀI

### 1.1, Giới thiệu chung

#### 1.1.1 Tính cấp thiết của đề tài

Trong bối cảnh ngành dệt may đang hướng tới tự động hóa và chuẩn hóa chất lượng theo xu thế Công nghiệp 4.0, việc kiểm soát chất lượng sợi cotton đầu vào, đặc biệt là loại bỏ tạp chất, trở nên đặc biệt quan trọng. Sợi cotton quyết định đến độ bền, thẩm mỹ và giá trị kinh tế của sản phẩm dệt may. Tuy nhiên, trong quá trình thu hoạch và chế biến, sợi dễ bị lẫn tạp chất như mảnh gỗ, xơ cây, nilon, lông vũ hay bụi bẩn. Những tạp chất này không chỉ ảnh hưởng đến chất lượng sản phẩm mà còn gây lỗi bề mặt vải, làm gián đoạn dây chuyền sản xuất và tổn hại đến uy tín thương hiệu.

Hiện nay, nhiều doanh nghiệp trong ngành vẫn sử dụng máy tách xơ ngoại lai như SP-FPU để loại bỏ tạp chất khỏi nguyên liệu. Nhưng hiệu suất loại bỏ của thiết bị này trong thực tế vẫn còn hạn chế, dẫn đến việc cần bổ sung thêm bước kiểm tra thủ công nhằm đảm bảo chất lượng sản phẩm. Phương pháp kiểm tra thủ công tuy cần thiết nhưng lại tiêu tốn nhiều thời gian, dễ xảy ra sai sót và làm gia tăng chi phí lao động.

Trước thực trạng này, đề tài “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi” là một giải pháp có tính ứng dụng cao. Bằng cách tích hợp công nghệ thị giác máy tính và robot Delta, hệ thống có khả năng tự động phát hiện – loại bỏ tạp chất theo thời gian thực với độ chính xác vượt trội nhờ các thuật toán AI hiện đại. Hệ thống không chỉ nâng cao năng suất so với phương pháp thủ công mà còn giảm đáng kể chi phí vận hành, góp phần thúc đẩy hiện đại hóa ngành dệt may và đáp ứng các tiêu chuẩn khắt khe của thị trường quốc tế.



Hình 1.1 Hình ảnh sơ ngoại lai được thu gom từ máy SP-FPU

### 1.1.2. Những vấn đề tồn tại trong kiểm tra và loại bỏ tạp chất lỗi

Trong quy trình sản xuất sợi, dù có bỏ qua các công đoạn trung gian như xử lý nguyên liệu, kéo sợi hay dệt vải, khâu kiểm tra chất lượng vẫn đóng vai trò then chốt, đảm bảo sản phẩm cuối cùng đạt tiêu chuẩn. Việc rà soát và loại bỏ tạp chất ngay từ nguyên liệu đầu vào trước khi đưa vào giai đoạn sản xuất chính phải được thực hiện nghiêm ngặt, nhằm ngăn chặn những yếu tố có thể làm giảm chất lượng và độ bền của sợi thành phẩm.

Mặc dù máy tách sơ ngoại lai (SP-FPU) là công nghệ phổ biến trong ngành, nhưng hiệu suất loại bỏ tạp chất của thiết bị này vẫn còn hạn chế. Vì vậy, đề tài này tập trung nghiên cứu và phát triển một hệ thống thị giác máy tính tích hợp robot, có khả năng phát hiện và loại bỏ hiệu quả các loại tạp chất thường gặp trong nguyên liệu đầu vào, bao gồm:

- *Xơ cây*: mảnh vụn xơ từ vỏ hoặc thân cây, còn sót lại sau thu hoạch và sơ chế.
- *Nhựa*: hạt hoặc mảnh nhựa nhỏ lẫn vào nguyên liệu trong quá trình vận chuyển, lưu kho.
- *Nilông*: vụn màng hoặc mảnh túi nilông, dễ bám dính do tính nhẹ và tích điện.
- *Gỗ và vỏ cây*: mảnh gỗ cứng hoặc lớp vỏ cây chưa được làm sạch hoàn toàn, ảnh hưởng đến độ đồng đều và khả năng kéo sợi.
- *Lông vũ*: lông động vật hoặc lông chim rơi trong không gian sản xuất, có thể lẫn vào nguyên liệu khi có gió hoặc trong môi trường mở.

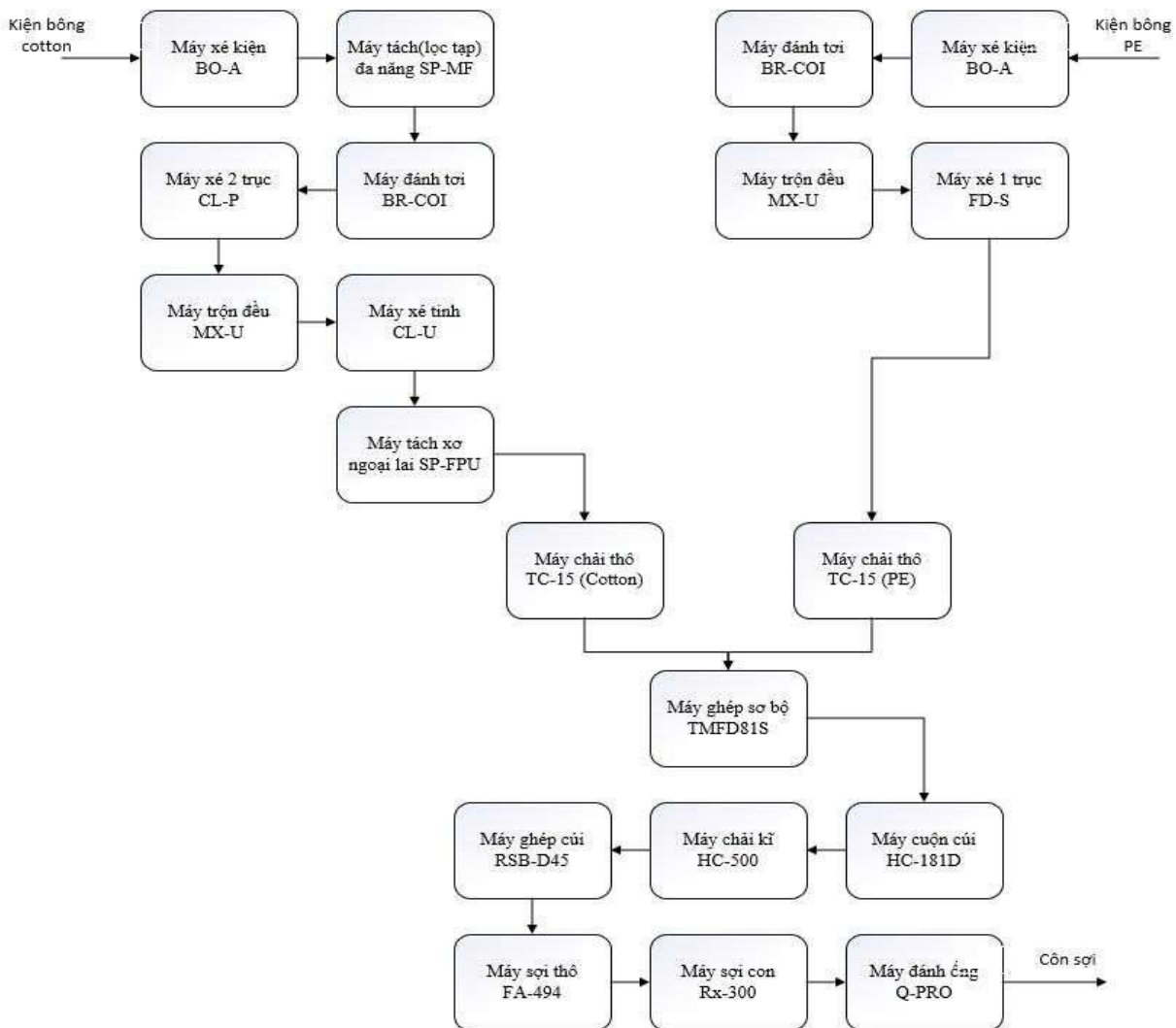


Hình 1.2 Các tạp chất thường xuất hiện trên sợi bông

## 1.2. Quy trình sản xuất tại nhà máy sợi

### 1.2.1 Chuỗi công đoạn chính từ nguyên liệu đến thành phẩm

Quy trình sản xuất sợi tại nhà máy bắt đầu bằng xử lý nguyên liệu thô, trong đó bông cotton trải qua các công đoạn xé kiện, tách đa năng, đánh toi, trộn đều và tách xơ ngoại lai SP-FPU để loại bỏ tạp chất, trong khi xơ PE được đánh toi, xé kiện và trộn đều tương tự; tiếp theo cả hai loại xơ được chải thô trên máy TC-15, sau đó ghép sơ bộ bằng TMFD81S để tạo cúi chải hỗn hợp. Sợi sau đó được phân nhánh: một phần tiếp tục qua RSB-D45 và FA-494 để hình thành ống thô, phần còn lại qua HC-500 và HC-181D để gia công chải kỹ, trước khi hội tụ tại máy kéo sợi con Rx-300 và được hoàn thiện trên Q-PRO nhằm loại bỏ sợi lỗi và tạo thành “con sợi” đáp ứng các yêu cầu về kích thước, cơ tính và chất lượng. Thiết bị SP-FPU của Trützschler đóng vai trò then chốt trong việc đảm bảo độ sạch nguyên liệu, góp phần nâng cao hiệu suất sản xuất và chất lượng sợi thành phẩm



Hình 1.2 Quy trình sản xuất sợi tại nhà máy Hoà Thọ

### **1.2.2 Vị trí và vai trò của khâu kiểm tra – loại bỏ tạp chất**

Khâu kiểm tra và loại bỏ tạp chất, thực hiện ngay sau bước xử lý nguyên liệu thô và trước khi đưa xơ vào công đoạn chải thô, đóng vai trò then chốt trong đảm bảo chất lượng sợi. Sau khi bông cotton và xơ PE được xé kiện, đánh toi và trộn đều, nguyên liệu sẽ được dẫn qua thiết bị tách xơ ngoại lai SP-FPU, nhằm loại bỏ các hạt bụi, mảnh gỗ, sợi nilon, vỏ hạt và những xơ ngoại lai còn sót lại. Việc bố trí SP-FPU tại vị trí này đảm bảo rằng toàn bộ xơ đầu vào cho máy chải thô TC-15 đều đã được làm sạch tối ưu, từ đó nâng cao hiệu suất và độ ổn định của các công đoạn tiếp theo.

Trước hết, bước làm sạch ban đầu này góp phần đảm bảo tính đồng nhất và độ tinh khiết của nguyên liệu, giảm thiểu các khiếm khuyết bề mặt trên sợi thành phẩm. Thứ hai, việc loại bỏ sớm các tạp chất có độ cứng hoặc kích thước lớn giúp bảo vệ hệ thống lưỡi cắt và trục chải, từ đó giảm thiểu chi phí bảo trì và tăng tuổi thọ thiết bị. Cuối cùng, độ ổn định về chất lượng nguyên liệu đầu vào làm giảm thiểu các sự cố vận hành, tránh gián đoạn dây chuyền và hạn chế nhu cầu phân loại thủ công, qua đó tối ưu hóa năng suất sản xuất. Như vậy, khâu kiểm tra – loại bỏ tạp chất không chỉ là bước đệm kỹ thuật mà còn quyết định trực tiếp đến hiệu quả kinh tế và khả năng cạnh tranh của nhà máy sợi.

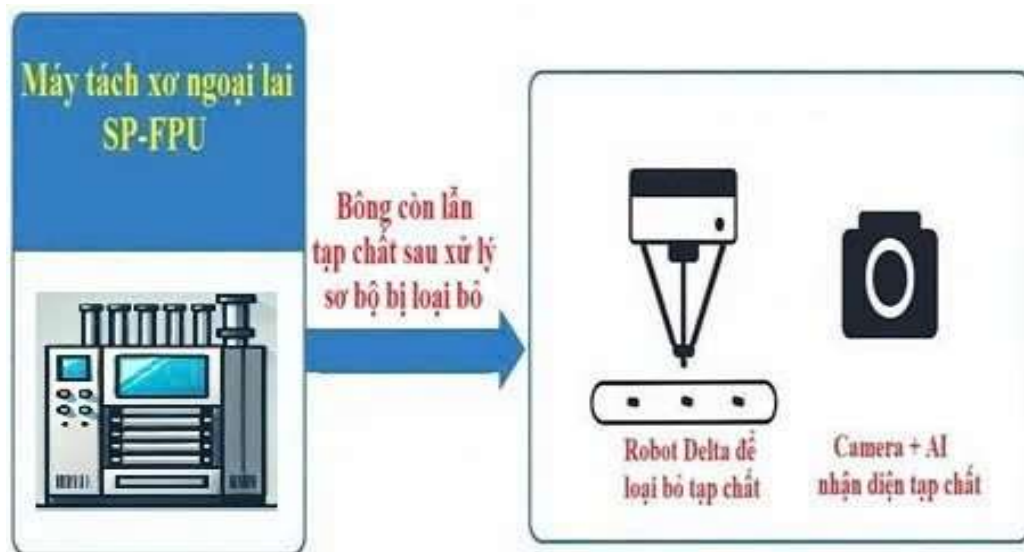
### **1.3. Phân tích thực trạng kiểm tra thủ công**

Đến nay, công đoạn loại bỏ tạp chất tại nhiều nhà máy sợi vẫn thực hiện bằng quan sát trực quan của công nhân. Phương thức này có ưu điểm về tính linh hoạt—công nhân có thể nhận biết những dạng tạp chất mới xuất hiện mà hệ thống tự động chưa được huấn luyện—và chi phí đầu tư thấp, phù hợp với quy mô nhỏ hoặc điều kiện vốn hạn hẹp. Tuy nhiên, việc phụ thuộc vào năng lực và sự tập trung của con người khiến tỷ lệ sai sót cao, đồng thời năng suất bị giới hạn do khối lượng nguyên liệu lớn và tốc độ xử lý chậm. Hơn nữa, điều kiện ánh sáng không đồng đều, bụi bẩn và sự mỏi mắt của công nhân trong ca làm việc kéo dài càng gia tăng nguy cơ bỏ sót hoặc nhật nhảm. Đào tạo công nhân để đạt được kỹ năng kiểm tra chính xác cũng tốn kém thời gian và chi phí.



Hình 1.3 Quá trình tách thủ công của công nhân để thu hồi bông có thể sử dụng

Từ những hạn chế nêu trên, có thể thấy rằng giải pháp tự động hóa sử dụng công nghệ thị giác máy tính kết hợp robot công nghiệp là hướng đi tất yếu để nâng cao độ chính xác, tăng năng suất và đảm bảo tính ổn định chất lượng trong sản xuất sợi.



Hình 1.4 Quy trình phát hiện và loại bỏ tạp chất bằng AI và Robot Delta

#### 1.4. Xác định bài toán nghiên cứu

Mặc dù các nhà máy sản xuất sợi đã áp dụng hệ thống tách xơ ngoại lai, trên thực tế vẫn còn một lượng tạp chất như mảnh gỗ, sợi nilon, xơ cây và lông vũ sót lại, buộc công nhân phải kiểm tra và loại bỏ thủ công ở khâu cuối cùng. Quy trình này không chỉ kéo dài thời gian xử lý và làm giảm hiệu suất chung, mà còn tiềm ẩn nguy cơ bỏ sót, dẫn đến chất lượng sản phẩm không đồng đều giữa các mẻ. Bên cạnh đó, sự khác biệt về kinh nghiệm và tay nghề, giới hạn thời gian làm việc theo ca, cùng với tình trạng mệt

mỗi hoặc nghỉ ốm của công nhân dễ gây gián đoạn hoạt động. Chi phí nhân công (lương, bảo hiểm, phúc lợi, đào tạo...) lại ngày càng tăng và khó kiểm soát, đặc biệt khi quy mô sản xuất mở rộng đòi hỏi phải bổ sung thêm lực lượng kiểm tra. Trước những hạn chế này, đề tài sẽ tập trung vào việc thiết kế và triển khai một hệ thống tự động tích hợp công nghệ thị giác máy tính và robot Delta, nhằm phát hiện, phân loại và loại bỏ tạp chất còn sót lại trên bề mặt sợi với tốc độ và độ chính xác cao, đồng thời đảm bảo tính ổn định của chất lượng sản phẩm đầu ra.

### **1.5. Đề xuất giải pháp tự động hóa**

Đề tài đề xuất xây dựng một hệ thống khép kín gồm: (1) thu thập và gán nhãn ảnh tạp chất (gỗ vụn, nilon, xơ cây, lông vũ) dưới điều kiện ánh sáng ổn định; (2) huấn luyện mô hình YOLOv11 đạt độ chính xác cao để phát hiện tạp chất trên nền bông; (3) thiết kế robot Delta nhỏ gọn trên nền SolidWorks, giải động học thuận/nghịch để biến đổi tọa độ ảnh sang tọa độ thực tế, điều khiển bởi Arduino và driver TB6600 với sai số nhỏ; (4) tích hợp qua module Python điều phối tuần tự “chụp ảnh – nhận diện – tính tọa độ – gắp–thả” và giao diện giám sát thời gian thực; (5) thử nghiệm đánh giá hiệu năng (thời gian xử lý, tỉ lệ loại bỏ chính xác) so với phương pháp thủ công để hoàn thiện và mở rộng ứng dụng.

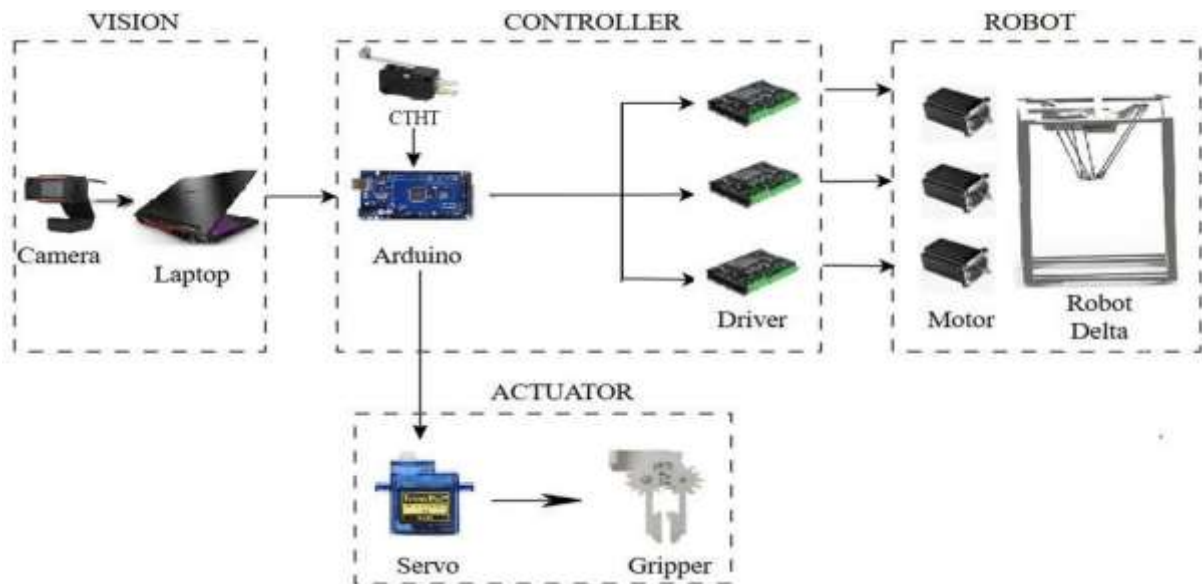
## **Kết luận chương 1**

Chương 1 đã cung cấp tổng quan về quy trình sản xuất sợi tại nhà máy, trong đó nhấn mạnh vai trò then chốt của khâu kiểm tra và loại bỏ tạp chất trước khi đưa nguyên liệu vào các công đoạn tiếp theo. Dựa trên khảo sát thực tế và phân tích, chương này chỉ ra các hạn chế của phương pháp thủ công—năng suất thấp, sai sót cao, khó mở rộng và chi phí nhân công tăng—từ đó xác định bài toán nghiên cứu: thiết kế một hệ thống tự động hóa khép kín. Giải pháp đề xuất kết hợp công nghệ thị giác máy tính (thu thập, gán nhãn dữ liệu ảnh tạp chất và huấn luyện mô hình YOLOv11) với robot công nghiệp Delta (thiết kế cơ khí, động học và điều khiển), tích hợp qua module Python và giao diện giám sát, nhằm tự động phân loại và loại bỏ tạp chất còn sót lại với tốc độ, độ chính xác và độ ổn định cao.

## Chương 2: THIẾT KẾ HỆ THỐNG, LỰA CHỌN THIẾT BỊ VÀ THI CÔNG PHẦN CỨNG

### 2.1 Tổng quan về hệ thống

Mục này tập trung vào việc thiết kế tổng quan về hệ thống giúp thực hiện quy trình công nghệ đã được mô tả mục trên. Hệ thống bao gồm các thành phần chính như khối xử lý ảnh, khối bộ điều khiển, khối cơ cấu chấp hành. Sơ đồ tổng quan về hệ thống được biểu diễn trong hình ảnh dưới.



Hình 2.1 Sơ đồ tổng quan về hệ thống

Khối xử lý ảnh: Thực hiện nhiệm vụ lấy thông tin đầu vào dưới dạng hình ảnh thông qua camera sau đó gửi đến trung tâm xử lý tín hiệu đầu tiên là máy tính. Mô hình YOLO sẽ phân loại và trích xuất ra vị trí tạp chất gửi cho khối điều khiển.

Khối điều khiển: Sau khi có được vị trí tạp chất khối điều khiển ở đây là Arduino sẽ tính toán và cấp xung để điều khiển động cơ và các thiết bị khác phối hợp theo quy trình để thực hiện phân loại.

Khối cơ cấu chấp hành: Động cơ servo hoạt động để gripper đóng lại, giữ chặt tạp chất. Khi Robot di chuyển đến vị trí cuối thì gripper sẽ mở ra để tạp chất sẽ rơi xuống.

### 2.2. Giới thiệu Robot

#### 2.2.1. Robot nối tiếp và robot song song

Trong robot công nghiệp, có hai kiến trúc động học chính:

Robot nối tiếp (ví dụ: robot 6 trục, SCARA): Sử dụng chuỗi khung hở, cho tầm với và độ linh hoạt lớn, nhưng do động cơ đặt ngay trên các khớp di động nên quán tính hệ thống cao, giới hạn tốc độ vận hành.

Robot song song (ví dụ: Robot Delta): Cơ cấu khép kín với các động cơ cố định vào đế, giảm đáng kể quán tính cho phần di động, từ đó đạt tốc độ, gia tốc và độ cứng vững vượt trội; bù lại, không gian làm việc sẽ hẹp hơn.

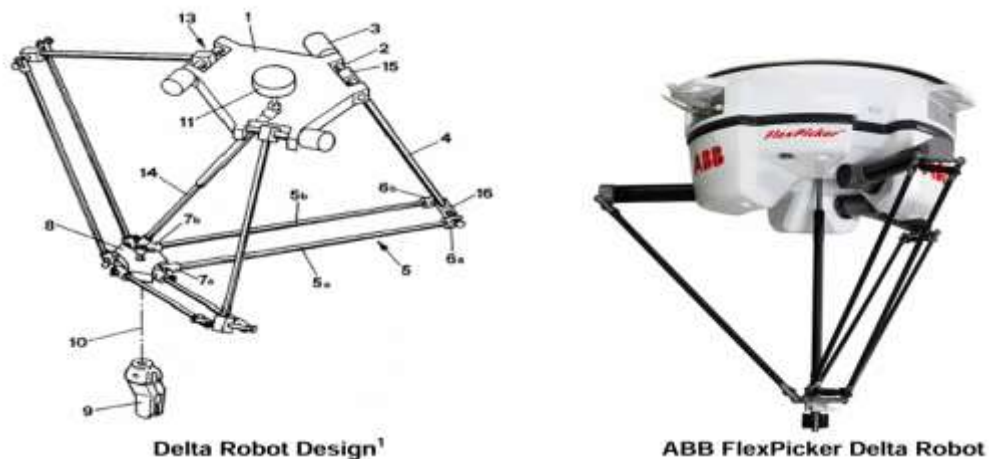
### 2.2.2. Giới thiệu Robot Delta

Robot song song kiểu Delta (từ đây trở về sau gọi tắt là robot Delta) lần đầu tiên được giáo sư Raymond Clavel phát minh vào năm 1985 [1]. Đến nay, rất nhiều hãng sản xuất robot nổi tiếng đã chế tạo thành công và đưa vào ứng dụng thực tiễn như Fanuc, ABB, Bosch Packaging...

Robot Delta sử dụng các cơ cấu hình bình hành (parallelogram) và một tấm đế di động (moving platform) có ba bậc tự do tịnh tiến so với tấm đế cố định (base).

Do tính ưu việt của Robot song song nên ngày càng thu hút được nhiều nhà khoa học nghiên cứu, đồng thời cũng được ứng dụng ngày càng rộng rãi vào nhiều lĩnh vực khác nhau như ngành vật lý, cơ khí, y tế, quân sự..

Với các yêu cầu đặc thù của đề tài "Nghiên cứu và Phát triển Hệ thống Thị giác Máy tính Tích hợp Robot để Loại bỏ Tạp chất trong Sản xuất Sợi," yếu tố tốc độ là quan trọng hàng đầu để xử lý các tạp chất nhỏ, nhẹ trên dây chuyền đang hoạt động. Do đó, với ưu thế tuyệt đối về tốc độ và gia tốc, robot delta chính là giải pháp tối ưu để đáp ứng hiệu quả bài toán này.



Hình 2.2 Một số mô hình robot delta

### 2.2.3. Phân loại

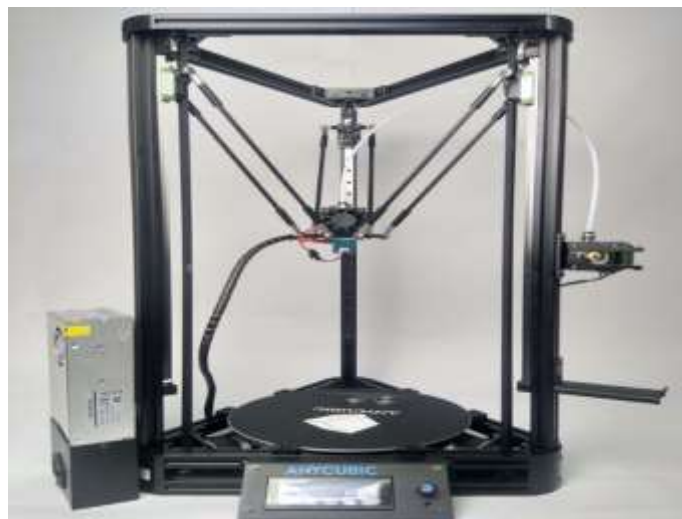
Robot Delta kiểu ba khớp xoay: Đây là một loại robot song song gồm ba tay đòn được điều khiển bởi ba động cơ quay gắn cố định trên khung. Mỗi tay đòn kết nối với

nền di động thông qua các khớp cầu và thanh nối, tạo ra chuyển động chính xác theo ba trục không gian (X, Y, Z). Nhờ cấu trúc nhẹ, cứng vững và khả năng phản hồi nhanh, robot này rất phù hợp cho các ứng dụng cần tốc độ và độ chính xác cao như gấp – đặt, phân loại và lắp ráp trong các ngành công nghiệp điện tử, thực phẩm, bao bì và dệt may.



Hình 2.3 Robot Delta kiểu ba khớp xoay

Robot Delta kiểu ba khớp trượt: Đây là robot song song gồm ba cơ cấu truyền động tuyến tính (khớp trượt) đặt thẳng đứng, điều khiển ba tay đòn gắn với nền di động thông qua các khớp nối. Khi các khớp trượt di chuyển lên xuống, nền di động dịch chuyển theo không gian 3 chiều (X, Y, Z). Cấu trúc này giúp robot đạt được độ chính xác cao, phản ứng nhanh và dễ điều khiển. Robot Delta ba khớp trượt thường được ứng dụng trong các nhiệm vụ gấp – đặt, phân loại sản phẩm và thao tác tự động tốc độ cao trong công nghiệp.



Hình 2-4:

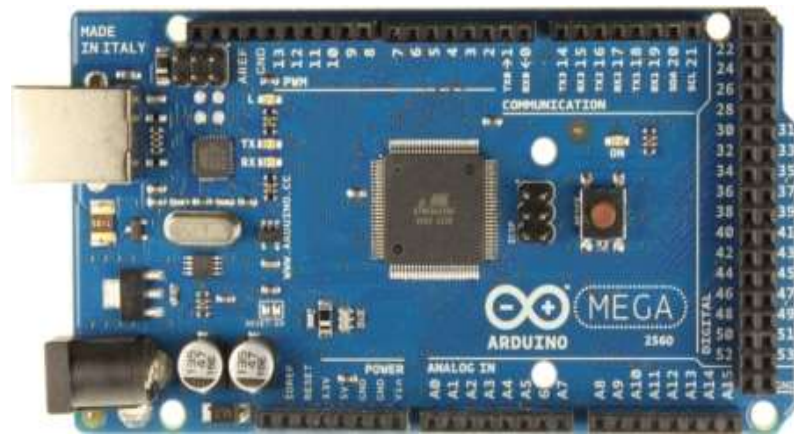
Hình 2.4 Robot Delta kiểu ba khớp trượt

## 2.3. Giới thiệu chung về các thiết bị

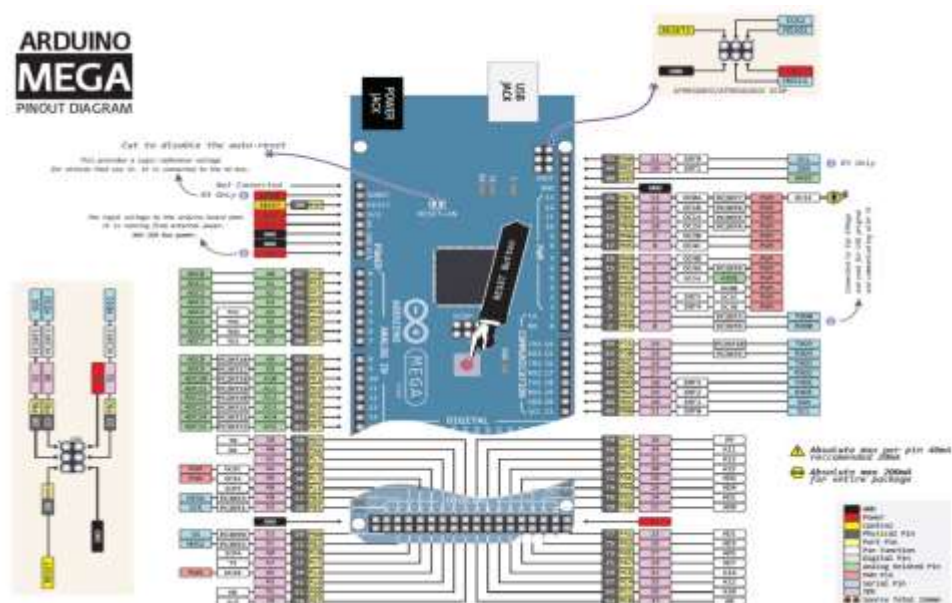
### 2.3.1. Giới thiệu chung về Arduino

Arduino là một board mạch vi điều khiển do một nhóm giáo sư và sinh viên nước Ý thiết kế và đưa ra sử dụng đầu tiên vào năm 2005. Arduino có thể thực hiện nhiều nhiệm vụ như lấy tín hiệu từ cảm biến đến điều khiển đèn, động cơ, và nhiều đối tượng khác. Ngoài ra, Arduino còn có khả năng kết nối, tương tác với các thiết bị GPS, Internet, Camera, điện thoại di động, ...

Arduino Mega 2560 R3 là phiên bản nâng cấp của Arduino Uno R3 với việc gia tăng số chân giao tiếp, ngoại vi và bộ nhớ. Mạch được thiết kế để giải quyết nhiều bài toán học búa, cần điều khiển nhiều loại động cơ và xử lý song song nhiều luồng dữ liệu số cũng như tương tự.



Hình 2.5 Arduino Mega 2560



Hình 2.6 Sơ đồ chân Arduino Mega 2560

Sơ đồ linh kiện Arduino Mega có 5 Chân GND, 3 chân 5V, 1 chân 3.3V, 1 nút reset, 16 chân analog, 4 chân UART, 54 Chân digital trong đó có 15 chân chúng ta có thể sử dụng như PWM, 6 chân lập trình ISP và nhiều thành phần khác...

### **2.3.2. Giới thiệu chung về động cơ bước và bộ điều khiển động cơ**

#### **2.3.2.1. Giới thiệu về động cơ bước**

Động cơ bước (còn gọi là Step Motor hay Stepping Motor) là một loại động cơ sử dụng điện. Động cơ bước quay theo từng bước một tùy theo số xung được cấp vào sẽ tạo thành một góc nào đó gọi là góc bước, nên có độ chính xác rất cao, đặc biệt là về mặt điều khiển. Nếu góc bước càng nhỏ thì số bước trên mỗi vòng quay của động cơ càng lớn và độ chính xác của vị trí càng lớn.

Các góc bước của động cơ thường được sử dụng phổ biến nhất là: góc 1,8 độ, góc 2,5 độ, góc 7,5 độ, góc 15 độ. Vì đây là các góc thuận lợi cho việc tính toán nhất.



Hình 2.7 Một số loại động cơ bước

Hiện nay, có 4 phương pháp để điều khiển động cơ bước được sử dụng phổ biến nhất, đó là:

- Điều khiển động cơ bước dạng sóng (Wave).
- Điều khiển động cơ bước đủ (Full step).
- Điều khiển động cơ nửa bước (Half step).
- Điều khiển động cơ vi bước (Microstep).

#### **2.3.2.2. Bộ điều khiển động cơ**

Bộ điều khiển động cơ bước (driver) là thiết bị không thể thiếu để vận hành động cơ bước. Trong đề tài này, nhóm chọn sử dụng driver TB6600 nhờ thiết kế nhỏ gọn, dễ dàng lắp đặt và khả năng đáp ứng tốt yêu cầu điều khiển xung trong hệ thống. Đây là loại driver phổ biến trong các ứng dụng như máy in 3D, máy CNC và robot. Driver TB6600 có khả năng điều khiển động cơ với điện áp từ 9V đến 42V, dòng điện tối đa lên đến 4A, đảm bảo độ chính xác và ổn định trong chuyển động – yếu tố quan trọng để điều khiển robot Delta hoạt động chính xác và hiệu quả



Hình 2.8 Driver TB 6600

## 2.4 Yêu cầu thiết kế phần cứng

Mục tiêu ban đầu của hệ thống mà nhóm đề ra là xây dựng một mô hình phát hiện tạp chất lẫn trong sợi bông. Dựa vào mô hình đã được thiết kế trên Solidworks, nhóm đã tiến hành gia công và lắp ráp các bộ phận robot lại với nhau. Dưới đây là một số yêu cầu khi thiết kế mô hình Robot Delta loại bỏ tạp chất:

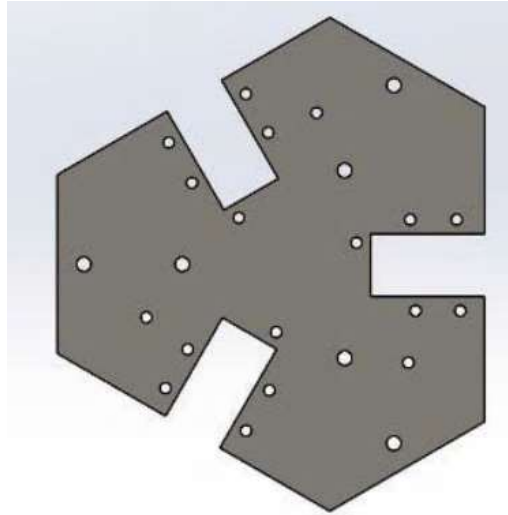
- Thiết kế mô hình Robot Delta hoạt động trong không gian làm việc của khung 800x350x350 (mm) chính xác, ổn định và đáp ứng được yêu cầu phân loại.
- Bố trí các thiết bị hợp lý cho ứng dụng phát hiện để loại bỏ tạp chất
- Lắp đặt tủ điện một cách hợp lý và mang tính thẩm mỹ, đảm bảo tính an toàn và thuận tiện trong việc sửa chữa, thay đổi thiết bị.

## 2.5 Thiết kế mô hình trên phần mềm Solidworks

Nhóm tham khảo các mô hình Robot Delta tiêu biểu và đề tài tốt nghiệp tại Đại Học Bách Khoa Đà Nẵng để xác định các thông số kỹ thuật, đảm bảo tốc độ và độ chính xác cao cho hệ thống phát hiện – loại bỏ tạp chất. Với yêu cầu cao về tốc độ và độ chính xác trong quá trình vận hành, robot đóng vai trò then chốt trong hệ thống tự động hóa, nhóm đã thiết kế tỉ mỉ gồm các thành phần: đế cố định, đế di động, khớp trên, khớp hình bình hành, khung robot và chi tiết phụ trợ, phối hợp hài hòa nhằm mang lại vận hành ổn định, chính xác và bền bỉ.

### 2.5.1 Đế cố định Robot

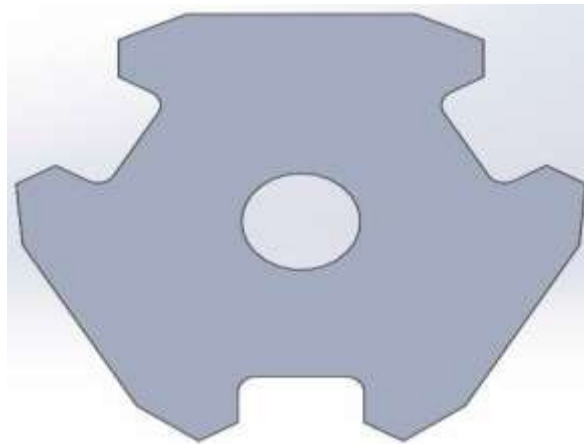
Đế cố định đóng vai trò là bộ phận liên kết giữa phần khung và ba động cơ, mỗi động cơ điều khiển một chuỗi động học của Robot Delta. Khi robot hoạt động, tải trọng của vật thể sẽ được phân bố đều lên ba cánh tay, đồng thời đế cũng phải chịu thêm trọng lượng từ các khớp nối.



Hình 2.9 Đế cố định robot

### 2.5.2 Đế di động Robot

Đế di động là bộ phận gắn với cơ cấu chấp hành (gripper) để thực hiện thao tác gắp và phân loại vật thể. Do đối tượng cần phân loại trong đề tài chủ yếu là các loại tạp chất nhỏ và nhẹ, nên yêu cầu thiết kế của đế di động không cần quá cứng mà ưu tiên trọng lượng nhẹ để giảm tải cho toàn bộ hệ thống robot.



Hình 0-10:

Hình 2.10 Đế di động robot

### 2.5.3 Khớp trên Robot

Khớp trên của robot đóng vai trò là khâu truyền động, được điều khiển trực tiếp từ động cơ thông qua hệ thống dây đai. Do đảm nhiệm chức năng truyền động, khớp này cần có khối lượng nhẹ nhằm giảm mô-men quán tính, giúp robot vận hành nhanh và chính xác hơn. Đồng thời, khớp cũng phải đảm bảo đủ độ cứng để hạn chế độ đàn hồi và rung lắc trong quá trình phân loại.



Hình 2.11 Khớp trên robot

#### 2.5.4 Khớp hình bình hành Robot

Khớp hình bình hành là chi tiết dùng để liên kết giữa cánh tay robot và đế di động, thông qua vòng bi mắt trâu loại SI8T. Do các thanh trục trong khớp là khâu thụ động, phải chịu lực truyền từ cả cánh tay robot và đế di động, nên cần được thiết kế chắc chắn. Nhóm đã lựa chọn vật liệu inox để gia công nhằm đảm bảo độ cứng vững, đồng thời giúp chi tiết có khả năng chống gỉ tốt khi làm việc trong môi trường bên ngoài.



Hình 2.12 Khớp hình bình hành robot

#### 2.5.5 Khung Robot

Robot Delta được lắp đặt trên một khung đỡ, vì vậy việc thiết kế khung cần đáp ứng các yêu cầu về độ cứng vững, sự ổn định và cân bằng nhằm đảm bảo hoạt động trơn tru của robot trong suốt quá trình thực hiện phân loại. Nhóm đã thiết kế khung có kích thước 800x350x350 (mm) và sử dụng chất liệu là nhôm định hình.

Tay gắp được nhóm lựa chọn và in 3D từ mẫu có sẵn, đảm bảo phù hợp với yêu cầu của bài toán đề ra.



Hình 2.13 Khung robot

## 2.6. Một số chi tiết khác

Để hoàn thiện mô hình Robot Delta, ngoài các chi tiết chính: đế cố định, khớp trên, khớp hình bình hành và đế di động, còn có một số linh kiện khác được liệt kê trong bảng bên dưới.

Bảng 2.1 Các chi tiết khác khi thi công robot

TT	Tên chi tiết	Hình ảnh	Thông tin chi tiết
1	Vòng bi mắt trâu SI8T		Ren trong M8, trục 8mm
2	Giá đỡ động cơ		In 3D Kích thước: 11x7 cm
3	Puly khớp trên robot		In 3D Đường kính : 3 cm Số răng: 180
4	Puly gắn với động cơ		In 3D Đường kính: 2cm Số răng: 32
5	Dây đai		Độ dài: 26 cm Số răng: 130

## 2.7. Lựa chọn thiết bị

### 2.7.1. Động cơ

Động cơ bước 57HS56 là loại động cơ bước 2 pha, được thiết kế với kích thước nhỏ gọn nhưng vẫn đảm bảo hiệu suất vận hành mạnh mẽ. Động cơ này được tích hợp hộp số giảm tốc Nema 17, giúp tăng lực kéo và độ chính xác. Nhờ thiết kế nhỏ gọn, độ bền cao và khả năng điều khiển chính xác, nó phù hợp cho các ứng dụng tự động hóa như Robot Delta.





Bảng 1.2 Động cơ bước 57HS56





Động cơ bước		Điện áp: 12 - 24V Dòng điện: 2.5 - 3A Số dây: 6 dây) Loại: 2 pha
--------------	--	---

### 2.7.2. Danh sách các thiết bị

Hệ thống sử dụng cầu dao tự động (CB) để ngắt mạch khi có sự cố, bảo vệ thiết bị và tăng độ bền. Nguồn chính là 24V–5A, kết hợp module hạ áp LM2596 để cấp điện ổn định cho các linh kiện như động cơ bước và servo mini. Bộ điều khiển trung tâm là Arduino Mega 2560, kết nối với driver TB6600 để điều khiển chính xác các động cơ bước. Hệ thống tích hợp công tắc hành trình giúp robot tự động về vị trí gốc, tránh vượt giới hạn chuyển động. Động cơ servo được lắp tại tay gắp để thực hiện thao tác kẹp/mở


Bảng 2.2 Danh sách các thiết bị trong hệ thống

STT	Tên thiết bị	Hình ảnh	Thông tin chi tiết
1	Arduino Mega 2560		Vi điều khiển AVR ATmega 2560 (8bit). Nguồn cung cấp 7-12V. Số chân I/O số 54. Số chân I/O tương tự 16.
2	Mạch Hạ Áp LM2596		IC LM2596. Nguồn đầu vào từ 4V-35V. Nguồn đầu ra điều chỉnh được trong khoảng từ 1V đến 30V. Dòng ra max 3A.
3	Nguồn 24V-5A		Input AC: 110/220VAC +/- 15% Output DC: 24V 5A
4	Mạch điều khiển động cơ bước TB6600		Nguồn đầu vào là 9V – 42V. Dòng cấp tối đa là 4A. Ngõ vào có cách ly quang, tốc độ cao.

			Có tích hợp đo quá dòng quá áp.
5	CB(Aptomat)		Tự động ngắt mạch điện hơn trong các trường hợp xảy ra quá tải giúp ngăn ngừa và giảm thiểu tác hại tới hệ thống
6	Nút nhấn		Sử dụng để điều khiển ON/OFF hệ thống
7	Động cơ Servo mini		Điện áp hoạt động: 4.8 ~ 6VDC Mô men xoắn dừng: 1,8kg/cm(4,8V), 2,2kg/cm(6V) Tốc độ hoạt động: 0,1 giây/60 độ(4,8V), 0,08 giây/60 độ(6V)
8	Công tắc hành trình V-156-1C25		Điện áp định mức 250V, dòng ma 15A

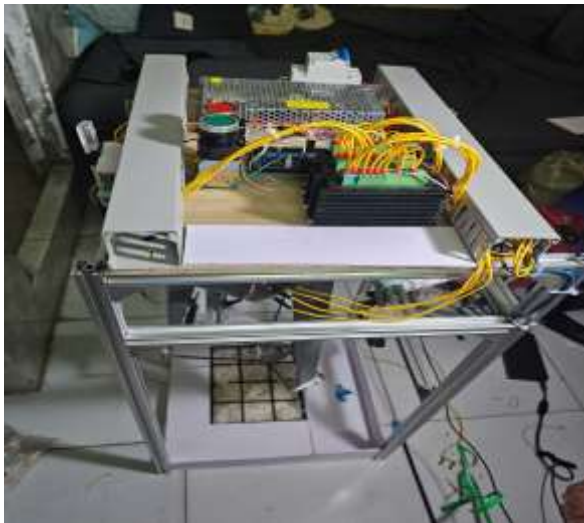
### 2.7.3. Camera

Bảng 2.4: Web cam máy tính

Hình ảnh	Thông tin	
	Tên thiết bị	Webcam máy tính
	Độ phân giải	1920x1080
	Tốc độ khung hình	20 FPS
	Góc xoay	60 độ
	Cảm biến hình ảnh	CMOS 1/2.7
	Kích thước	8 * 3 * 11cm

## 2.8. Thi công phần cứng

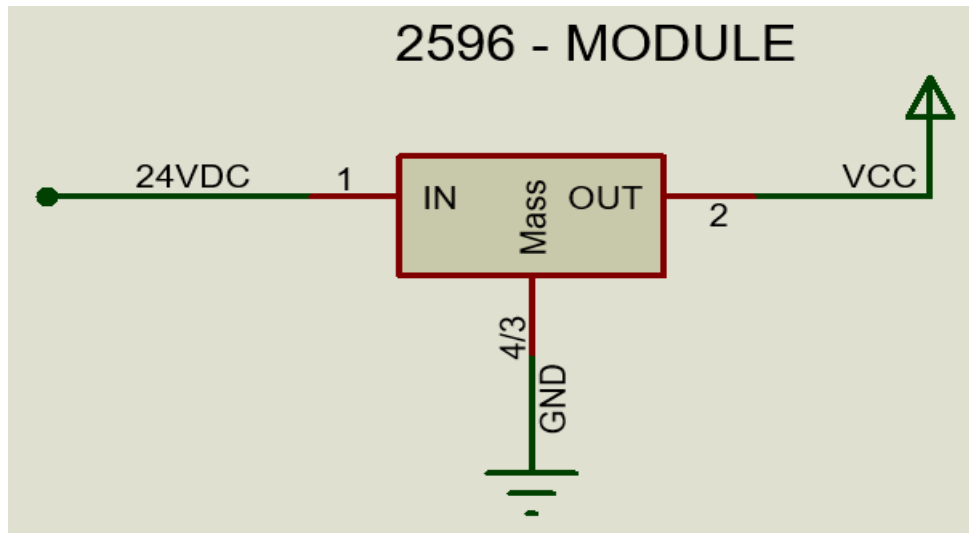
### 2.8.1. Mô hình thực tế sau khi thi công



Hình 5 Phần cứng sau thi công và tay gắp

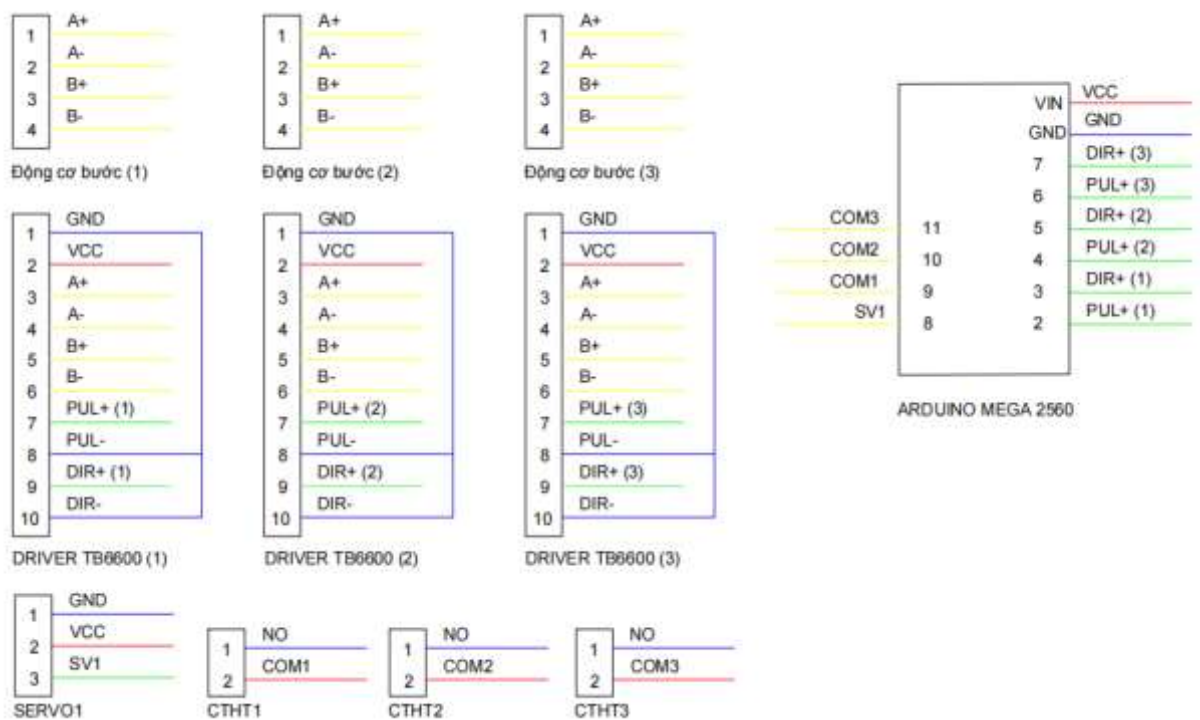
### 2.8.2. Sơ đồ hệ thống

Nguồn cấp cho hệ thống: Từ nguồn điện xoay chiều 220VAC, nhóm em sử dụng nguồn tổ ong 24V-5A để hạ áp xuống cấp nguồn cho động cơ. Nhóm em sử dụng mạch LM2596 để hạ áp xuống 5V cấp nguồn cho Arduino mega 2560 và động cơ servo



Hình 2.14 Module MLS2596

Sơ đồ đi dây tín hiệu điều khiển từ Arduino:



Hình 2.15 Sơ đồ đi dây tín hiệu điều khiển

### 2.8.3. Thi công tủ điện



Hình 2.16 Hệ thống điện sau khi hoàn tất đấu nối

#### **Kết luận chương 2:**

Chương 2 tập trung vào thiết kế và triển khai phần cứng cho hệ thống tự động loại bỏ tạp chất. Trước hết, cơ cấu robot Delta được mô hình hóa và thiết kế chi tiết trên SolidWorks, dựa trên lý thuyết cấu trúc song song để đảm bảo quán tính thấp và độ cứng cao. Tiếp đó, chương tích hợp các nội dung về động cơ bước và driver TB6600, vi điều khiển Arduino Mega 2560, cũng như lựa chọn camera công nghiệp phù hợp, từ đó xây dựng hoàn chỉnh khối điều khiển và khối chấp hành sẵn sàng cho giai đoạn hiệu chỉnh và tích hợp phần mềm.

### Chương 3: CƠ SỞ LÝ THUYẾT VỀ ROBOT DELTA

Trong chương này trình bày về các thông số của Robot Delta, tính toán động học thuận nghịch và tìm không gian làm việc trên phần mềm Matlab.

#### 3.1. Các thông số của Robot Delta

Trước khi tính toán động học ta phải xác định 1 số thông số cố định của Robot Delta bao gồm:  $f$ ,  $e$ ,  $r_f$ ,  $r_e$ . Đây là 4 thông số chính của các bộ phận Robot Delta.

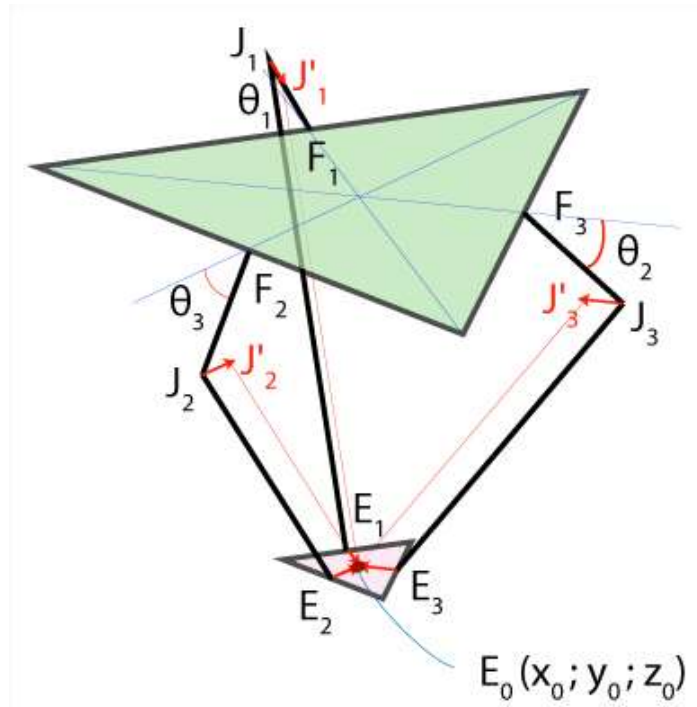
Bảng 3.1 Thông số các chi tiết Robot

Tên bộ phận	Ký hiệu	Kích thước (mm)
Cạnh tam giác đế cố định	$f$	248
Cạnh tam giác đế di động	$e$	58
Khớp trên	$r_f$	310
Khớp hình bình hành	$r_e$	380

#### 3.2. Tính toán động học cho Robot Delta

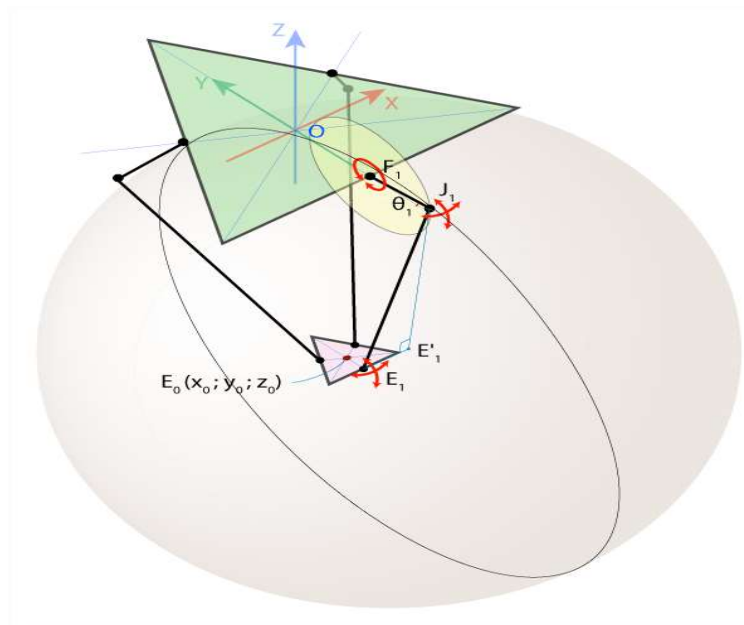
##### 3.2.1. Động học nghịch

Với bài toán động học nghịch chúng ta sẽ biết được tọa độ điểm cuối cùng của Robot  $E_0(x_0, y_0, z_0)$  là trọng tâm của tam giác đế di động và các thông số cần xác định là 3 góc  $\theta$  được tạo bởi các khớp trên và mặt phẳng của đế cố định [2]. Với hệ quy chiếu sẽ được chọn có gốc tọa là trọng tâm của tam giác đế cố định.



Hình 3.1 Mô hình động học robot delta

Do khớp nối thiết kế của robot  $F_1J_1$  chỉ có thể xoay trong mặt phẳng  $YZ$ , tạo thành hình tròn với tâm tại điểm  $F_1$  có bán kính  $r_f$ , đồng thời khớp hình bình hành  $J_1E_1$  có thể xoay tự do tương đối với  $E_1$ , tạo thành hình cầu với tâm tại điểm  $E_1$  và bán kính  $r_e$ .

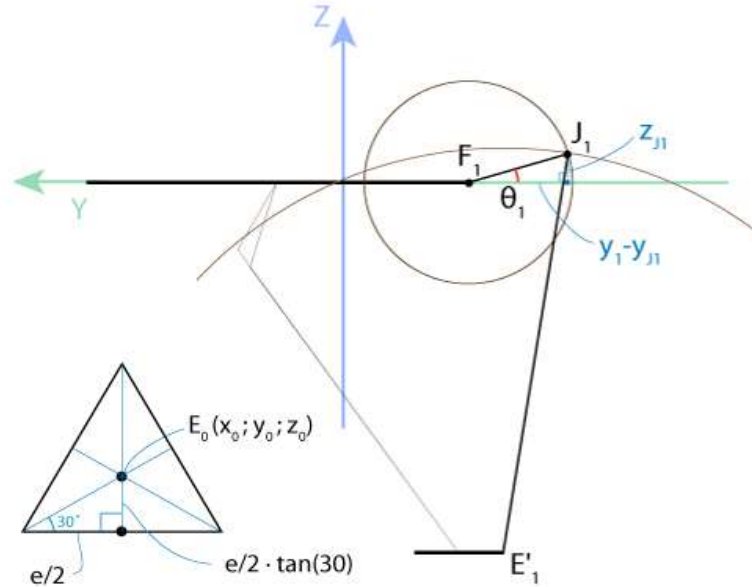


Hình 3.2 Giao tuyến của hình cầu và mặt phẳng YZ

Giao tuyến của hình cầu này và mặt phẳng  $YZ$  là một đường tròn có tâm tại điểm  $E'_1$  và bán kính  $E'_1J_1$  (trong đó  $E'_1$  là phép chiếu của điểm  $E_1$  trên mặt phẳng  $YZ$ ). Điểm  $J_1$  có thể được tìm thấy bây giờ như là giao điểm của các đường tròn bán

kính đã biết với tâm là  $E'_1$  và  $F_1$  (chúng ta chỉ nên chọn một điểm giao nhau với tọa độ Y nhỏ hơn). Và nếu ta biết  $J_1$ , ta có thể tính được góc  $\theta_1$ .

Xét mặt phẳng YZ:



Hình 3.3 Mặt phẳng YZ

Xét tam giác để di động, ta tìm được:

$$EE_1 = \frac{e}{2} \tan 30^\circ = \frac{e}{2\sqrt{3}}$$

Nên

$$E_1 \left( x_0; y_0 - \frac{e}{2\sqrt{3}}; z_0 \right) \Rightarrow E'_1 \left( 0; y_0 - \frac{e}{2\sqrt{3}}; z_0 \right)$$

Và  $EE_1 = x_0 \rightarrow E'_1 J_1 = \sqrt{E_1 J_1^2 - E_1 E_1'^2} = \sqrt{r_e^2 - x_0^2}$

Xét tam giác để cố định, ta tìm được:

$$OF_1 = \frac{f}{2} \tan 30^\circ = \frac{f}{2\sqrt{3}}$$

Nên:

$$F_1 \left( 0; -\frac{f}{2\sqrt{3}}; 0 \right)$$

Áp dụng định lý Pytago cho 2 tam giác vuông  $F_1 J_1 J_1'$  và  $J_1 E_1 E_1'$ :

$$\begin{cases} (y_{J_1} - y_{F_1})^2 + (z_{J_1} - z_{F_1})^2 = r_f^2 \\ (y_{J_1} - y_{E_1'})^2 + (z_{J_1} - z_{E_1'})^2 = r_e^2 - x_0^2 \end{cases}$$

Ta được:

Capstone Project: “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”

$$\begin{cases} (y_{J_1} + \frac{f}{2\sqrt{3}})^2 + z_{J_1}^2 = r_f^2 \\ (y_{J_1} - y_0 + \frac{e}{2\sqrt{3}})^2 + (z_{J_1} - z_0)^2 = r_e^2 - x_0^2 \end{cases}$$

Tọa độ điểm  $J_1$  là nghiệm hệ phương trình giao điểm của 2 đường tròn:

$$\Rightarrow J_1(0; y_{J_1}; z_{J_1})$$

Từ lấy phương trình thứ nhất trừ phương trình thứ hai ta thu được:

$$2m_1y_{J_1} + 2z_0z_{J_1} + m_2 = 0$$

Trong đó:

$$m_1 = \frac{f}{2\sqrt{3}} + y_0 - \frac{e}{2\sqrt{3}} \quad m_2 = \left(\frac{f}{2\sqrt{3}}\right)^2 - \left(y_0 - \frac{e}{2\sqrt{3}}\right)^2 - z_0^2 - r_f^2 + (r_e^2 - x_0^2)$$

Từ biến đổi ta được:

$$z_{J_1} = \frac{-2y_{J_1}m_1 - m_2}{2z_0}$$

Thay vào phương trình đầu của ta được:

$$\left(1 + \frac{m_1^2}{z_0^2}\right)y_{J_1}^2 + \frac{f}{\sqrt{3}} + \frac{m_1m_2}{z_0^2}y_{J_1} - \left(\frac{f}{2\sqrt{3}}\right)^2 + \left(\frac{m_2}{2z_0}\right)^2 - r_f^2 = 0$$

Đặt:

$$\Delta = b^2 - 4ac$$

Với:

$$\begin{cases} a = 1 + \frac{m_1^2}{z_0^2} \\ b = \frac{f}{\sqrt{3}} + \frac{m_1m_2}{z_0^2} \\ c = \left(\frac{f}{2\sqrt{3}}\right)^2 + \left(\frac{m_2}{2z_0}\right)^2 - r_f^2 \end{cases}$$

Vì điểm  $J_1$  luôn ở vị trí âm của trục y nên ta chọn nghiệm:

$$y_{J_1} = \frac{-b - \sqrt{\Delta}}{2a}$$

Thay nghiệm phương trình vào phương trình trên ta được:

$$z_{J_1} = \frac{-2 \left( y_{J_1} = \frac{-b - \sqrt{\Delta}}{2a} \right) m_1 - m_2}{2z_0}$$

Góc  $\theta_1$  là góc hợp bởi khớp trên  $F_1 J_1$  và mặt phẳng đế cố định.

Ta có:

$$J_1(0; y_{J_1}; z_{J_1})$$

Ta tìm được:

$$\theta_1 = a \tan 2 \left( \frac{z_{J_1}}{y_{F_1} - y_{J_1}} \right) \quad (3.2.1.a)$$

Khi đã tính được góc  $\theta_1$  nhóm lợi dụng tính đối xứng của Robot Delta để tính toán góc  $\theta_2$  và  $\theta_3$ . Bằng cách xoay hệ trục xyz  $120^\circ$ . với tọa độ điểm cuối, cùng của Robot mới  $E'_0(x'_0, y'_0, z'_0)$  và sử dụng cùng những thuật toán ở trên để tìm góc  $\theta_2$ . Và để tìm góc  $\theta_3$  chỉ cần xoay hệ trục xyz 1 góc  $-120^\circ$ .

Nhóm đặt hệ tọa độ ban đầu là A thì điểm  $E_0$ , nhìn từ hệ tọa độ A sẽ là  $A_{E_0} = (x_0, y_0, z_0)$ . Khi xoay từ hệ trục tọa độ A một góc theo trục z thì sẽ nhận được hệ tọa độ mới B với tọa độ điểm cuối cùng  $E_0$  nhìn từ hệ tọa độ  $B_{E_0} = (x'_0, y'_0, z'_0)$ . Ta có công thức của phép xoay từ hệ tọa độ A qua hệ tọa độ B như sau:

$$A_{E_0} = {}^A R \cdot B_{E_0}$$

Để tìm được  $B_{E_0} = (X'_0, Y'_0, Z'_0)$ , ta cần nhân 2 vế của phương trình cho  $({}^A R)^T$

$$({}^A R)^T \cdot A_{E_0} = ({}^A R)^T \cdot {}^A R \cdot B_{E_0}$$

Với

$$({}^A R)^T \cdot {}^A R = 1$$

Từ suy ra được:

$$B_{E_0} = ({}^A R)^T \cdot A_{E_0}$$

Bằng cách xoay từ hệ tọa độ A sang hệ tọa độ B quanh trục z với góc  $120^\circ$  ta suy được phương trình:

$$\left( \begin{matrix} A \\ B \end{matrix} R \right)^T = (R_z(120^\circ))^T = \begin{bmatrix} \cos(120^\circ) & \sin(120^\circ) & 0 \\ -\sin(120^\circ) & \cos(120^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

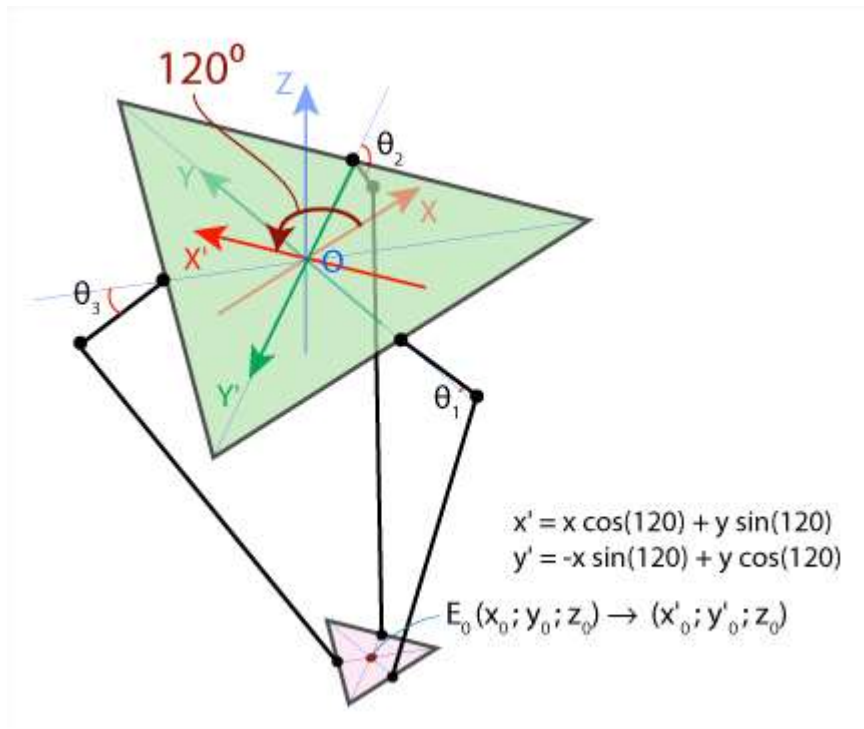
Ngoài ra, ta có ma trận chuyển đổi hệ tọa độ tổng quát từ hệ tọa độ A sang hệ tọa độ B. Với lần lượt là phép xoay quanh trục z 1 góc  $120^\circ$  và vì chỉ xoay tại chỗ mà không tịnh tiến. Từ phương trình (4.28) suy ra phương trình như sau:

$B_{E_0} = (R_z(120^\circ))^T \cdot A_{E_0}$  Tiếp tục thay phương trình vào phương trình ta được phương trình:

$$B_{E_0} = (R_z(120^\circ))^T \cdot A_{E_0} = \begin{bmatrix} x_0 \cos(120)^\circ + y_0 \sin(120)^\circ \\ -x_0 \sin(120)^\circ + y_0 \cos(120)^\circ \\ z_0 \\ 1 \end{bmatrix}$$

Vậy nhóm đã tìm được hệ phương trình để tính góc  $\theta_2$  khi xoay quanh hệ trục tọa độ 1 góc  $120^\circ$  theo trục z:

$$\begin{cases} x_0' = x_0 \cos(120)^\circ + y_0 \sin(120)^\circ \\ y_0' = -x_0 \sin(120)^\circ + y_0 \cos(120)^\circ \\ z_0' = z_0 \end{cases} \quad (3.2.1.b)$$



Hình 3.4 Hệ tọa độ mới  $(x'_0, y'_0, z'_0)$

Ngược lại với góc  $\theta_2$ , tiến hành xoay hệ tọa độ A quanh trục z 1 góc  $-120^\circ$  tính toán tương tự cho góc  $\theta_3$ , ta sẽ thu được hệ phương trình sau:

$$\begin{cases} x_0' = x_0 \cos(120)^\circ - y_0 \sin(120)^\circ \\ y_0' = x_0 \sin(120)^\circ + y_0 \cos(120)^\circ \\ z_0' = z_0 \end{cases} \quad (3.2.1.c)$$

### 3.2.2. Động học thuận

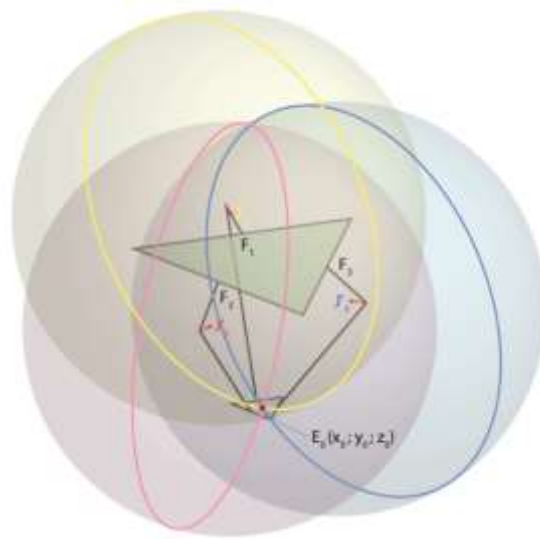
Mục đích của bài toán động học thuận là xác định vị trí của khâu tác động cuối của robot khi biết các biến khớp của robot [2]. Từ các góc là trọng tâm của tam giác để di động.

Từ các góc  $\theta_1, \theta_2, \theta_3 \rightarrow$  Tọa độ  $(x_0, y_0, z_0)$  của điểm  $E_0$ .

Nếu ta đã biết góc  $\theta_x$ , chúng ta có thể dễ dàng tìm thấy tọa độ của các điểm  $J_1, J_2$  và  $J_3$ . Các khớp nối  $J_1E_1, J_2E_2$  và  $J_3E_3$  có thể tự do xoay quanh các điểm  $J_1, J_2$  và  $J_3$ . tương ứng, tạo thành ba hình cầu có bán kính  $r_e$ .

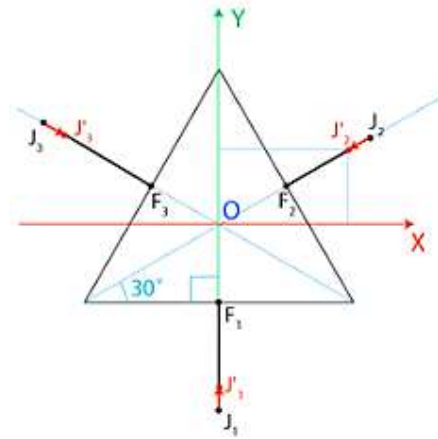
Ta di chuyển các tâm của các hình cầu từ các điểm  $J_1, J_2$  và  $J_3$  đến các điểm  $J'_1, J'_2$  và  $J'_3$  bằng cách sử dụng các vector chuyển tiếp  $E_1E_0, E_2E_0$  và  $E_3E_0$  tương ứng.

Sau quá trình chuyển đổi này, tất cả ba hình cầu sẽ giao nhau tại một điểm:  $E_0$ , như được thể hiện trong hình phía dưới:



Hình 3.5 Mô hình động học robot delta

Vì vậy, để tìm tọa độ  $(x_0, y_0, z_0)$  của điểm  $E_0$ , chúng ta cần giải quyết tập hợp ba phương trình như  $(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2 = r_e^2$ , nơi tọa độ các tâm hình cầu  $(x_j, y_j, z_j)$  và bán kính  $r_e$  đã biết.



Hình 3-6:

Hình 3.6 Tọa độ các điểm  $J'_1, J'_2, J'_3$  trên mặt phẳng  $Oxy$

Theo hình học:

$$OF_1 = OF_2 = OF_3 = \frac{f}{2} \tan(30^\circ) = \frac{f}{2\sqrt{3}}$$

Và

$$J_1J'_1 = J_2J'_2 = J_3J'_3 = \frac{e}{2} \tan(30^\circ) = \frac{e}{2\sqrt{3}}$$

$$\begin{cases} F_1J_1 = r_f \cos(\theta_1) \\ F_2J_2 = r_f \cos(\theta_2) \\ F_3J_3 = r_f \cos(\theta_3) \end{cases}$$

Từ

$$\begin{cases} J'_1 \left( 0; -\frac{f-e}{2\sqrt{3}} - r_f \cos \theta_1; -r_f \sin \theta_1 \right) \\ J'_2 \left( \left( \frac{f-e}{2\sqrt{3}} + r_f \cos \theta_2 \right) \cos 30^\circ; \left( \frac{f-e}{2\sqrt{3}} + r_f \cos \theta_2 \right) \sin 30^\circ; -r_f \sin \theta_2 \right) \\ J'_3 \left( \left( \frac{f-e}{2\sqrt{3}} + r_f \cos \theta_3 \right) \cos 30^\circ; \left( \frac{f-e}{2\sqrt{3}} + r_f \cos \theta_3 \right) \sin 30^\circ; -r_f \sin \theta_3 \right) \end{cases}$$

Trong các phương trình sau đây ta sẽ chỉ định tọa độ của các điểm  $J'_1, J'_2, J'_3$  là  $(x_1, y_1, z_1)$ ,  $(x_2, y_2, z_2)$  và  $(x_3, y_3, z_3)$ . Lưu ý rằng  $x_1 = 0$ . Dưới đây là phương trình của ba mặt cầu với giao điểm là  $E_0(x_0, y_0, z_0)$

$$\begin{cases} x^2 + (y - y_1)^2 + (z - z_1)^2 = r_e^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = r_e^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = r_e^2 \end{cases}$$

$$\Rightarrow \begin{cases} x^2 + y^2 + z^2 - 2y_1y - 2z_1z = r_e^2 - y_1^2 - z_1^2 & (1) \\ x^2 + y^2 + z^2 - 2x_2x - 2y_2y - 2z_2z = r_e^2 - x_2^2 - y_2^2 - z_2^2 & (2) \\ x^2 + y^2 + z^2 - 2x_3x - 2y_3y - 2z_3z = r_e^2 - x_3^2 - y_3^2 - z_3^2 & (3) \end{cases}$$

Đặt:  $w_i = x_i^2 + y_i^2 + z_i^2$ , ta được:

$$\begin{cases} x_2x + (y_1 - y_2)y + (z_1 - z_2)z = \frac{(w_1 - w_2)}{2} & (4) = (1) - (2) \\ x_3x + (y_1 - y_3)y + (z_1 - z_3)z = \frac{(w_1 - w_3)}{2} & (5) = (1) - (3) \\ (x_2 - x_3)x + (y_2 - y_3)y + (z_2 - z_3)z = \frac{(w_2 - w_3)}{2} & (6) = (2) - (3) \end{cases}$$

Từ (4) - (5):

$$x = a_1z + b_1 \quad (7)$$

$$y = a_2z + b_2 \quad (8)$$

Với:

$$a_1 = \frac{1}{d} [(z_2 - z_1)(y_3 - y_1) - (z_3 - z_1)(y_2 - y_1)]$$

$$b_1 = -\frac{1}{2d} [(w_2 - w_1)(y_3 - y_1) - (w_3 - w_1)(y_2 - y_1)]$$

$$a_2 = -\frac{1}{d} [(z_2 - z_1)x_3 - (z_3 - z_1)x_2]$$

$$b_2 = \frac{1}{2d} [(w_2 - w_1)x_3 - (w_3 - w_1)x_2]$$

$$d = (y_2 - y_1)x_3 - (y_3 - y_1)x_2$$

Thế (7), (8) vào (1) ta được:

$$(a_1^2 + a_2^2 + 1)z^2 + 2(a_1 + a_2(b_2 - y_1) - z_1)z + (b_1^2 + (b_2 - y_1)^2 + z_1^2 - r_e^2) = 0$$

Cuối cùng, ta cần giải phương trình này và tìm  $z_0$  (chúng ta nên chọn gốc phương trình âm nhỏ nhất), và sau đó tính  $x_0$  và  $y_0$  từ (7) và (8).

$$az^2 + bz + c = 0$$

Với:

$$a = a_1^2 + a_2^2 + 1$$

$$b = 2(a_1b_1 + a_2(b_2 - y_1) - z_1)$$

$$c = b_1^2 + (b_2 - y_1)^2 + z_1^2 - r_e^2$$

$$\text{Đặt } \Delta = b^2 - 4ac$$

Do mô hình Robot Delta có khâu cuối luôn ở vị trí âm của trục z nên ta lấy nghiệm âm:

$$y_0 = \frac{-b - \sqrt{\Delta}}{2a} \quad (3.2.2.a)$$

### 3.3 Động lực học Robot Delta

Hệ gồm  $p=7$  vật rắn, số bậc tự do  $f=3$ , số tọa độ suy rộng dư  $m=6$ ,  $r=3$  phương trình liên kết với ( $j = 1, \dots, r$ )

$$f_j = f(s, t) = f(q_1, q_2, q_3, \dots, q_m, t) = 0 \quad (3.3.a)$$

Phương trình Lagrange dạng nhân tử:

$$M(s)\ddot{s} + C(s, \dot{s})\dot{s} + g(s) + J_s^T(s)\lambda = Q^{np} \quad (3.3.b)$$

Trong đó:

$M(s)$  là ma trận khối lượng suy rộng  $6 \times 6$

$C(s, \dot{s})$  là ma trận quán tính và Coriolis  $6 \times 6$

$$C(s, \dot{s}) = \frac{dM(s)}{dt} - \frac{1}{2} \left( \frac{\partial(M(s)\dot{s})}{\partial s} \right)^T \quad (3.3.c)$$

$g(s)$  là vector do trọng trường gây ra  $6 \times 1$

$$g(s) = \left( \frac{\partial \pi}{\partial s} \right)^T \quad (3.3.d)$$

$J_s$  là ma trận Jacobian của tọa độ suy rộng đầy đủ  $3 \times 6$

$\lambda$  là véctơ các nhân tử Lagrange  $3 \times 1$

$Q^{np}$  là véctơ chứa lực suy rộng của các lực không thế  $6 \times 1$ .

Các mômen dẫn động đặt tại các khớp chủ động  $\vec{\tau}_1, \vec{\tau}_2, \vec{\tau}_3$

$$Q^{np} = [\tau_1 \ \tau_2 \ \tau_3 \ 0 \ 0 \ 0]^T \quad (3.3.e)$$

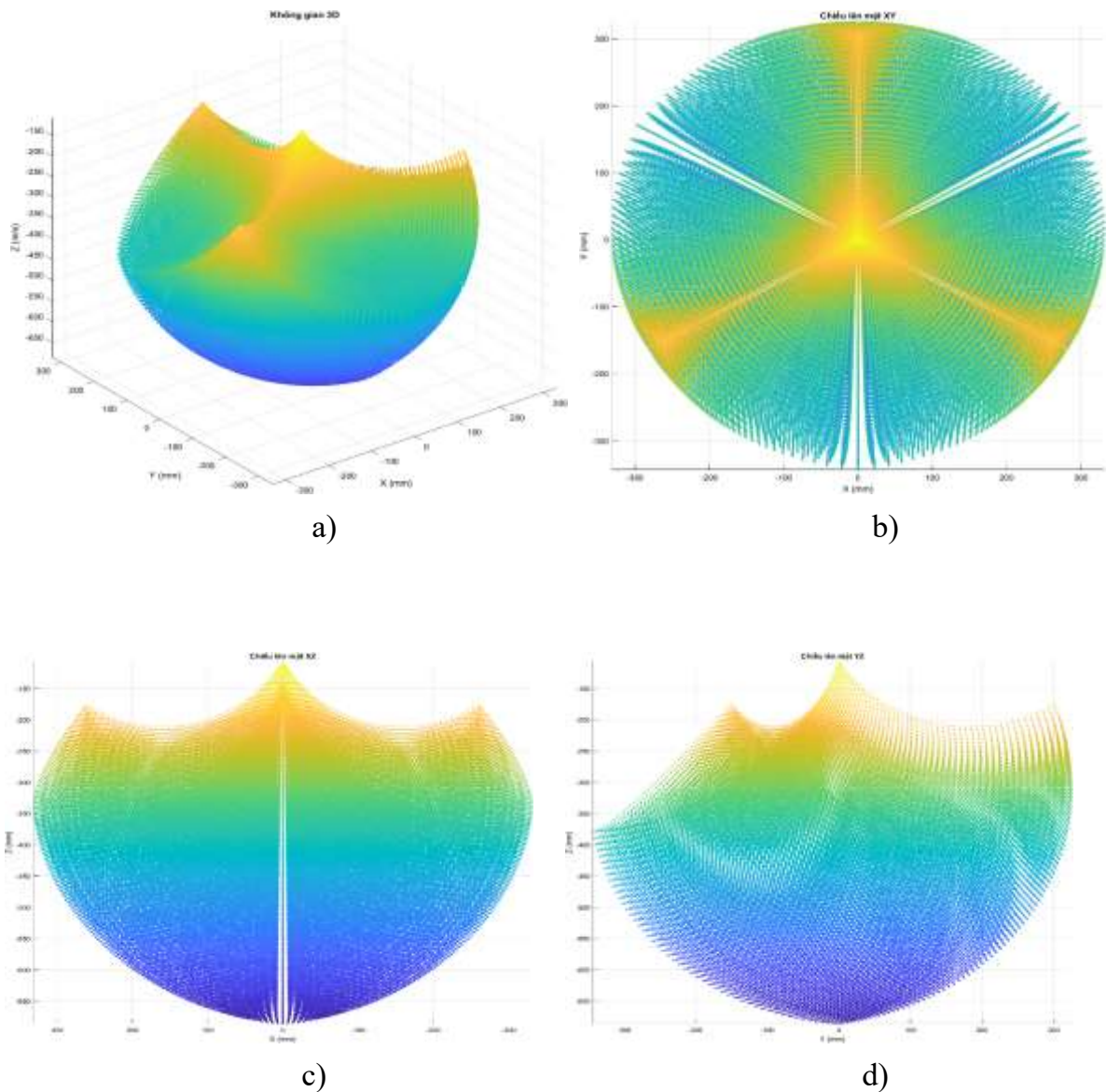
Bài toán động lực học ngược có thể được giải trực tiếp hoặc bằng cách biến đổi về các tọa độ suy rộng độc lập.

### 3.4. Không gian làm việc của Robot Delta

Không gian làm việc của Robot Delta là toàn bộ thể tích mà cơ cấu chấp hành cuối có thể hoạt động khi Robot thực hiện tất cả các chuyển động. Thể tích không gian làm việc phụ thuộc vào kết cấu của khớp trên và khớp hình bình hành của Robot. Bên cạnh đó, không gian làm việc cũng phụ thuộc vào miền giá trị của các biến khớp. Biết được vùng làm việc của Robot giúp nhóm dễ dàng và thuận tiện trong việc điều khiển và phân loại. Nhóm đã sử dụng phần mềm Matlab và kết quả của bài toán động học thuận, tiến hành cho các điểm tọa độ của cơ cấu chấp hành cuối chạy trong không gian theo các góc  $\theta_1, \theta_2, \theta_3$  đã giới hạn.

Trong trường hợp này, nhóm sinh viên đã giới hạn các góc  $\theta_1, \theta_2, \theta_3$  như sau:

$$0^\circ < \theta_1, \theta_2, \theta_3 < 90^\circ$$



Hình 3.7 Không gian làm việc của Robot Delta trong. a) Không gian 3 chiều xyz; Không gian 2 chiều b) xy; c) xz; d) yz

### Kết luận chương 3

Chương 3 trình bày quy trình thiết kế và triển khai thuật toán điều khiển chuyển động cho robot Delta, bắt đầu từ việc tính toán động học nghịch để xác định ba góc khớp tương ứng với tọa độ đầu chấp hành và động học thuận để ngược lại tính toán vị trí thực của cơ cấu gấp, tiếp đó sử dụng Matlab mô phỏng không gian làm việc nhằm đánh giá phạm vi hoạt động. Nhờ kết hợp chặt chẽ lý thuyết động học với các

*Capstone Project: “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”*

bước mô phỏng và thực thi phân cứng, thuật toán vị trí đạt được tính ổn định và chính xác, đáp ứng yêu cầu vận hành của robot Delta.

## Chương 4: XÂY DỰNG VÀ TRIỂN KHAI MÔ HÌNH THỊ GIÁC MÁY TÍNH

### 4.1. Tổng quan về thị giác máy tính, xử lý ảnh và các phần mềm

Thị giác máy tính và xử lý ảnh là hai lĩnh vực liên quan mật thiết với nhau, áp dụng một số kỹ thuật trí tuệ nhân tạo để lấy thông tin có ý nghĩa từ hình ảnh, video và một vài đầu vào trực quan khác.

#### 4.1.1. Giới thiệu về thị giác máy tính

Thị giác máy tính (tiếng Anh: computer vision) là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh và, nói chung là dữ liệu đa chiều từ thế giới thực để cho ra các thông tin số hoặc biểu tượng, ví dụ trong các dạng quyết định. Việc phát triển lĩnh vực này có bối cảnh từ việc sao chép các khả năng thị giác con người bởi sự nhận diện và hiểu biết một hình ảnh mang tính điện tử. Sự nhận diện hình ảnh có thể xem là việc giải quyết vấn đề của các biểu tượng thông tin từ dữ liệu hình ảnh qua cách dùng các mô hình được xây dựng với sự giúp đỡ của các ngành lý thuyết học, thống kê, vật lý và hình học. Thị giác máy tính cũng được mô tả là sự tổng thể của một dải rộng các quá trình tự động và tích hợp và các thể hiện cho các nhận thức thị giác.

Thị giác máy tính là một môn học khoa học liên quan đến lý thuyết đằng sau các hệ thống nhân tạo có trích xuất các thông tin từ các hình ảnh. Dữ liệu hình ảnh có thể nhiều dạng, chẳng hạn như chuỗi video, các cảnh từ đa camera, hay dữ liệu đa chiều từ máy quét y học. Thị giác máy tính còn là một môn học kỹ thuật, trong đó tìm kiếm việc áp dụng các mô hình và các lý thuyết cho việc xây dựng các hệ thống thị giác máy tính.

Các lĩnh vực con của thị giác máy tính bao gồm tái cấu trúc cảnh, dò tìm sự kiện, theo dõi video, nhận diện bố cục đối tượng, học, chỉ mục, đánh giá chuyển động và phục hồi ảnh [4].

#### 4.1.2. Giới thiệu về xử lý ảnh

Xử lý ảnh là đầu tàu trong các lĩnh vực 4.0 và là nền tảng của lĩnh vực thị giác máy tính. Xử lý ảnh là quá trình biến đổi từ một ảnh ban đầu sang ảnh khác tùy theo nhu cầu và mong muốn của người sử dụng. Các thuật toán xử lý hình ảnh được sử dụng để trích xuất thông tin từ hình ảnh, khôi phục và nâng cao dữ liệu hình ảnh và video.

Mục đích trong việc sử dụng xử lý ảnh bao gồm:

- Sử dụng một số kỹ thuật khác nhau để phân loại đối tượng.
- Phát triển các thuật toán trong xử lý ảnh để tăng hoặc giảm chất lượng ảnh đầu vào, lọc nhiễu, làm sắc nét, ...

### 4.1.3. Giới thiệu phần mềm sử dụng trong quá trình huấn luyện

Roboflow là một nền tảng end-to-end cho bài toán phát hiện đối tượng, cho phép thu thập và tổ chức dữ liệu hình ảnh, thực hiện chú thích bounding box hoặc polygon, đồng thời áp dụng các kỹ thuật tiền xử lý và tăng cường dữ liệu để cải thiện tính đa dạng và chất lượng mẫu. Roboflow tích hợp sẵn các thư viện như EfficientNet, MobileNet, YOLO, TensorFlow, PyTorch... giúp triển khai nhanh mô hình thị giác máy tính trên nhiều nền tảng. [5]

The logo for Roboflow, featuring the word "roboflow" in a bold, lowercase, purple sans-serif font.

Hình 4.1 Phần mềm roboflow

Kaggle là một nền tảng trực tuyến cho cộng đồng Machine Learning (ML) và Khoa học dữ liệu. Kaggle cho phép người dùng chia sẻ, tìm kiếm các bộ dữ liệu; tìm hiểu và xây dựng models, tương tác với những nhà khoa học và kỹ sư ML trên toàn thế giới; tham gia các cuộc thi để có cơ hội chiến thắng những giải thưởng giá trị. Người dùng Kaggle sẽ được hỗ trợ Graphic Processing Unit (GPU) và gần đây có thêm Tensor Processing Unit (TPU) để tăng tốc độ tính toán trong quá trình training cũng như inference.[6]

The logo for Kaggle, featuring the word "kaggle" in a bold, lowercase, blue sans-serif font.

Hình 4.2 Phần mềm Kaggle

#### *Ưu điểm:*

- Kho dữ liệu đa dạng, dễ tải và xem ngay trên web.
- Hỗ trợ chạy notebook Python/R có GPU/TPU miễn phí.
- Cộng đồng sôi nổi, nhiều code mẫu và giải pháp tham khảo.

#### *Nhược điểm:*

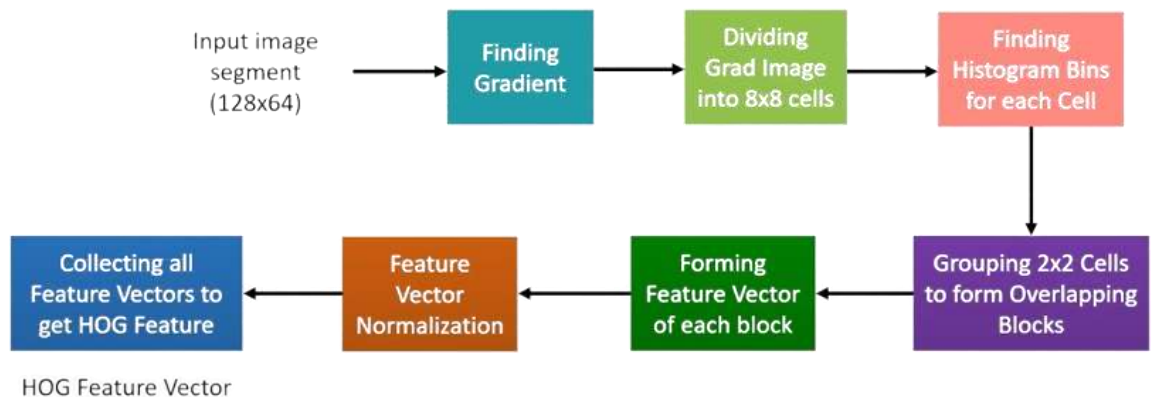
- Cạnh tranh cao, dễ “ngợp” với người mới.
- Tài nguyên GPU/TPU có hạn theo quota.
- Một số dataset chỉ dùng cho học tập, không thương mại

## 4.2 Các thuật toán phát hiện đối tượng phổ biến

### 4.2.1. Phương pháp mô tả đặc trưng (Histogram of Oriented Gradients – HOG)

Phương pháp mô tả đặc trưng (Histogram of Oriented Gradient – HOG) là một trong những phương pháp phát hiện đối tượng lâu đời nhất. Nó được giới thiệu lần đầu tiên vào năm 1986. HOG được sử dụng như 1 thuật toán trích chọn đặc trưng của đối tượng trong ảnh.

Mục đích của HOG là trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích. Vì vậy, HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh. HOG dựa trên việc chia ảnh đầu vào thành các ảnh con, tính toán histogram của ảnh để tổng hợp và trích rút ra các vector gọi là vector đặc trưng ứng với từng đối tượng.

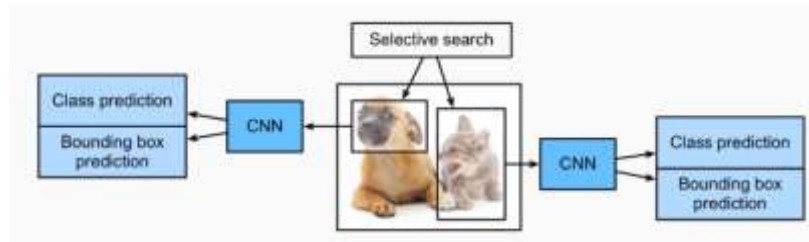


Hình 4.3 Biểu đồ phương pháp HOG.

Bản chất của phương pháp HOG là sử dụng thông tin về sự phân bố của các cường độ gradient (intensity gradient) hoặc của hướng biên (edge directions) để mô tả các đối tượng cục bộ trong ảnh. Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con, được gọi là cell và với mỗi cell, ta sẽ tính toán một histogram về các hướng của gradients cho các điểm nằm trong cell. Ghép các histogram lại với nhau ta sẽ có một biểu diễn cho bức ảnh ban đầu. Để tăng cường hiệu năng nhận dạng, các histogram cục bộ có thể được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một vùng lớn hơn cell, gọi là các khối (blocks) và sử dụng giá trị ngưỡng đó để chuẩn hóa tất cả các cell trong khối. Kết quả sau bước chuẩn hóa sẽ là một vector đặc trưng có tính bất biến đối với các thay đổi về điều kiện ánh sáng.[16]

#### 4.2.2. Mạng nơ-ron tích chập theo vùng (R-CNN)

Mạng nơ-ron tích chập theo vùng (R-CNN) là một cải tiến mới trong kỹ thuật phát hiện đối tượng từ các phương pháp trước đây của HOG và SIFT. Trong các mô hình R-CNN thường trích xuất các đặc trưng cần thiết nhất của đối tượng (thường là khoảng 2000 đặc trưng) bằng cách sử dụng 1 giải thuật chọn lọc (gọi là selective search). Quá trình lựa chọn các đặc trưng quan trọng nhất có thể được tính toán với sự trợ giúp của thuật toán tìm kiếm chọn lọc.

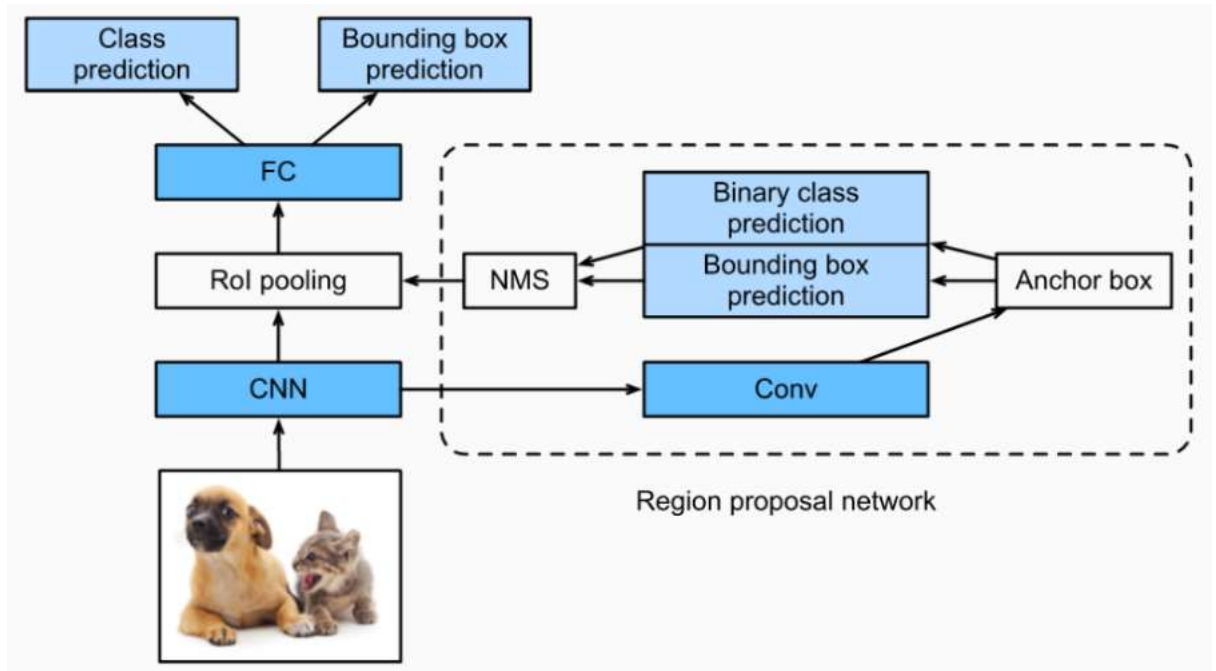


Hình 4.4: Sơ đồ mạng nơ-ron R-CNN.

Với R-CNN, việc trích xuất các vùng region proposal được thực hiện thông qua thuật toán Selective Search để trích chọn ra các vùng có khả năng chứa đối tượng (khoảng 2000 vùng). Sau đó, các vùng (ảnh) này được resize về 1 kích thước cố định và đưa qua 1 pretrained CNN model (imagenet), rồi từ đó tiến hành xác định offset và nhãn đối tượng. Tuy nhiên, việc đưa các vùng region proposal qua mạng CNN 2000 lần khiến tốc độ thực thi của model cực kì chậm![16]

#### 4.2.3. Faster R-CNN

Mặc dù mô hình R-CNN có thể thực hiện tính toán phát hiện đối tượng và đạt được hiệu quả nhưng vẫn tồn đọng nhiều điểm bất cập, điển hình là tốc độ mô hình. Vì vậy, một số phương pháp đã được phát triển để giải quyết vấn đề này và xử lý các nhược điểm của R-CNN. Trong số đó nổi bật nhất chính là mô hình Fast R-CNN và Faster R-CNN.



Hình 4.5 Sơ đồ mạng Faster-RCNN.[16]

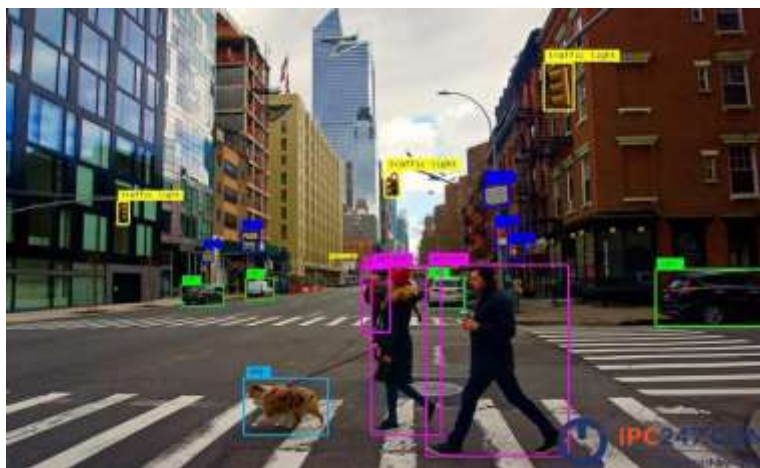
Với Faster-RCNN, thay vì việc sử dụng Selective Search, mô hình được thiết kế thêm 1 mạng con gọi là RPN (Region Proposal Network) để trích rút các vùng có khả năng chứa đối tượng của ảnh. Nhìn chung, sau khi thực hiện RPN, các bước xử lý sau tương tự như Fast-RCNN nhưng nhanh hơn nhiều (vì không sử dụng Selective Search) và được thiết kế như 1 mạng end-to-end trainable network![16]

### 4.3. Giới thiệu bài toán Object Detection và mô hình Yolo

#### 4.3.1. Bài toán Object Detection

Object detection (phát hiện vật thể) là một nhiệm vụ cốt lõi trong thị giác máy tính, không chỉ đơn thuần nhận biết các vật thể có trong hình ảnh mà còn xác định vị trí chính xác của chúng bằng các hộp giới hạn (bounding boxes) [7].

Khác với phân loại ảnh (image classification), vốn chỉ xác định xem một đối tượng cụ thể có xuất hiện trong ảnh hay không, object detection có khả năng nhận diện nhiều đối tượng và xác định vị trí từng đối tượng đó.



Hình 4.6 Xe tự lái cũng phụ thuộc nhiều vào computer vision

Xuất phát từ ý tưởng bài toán này, nhóm đã triển khai đề tài “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”. Trong đó, mô hình YOLOv11 đảm nhiệm việc nhận diện và xác định tọa độ chính xác của các tạp chất—xơ cây, nhựa, nilon, mảnh gỗ và lông vũ—trong khi robot Delta sử dụng thông tin đó để tính toán quỹ đạo gấp, đảm bảo quá trình loại bỏ tạp chất diễn ra tự động, liên tục và đạt độ chính xác cao.

#### 4.3.2. **Mô hình YOLO ( YOU ONLY LOOK ONCE )**

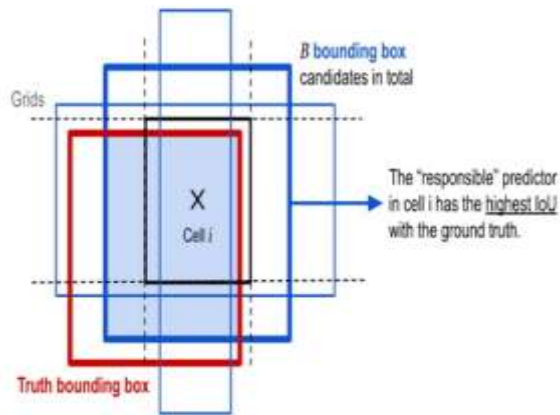
Mô hình YOLO, được Joseph Redmon giới thiệu lần đầu vào năm 2016, là một kiến trúc mạng CNN đột phá, ứng dụng cho việc nhận dạng và phân loại đối tượng dựa trên việc dự đoán các khung giới hạn (bounding box). Dù có thể không phải lúc nào cũng là thuật toán đạt độ chính xác cao nhất tuyệt đối so với một số mô hình phát hiện đối tượng khác, điểm mạnh cốt lõi của YOLO là khả năng không chỉ phát hiện và gán nhãn cho bài toán phân loại mà còn xác định được chính xác vị trí của vật thể trong không gian.

Trải qua nhiều phiên bản cải tiến liên tục, dòng mô hình YOLO ngày càng được hoàn thiện.[8] Trong đề tài này, nhận thấy những ưu thế tiềm năng về độ chính xác và tốc độ xử lý được kỳ vọng từ các phiên bản mới, nhóm đã quyết định sử dụng YOLO11 cho việc huấn luyện mô hình phân loại. Lựa chọn này nhằm khai thác những cải tiến về kiến trúc và hiệu suất mà YOLO11 hứa hẹn mang lại, giúp tối ưu hóa khả năng nhận diện và định vị đối tượng của hệ thống.

#### 4.3.3. **Anchor Box**

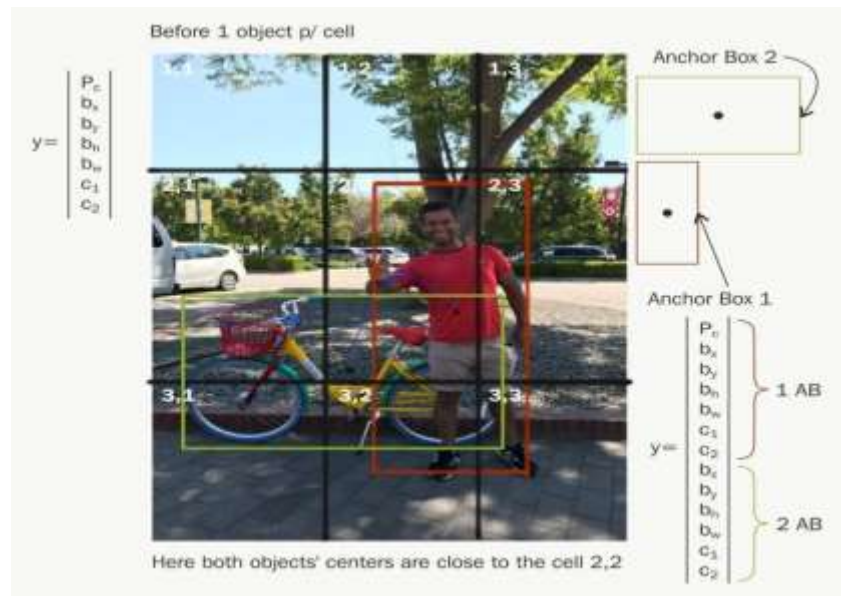
YOLO sẽ cần các anchor box để ước lượng và tìm bounding box cho vật thể. Ban đầu các anchor box này sẽ xác định và bao quanh vật thể, sau đó thuật toán regression bounding box sẽ tìm ra bounding box dự đoán cho vật thể. Mỗi vật thể sẽ được phân bổ về một anchor box. Trong trường hợp có nhiều anchor box cùng bao quanh vật thể thì thuật toán sẽ chọn ra anchor

box có IOU với ground truth cao nhất.



Hình 4.7 Cách xác định anchor box cho vật thể

Mô tả: Từ cell  $i$  ta xác định được 3 anchor box viền xanh. Tuy nhiên chỉ anchor box viền xanh đậm là được chọn vì có IOU với ground truth bounding box là cao nhất. Bên cạnh đó, mỗi vật thể còn được phân bố về một cell trên feature map chứa điểm mid point của vật thể. Từ cell đó ta có thể xác định các anchor box của đối tượng. Tóm lại, để xác định vật thể ta cần quan tâm đến anchor box và cell.



Hình 4.8 Cách xác định vật thể của YOLO

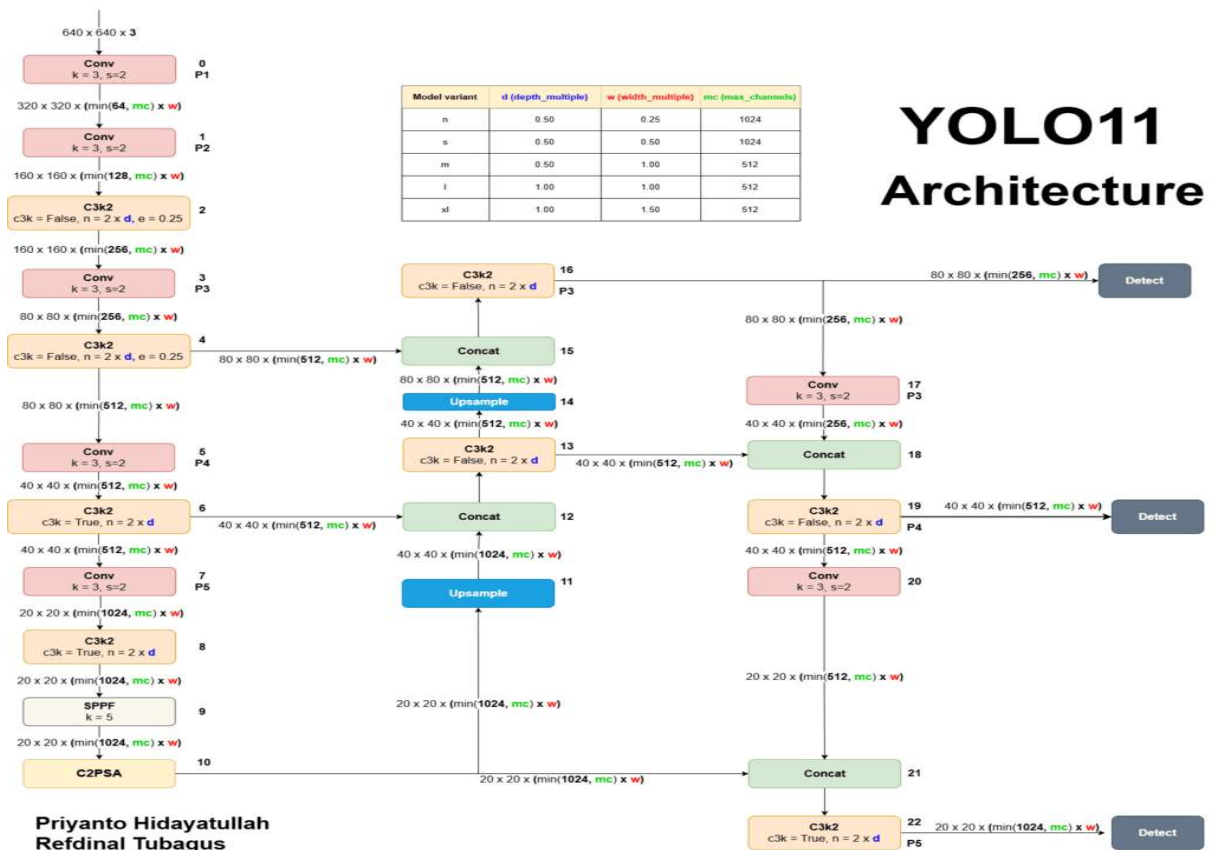
#### 4.3.4. Phân loại YOLOv11

Có 5 mô hình khác nhau: YOLO11n, YOLO11s, YOLO11m, YOLO11l, YOLO11x. Đầu tiên là nhỏ nhất và kém chính xác nhất, cuối cùng là lớn nhất với độ chính xác lớn nhất. Tất cả các mô hình đều chạy trên PyTorch

Model	size (pixels)	mAP <sup>val</sup> 50-95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

Hình 4.9 So sánh 5 mô hình YOLOv11

#### 4.3.5. Cấu trúc của YOLOv11



Hình 4.10 Cấu trúc của YOLOv11

**Backbone:** trích xuất đặc trưng ở nhiều tỉ lệ không gian.

**Lớp convolution:** YOLOv11 giữ nguyên chiến lược downsampling qua các lớp convolution như các phiên bản trước, nhưng thay thế khối C2f bằng khối C3k2 hiệu quả hơn. Khối C3k2 cải thiện hiệu suất tính toán bằng cách sử dụng hai phép convolution nhỏ thay vì một phép lớn, tương tự như trong YOLOv8.

**Khối SPPF và C2PSA:** YOLOv11 vẫn sử dụng khối Spatial Pyramid Pooling – Fast (SPPF) để trích xuất đặc trưng ở nhiều tỉ lệ khác nhau. Sau SPPF, bổ sung khối

Cross Stage Partial with Spatial Attention (C2PSA) để tăng cường cơ chế chú ý không gian.

*Neck:* kết hợp đặc trưng từ nhiều tầng và truyền xuống head.

YOLOv11 thay thế khối C2f trong neck bằng khối C3k2, giúp tăng tốc quá trình xử lý.

Mạng cũng mở rộng việc sử dụng cơ chế chú ý không gian thông qua mô-đun C2PSA, điểm khác biệt so với các phiên bản trước như YOLOv8, vốn không có khối chú ý này.

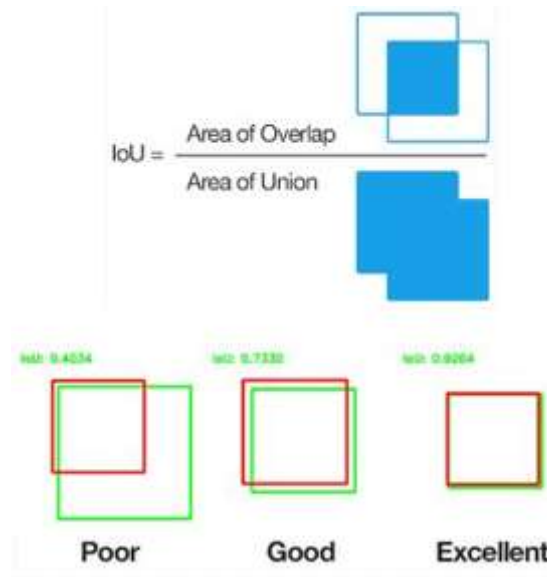
*Head:*

- Các khối C3k2 được tích hợp vào nhiều nhánh xử lý, giúp tinh chỉnh feature map hiệu quả ở các độ sâu khác nhau.
  - Sau các khối C3k2, YOLOv11 sử dụng các khối CBS (Convolution → BatchNorm → SiLU) để:
    - Trích xuất các đặc trưng quan trọng cho độ chính xác cao,
    - Ổn định feature map nhờ Batch Normalization,
    - Tăng cường phi tuyến bằng hàm kích hoạt SiLU, từ đó cải thiện hiệu năng mô hình.
- YOLOv11 tối ưu hoá toàn bộ mạng bằng cách thay khối C2f bằng C3k2 tiết kiệm tham số, giữ SPPF để thu nhận đa tỉ lệ, thêm C2PSA tăng cường chú ý không gian và sử dụng CBS để ổn định, phi tuyến hoá ở phần head, từ đó cải thiện tốc độ và hiệu năng phát hiện. [10]

#### 4.3.6. Phương pháp đánh giá mô hình phát hiện đối tượng

##### a. Intersection Over Union (IoU)

Intersection Over Union (IoU) là chỉ số đánh giá độ chính xác của model Object Detection trên một bộ dữ liệu cụ thể. Được xác định bởi tỷ lệ giữa vùng “overlap” và vùng “combine” giữa kết quả dự đoán của model và Ground Truth. IoU càng lớn thì chứng tỏ mô hình phân loại càng tốt, đồng nghĩa với việc bounding box dự đoán gần giống với bounding box thực tế.



Hình 4.11 Chỉ số đánh giá độ chính xác IoU

### b. Confusion Matrix

Confusion Matrix là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét các chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng đối tượng. Với bài toán nhận dạng với 2 đối tượng là Positive và Negative, Confusion Matrix có 4 thành phần:

- True Positive (TP): Các điểm positive được nhận đúng là positive.
- False Positive (FP): Các điểm negative thực được nhận sai là positive.
- False Negative (FN): Các điểm positive được phát hiện sai là negative.
- True Negative (TN): Các điểm negative được nhận đúng là negative.

Theo 4 thông số này được dùng để tính toán các chỉ số đánh giá phân loại như Precision và Recall. Cả 2 thông số này đều quan trọng trong mô hình đánh giá phân loại, tùy thuộc vào mục đích ứng dụng của người huấn luyện mà ưu tiên thông số nào hơn.

### c. Precision, Recall và F1-score

Precision được định nghĩa là tỷ lệ số điểm true positive (TP) trong số những điểm được phân loại là positive (TP + FP). Được dùng để đánh giá xem trong những dự đoán positive mà mô hình đã thực hiện có bao nhiêu là đúng.

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

Recall được định nghĩa là tỷ lệ số điểm true positive (TP) trong số những điểm thực sự là positive (TP + FN). Được dùng để đánh giá xem trong những điểm thực sự là positive thì mô hình dự đoán đúng bao nhiêu. Đây là chỉ số cho thấy độ bao quát của mô hình.

$$\text{Recall} = \frac{TP}{TP + FN}$$

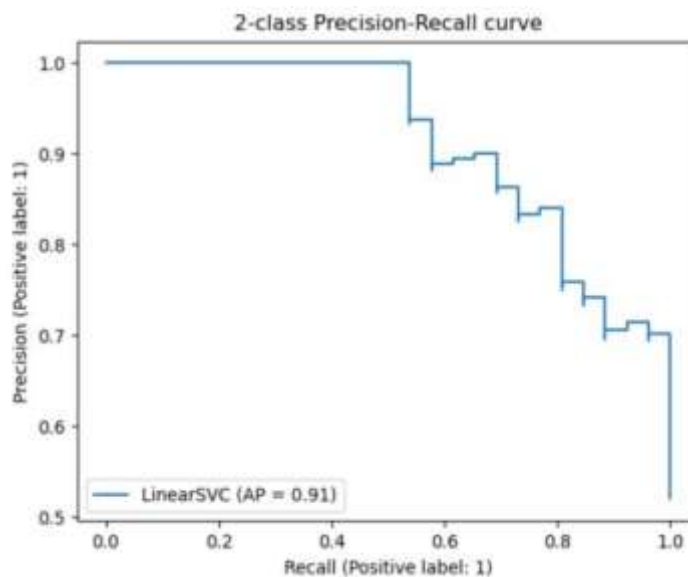
Precision và Recall càng cao thì mô hình huấn luyện càng tốt, tuy nhiên trong thực tế 2 chỉ số này thường đối lập nhau – không thể cùng tăng hoặc cùng giảm. Vì vậy, ta cần 1 chỉ số thứ 3 để đánh giá tổng quan mô hình: F1-score. F1-score càng cao thì mô hình càng tốt, lý tưởng nhất là F1-score = 1 (khi Precision = 1 và Recall = 1)

$$F1 = 2 \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

#### d. Precision-Recall curve và Average precision

Precision-Recall curve là một cách tốt để đánh giá hiệu suất của một mô hình phát hiện đối tượng cũng như độ tin cậy đã được biểu đồ hóa bởi một đường cong cho mỗi lớp. Với Precision-Recall curve, AUC còn có một tên khác là Average precision (AP).

Giả sử có N ngưỡng để tính precision và recall, mỗi ngưỡng cho một cặp giá trị precision, recall là  $P_n, R_n, n=1,2, \dots, N$ . Precision-Recall curve được vẽ bằng cách vẽ từng điểm có tọa độ  $(P_n, R_n)$  trên trục tọa độ và nối chúng với nhau.



Hình 4.12 Biểu đồ Precision và Recall

#### e. Mean Average Precision

mAP (mean Average Precision) là một chỉ số phổ biến để đánh giá hiệu suất của mô hình phát hiện đối tượng. Nó cung cấp một điểm số tổng hợp phản ánh khả năng nhận diện và định vị chính xác các đối tượng thuộc nhiều lớp khác nhau, ở các mức độ tin cậy (confidence) khác nhau.

Average Precision (AP) được tính bằng diện tích dưới đường cong Precision-Recall của từng lớp đối tượng. Sau đó, mAP là giá trị trung bình của các AP trên toàn bộ các lớp.

mAP: Là trung bình của AP trên tất cả các lớp trong tập dữ liệu.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

Các biến thể phổ biến:

- mAP@0.5 (mAP50): Đo lường mAP tại ngưỡng IoU = 0.5, chấp nhận dự đoán có  $\geq 50\%$  diện tích giao với ground truth; chuẩn mực của PASCAL VOC.
- mAP@0.5:0.95 (mAP[.5:.95]): Trung bình mAP qua các ngưỡng IoU từ 0.5 đến 0.95 (bước 0.05), đánh giá toàn diện hơn về định vị; tiêu chuẩn của COCO.

Tầm quan trọng của mAP:

Chỉ số mAP cung cấp một thước đo toàn diện khi kết hợp cả độ chính xác phân loại và định vị, đồng thời duy trì tính công bằng trên các tập dữ liệu mất cân bằng hoặc đa lớp, cho phép so sánh hiệu năng đồng đều giữa các đối tượng; điểm mAP cao cũng là chỉ báo tin cậy, cho thấy mô hình ít bỏ sót và ít sai nhận dạng, từ đó nâng cao mức độ chắc chắn trong các ứng dụng thực tế.

#### 4.3.7. Những cải tiến của YOLOv11 so với các phiên bản trước

YOLO11 được triển khai trên nền tảng PyTorch thông qua gói Ultralytics Python, tận dụng hệ sinh thái thư viện đa dạng và khả năng triển khai dễ dàng trên nhiều môi trường. Mô hình sử dụng detection head mới, giảm độ trễ và tăng tốc độ suy luận mỗi khung hình mà vẫn giữ vững độ chính xác nhờ cải tiến ở lớp dự đoán cuối cùng. Kiến trúc backbone và neck cũng được tái thiết kế để nâng cao khả năng trích xuất đặc trưng, cải thiện mAP trên COCO với các vật thể có kích thước và hình dạng đa dạng, dù giảm khoảng 22 % số tham số so với YOLOv8m [11].

YOLO11 tương thích hoàn toàn với quy trình làm việc của YOLOv8, cho phép chuyển đổi mượt mà mà không cần điều chỉnh pipeline hiện có. Đồng thời, nó hỗ trợ đầy đủ các tác vụ Vision AI như phát hiện đối tượng, phân đoạn instance và phân loại ảnh, mang lại tính đa năng cao cho cả nghiên cứu và ứng dụng thực tế. Nhờ tối ưu hóa trên cả edge devices, cloud servers và GPU NVIDIA, YOLO11 cân bằng tốt giữa tốc độ xử lý, độ chính xác và khả năng triển khai linh hoạt.

### 4.4. Phương pháp Homography

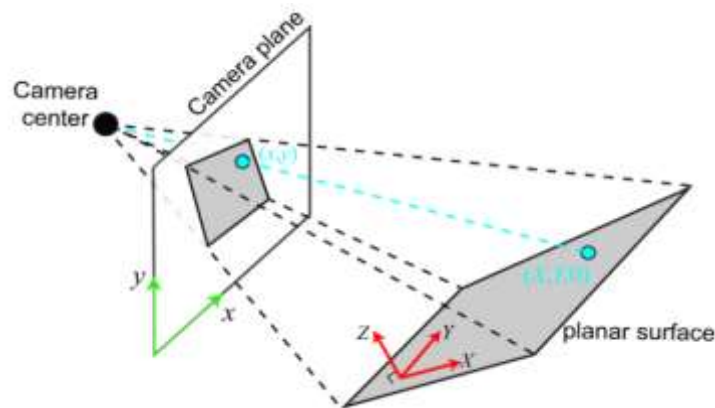
#### 4.4.6. Giới thiệu

[13] Homography là một kỹ thuật trong thị giác máy tính, mô tả một loại phép biến đổi hình học giữa hai ảnh chụp của cùng một bề mặt phẳng hoặc giữa các ảnh được

chụp bởi các camera chỉ khác nhau về góc quay. Đây là một phép biến đổi chiếu (projective transformation) có khả năng bảo toàn tính thẳng hàng của các điểm, được biểu diễn bởi một ma trận  $3 \times 3$ , mã hóa các biến đổi như tịnh tiến, xoay, và phối cảnh. Phương pháp này thường được dùng trong ghép ảnh, hiệu chuẩn camera, và thực tế tăng cường.

Đặc trưng của phép homography:

- Bảo toàn tính thẳng hàng của các điểm.
- Ánh xạ các đường thẳng thành đường thẳng khác.
- Không bảo toàn góc, khoảng cách hay song song.
- Là phép biến đổi tuyến tính trong hệ tọa độ đồng nhất (homogeneous coordinates).



Hình 4.13 Mối quan hệ giữa hệ tọa độ ảnh và mặt phẳng làm việc trong phép biến đổi Homography

Vì phép chiếu đồng dạng được biểu diễn bởi một ma trận kích thước  $3 \times 3$ , ta có thể biểu diễn nó dưới dạng:

$$H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix}$$

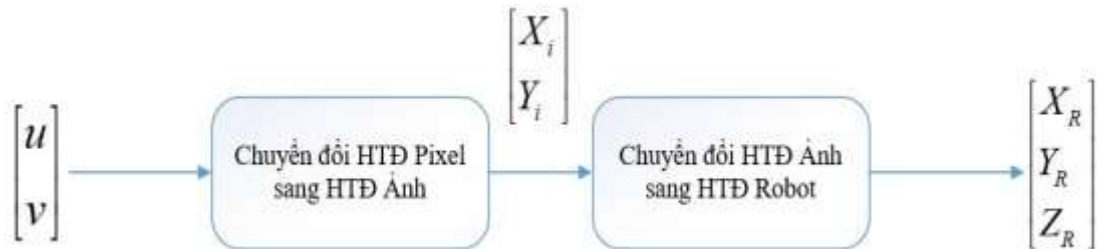
Giả sử ta lấy cặp điểm tương ứng đầu tiên — điểm  $u, v$  trên ảnh thứ nhất  $X, Y$  và trên ảnh thứ hai. Khi đó, phép chiếu đồng dạng  $H$  sẽ biến đổi chúng theo công thức sau:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

#### 4.4.7. Phương pháp chuyển đổi hệ tọa độ xác định vị trí đối tượng

Mục đích của việc chuyển đổi tọa độ là để xác định chính xác vị trí tâm của tạp chất từ hình ảnh camera (tọa độ pixel) sang hệ tọa độ thực tế của robot Delta, nhằm điều khiển robot đến đúng vị trí để gắp và loại bỏ.

Nhóm sử dụng phép biến đổi homography, cho phép ánh xạ các điểm từ mặt phẳng ảnh sang mặt phẳng làm việc dựa trên các cặp điểm mốc cố định. Phương pháp này không cần thông số nội tại của camera, chỉ dựa vào các điểm tương ứng thực tế giữa ảnh và bề mặt làm việc.



Hình 4.14 Quy trình chuyển đổi hệ tọa độ

#### Ưu điểm

- Đơn giản, không cần hiệu chuẩn camera phức tạp.
- Chính xác cho bề mặt phẳng, phù hợp với bàn làm việc của robot delta.
- Linh hoạt khi thay đổi vị trí camera.

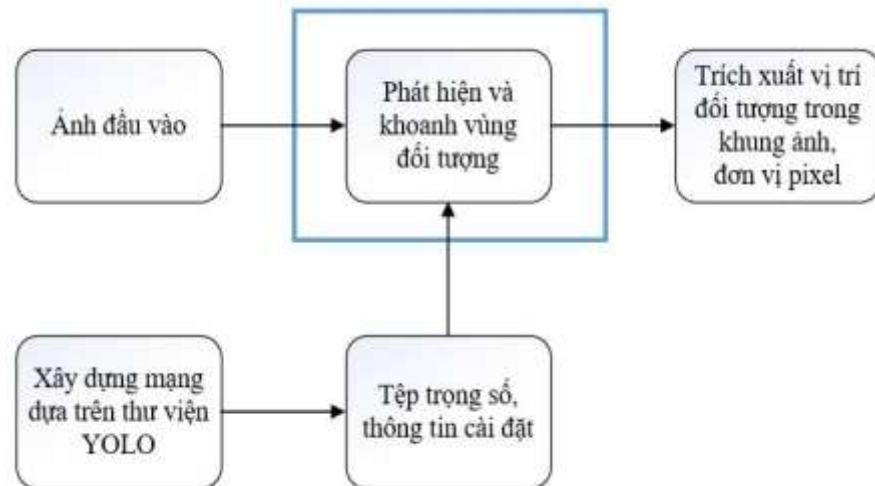
#### Hạn chế

- Chỉ hiệu quả với bề mặt phẳng.
- Yêu cầu các điểm mốc rõ ràng và chính xác.

### 4.5. Quy trình huấn luyện YOLO

Sử dụng mô hình mạng nơ-ron đã huấn luyện trên thư viện YOLO để phát hiện đối tượng và trích xuất thông tin từ các khung bao trong ảnh.

Capstone Project: “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”



Hình 4.15 Quy trình phát hiện và xác định vị trí đối tượng bằng mô hình YOLO

Nhóm sử dụng mô hình YOLOv11 – một trong những phiên bản mới nhất của You Only Look Once, khai thác kiến trúc CNN tối ưu để đạt tốc độ và độ chính xác cao trong phát hiện đối tượng.



Hình 4.16 Quy trình huấn luyện YOLO

## 4.6. Chuẩn bị tập dữ liệu

### 4.6.6. Thu thập dữ liệu

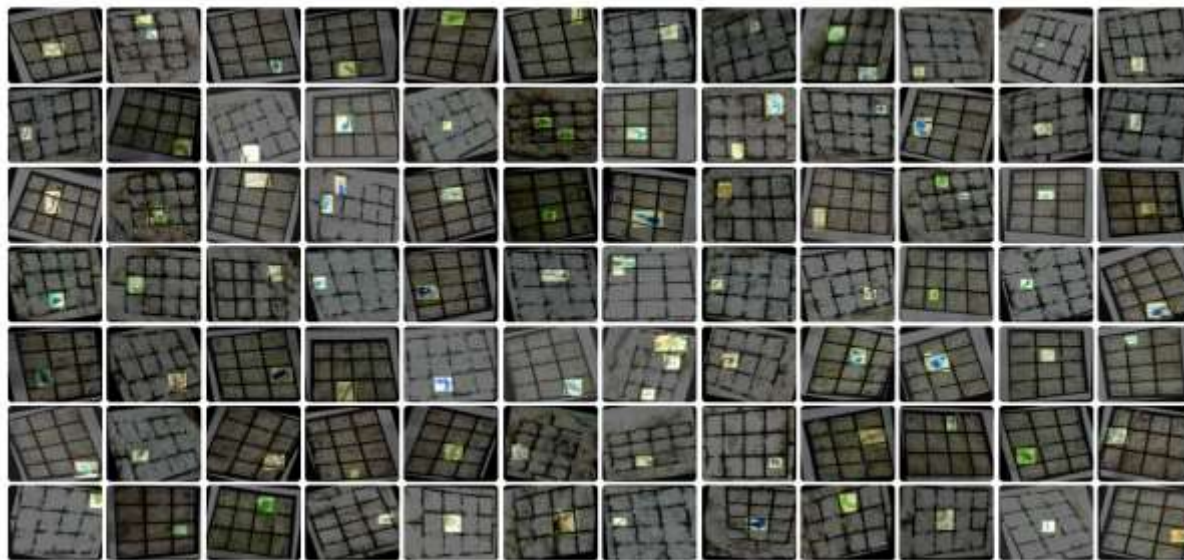
Tập dữ liệu cụ thể là các hình ảnh dùng để huấn luyện mạng nơ-ron sẽ được chụp trực tiếp từ máy ảnh, đây là các hình ảnh đầu vào chưa qua xử lý và định dạng. Tùy vào chất lượng của thiết bị chụp và điều kiện ánh sáng sẽ ảnh hưởng đến chất lượng ảnh đầu vào, qua đó có thể ảnh hưởng phần nào đến quá trình huấn luyện.

Sau khi có ảnh đầu vào, nhóm đã tạo một tập dữ liệu riêng trên phần mềm Roboflow và tải tất cả ảnh lên đó.

### 4.6.7. Đán nhãn dữ liệu

Sau khi đã tải dữ liệu lên, nhóm sẽ tiến hành bước đầu tiên của xử lý ảnh là đán nhãn dữ liệu. Đây sẽ là quá trình vẽ các đường bounding box bao quanh vật thể và chú thích vật thể sẽ thuộc nhóm đối tượng cần phân loại nào. Tùy vào số lượng đối tượng mong muốn phân loại mà quá trình này sẽ mất nhiều thời gian hơn. Ở đề tài này, nhóm đưa tất cả các tạp chất xuất hiện với một class là object

Roboflow sẽ hỗ trợ phương pháp dán nhãn bounding box đó là: dán nhãn theo hình chữ nhật.



Hình 4.17 Dán nhãn dữ liệu

#### 4.6.8. Phân chia tập dữ liệu

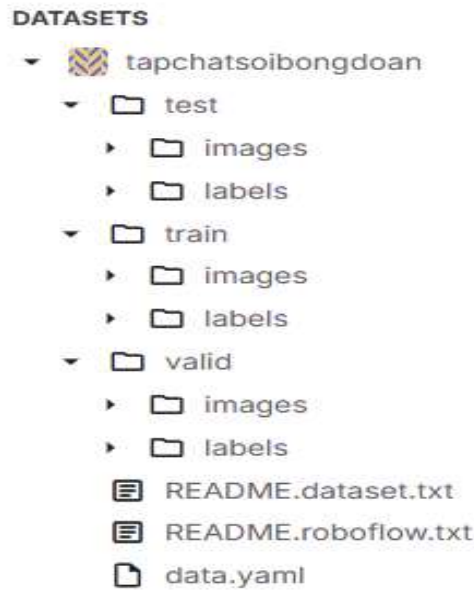
Ở bước này, ta sẽ phân chia tập dữ liệu hình ảnh đã dán nhãn ra 3 mục:

- Training: Sẽ được dùng làm đầu vào huấn luyện.
- Validation: Sẽ được dùng để kiểm chứng qua đó tìm kiếm ra phương pháp huấn luyện tốt nhất cho mô hình.
- Test: Sẽ được dùng để đánh giá kết quả huấn luyện.

Nhóm tiến hành chia tập dữ liệu ban đầu theo tỷ lệ 60% cho huấn luyện, 20% cho kiểm định và 20% cho kiểm tra.



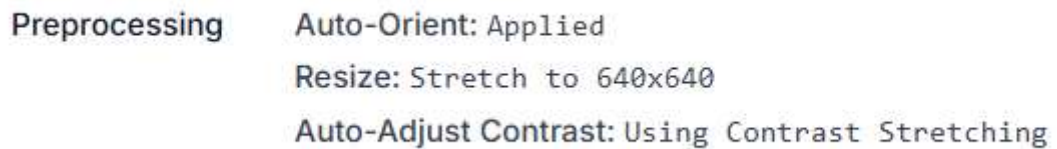
Hình 4.18 Tỷ lệ chia tập train/validation/test trong hệ thống



Hình 4.19 Cấu trúc thư mục dữ liệu cho hệ thống

#### 4.6.9. Chuẩn hoá và làm giàu dữ liệu

Trước khi đưa vào mô hình, tất cả ảnh được đưa về cùng kích thước 640×640 và định dạng thống nhất thông qua công cụ Roboflow



Hình 4.20 Cấu hình chuẩn hóa dữ liệu trong Roboflow

Để có thể tiết kiệm thời gian trong việc thu thập dữ liệu cũng như đa dạng hóa dữ liệu đầu vào, phần mềm Roboflow hỗ trợ các chức năng làm giàu dữ liệu như xoay ảnh, làm mờ, tăng/giảm độ sáng, và thêm nhiễu. Ở bước này, ta có thể tăng số lượng ảnh của tập dữ liệu lên gấp nhiều lần tùy theo mong muốn của người dùng.

Các phép biến đổi (augmentations) được sử dụng để làm giàu dữ liệu bao gồm:

- Lật ảnh (Flip): ngang
- Xoay 90° (90° Rotate): thuận và ngược chiều kim đồng hồ
- Cắt và phóng to (Crop): zoom tối thiểu 0%, tối đa 18%
- Xoay nhẹ (Rotation): góc từ -15° đến +15°
- Cắt xiên (Shear): ±10° theo phương ngang và ±10° theo phương dọc
- Điều chỉnh màu sắc:
- Hue: ±15°

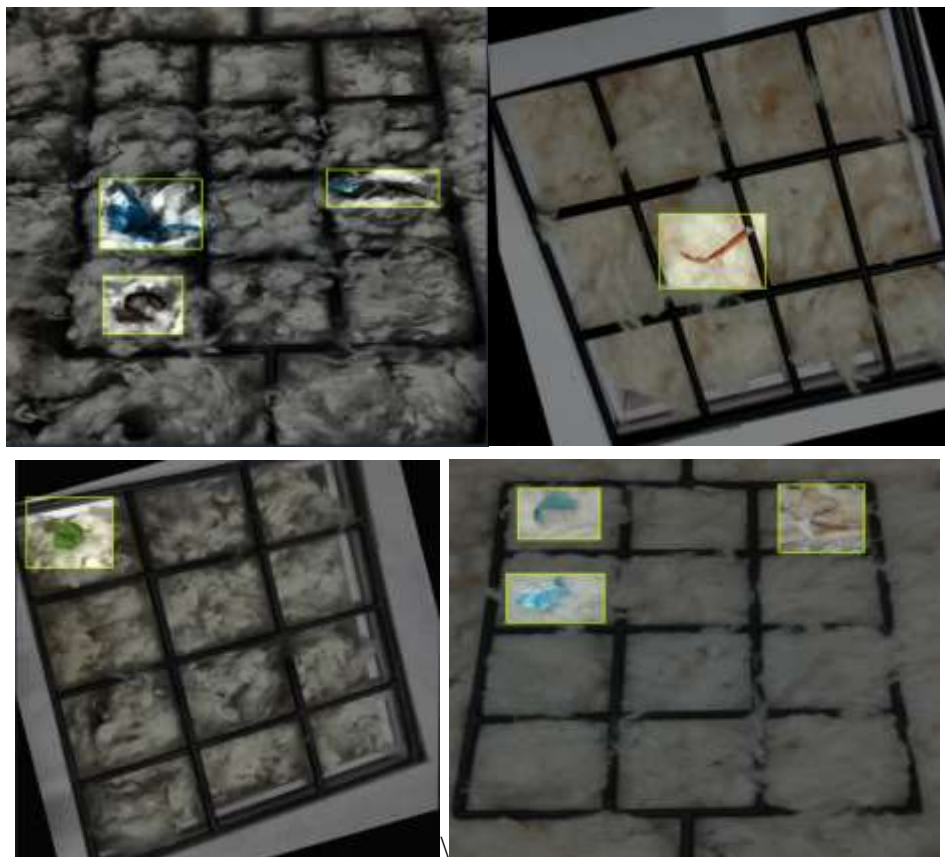
Capstone Project: “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”

- Saturation:  $\pm 20\%$
- Brightness:  $\pm 21\%$
- Exposure:  $\pm 15\%$
- Làm mờ (Blur): tối đa 0.6 px
- Thêm nhiễu (Noise): tối đa 0.05% số pixel

Việc phân chia dữ liệu theo tỷ lệ 80% huấn luyện, 10% kiểm định và 10% kiểm tra. Giúp đảm bảo mô hình có đủ mẫu để học các đặc trưng đa dạng, đồng thời giữ lại tập validation vừa đủ để điều chỉnh siêu tham số và phát hiện sớm hiện tượng overfitting, rồi dùng tập test độc lập để đánh giá khách quan khả năng tổng quát hóa của mô hình trên dữ liệu mới.



Hình 4.21 Dữ liệu được mở rộng và tái phân phối theo tỷ lệ 80-10-10



Hình 4.22 Các hình ảnh mới được tạo sau khi đã làm giàu

## 4.7. Quá trình huấn luyện

### 4.7.1 Quá trình huấn luyện trên phần mềm Kaggle

Trong giai đoạn đầu, nhóm đã sử dụng Google Colab để huấn luyện mô hình nhờ ưu điểm truy cập nhanh và hỗ trợ GPU miễn phí. Tuy nhiên, do giới hạn về thời gian sử dụng GPU liên tục và dung lượng bộ nhớ trên Colab không đáp ứng được yêu cầu cho tập dữ liệu lớn và kiến trúc mô hình YOLOv11m, nhóm đã chuyển sang nền tảng Kaggle.

	Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	100%	38/38	[00:05<00:00, 7.041t/s]
	all	600	742	0.921	0.861	0.916	0.44			
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
46/50	1.87G	1.391	0.5775	1.345	4	640: 100% 1745/1745 [04:43<00:00, 6.16it/s]				
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	100%	38/38	[00:04<00:00, 8.03it/s]
	all	600	742	0.915	0.861	0.911	0.438			
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
47/50	1.88G	1.388	0.5761	1.339	7	640: 100% 1745/1745 [04:41<00:00, 6.19it/s]				
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	100%	38/38	[00:05<00:00, 7.17it/s]
	all	600	742	0.917	0.868	0.912	0.441			
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
48/50	1.89G	1.377	0.5703	1.337	4	640: 100% 1745/1745 [04:45<00:00, 6.12it/s]				
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	18%	7/38	[00:01<00:04, 7.38it/s]

Hình 4.23 Nhiều lần chương trình tự dừng khi quá trình huấn luyện sắp hoàn thành

Sau epoch 41, các thành phần loss bao gồm box ( $\approx 1.34-1.35$ ), cls ( $\approx 0.66-0.68$ ) và dfl ( $\approx 1.30-1.31$ ) cùng với chỉ số mAP ( $mAP@0.5 \approx 0.915$ ,  $mAP@[0.5:0.95] \approx 0.46$ ) gần như không có sự cải thiện đáng kể. Do đó, cơ chế Early Stopping với hệ số độ trễ (patience) bằng 15 đã ngừng quá trình huấn luyện ở epoch 56 và lưu lại checkpoint tốt nhất tại epoch 41. Việc này vừa tối ưu hóa thời gian đào tạo, vừa giảm thiểu nguy cơ overfitting.

Capstone Project: “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
52/100	8.1G	1.364	0.6877	1.314	12	640: 100% [10:07<00:00, 1.82it/s]
s]	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% [1105/1105 [00:10<00:00, 2.13it/
	all	734	1101	0.924	0.851	0.917 0.463
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
53/100	8.1G	1.355	0.6787	1.307	13	640: 100% [10:07<00:00, 1.82it/s]
s]	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% [1105/1105 [00:10<00:00, 2.13it/
	all	734	1101	0.933	0.851	0.917 0.464
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
54/100	8.02G	1.35	0.6774	1.305	14	640: 100% [10:07<00:00, 1.82it/s]
s]	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% [1105/1105 [00:10<00:00, 2.14it/
	all	734	1101	0.922	0.852	0.915 0.462
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
55/100	8.1G	1.345	0.6695	1.299	18	640: 100% [10:07<00:00, 1.82it/s]
s]	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% [1105/1105 [00:10<00:00, 2.13it/
	all	734	1101	0.926	0.847	0.915 0.462
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
56/100	8.09G	1.352	0.6655	1.304	17	640: 100% [10:07<00:00, 1.82it/s]
s]	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% [1105/1105 [00:10<00:00, 2.13it/
	all	734	1101	0.928	0.848	0.915 0.461

EarlyStopping: Training stopped early as no improvement observed in last 15 epochs. Best results observed at epoch 41, best model saved as best.pt.]

Hình 4.24: Quá trình huấn luyện mô hình trên phần mềm Kaggle

#### 4.7.2 Kết quả huấn luyện

Sau khi huấn luyện thành công trên Kaggle. Các hình ảnh này đã phát được phát hiện và định vị bằng các bounding box bởi mô hình mạng đã được huấn luyện

##### Kết quả huấn luyện với 2052 ảnh đầu vào

Trong quá trình huấn luyện với hình ảnh tạp chất trong sợi bông với số lượng ảnh 2052 ảnh, mô hình cho kết quả khá khiêm tốn. Trên tập kiểm thử, tỉ lệ phát hiện (detection rate) đạt mức thấp, hầu hết các đối tượng không được đánh dấu chính xác hoặc hoàn toàn bị bỏ sót. Một số hạn chế mà nhóm ghi nhận được:

- Độ chính xác thấp và hiện tượng “nhảy khung bao” (bounding box jitter). Một số đối tượng bị nhận diện sai vị trí hoặc nhận diện “ma” (false positive) ngay cả khi trong ảnh không có dị vật.
- Mô hình thường lỏng khung bao chung hoặc chỉ phát hiện một trong số chúng, dẫn đến tỉ lệ false negative gia tăng. Hiện tượng này cho thấy khả năng phân biệt biên giới giữa các đối tượng cùng lớp còn kém.
- Mô hình dễ bị “dẫn dắt” bởi những vùng nhiễu có nền sáng – tối tương phản cao, xem đó như là dị vật. Các chi tiết nhỏ, cỡ dưới 20×20 pixel hầu như không được phát hiện.

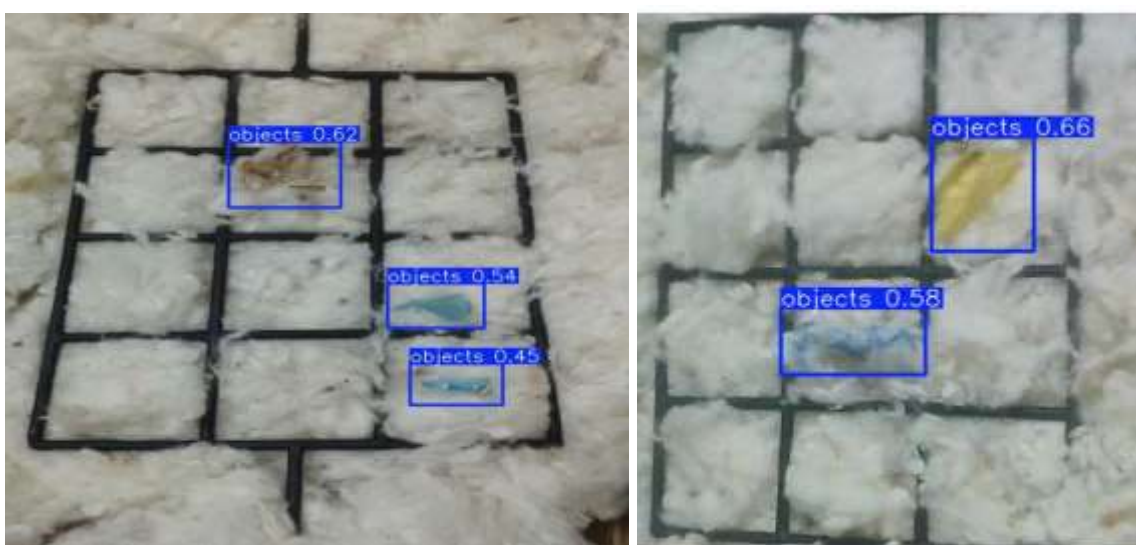


Hình 4.25 Kết quả huấn luyện thời gian thực với 2052 ảnh đầu vào

#### Kết quả huấn luyện với 8554 ảnh đầu vào

Mặc dù đã có những cải thiện đáng kể so với kết quả ban đầu trên ảnh 3052, hiệu suất của mô hình vẫn thể hiện xu hướng dao động trong khoảng từ **40%** đến **65%**, chưa đạt tính ổn định mong muốn. Cụ thể:

- Đã khắc phục được một số hạn chế khi huấn luyện với 2052 ảnh, nhưng thỉnh thoảng vẫn còn gặp phải
- Có thể nhận biết được tương đối tốt các tạp chất nhựa và ni lông, song khả năng phát hiện vỏ cây chưa thật đồng nhất, còn xơ cây gần như không được ghi nhận do màu sắc quá tương đồng với sợi bông.



Hình 4.26 Kết quả huấn luyện thời gian thực với 8554 ảnh đầu vào

#### Kết quả huấn luyện với 13083 ảnh đầu vào

Nhận thấy kết quả huấn luyện chưa tốt, nhóm đã tiến hành thu thập và mở rộng quy mô dữ liệu huấn luyện lên 13082 ảnh, tập trung hơn vào dữ liệu là vỏ cây và xơ cây. Mô hình đã khắc phục hầu hết các nhược điểm trước đây và đạt được những cải thiện đáng kể:

- Tỷ lệ nhận dạng chính xác gần như 90%
  - Vẫn duy trì khả năng phân biệt rõ ràng khi các đối tượng nằm sát nhau.
  - Phân phối độ tin cậy (confidence) chủ yếu rơi vào khoảng 80–90%; mặc dù đôi khi xuất hiện các vật thể lạ khiến xác suất giảm, mô hình vẫn nhận diện thành công.
- Ngoài ra, Kaggle cho phép lưu trữ thư mục chứa các trọng số mô hình sau khi huấn luyện; người dùng có thể cam kết (commit) notebook và tải về máy cá nhân để thực hiện phân loại trực tiếp qua camera hoặc phục vụ cho các dự án phát triển tiếp theo.



Hình 4.23 Kết quả huấn luyện với thời gian thực

#### 4.7.3 Đánh giá huấn luyện

YOLOv11n đạt  $mAP@50 = 0.907$ , trong khi YOLOv11m (cấu hình cơ bản) chỉ đạt 0.872. Sau khi tối ưu hóa hyper-parameters trên YOLOv11m,  $mAP@50$  được nâng lên 0.912, khẳng định vai trò then chốt của việc hiệu chỉnh tham số trong việc khai thác tối đa năng lực mô hình. Tuy nhiên, cải thiện này đi kèm với yêu cầu tài nguyên tính toán

Capstone Project: “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”

tăng, đòi hỏi cân nhắc giữa độ chính xác và khả năng triển khai thực tế trên hệ thống thị giác máy tính tích hợp robot delta.

```
50 epochs completed in 2.104 hours.
Optimizer stripped from /kaggle/working/runs/yolov11n/weights/last.pt, 5.5MB
Optimizer stripped from /kaggle/working/runs/yolov11n/weights/best.pt, 5.5MB

Validating /kaggle/working/runs/yolov11n/weights/best.pt...
Ultralytics 8.3.155 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
YOLO11n summary (fused): 100 layers, 2,582,347 parameters, 0 gradients, 6.3 GFLOPs

Class Images Instances Box(P. R. mAP50 mAP50-95): 100% |██████████| 44/44 [00:09<00:00, 4.62it/s]
/usr/local/lib/python3.11/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered in less
  xa[xa < 0] = -1
/usr/local/lib/python3.11/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered in less
  xa[xa < 0] = -1

all 1384 1843 0.852 0.881 0.907 0.463

Speed: 0.2ms preprocess, 2.3ms inference, 0.8ms loss, 1.2ms postprocess per image
Results saved to /kaggle/working/runs/yolov11n
```

Hình 4.24: Kết quả sau khi huấn luyện với yolov11n

```
50 epochs completed in 8.761 hours.
Optimizer stripped from /kaggle/working/runs/yolov11m2/weights/last.pt, 40.5MB
Optimizer stripped from /kaggle/working/runs/yolov11m2/weights/best.pt, 40.5MB

Validating /kaggle/working/runs/yolov11m2/weights/best.pt...
Ultralytics 8.3.155 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
YOLO11m summary (fused): 125 layers, 20,030,803 parameters, 0 gradients, 67.6 GFLOPs

Class Images Instances Box(P. R. mAP50 mAP50-95): 100% |██████████| 47/47 [00:29<00:00, 1.62it/s]
/usr/local/lib/python3.11/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered in less
  xa[xa < 0] = -1
/usr/local/lib/python3.11/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered in less
  xa[xa < 0] = -1

all 1499 1568 0.856 0.851 0.872 0.434

Speed: 0.2ms preprocess, 15.5ms inference, 0.0ms loss, 1.8ms postprocess per image
Results saved to /kaggle/working/runs/yolov11m2
+ Code + Markdown
```

Hình 4.25 Kết quả sau khi huấn luyện với yolov11m ( cấu hình cơ bản)

```
56 epochs completed in 9.623 hours.
Optimizer stripped from /kaggle/working/runs/yolov11m_optimized/weights/last.pt, 40.5MB
Optimizer stripped from /kaggle/working/runs/yolov11m_optimized/weights/best.pt, 40.5MB

Validating /kaggle/working/runs/yolov11m_optimized/weights/best.pt...
Ultralytics 8.3.155 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
YOLO11m summary (fused): 125 layers, 20,030,803 parameters, 0 gradients, 67.6 GFLOPs

Class Images Instances Box(P. R. mAP50 mAP50-95): 100% |██████████| 23/23 [00:11<00:00, 2.06it/s]
/usr/local/lib/python3.11/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered in less
  xa[xa < 0] = -1
/usr/local/lib/python3.11/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered in less
  xa[xa < 0] = -1

all 734 1101 0.92 0.863 0.912 0.467

Speed: 0.2ms preprocess, 11.0ms inference, 0.0ms loss, 1.4ms postprocess per image
Results saved to /kaggle/working/runs/yolov11m_optimized
```

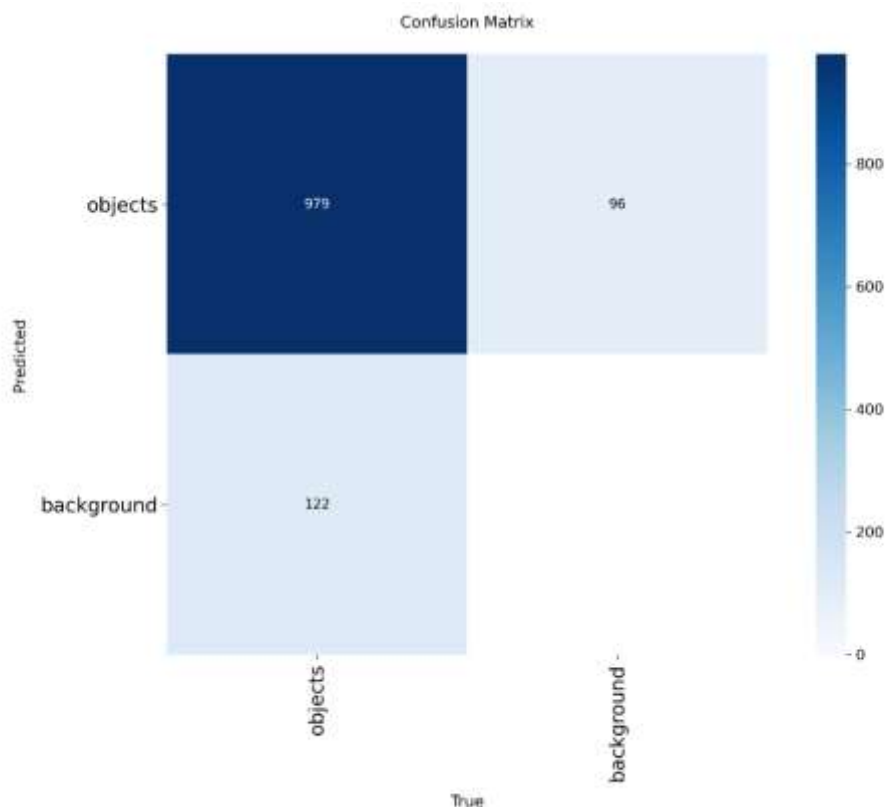
Hình 4.26 Kết quả sau khi huấn luyện với yolov11m ( mở rộng tham số)

Nhóm đã triển khai YOLOv11m sau khi tối ưu hyper-parameters, tích hợp trực tiếp với camera để đảm bảo cả độ chính xác (mAP cao) và khả năng xử lý real-time. Khung hình được truyền liên tục đến bộ xử lý, nơi mô hình xác định và định vị tạp chất, rồi ngay lập tức gửi tọa độ đến cơ cấu chấp hành của robot Delta, từ đó thực hiện loại bỏ chính xác và duy trì hiệu suất ổn định cho quy trình sản xuất sợi.

Sau khi hoàn tất quá trình huấn luyện tự động, bước đánh giá mô hình trên tập dữ liệu chưa từng quan sát là thiết yếu để xác định khả năng tổng quát hóa và phát hiện giới hạn của mô hình. Trong đó, các chỉ số thường được ứng dụng bao gồm:

- **Accuracy (độ chính xác):** tỷ lệ giữa số dự đoán đúng và tổng số dự đoán. Mặc dù trực quan và dễ tính toán, accuracy có thể đánh giá thiếu công bằng khi phân bố nhãn mất cân bằng.
- **Precision (độ chuẩn xác):** tỷ lệ của các kết quả dự đoán dương tính chính xác trên tổng số dự đoán dương tính. Chỉ số này đặc biệt quan trọng khi chi phí của kết quả dương tính giả cao.
- **Recall (khả năng thu hồi):** tỷ lệ của các dự đoán dương tính đúng trên tổng số mẫu dương tính thực tế. Khi bỏ sót các trường hợp dương tính gây hậu quả nghiêm trọng, recall trở thành tiêu chí ưu tiên.
- **F1-Score:** đạo hàm điều hòa giữa precision và recall, cho phép cân bằng đồng thời hai khía cạnh độ an toàn và độ bao phủ của mô hình.

Tóm lại, đánh giá mô hình không chỉ giúp lượng hóa hiệu năng mà còn chỉ ra điểm mạnh, điểm yếu cần khắc phục thông qua các lần lặp huấn luyện tiếp theo, từ đó tối ưu hóa chất lượng dự đoán.

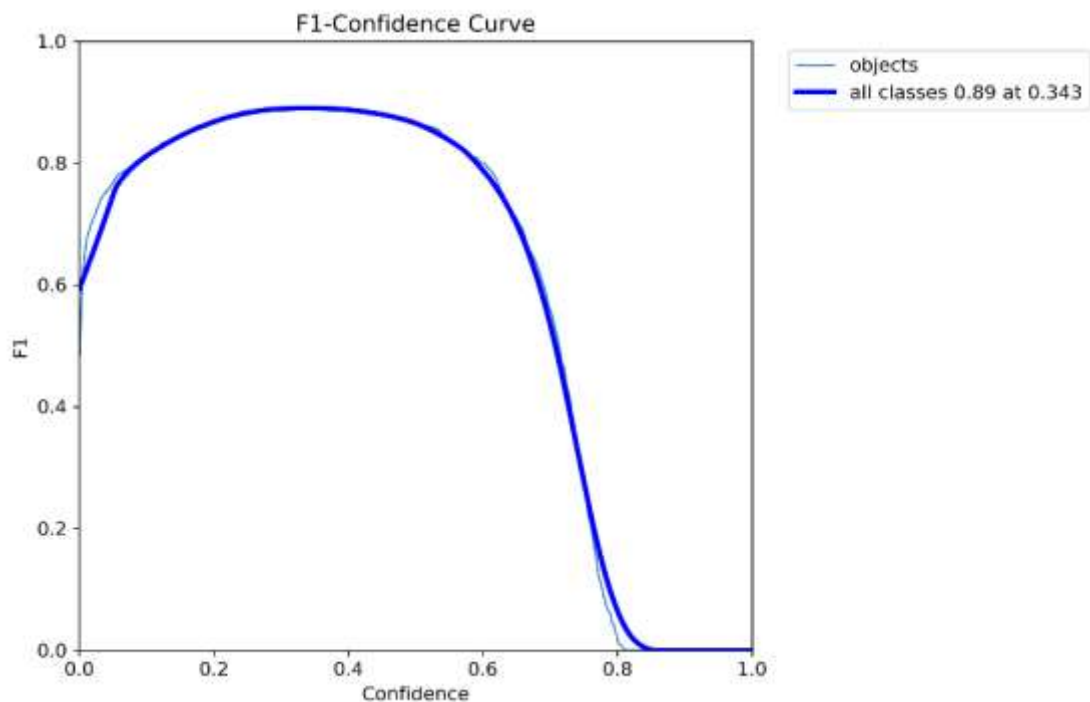


Hình 4.27 Ma trận Confusion

**Nhận xét:** Ma trận này đánh giá hiệu suất của mô hình trong việc phân loại "objects" (đối tượng) và "background" (nền). True Positives (TP): 979 (mô hình dự

đoán là đối tượng và thực tế là đối tượng). Đây là con số rất cao, cho thấy mô hình nhận diện đối tượng rất tốt. False Negatives (FN): 122 (mô hình dự đoán là nền nhưng thực tế là đối tượng). Đây là các lỗi bỏ sót đối tượng. False Positives (FP): 96 (mô hình dự đoán là đối tượng nhưng thực tế là nền). Đây là các lỗi nhận diện sai đối tượng. True Negatives (TN): Không hiển thị trực tiếp nhưng có thể suy ra từ tổng số mẫu hoặc được tính toán là số lượng nền được phân loại đúng. (Trong trường hợp này, cột 'background' của 'True' không có giá trị, có thể do ma trận này tập trung vào một lớp cụ thể hoặc đơn giản hóa).

**Ý nghĩa:** Ma trận này cung cấp cái nhìn chi tiết về các loại lỗi mà mô hình mắc phải. Các giá trị TP, FN, FP này cho thấy mô hình có độ chính xác và độ thu hồi cao đối với việc phát hiện đối tượng, với số lượng lỗi tương đối thấp.

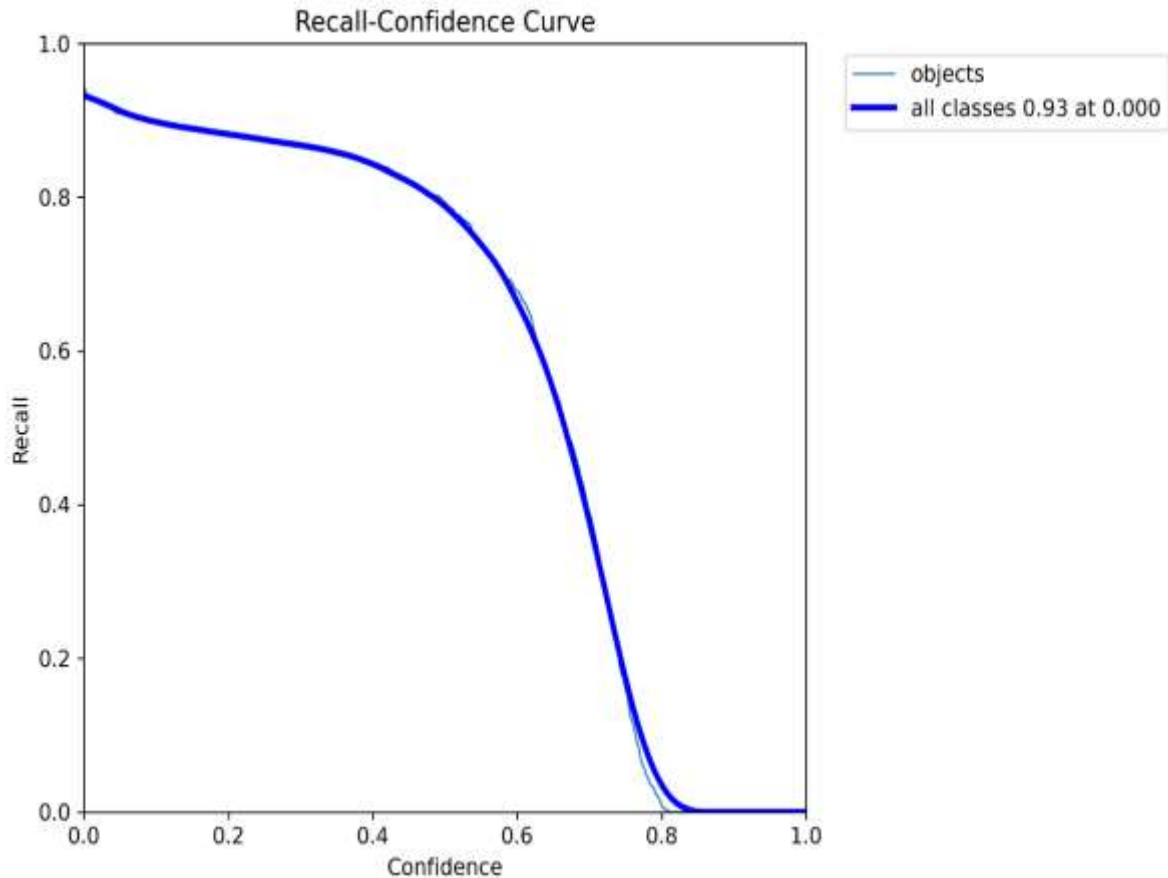


Hình 4.28 Đồ thị F1-Confidence Curve.

**Nhận xét:** Đồ thị này hiển thị điểm F1 (là trung bình điều hòa của precision và recall) theo ngưỡng tin cậy. Điểm F1 đạt đỉnh cao nhất (khoảng 0.89-0.9) ở ngưỡng tin cậy trung bình (khoảng 0.3-0.4), sau đó giảm dần khi ngưỡng tin cậy tăng hoặc giảm. Giá trị "all classes 0.89 at 0.343" cho thấy điểm F1 cao nhất của tất cả các lớp là 0.89 khi ngưỡng tin cậy là 0.343.

**Ý nghĩa:** Điểm F1 là một thước đo cân bằng giữa Precision và Recall. Đỉnh của đường cong F1-Confidence cho biết ngưỡng tin cậy tối ưu để đạt được sự cân bằng tốt

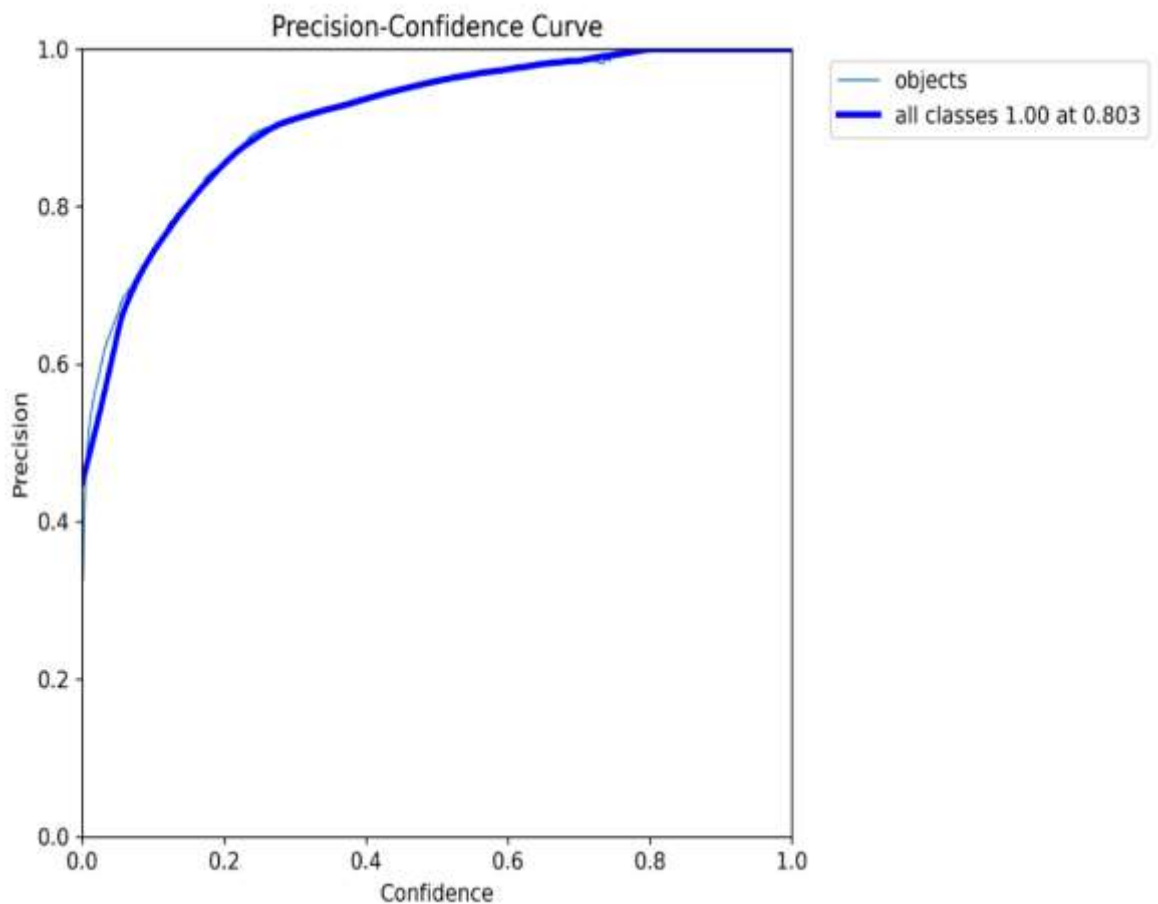
nhất giữa Precision và Recall. Giá trị F1 cao nhất là 0.89 là một kết quả rất tốt, điều này chứng tỏ mô hình có sự cân bằng hiệu suất rất tốt.



Hình 4.29 Đồ thị độ phủ Recall - Confidence.

**Nhận xét:** Đồ thị này cho thấy mối quan hệ nghịch biến giữa Recall (độ thu hồi) và Confidence (ngưỡng tin cậy). Khi ngưỡng tin cậy tăng, recall giảm mạnh. Đường cong thể hiện rằng ở ngưỡng tin cậy thấp (gần 0), độ thu hồi rất cao (gần 0.95), nhưng khi ngưỡng tin cậy tăng lên khoảng 0.7-0.8, độ thu hồi giảm về gần 0. Giá trị "all classes 0.93 at 0.000" cho thấy độ thu hồi chung của tất cả các lớp khi ngưỡng tin cậy bằng 0 là 0.93.

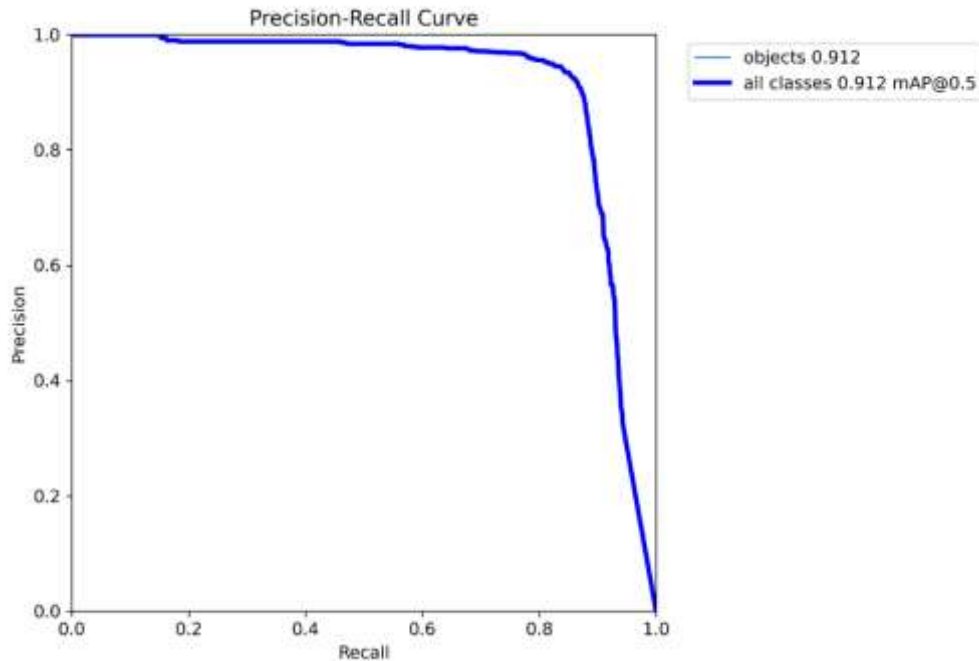
**Ý nghĩa:** Đường cong này giúp đánh giá khả năng của mô hình trong việc tìm ra tất cả các đối tượng liên quan khi các ngưỡng tin cậy khác nhau được áp dụng.



Hình 4.30: Đồ thị độ chính xác Precision - Confidence.

**Nhận xét:** Đồ thị này biểu diễn mối quan hệ giữa Precision và Confidence. Đường cong cho thấy khi ngưỡng tin cậy tăng, độ chính xác của mô hình cũng tăng lên và đạt gần 1.0 ở ngưỡng tin cậy cao (khoảng 0.7-0.8). Giá trị "all classes 1.00 at 0.803" cho thấy mô hình đạt độ chính xác hoàn hảo (1.00) khi ngưỡng tin cậy là 0.803.

**Ý nghĩa:** Đường cong này giúp đảm bảo các phát hiện của mô hình có độ tin cậy cao nhất.



Hình 4.31 Precision-Recall Curve

**Nhận xét:** Đây là đồ thị rất quan trọng trong đánh giá mô hình phân loại và phát hiện đối tượng. Đường cong cho thấy mối quan hệ giữa Precision (độ chính xác) và Recall. Mô hình đạt độ chính xác gần 1.0 khi độ thu hồi thấp (dưới 0.8), sau đó độ chính xác giảm nhanh chóng khi độ thu hồi tăng lên gần 1.0. Giá trị "all classes 0.912 mAP@0.5" chỉ ra rằng giá trị mAP (mean Average Precision) của tất cả các lớp tại ngưỡng IoU (Intersection over Union) 0.5 là 0.912. Đây là một con số rất tốt, cho thấy hiệu suất tổng thể cao của mô hình.

**Ý nghĩa:** Đường cong này là chỉ số mạnh mẽ về khả năng phân biệt của mô hình. mAP@0.5 là một metric tiêu chuẩn và giá trị 0.912 cho thấy mô hình hoạt động rất hiệu quả trong việc phát hiện và phân loại đối tượng chính xác

#### 4.8. Tính toán và xác định vị trí đối tượng

Để chuyển đổi chính xác tọa độ tâm tạp chất từ ảnh (tọa độ pixel) về hệ tọa độ làm việc thực tế của robot Delta, Nhóm đã áp dụng phép biến đổi đồng nhất hình học (Homography). Phương pháp này cho phép ánh xạ một cách tuyến tính các điểm trong mặt phẳng ảnh sang mặt phẳng thực khi ta giả thiết tạp chất nằm trên một mặt phẳng chung ( $Z = 0$ ). Việc chọn  $Z = 0$  tương ứng với bề mặt bàn làm việc giúp đơn giản hóa mô hình và giảm thiểu sai số do chiều sâu không đổi.

Quy trình thực hiện:

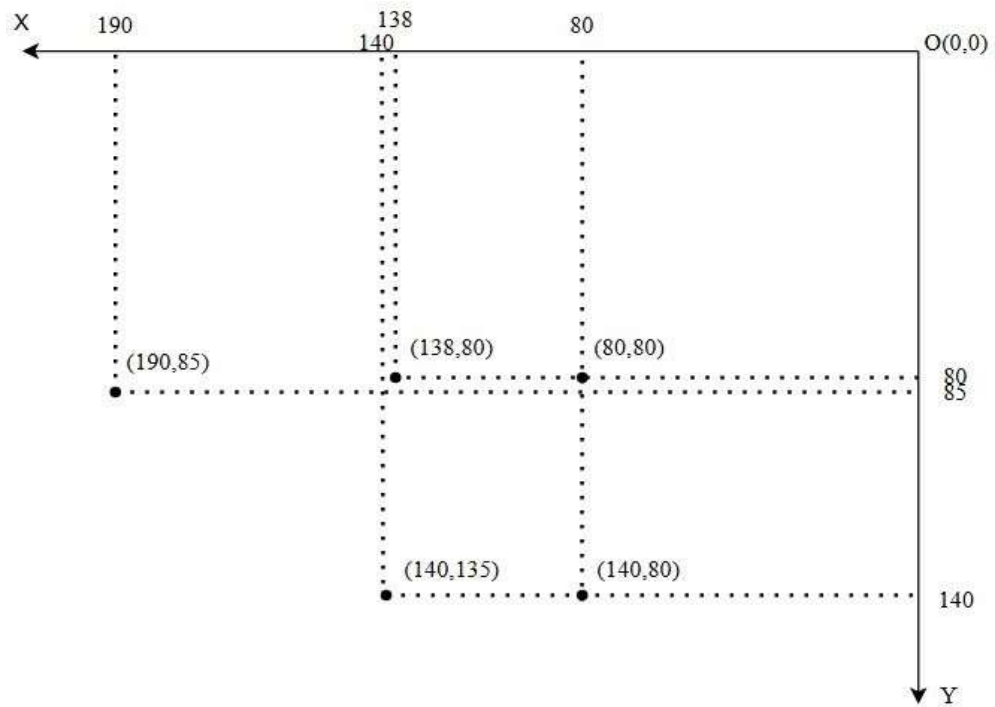
**Bước 1:** Chuẩn bị dữ liệu chuẩn

Capstone Project: “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”

Gán trục tọa độ XY như hình dưới gốc O, hình ảnh sau khi đã dùng Yolo phát hiện, và tính toán tâm của bounding box



Hình 4.32 Gán tọa độ Oxy với ảnh chứa tâm pixel



Hình 4.33 Lấy tọa độ 5 điểm trên thực tế

Với phương pháp Homography, phải chọn tối thiểu trên 4, phân bố đều trên mặt bàn (bề mặt làm việc) và ghi lại tọa độ thực (X, Y, Z = 0) bằng thước kỹ thuật, nhóm đã chọn 5 điểm như bảng dưới đây

Bảng 4.1: Tọa độ thực các điểm đã chọn

111,237	190,85
441,104	140,80
264,243	80,138
265,115	140,135
443,265	80,80

**Bước 2:** Sử dụng hàm `cv2.findHomography()` trong OpenCV để tính toán ma trận Homography.

$$\begin{bmatrix} -3.94649156e^{-1} & 2.05770145e^{-2} & 2.84498029e^2 \\ 2.32319639e^{-1} & 3.40576683e^{-1} & 3.19963130 \\ 1.57701646e^{-3} & 7.55774409e^{-5} & 1 \end{bmatrix}$$

**Bước 3:** Áp dụng biến đổi Homography lên tọa độ phát hiện

Nhận kết quả detection từ mô hình YOLO dưới dạng từ tâm của đối tượng, chuyển các điểm này vào dạng ma trận (x, y, 1) và nhân với ma trận H:

$$\begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = H \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

**Bước 4:** Hiệu chỉnh tọa độ giữa gốc O và tâm robot ( $x_d, y_d$ )

Tính sai lệch đến góc

$$\Delta x = x_d - x_o, \Delta y = y_d - y_o$$

Tính tọa độ tương đối đến tâm robot

$$x_{rel} = \Delta x - x_r, y_{rel} = \Delta y - y_r$$

**Nhận xét:**

- Độ tin cậy của mô hình suy giảm khi ánh sáng quá yếu hoặc chói gắt, dẫn đến tăng tỉ lệ bỏ sót (false negatives) và báo động nhầm (false positives).
- Khi camera quan sát ở góc nghiêng hoặc tạp chất không đồng nhất, tọa độ xác định tạp chất có thể bị sai lệch, ảnh hưởng đến khả năng gấp chính xác của robot Delta.

**Hướng phát triển trong tương lai**

- Kết hợp dữ liệu độ sâu (depth map) hoặc stereo-vision để xác định không chỉ vị trí mà còn hình dạng và kích thước tạp chất, giúp robot điều chỉnh góc gấp phù hợp.

- Nghiên cứu và huấn luyện phiên bản siêu nhẹ (dung lượng < 1 MB) sử dụng các kỹ thuật pruning, quantization hoặc kiến trúc Tiny Neural Network, đảm bảo chạy trực tiếp trên thiết bị nhúng với bộ nhớ và hiệu năng hạn chế.

#### **Kết luận chương 4:**

Chương 4 trình bày quá trình xây dựng và triển khai mô hình thị giác máy tính để tự động phát hiện và loại bỏ tạp chất, khởi đầu bằng các khái niệm xử lý ảnh cơ bản và chọn lọc công cụ (Arduino IDE, Roboflow, Kaggle) cho giai đoạn thu thập, chú thích và huấn luyện dữ liệu. Trọng tâm là bài toán phát hiện đối tượng với kiến trúc YOLOv11, từ mô tả cấu trúc mạng, cơ chế anchor box, đến quy trình huấn luyện (tăng cường dữ liệu, thiết lập hyper-parameters, early stopping) và đánh giá qua IoU, Precision, Recall, F1-score, mAP. Cuối cùng, chương giới thiệu phép biến đổi Homography để chuyển tọa độ pixel sang tọa độ thực cho robot Delta, đồng thời chỉ ra hạn chế hiện tại và kiến nghị hướng tối ưu (ví dụ tích hợp depth map, sử dụng mô hình nhẹ) để đảm bảo hiệu năng và khả năng vận hành trên nền tảng nhúng.

## CHƯƠNG 5: TÍCH HỢP HỆ THỐNG VÀ KIỂM THỬ THỰC NGHIỆM

### 5.1. Mô tả quy trình hoạt động và lưu đồ giải thuật của hệ thống

Quy trình hoạt động của hệ thống bao gồm 5 bước:

*Bước 1:* Người dùng thao tác trên giao diện điều khiển được thiết kế bằng Visual Studio Code để bật hệ thống và các thiết bị liên quan. Khi chế độ phát hiện được kích hoạt, camera sẽ bắt đầu thu nhận hình ảnh vùng làm việc.

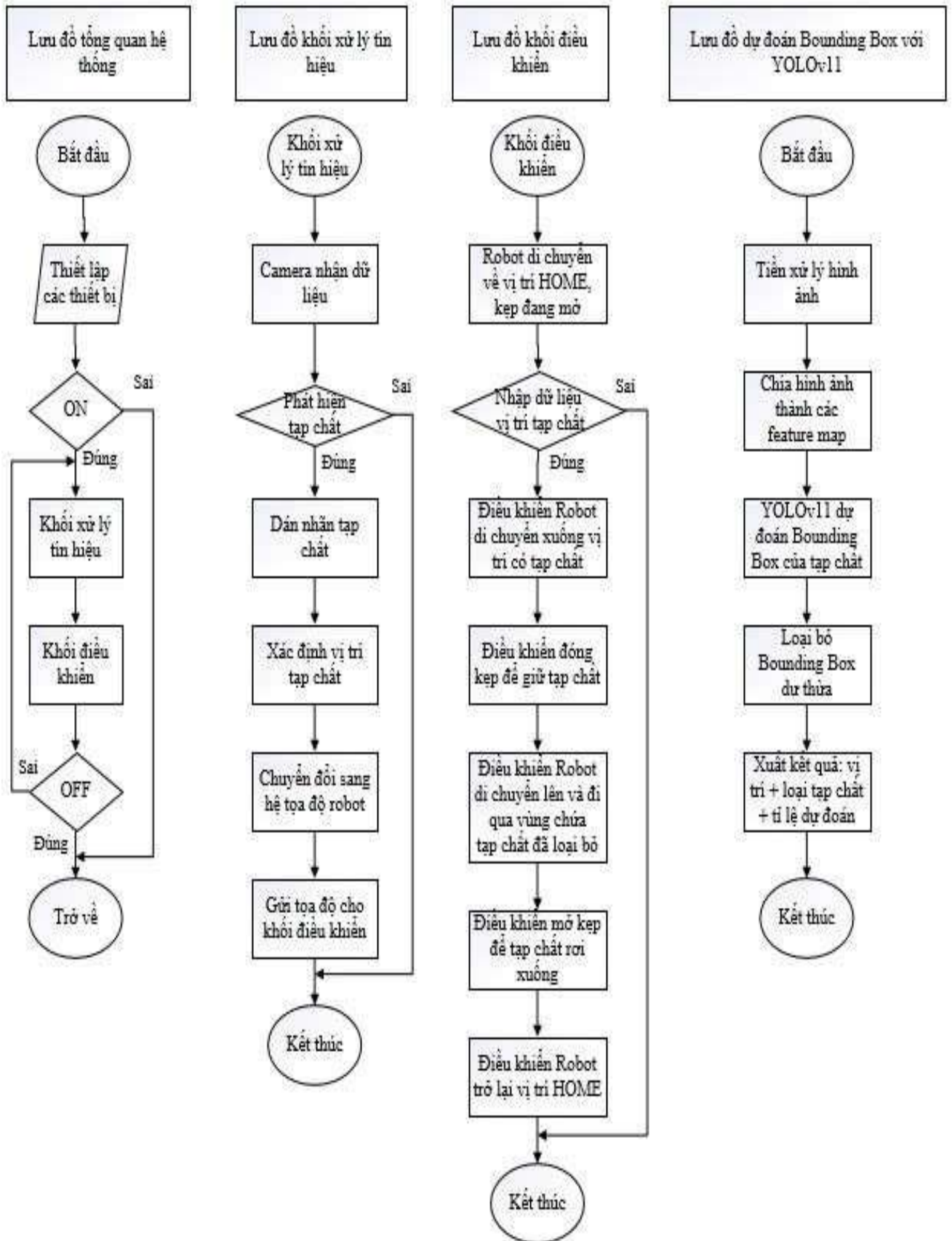
*Bước 2:* Camera truyền hình ảnh thu được vào mô hình YOLOv11. Mô hình sẽ thực hiện phân tích, phát hiện tạp chất và xuất ra vị trí của bounding box tương ứng với đối tượng.

*Bước 3:* Hệ thống chuyển đổi tọa độ từ hệ camera sang hệ tọa độ robot. Dựa vào vị trí của tạp chất, Arduino sẽ tính toán góc quay và số xung cần thiết, sau đó gửi tín hiệu điều khiển robot di chuyển chính xác đến vị trí mục tiêu.

*Bước 4:* Tín hiệu điều khiển được gửi đến driver động cơ bước (TB6600) để cấp xung cho các động cơ của robot. Đồng thời, các tín hiệu đóng/mở kẹp gấp cũng được điều khiển theo thứ tự nhằm đảm bảo quá trình gấp – thả tạp chất diễn ra đúng quy trình.

*Bước 5:* Robot Delta di chuyển đến vị trí tạp chất, đóng kẹp để gấp vật, sau đó di chuyển đến vùng chứa rác để thả tạp chất. Cuối cùng, robot quay trở lại vị trí HOME và chờ lệnh xử lý tạp chất tiếp theo.

Capstone Project: “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”



Hình 5.1 Các lưu đồ thuật toán của hệ thống

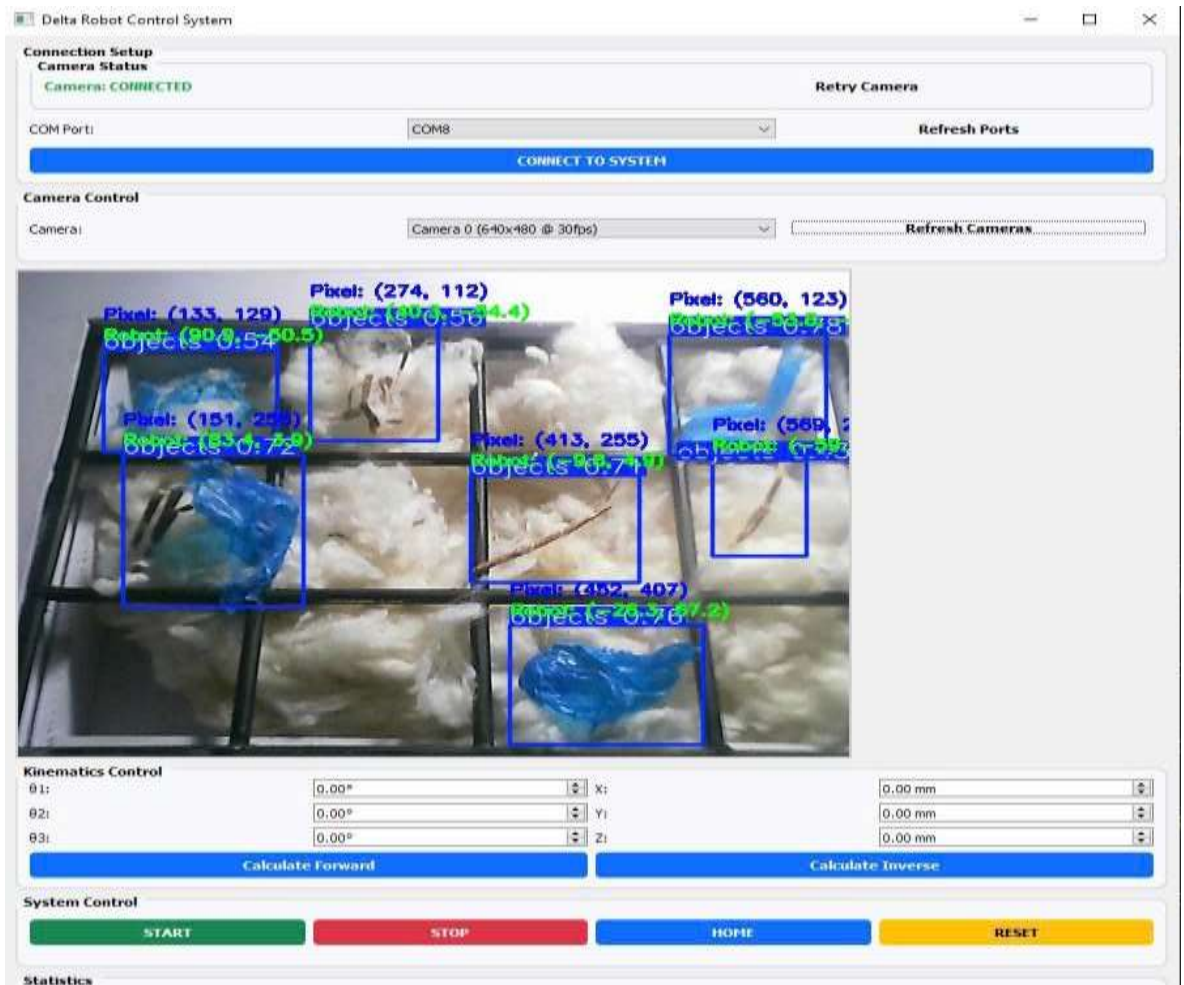
## 5.2. Lập trình, thiết kế phần mềm điều khiển và giao diện người dùng

Trong chương này, nhóm đã xây dựng phần mềm điều khiển và giao diện người dùng cho hệ thống bằng ngôn ngữ Python. Quá trình này bao gồm tạo dữ liệu huấn luyện, xây dựng và huấn luyện mô hình xử lý ảnh YOLOv11, sau đó sử dụng file trọng số thu được để nhận diện đối tượng trong thực tế.

Giao diện do nhóm thiết kế cung cấp các chức năng chính:

- Kết nối và kiểm tra trạng thái camera
- Kết nối robot Delta
- Hiển thị luồng hình ảnh, thực hiện nhận diện đối tượng
- Điều khiển robot loại bỏ tạp chất
- Hiển thị các thông số thống kê về quá trình xử lý

Những thành phần này giúp người dùng theo dõi, vận hành và đánh giá hiệu quả hệ thống một cách trực quan và thuận tiện.



Hình 5.2 Hệ thống thị giác máy tính tích hợp robot Delta trong quá trình loại bỏ tạp chất sợi

Giao diện điều khiển robot Delta được thiết kế thành bốn khu vực chức năng chính, sắp xếp khoa học theo chiều ngang để người vận hành dễ dàng tiếp cận. Ở khung Camera Control, người dùng có thể chọn và làm mới danh sách camera thông qua dropdown và nút “Refresh”, trong khi khung xem video chính nằm ở giữa màn hình, hiển thị trực tiếp hình ảnh kèm khung nhận diện đối tượng và thông tin vị trí (tọa độ pixel) cùng xác suất nhận dạng ngay trên khung hình.

Ngay bên cạnh là khung Kinematics Control, bao gồm hai nhóm ô nhập liệu cho góc khớp ( $\theta_1, \theta_2, \theta_3$  và tọa độ không gian (X, Y, Z) với hai nút “Calculate Forward” và “Calculate Inverse” để thực hiện tính toán động học thuận – nghịch nhanh chóng. Phía bên phải, khung System Control tập trung các nút lệnh quan trọng: “START” và “STOP” để khởi chạy hoặc tạm dừng quy trình tự động, “HOME” để đưa robot về vị trí gốc an toàn và “RESET” khởi động lại hệ thống khi cần thiết

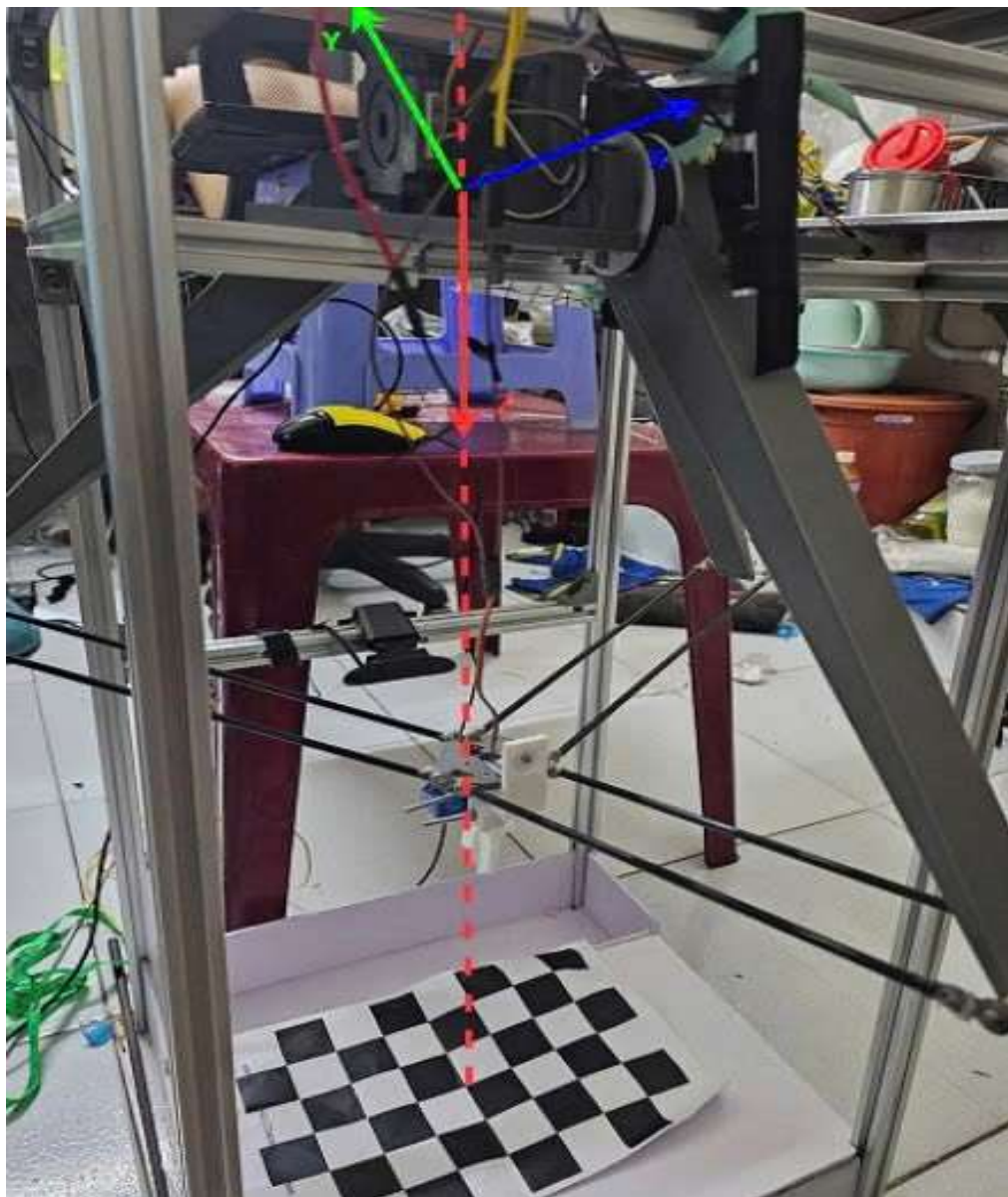
### **Kết luận chương 5**

Chương 5 tập trung vào việc tích hợp và kiểm thử toàn bộ hệ thống tự động loại bỏ tạp chất, gồm hai phần chính: quy trình vận hành năm bước “chụp ảnh – nhận diện – chuyển tọa độ – điều khiển động cơ – gấp thả” được điều phối liên tục qua giao diện Python trên Visual Studio Code, và ứng dụng phần mềm với các tính năng kết nối camera, hiển thị kết quả nhận diện, tính toán động học và điều khiển lệnh “START/STOP/HOME/RESET”. Việc đồng bộ giữa Arduino Mega, driver TB6600, cơ cấu servo/motor, mô hình YOLOv11 và giao diện người dùng đã được thử nghiệm trên mô hình thực tế, cho thấy hệ thống vận hành ổn định, chính xác và đáp ứng đầy đủ yêu cầu tự động hóa trong sản xuất sợi.

## Chương 6: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ


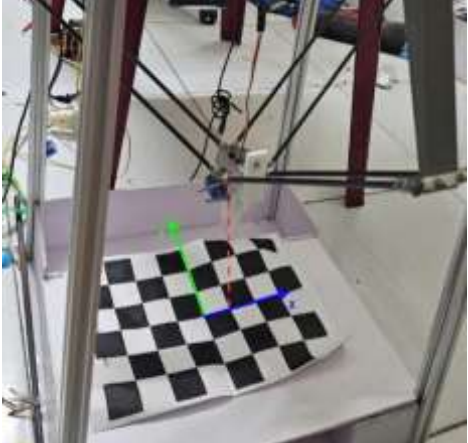

### 6.1. Kiểm chứng động học nghịch trên checkerboard

Khoảng cách từ gốc tọa độ Robot Delta đến vị trí checkerboard nhóm chọn là tâm của checkerboard theo trục (x, y, z) lần lượt là (0, 0, 340) (mm) với mỗi ô của checkerboard với kích thước thực là 320x320 (mm). Đối với cách kiểm chứng động học thực tế trên checkerboard, nhóm sẽ tiến hành nhập lần lượt 2 vị trí bất kỳ và xem đối chiếu kết quả thực tế của cơ cấu chấp hành cuối so với checkerboard.



Hình 6.1 Hệ tọa độ Robot so với checkerboard

Bảng 6.1: Kiểm chứng động học nghịch

Điểm	Kết quả thực tế
Điểm 1: $P_x, P_y, P_z = (0, 0, 440)$	
Điểm 2: $P_x, P_y, P_z = (32, 0, 440)$	
Điểm 3: $P_x, P_y, P_z = (32, -32, 440)$	

Hình 6.1 Kiểm chứng động học nghịch

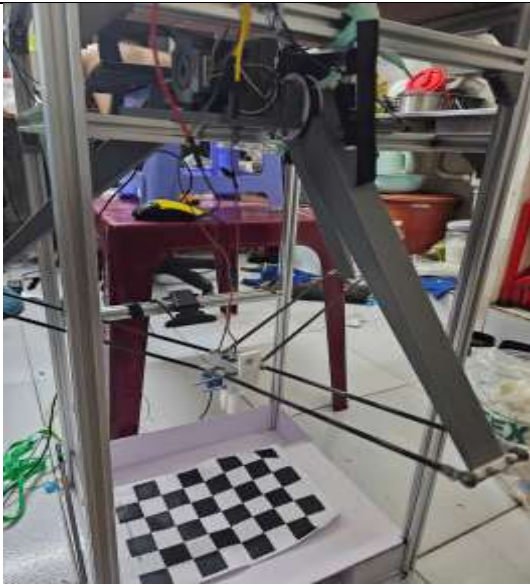
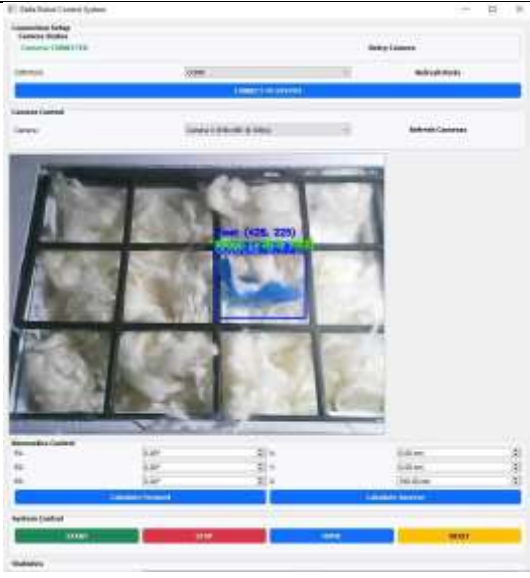
Nhận xét: Kết quả kiểm tra cho thấy, khi nhập vào các tọa độ bất kỳ và tiến hành tính toán động học nghịch cho Robot Delta, vị trí thực tế của cơ cấu cuối (end-effector) mà robot di chuyển đến có sai số trong khoảng  $\pm 1$  mm so với giá trị đặt. Điều này chứng

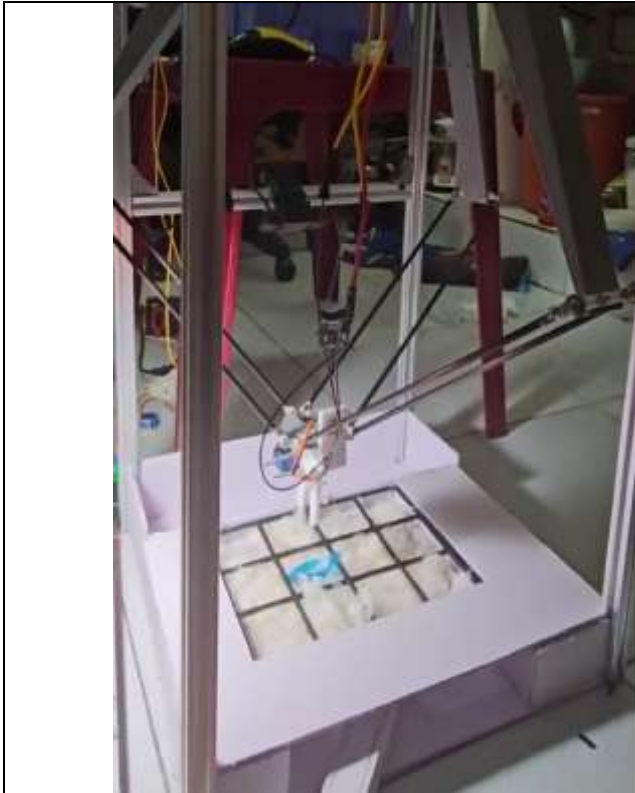
tổ hệ thống điều khiển và mô hình động học được xây dựng chính xác, đảm bảo độ tin cậy trong các ứng dụng thực tế yêu cầu độ chính xác cao.

### 6.1.1. Đánh giá kết quả toàn hệ thống

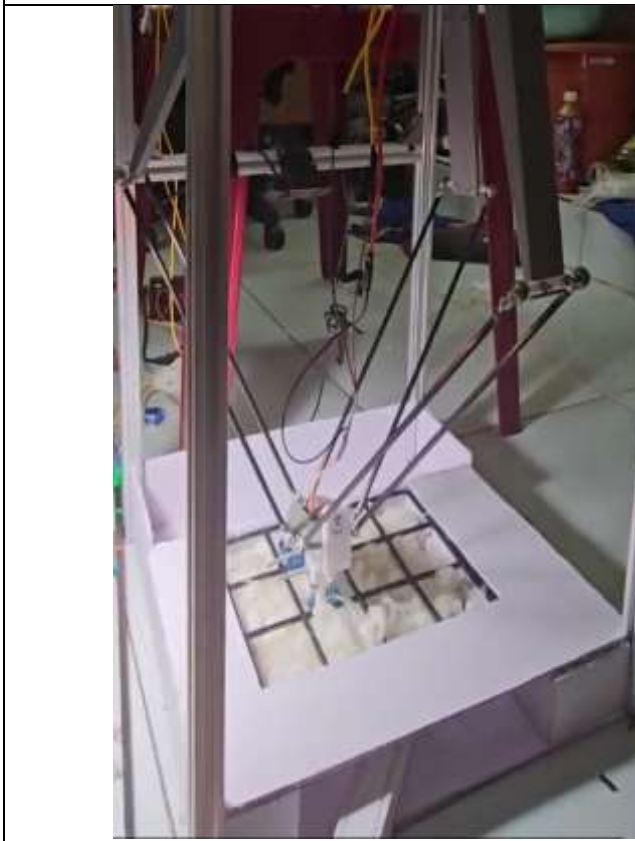
Nhóm tiến hành chạy thực nghiệm mô hình loại bỏ tạp chất trong bông để kiểm chứng tính liên tục và ổn định của hệ thống. Trong quá trình thử nghiệm, các mẫu bông có lẫn tạp chất được đặt lên khay xử lý một cách xen kẽ nhằm mô phỏng điều kiện làm việc thực tế. Hệ thống Robot Delta kết hợp với thị giác máy tính nhận diện và thực hiện thao tác loại bỏ tạp chất một cách tự động. Dưới đây là bảng quy trình hoạt động của hệ thống trong quá trình kiểm chứng.

Bảng 6.2: Quy trình giai đoạn hoạt động của hệ thống

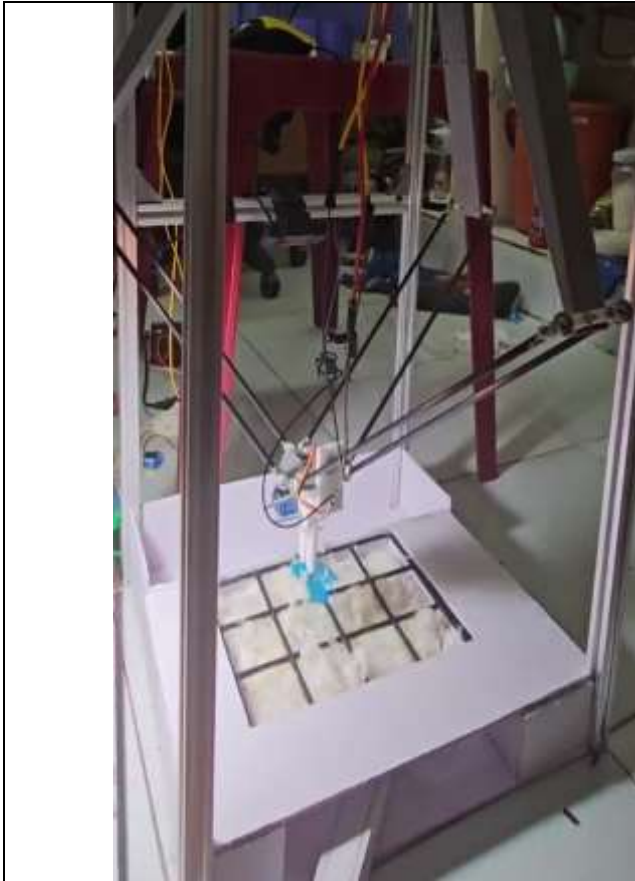
Giai đoạn trong hệ thống	Mô tả
	<p>Khi hệ thống được bật nguồn, Robot Delta sẽ tự động di chuyển về vị trí Home — điểm xuất phát mặc định được xác định bởi 3 cảm biến hành trình, đảm bảo độ chính xác trong các bước tiếp theo. Sau đó, bông được đặt lên khay chứa để bắt đầu quá trình xử lý.</p>
	<p>Hệ thống thị giác máy tính sử dụng camera để quét toàn bộ bề mặt khay, nhận diện các tạp chất dựa trên sự khác biệt về màu sắc, hình dạng và kích thước so với bông. Tọa độ các vị trí chứa tạp chất sau đó được xác định và gửi về bộ điều khiển của Robot Delta.</p>



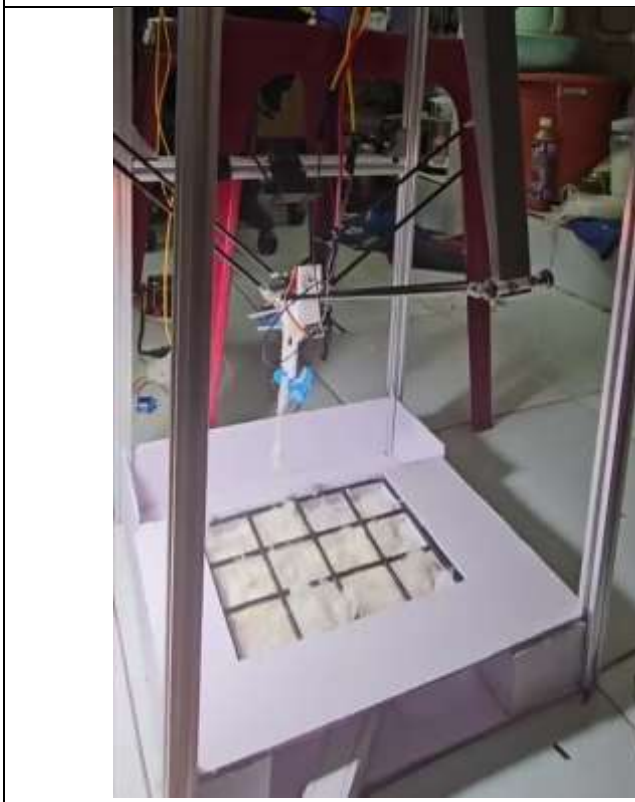
Robot Delta di chuyển đến vị trí cần gắp với độ cao  $Z = 5\text{cm}$  phía trên điểm đích, sau đó hạ xuống để đưa đầu kẹp vào vị trí chính xác. Đầu kẹp được thiết kế chuyên dụng nhằm chỉ kẹp chính xác tạp chất mà không làm ảnh hưởng đến bông xung quanh.



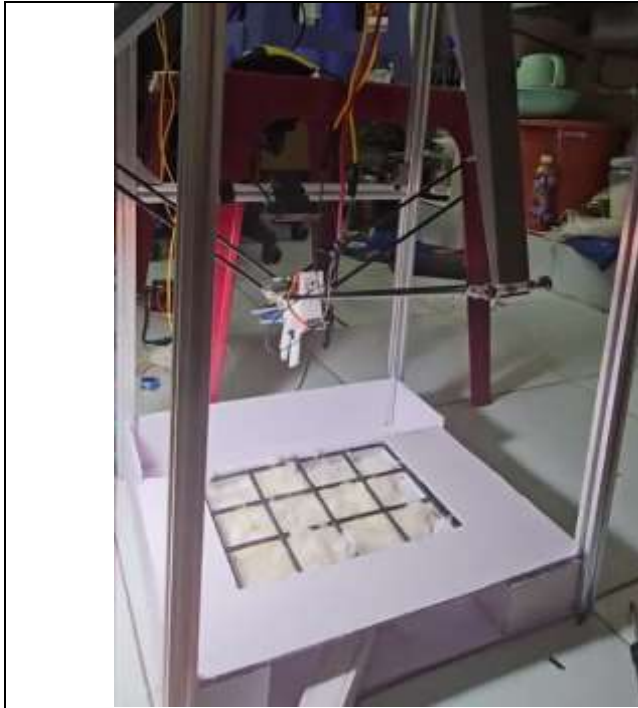
Khi Robot Delta đến đúng vị trí, đầu kẹp sẽ hoạt động và tiến hành kẹp lấy tạp chất ra khỏi khay. Đầu kẹp được thiết kế đặc biệt để chỉ nhắm mục tiêu vào tạp chất mà không làm ảnh hưởng đến bông xung quanh.



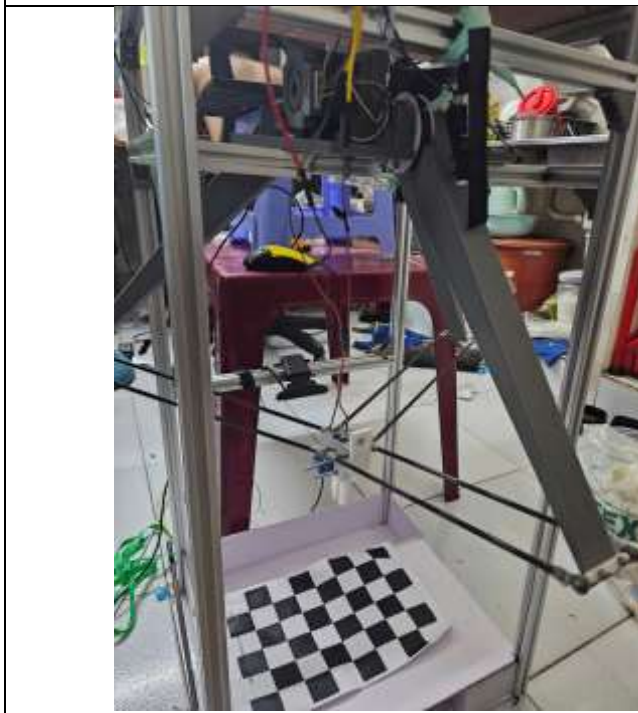
Sau khi kẹp thành công, robot nâng vật thể lên với độ cao  $Z = 5\text{cm}$



Sau khi tạp chất được kẹp lên, Robot Delta di chuyển đến khu vực xử lý rác (thường là một khay hoặc thùng riêng biệt) và thả tạp chất xuống.



Sau khi tạp chất được kẹp lên, Robot Delta di chuyển đến khu vực xử lý rác và thả tạp chất xuống.



Khi đã hoàn tất 1 chu trình thì Robot sẽ về vị trí Home để đợi đối tượng tiếp theo

Đánh giá hệ thống phân loại với cấu trúc mô hình YOLOV11 với hiệu suất cao nhất:

Với cách dán nhãn Bounding box, tập dữ liệu ảnh 13082 ảnh được chia theo tỉ lệ (training set = 80%, valid set = 10%, test set = 10%). Quá trình huấn luyện và kiểm thử được thực hiện trong điều kiện ánh sáng bình thường nhằm đánh giá khả năng hoạt động của hệ thống trong môi trường thực tế.

Kết quả thực nghiệm:

Bảng 6.3: Đánh giá độ nhận diện hệ thống

Lần thử	Tổng số mẫu	Nhận đúng	Bỏ sót	Sai lệch
1	120	113	5	2
2	120	115	4	1
3	120	114	3	3

Đánh giá tham số:

Chỉ số trung bình qua các lần thử nghiệm cho thấy mô hình đạt hiệu suất cao:

- Precision = 0.986: Đây là chỉ số rất ấn tượng, cho thấy tỷ lệ dự đoán sai (false positives) cực thấp. Mô hình gần như không đánh nhầm bông thành tạp chất, đảm bảo chất lượng sản phẩm đầu ra.
- Recall = 0.955: Tỷ lệ bỏ sót (miss rate) chỉ khoảng 4.5% , nghĩa là mô hình phát hiện được 95.5% các tạp chất thực tế. Tuy nhiên, với các vật thể nhỏ hoặc mờ (ví dụ: lông vũ), tỷ lệ bỏ sót tăng lên đáng kể.
- F1-score = 0.961: Sự cân bằng giữa Precision và Recall cho thấy mô hình vừa đảm bảo độ chính xác cao, vừa duy trì khả năng phát hiện ổn định. Đây là kết quả lý tưởng cho bài toán tự động hóa, nơi yêu cầu cả độ tin cậy và tính toàn diện.

Các chỉ số này phản ánh khả năng hoạt động mạnh mẽ của mô hình trong môi trường kiểm soát, nhưng cũng cho thấy tiềm năng cải tiến nếu tối ưu hóa các yếu tố kỹ thuật như ánh sáng, tốc độ chụp ảnh, và độ phân giải của camera.

*Nhận xét:*

Kết quả thực nghiệm cho thấy mô hình YOLOv11 được huấn luyện trên tập dữ liệu mở rộng 13082 ảnh hoạt động ổn định và hiệu quả trong điều kiện ánh sáng bình thường. Mô hình thể hiện khả năng nhận diện tốt hầu hết các loại tạp chất phổ biến như xơ cây, vỏ cây, nhựa, và gỗ, với tỷ lệ chính xác cao và ít xảy ra sai lệch. Đặc biệt, mô hình vẫn duy trì hiệu suất ổn định ngay cả khi đối tượng có hình dạng hoặc màu sắc tương đồng với bông, như nilon trắng hoặc lông vũ mềm, nhờ vào việc tối ưu hóa đặc trưng trong quá trình huấn luyện. Tuy nhiên, một số hạn chế vẫn tồn tại, bao gồm:

- Bỏ sót một số vật thể khó phân biệt : Các vật thể như lông vũ hoặc nilon mỏng đôi khi bị bỏ sót do màu sắc hòa lẫn với bông hoặc bị biến dạng bởi chuyển động nhẹ.
- Ảnh hưởng của ánh sáng môi trường: Trong điều kiện ánh sáng yếu hoặc phản quang, độ tương phản giữa bông và tạp chất giảm, dẫn đến một số trường hợp gán nhãn sai.
- Hạn chế phân cứng: Camera hiện tại chưa đủ độ phân giải và tốc độ chụp để bắt kịp các vật thể nhỏ, nhẹ hoặc chuyển động nhanh.

Những hạn chế này cho thấy cần cải thiện hệ thống chiếu sáng, nâng cấp phần cứng, và mở rộng tập dữ liệu với các tình huống phức tạp để nâng cao tính tổng quát của mô hình.

## **Tổng kết chương 6**

Chương 6 đã chứng minh hiệu quả của hệ thống kiểm chứng và đánh giá thực nghiệm: từ việc xác minh độ chính xác động học nghịch của Robot Delta (sai số dưới 1 mm) đến thử nghiệm mô hình YOLO trên ba giai đoạn dữ liệu (từ hơn 2.000 lên 13.000 ảnh) với tỉ lệ nhận diện cuối cùng đạt gần 90% và độ tin cậy 80–90%. Quy trình vận hành tích hợp camera và robot cho khả năng phát hiện, kẹp và loại bỏ tạp chất chính xác, với precision 98,6%, recall 95,5% và F1-score 96,1%. Dù vẫn còn hạn chế khi gặp tạp chất nhỏ, ánh sáng kém hoặc phản xạ cao, chương này khuyến nghị cải thiện hệ thống chiếu sáng, nâng cấp camera và mở rộng dữ liệu huấn luyện để nâng cao tính ổn định và ứng dụng thực tế.

## KẾT LUẬN

Sau quá trình nghiên cứu và thực hiện đề tài: “Nghiên cứu và phát triển hệ thống thị giác máy tính tích hợp robot để loại bỏ tạp chất trong sản xuất sợi”, nhóm đã đạt được những kết quả như sau:

- Tìm hiểu cấu trúc và thiết kế mô hình Robot Delta hoàn chỉnh trên phần mềm Solidworks.
- Thi công phần cứng Robot Delta và tủ điện đảm bảo an toàn, thẩm mỹ, sáng tạo.
- Tính toán động học thuận, nghịch và kiểm nghiệm động học trên phần mềm Visual Studio Code. Xác định không gian làm việc cho Robot trên phần mềm Matlab.
- Điều khiển Robot trên phần mềm Arduino IDE và tính toán, thiết kế giao diện người dùng trên phần mềm Visual Studio Code.
- Xây dựng và huấn luyện mô hình YOLOv11 để phát hiện tạp chất trong sợi bông
- Hoàn thành luận văn.

Những hạn chế còn tồn tại trong quá trình thực hiện đề tài như sau:

- Tốc độ loại bỏ tạp chất chưa cao, khi phân loại xong 1 đối tượng thì mới bắt đầu tiếp nhận đối tượng tiếp theo.
- Độ chính xác của quá trình loại bỏ chịu ảnh hưởng bởi độ tương phản ánh sáng và màu sắc của môi trường xung quanh.
- Việc thiết kế và thi công phần cứng chưa tối ưu, khi Robot hoạt động ở tốc độ cao thì Robot có hiện tượng rung.
- Đầu gấp chưa được tối ưu.

### Hướng phát triển

- Điều chỉnh lại phần cứng (thiết kế lại một số chi tiết Robot) và thay thế một số thiết bị (động cơ, driver) để tốc độ hoạt động của Robot tối ưu.
- Thiết kế lại mô hình điều khiển và quy hoạch cơ cấu dây để tăng độ ổn định cho hệ thống phát hiện và loại bỏ.
- Thay đổi cơ cấu phần cứng để cải thiện tính ứng dụng phát hiện và loại bỏ tạp chất tối ưu hơn
- Tăng số lượng đối tượng huấn luyện để phân loại đa dạng hơn.
- Cải thiện phần cứng để có thể có hiệu suất loại bỏ tạp chất trong bông cao hơn

## TÀI LIỆU THAM KHẢO

- [1] Robot Delta, wikipedia
- [2] A. Vina, *So sánh Ultralytics YOLO11 với các mẫu YOLO trước đó*, Ultralytic, Apr. 2, 2025. [Online]. Available: <https://www.ultralytics.com/vi/blog/comparing-ultralytics-yolo11-vs-previous-yolo-models>
- [3] A. Vina, *Vision AI để phát hiện bất thường*, Ultralytic, Feb. 19, 2025. [Online]. Available: <https://www.ultralytics.com/vi/blog/vision-ai-for-anomaly-detection-a-quick-overview>
- [4] Jye Sawtell-Rickson, What Is Computer Vision  
<https://builtin.com/machine-learning/computer-vision>
- [5] AITHIETKE, *Sử dụng Roboflow cho bài toán Object Detection đơn giản*, Jun.23, 2022. [Online]. Available: <https://aithietke.com/su-dung-roboflow-cho-bai-toan-object-detection-don-gian/>
- [6] Hòa, *Kaggle Tutorial*, Kaggle , 2025. [Online]. Available: <https://www.kaggle.com/code/nguyenhoa/kaggle-tutorial>
- [7] A. Vina, *Cách sử dụng Ultralytics YOLO11 để phát hiện đối tượng*, Ultralytic, Nov. 29, 2025. [Online]. Available: <https://www.ultralytics.com/vi/blog/how-to-use-ultralytics-yolo11-for-object-detection>
- [8] Phạm Đình Khánh, *YOLO You Only Look Once*, Mar. 09, 2020. [Online]. Available: <https://phamdinhhkhanh.github.io/2020/03/09/DarknetAlgorithm.html>
- [9] Ultralytics YOLO Tài liệu, *Ultralytics YOLO11*, Ultralytic. [Online]. Available: <https://docs.ultralytics.com/vi/models/yolo11/>
- [10] Soumyadip, Subham, Bhomik Sharma., *Mastering All YOLO Models from YOLOv1 to YOLOv12: Papers Explained*, LearnOpenCV, Dec. 26, 2023. [Online]. Available: <https://learnopencv.com/mastering-all-yolo-models/#yolo11>
- [11] Rahima Khanam\* and Muhammad Hussain , *YOLOV11: An Overview Of The Key Architectural Enhancements* -
- [12] Robert L. Williams , *"The Delta Parallel Robot: Kinematics Solutions"*, Ohio University, October 2016.
- [13] Antonio Torralba, Phillip Isola, and William Freeman, *Foundations of Computer Vision*

[14] Juan R. Terven and Diana M. Cordova-Esparza, A Comprehensive Review Of Yolo Architectures In Computer Vision: From Yolov1 To Yolov8 And Yolo-Nas

<https://arxiv.org/pdf/2304.00501>

[15] Ths. Trương Phước Thọ Vũ Minh Tuấn, Phạm Đâu Huy Hoàng, Phạm Xuân Giao, Hệ Thống Robot Delta Phân Loại Sản Phẩm Bằng Xử Lý Ảnh

[16] Robert L. Williams II, Ph.D, The Delta Parallel Robot: Kinematics Solutions, , October 2016

[17] Shiva Shrestha, Samman Shrestha, Prateek Paudel, Sudarshan Gurung, Smita Adhikari, Automated Waste Sorting with Delta Arm and YOLOv8 Detection

[18] Paul Zsombor-Murray, Descriptive Geometric Kinematic Analysis of Clavel’s “Delta” Robot, 2004

[18] N. N. Trường and N. T. B. Thạch, “Thiết kế mô hình Robot Delta ứng dụng phân loại rác thải,” Trường Đại học Sư phạm Kỹ thuật TP. Hồ Chí Minh, TP. Hồ Chí Minh, 2023.

[19] Wikipedia, “Robot Delta,” 10/2009. [Online]. Available: [https://en.wikipedia.org/wiki/Delta\\_robot](https://en.wikipedia.org/wiki/Delta_robot).

[20] L. T. a. L. N. V. Vinh, “Thiết kế và chế tạo robot Delta,” Đại học Bách khoa – Đại học Đà Nẵng, Đà Nẵng, 2019.

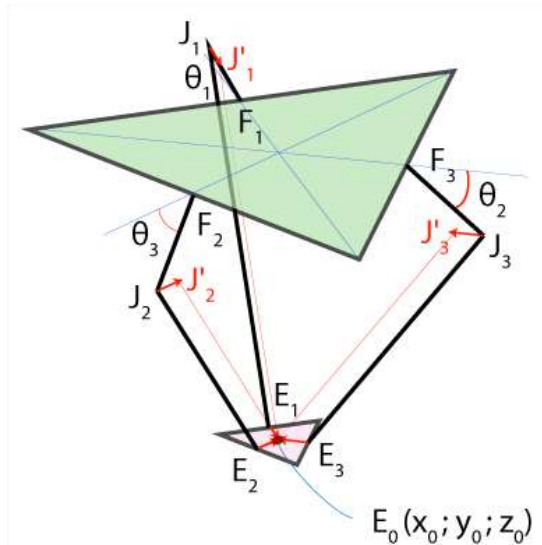


# PHỤ LỤC 1

## 1.. Tính toán động học cho Robot Delta

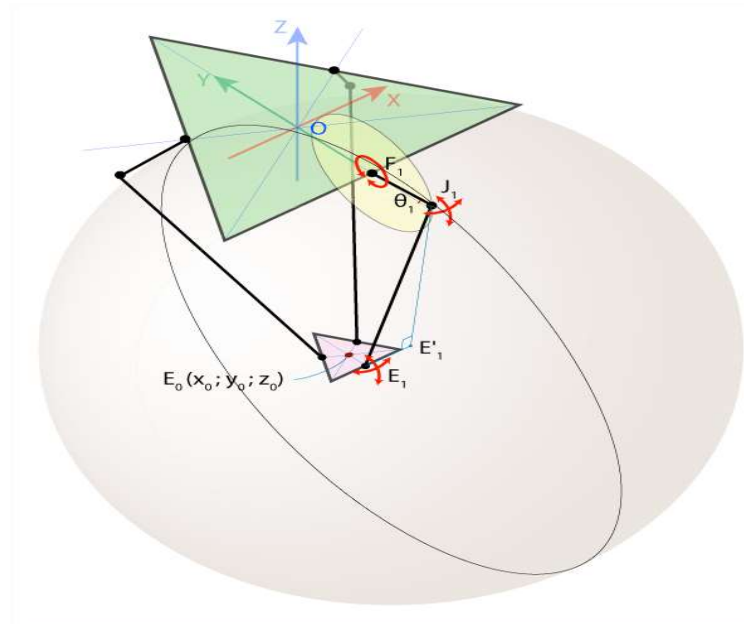
### Động học nghịch

Với bài toán động học nghịch chúng ta sẽ biết được tọa độ điểm cuối cùng của Robot  $E_0(x_0, y_0, z_0)$  là trọng tâm của tam giác để di động và các thông số cần xác định là 3 góc  $\theta$  được tạo bởi các khớp trên và mặt phẳng của đế cố định. Với hệ quy chiếu sẽ được chọn có gốc tọa là trọng tâm của tam giác đế cố định.



Hình 3-1: Mô hình động học robot delta

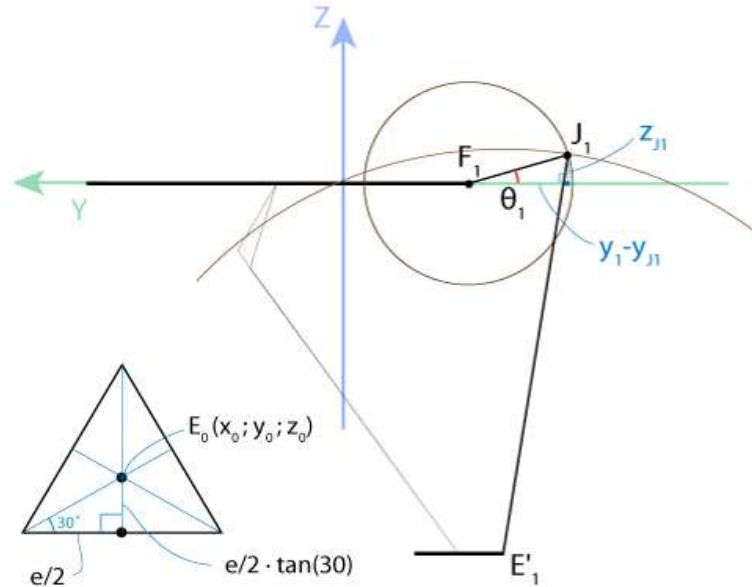
Do khớp nối thiết kế của robot  $F_1J_1$  chỉ có thể xoay trong mặt phẳng  $YZ$ , tạo thành hình tròn với tâm tại điểm  $F_1$  và bán kính  $r_f$  đồng thời khớp hình bình hành  $J_1E_1$  có thể xoay tự do tương đối với  $E_1$ , tạo thành hình cầu với tâm tại điểm  $E_1$  và bán kính  $r_e$ .



Hình 3-2: Giao tuyến của hình cầu và mặt phẳng  $YZ$

Giao tuyến của hình cầu này và mặt phẳng YZ là một đường tròn có tâm tại điểm  $E'_1$  và bán kính  $E'_1J_1$  (trong đó  $E'_1$  là phép chiếu của điểm  $E_1$  trên mặt phẳng YZ). Điểm  $J_1$  có thể được tìm thấy bây giờ như là giao điểm của các đường tròn bán kính đã biết với tâm là  $E'_1$  và  $F_1$  (chúng ta chỉ nên chọn một điểm giao nhau với tọa độ Y nhỏ hơn). Và nếu ta biết  $J_1$ , ta có thể tính được góc  $\theta_1$ .

Xét mặt phẳng YZ:



Hình 3-3: Mặt phẳng YZ

Xét tam giác để di động, ta tìm được:

$$E_0E_1 = \frac{e}{2} \tan 30^\circ = \frac{e}{2\sqrt{3}}$$

Nên

$$E_1 \left( x_0; y_0 - \frac{e}{2\sqrt{3}}; z_0 \right) \Rightarrow E'_1 \left( 0; y_0 - \frac{e}{2\sqrt{3}}; z_0 \right)$$

Và 
$$E_1J_1 = x_0 \rightarrow E'_1J_1 = \sqrt{E_1J_1^2 - E_1E_1'^2} = \sqrt{r_e^2 - x_0^2}$$

Xét tam giác để cố định, ta tìm được:

$$OF_1 = \frac{f}{2} \tan 30^\circ = \frac{f}{2\sqrt{3}}$$

Nên:

$$F_1 \left( 0; -\frac{f}{2\sqrt{3}}; 0 \right)$$

Áp dụng định lý Pytago cho 2 tam giác vuông  $F_1J_1J'_1$  và  $J_1E_1E'_1$ :

$$\begin{cases} (y_{J_1} - y_{F_1})^2 + (z_{J_1} - z_{F_1})^2 = r_f^2 \\ (y_{J_1} - y_{E_1})^2 + (z_{J_1} - z_{E_1})^2 = r_e^2 - x_0^2 \end{cases}$$

Ta được:

$$\begin{cases} (y_{J_1} + \frac{f}{2\sqrt{3}})^2 + z_{J_1}^2 = r_j^2 \\ (y_{J_1} - y_0 + \frac{e}{2\sqrt{3}})^2 + (z_{J_1} - z_0)^2 = r_e^2 - x_0^2 \end{cases}$$

Tọa độ điểm  $J_1$  là nghiệm hệ phương trình giao điểm của 2 đường tròn:

$$\Rightarrow J_1(0; y_{J_1}; z_{J_1})$$

Từ lấy phương trình thứ nhất trừ phương trình thứ hai ta thu được:

$$2m_1 y_{J_1} + 2z_0 z_{J_1} + m_2 = 0$$

Trong đó:

$$m_1 = \frac{f}{2\sqrt{3}} + y_0 - \frac{e}{2\sqrt{3}} \quad m_2 = \left(\frac{f}{2\sqrt{3}}\right)^2 - \left(y_0 - \frac{e}{2\sqrt{3}}\right)^2 - z_0^2 - r_j^2 + (r_e^2 - x_0^2)$$

Từ biến đổi ta được:

$$z_{J_1} = \frac{-2y_{J_1}m_1 - m_2}{2z_0}$$

Thay vào phương trình đầu của ta được:

$$\left(1 + \frac{m_1^2}{z_0^2}\right) y_{J_1}^2 + \frac{f}{\sqrt{3}} + \frac{m_1 m_2}{z_0^2} y_{J_1} - \left(\frac{f}{2\sqrt{3}}\right)^2 + \left(\frac{m_2}{2z_0}\right)^2 - r_j^2 = 0$$

Đặt:

$$\Delta = b^2 - 4ac$$

Với:

$$\begin{cases} a = 1 + \frac{m_1^2}{z_0^2} \\ b = \frac{f}{\sqrt{3}} + \frac{m_1 m_2}{z_0^2} \\ c = \left(\frac{f}{2\sqrt{3}}\right)^2 + \left(\frac{m_2}{2z_0}\right)^2 - r_j^2 \end{cases}$$

Vì điểm  $J_1$  luôn ở vị trí âm của trục y nên ta chọn nghiệm:

$$y_{J_1} = \frac{-b - \sqrt{\Delta}}{2a}$$

Thay nghiệm phương trình vào phương trình ta được:

$$z_{J_1} = \frac{-2 \left( y_{J_1} = \frac{-b - \sqrt{\Delta}}{2a} \right) m_1 - m_2}{2z_0}$$

Góc  $\theta_1$  là góc hợp bởi khớp trên  $F_1 J_1$  và mặt phẳng đế cố định.

Ta có:

$$J_1(0; y_{J_1}; z_{J_1})$$

Ta tìm được:

$$\theta_1 = a \tan 2 \left( \frac{z_{J_1}}{y_{F_1} - y_{J_1}} \right)$$

Khi đã tính được góc  $\theta_1$  nhóm lợi dụng tính đối xứng của Robot Delta để tính toán góc  $\theta_2$  và  $\theta_3$ . Bằng cách xoay hệ trục xyz  $120^\circ$  với tọa độ điểm cuối, cùng của Robot mới  $E'_0(x'_0, y'_0, z'_0)$  và sử dụng cùng những thuật toán ở trên để tìm góc  $\theta_2$ . Và để tìm góc  $\theta_3$  chỉ cần xoay hệ trục xyz 1 góc  $-120^\circ$ .

Nhóm đặt hệ tọa độ ban đầu là A thì điểm  $E_0$ , nhìn từ hệ tọa độ A sẽ là  $A_{E_0} = (x_0, y_0, z_0)$ . Khi xoay từ hệ trục tọa độ A một góc theo trục z thì sẽ nhận được hệ tọa độ mới B với tọa độ điểm cuối cùng  $E_0$  nhìn từ hệ tọa độ  $B_{E_0} = (x'_0, y'_0, z'_0)$ . Ta có công thức của phép xoay từ hệ tọa độ A qua hệ tọa độ B như sau:

$$A_{E_0} = {}^A R \cdot B_{E_0}$$

Để tìm được  $B_{E_0} = (X'_0, Y'_0, Z'_0)$ , ta cần nhân 2 vế của phương trình cho  $({}^A R)^T$

$$({}^A R)^T \cdot A_{E_0} = ({}^A R)^T \cdot {}^A R \cdot B_{E_0}$$

Với

$$({}^A R)^T \cdot {}^A R = 1$$

Từ suy ra được:

$$B_{E_0} = ({}^A R)^T \cdot A_{E_0}$$

Bằng cách xoay từ hệ tọa độ A sang hệ tọa độ B quanh trục z với góc  $120^\circ$  ta suy được phương trình:

$$({}^A R)^T = (R_z(120^\circ))^T = \begin{bmatrix} \cos(120^\circ) & \sin(120^\circ) & 0 \\ -\sin(120^\circ) & \cos(120^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ngoài ra, ta có ma trận chuyển đổi hệ tọa độ tổng quát từ hệ tọa độ A sang hệ tọa độ B. Với lần lượt là phép xoay quanh trục z 1 góc  $120^\circ$  và vì chỉ xoay tại chỗ mà không tính tiến. Từ phương trình (4.28) suy ra phương trình như sau:

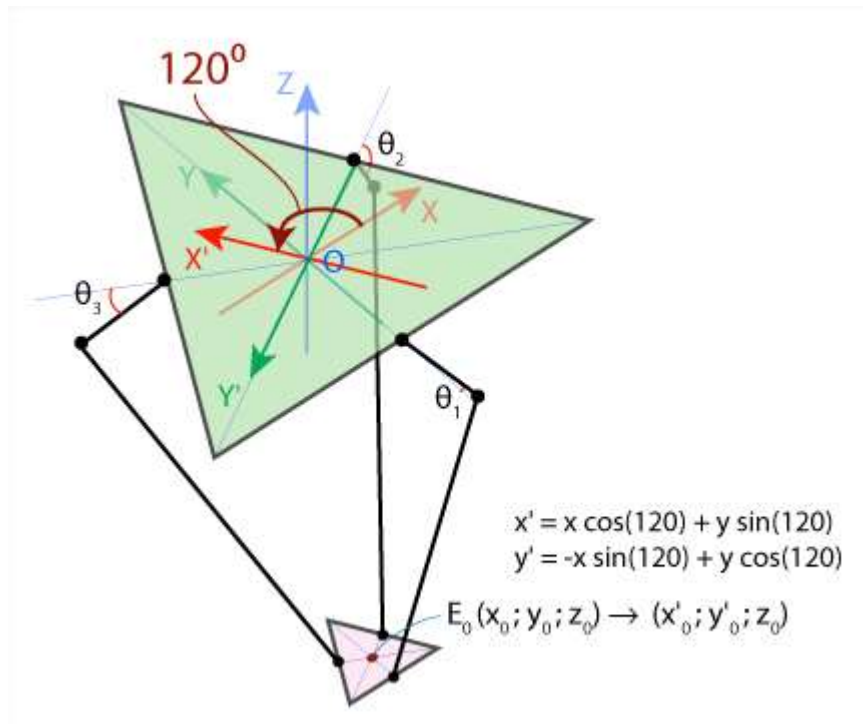
$B_{E_0} = (R_z(120^\circ))^T \cdot A_{E_0}$  Tiếp tục thay phương trình vào phương trình ta được phương

trình:

$$B_{E_0} = (R_z(120^\circ))^T \cdot A_{E_0} = \begin{bmatrix} x_0 \cos(120)^\circ + y_0 \sin(120)^\circ \\ -x_0 \sin(120)^\circ + y_0 \cos(120)^\circ \\ z_0 \\ 1 \end{bmatrix}$$

Vậy nhóm đã tìm được hệ phương trình để tính góc  $\theta_2$  khi xoay quanh hệ trục tọa độ 1 góc  $120^\circ$  theo trục z:

$$\begin{cases} x_0' = x_0 \cos(120)^\circ + y_0 \sin(120)^\circ \\ y_0' = -x_0 \sin(120)^\circ + y_0 \cos(120)^\circ \\ z_0' = z_0 \end{cases}$$



Hình 3-4: Hệ tọa độ mới  $(x'_0, y'_0, z'_0)$

Ngược lại với góc  $\theta_2$ , tiến hành xoay hệ tọa độ A quanh trục z 1 góc  $-120^\circ$  tính toán tương tự cho góc  $\theta_3$ , ta sẽ thu được hệ phương trình sau:

$$\begin{cases} x_0' = x_0 \cos(120)^\circ - y_0 \sin(120)^\circ \\ y_0' = x_0 \sin(120)^\circ + y_0 \cos(120)^\circ \\ z_0' = z_0 \end{cases}$$

### . Động học thuận

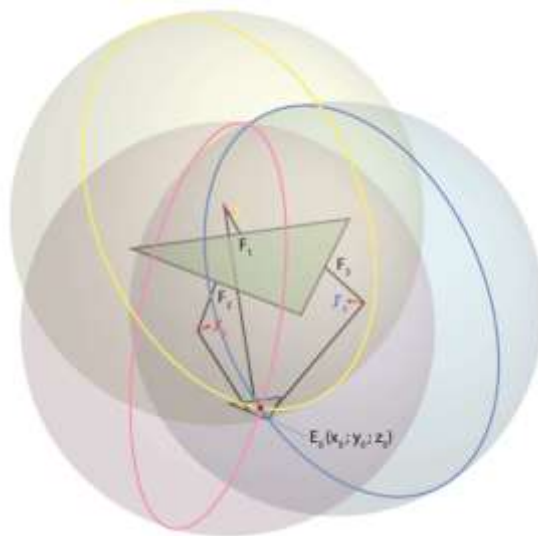
Mục đích của bài toán động học thuận là xác định vị trí của khâu tác động cuối của robot khi biết các biến khớp của robot. Từ các góc là trọng tâm của tam giác để di động.

Từ các góc  $\theta_1, \theta_2, \theta_3 \rightarrow$  Tọa độ  $(x_0, y_0, z_0)$  của điểm  $E_0$ .

Nếu ta đã biết góc  $\theta_x$ , chúng ta có thể dễ dàng tìm thấy tọa độ của các điểm  $J_1, J_2$  và  $J_3$ . Các khớp nối  $J_1E_1, J_2E_2$  và  $J_3E_3$  có thể tự do xoay quanh các điểm  $J_1, J_2$  và  $J_3$ . tương ứng, tạo thành ba hình cầu có bán kính  $r_e$ .

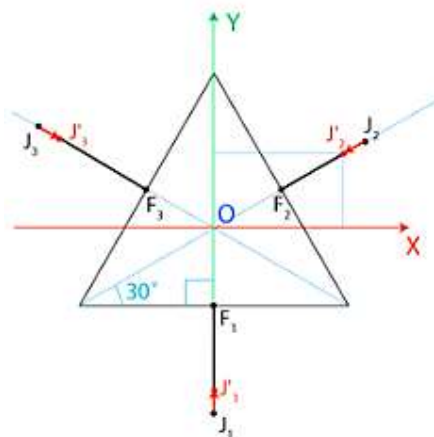
Ta di chuyển các tâm của các hình cầu từ các điểm  $J_1, J_2$  và  $J_3$  đến các điểm  $J'_1, J'_2$  và  $J'_3$  bằng cách sử dụng các vector chuyển tiếp  $E_1E_0, E_2E_0$  và  $E_3E_0$  tương ứng.

Sau quá trình chuyển đổi này, tất cả ba hình cầu sẽ giao nhau tại một điểm:  $E_0$ , như được thể hiện trong hình phía dưới:



Hình 3-5: Mô hình động học robot delta

Vì vậy, để tìm tọa độ  $(x_0, y_0, z_0)$  của điểm  $E_0$ , chúng ta cần giải quyết tập hợp ba phương trình như  $(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2 = r_e^2$ , nơi tọa độ các tâm hình cầu  $(x_j, y_j, z_j)$  và bán kính  $r_e$  đã biết.



Hình 3-6: Tọa độ các điểm  $J'_1, J'_2, J'_3$  trên mặt phẳng  $Oxy$

Theo hình học:

$$OF_1 = OF_2 = OF_3 = \frac{f}{2} \tan(30^\circ) = \frac{f}{2\sqrt{3}}$$

Và

$$J_1 J'_1 = J_2 J'_2 = J_3 J'_3 = \frac{e}{2} \tan(30^\circ) = \frac{e}{2\sqrt{3}}$$

$$\begin{cases} F_1 J_1 = r_f \cos(\theta_1) \\ F_2 J_2 = r_f \cos(\theta_2) \\ F_3 J_3 = r_f \cos(\theta_3) \end{cases}$$

Từ

$$\begin{cases} J'_1 \left( 0; -\frac{f-e}{2\sqrt{3}} - r_f \cos \theta_1; -r_f \sin \theta_1 \right) \\ J'_2 \left( \left( \frac{f-e}{2\sqrt{3}} + r_f \cos \theta_2 \right) \cos 30^\circ; \left( \frac{f-e}{2\sqrt{3}} + r_f \cos \theta_2 \right) \sin 30^\circ; -r_f \sin \theta_2 \right) \\ J'_3 \left( \left( \frac{f-e}{2\sqrt{3}} + r_f \cos \theta_3 \right) \cos 30^\circ; \left( \frac{f-e}{2\sqrt{3}} + r_f \cos \theta_3 \right) \sin 30^\circ; -r_f \sin \theta_3 \right) \end{cases}$$

Trong các phương trình sau đây ta sẽ chỉ định tọa độ của các điểm  $J'_1, J'_2, J'_3$  là  $(x_1, y_1, z_1), (x_2, y_2, z_2)$  và  $(x_3, y_3, z_3)$ . Lưu ý rằng  $x_1 = 0$ . Dưới đây là phương trình của ba mặt cầu với giao điểm là  $E_0(x_0, y_0, z_0)$

$$\begin{cases} x^2 + (y - y_1)^2 + (z - z_1)^2 = r_e^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = r_e^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = r_e^2 \end{cases}$$

$$\Rightarrow \begin{cases} x^2 + y^2 + z^2 - 2y_1 y - 2z_1 z = r_e^2 - y_1^2 - z_1^2 & (1) \\ x^2 + y^2 + z^2 - 2x_2 x - 2y_2 y - 2z_2 z = r_e^2 - x_2^2 - y_2^2 - z_2^2 & (2) \\ x^2 + y^2 + z^2 - 2x_3 x - 2y_3 y - 2z_3 z = r_e^2 - x_3^2 - y_3^2 - z_3^2 & (3) \end{cases}$$

Đặt:  $w_i = x_i^2 + y_i^2 + z_i^2$ , ta được:

$$\begin{cases} x_2 x + (y_1 - y_2)y + (z_1 - z_2)z = \frac{(w_1 - w_2)}{2} & (4) = (1) - (2) \\ x_3 x + (y_1 - y_3)y + (z_1 - z_3)z = \frac{(w_1 - w_3)}{2} & (5) = (1) - (3) \\ (x_2 - x_3)x + (y_2 - y_3)y + (z_2 - z_3)z = \frac{(w_2 - w_3)}{2} & (6) = (2) - (3) \end{cases}$$

Từ (4) - (5):

$$x = a_1 z + b_1 \quad (7)$$

$$y = a_2 z + b_2 \quad (8)$$

Với:

$$a_1 = \frac{1}{d} [(z_2 - z_1)(y_3 - y_1) - (z_3 - z_1)(y_2 - y_1)]$$

$$b_1 = -\frac{1}{2d}[(w_2 - w_1)(y_3 - y_1) - (w_3 - w_1)(y_2 - y_1)]$$

$$a_2 = -\frac{1}{d}[(z_2 - z_1)x_3 - (z_3 - z_1)x_2]$$

$$b_2 = \frac{1}{2d}[(w_2 - w_1)x_3 - (w_3 - w_1)x_2]$$

$$d = (y_2 - y_1)x_3 - (y_3 - y_1)x_2$$

Thế (7), (8) vào (1) ta được:

$$(a_1^2 + a_2^2 + 1)z^2 + 2(a_1 + a_2(b_2 - y_1) - z_1)z + (b_1^2 + (b_2 - y_1)^2 + z_1^2 - r_e^2) = 0$$

Cuối cùng, ta cần giải phương trình này và tìm  $z_0$  (chúng ta nên chọn gốc phương trình âm nhỏ nhất), và sau đó tính  $x_0$  và  $y_0$  từ (7) và (8).

$$az^2 + bz + c = 0$$

Với:

$$a = a_1^2 + a_2^2 + 1$$

$$b = 2(a_1b_1 + a_2(b_2 - y_1) - z_1)$$

$$c = b_1^2 + (b_2 - y_1)^2 + z_1^2 - r_e^2$$

$$\text{Đặt } \Delta = b^2 - 4ac$$

Do mô hình Robot Delta có khâu cuối luôn ở vị trí âm của trục  $z$  nên ta lấy nghiệm âm:

$$y_0 = \frac{-b - \sqrt{\Delta}}{2a}$$

## PHỤ LỤC 2

### 1. OUTPUT của YOLO:

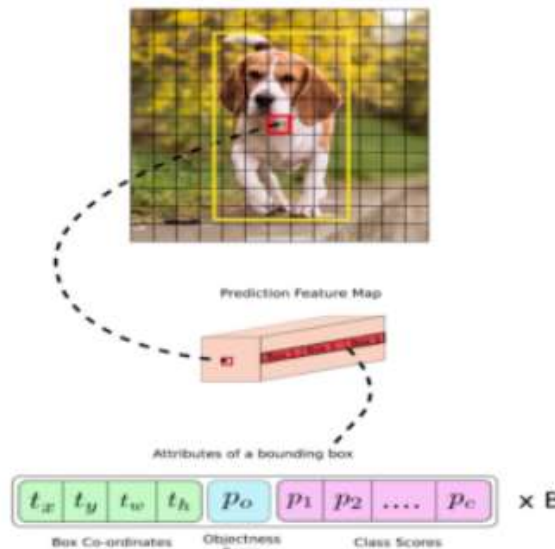
Output của mô hình YOLO là một véc tơ sẽ bao gồm các thành phần:

$$y^T = [p_0, \langle t_x, t_y, t_w, t_h \rangle, \langle p_1, p_2, \dots, p_c \rangle]$$

Trong đó:

- $\theta_1 = a \tan 2 \left( \frac{z_{J_1}}{y_{F_1} - y_{J_1}} \right)$ : xác suất dự báo vật thể xuất hiện trong bounding box.
- $\langle t_x, t_y, t_w, t_h \rangle$ : giúp xác định bounding box. Trong đó  $t_x, t_y$  là tọa độ tâm và  $t_w, t_h$  là kích thước rộng, dài của bounding box.
- $\langle p_1, p_2, \dots, p_c \rangle$ : véc tơ phân phối xác suất dự báo của các classes.

Việc hiểu output khá là quan trọng để chúng ta cấu hình tham số chuẩn xác khi huấn luyện model qua các open source như darknet. Như vậy output sẽ được xác định theo số lượng classes theo công thức (n\_class+5). Nếu huấn luyện 80 classes thì bạn sẽ có output là 85. Trường hợp bạn áp dụng 3 anchors/cell thì số lượng tham số output sẽ là:



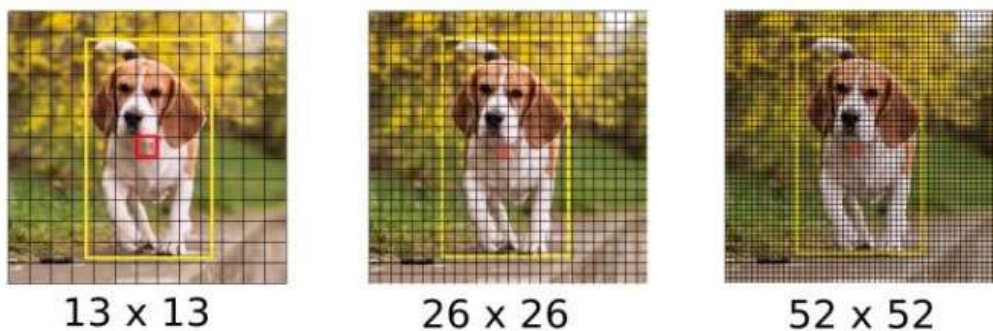
Hình 1: Kiến trúc một output của model

### 2. Feature map

Tương tự như SSD, YOLO cũng dự đoán vật thể trên nhiều feature map. Trên mỗi cell của feature map sẽ có 3 anchor box để dự đoán vật thể, như vậy ta tính được số lượng các anchor box trong mô hình YOLO là (3 feature map x 3 anchor box) = 9. Từ đó ta có thể tính được số lượng anchor box trên một bức ảnh là:

$$(13 \times 13 + 26 \times 26 + 52 \times 52) \times 3 = 10647 \text{ (anchor box)}$$

Với số lượng lớn thể này thì quá trình huấn luyện sẽ rất chậm bởi ta phải dự báo nhãn và bounding box trên từng anchor box.



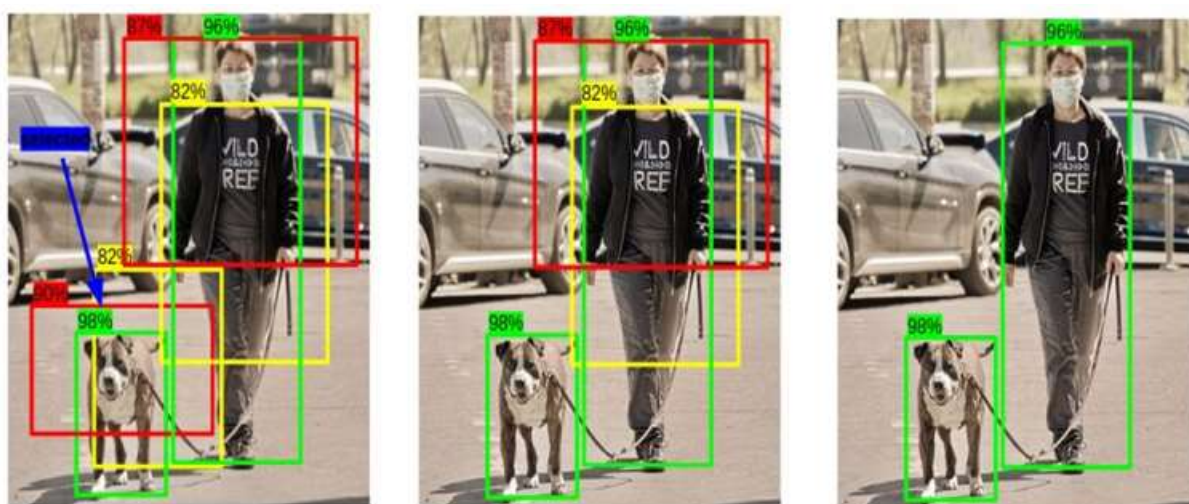
Hình 2: Các feature maps của mạng YOLOv3 với input shape là 416x416, output là 3 feature maps

### 3. Non-max suppression

Non Maximum Suppression (NMS) là một bước hậu xử lý mà hầu hết các thuật toán Object Detection sau này đều sử dụng. Mục tiêu của NMS là lựa chọn một bounding box thích hợp nhất cho đối tượng.

NMS tính theo 2 tiêu chí: Objectiveness score được trả về bởi model -Overlap hoặc IOU giữa các bounding box.

Do thuật toán YOLO dự báo ra rất nhiều bounding box trên một bức ảnh nên đối với những cell có vị trí gần nhau, khả năng các khung hình bị overlap là rất cao. Trong trường hợp đó YOLO sẽ cần đến non-max suppression để giảm bớt số lượng các khung hình được sinh ra



Hình 3: Non-max suppression từ 3 bounding box xuống còn 1 bounding box

Nếu chỉ số này lớn hơn ngưỡng threshold thì điều đó chứng tỏ 2 bounding boxes đang overlap nhau rất cao. Ta sẽ xóa các bounding có xác suất thấp hơn và giữ lại bounding có xác

suất cao nhất. Cuối cùng, ta thu được một bounding box duy nhất cho một vật thể.

#### 4. Loss Function

Hàm loss function của YOLO chia thành 2 phần:  $z_{loc}$  (localization loss) đo lường sai số của bounding box và  $z_{cls}$  (confidence loss) đo lường sai số của phân phối xác suất các classes.

$$z_{loc} = l_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} \left( (x_i - \mu_i)^2 + (y_i - \nu_i)^2 + (\sqrt{w_i} - \sqrt{w_i^g})^2 + (\sqrt{h_i} - \sqrt{h_i^g})^2 \right)$$

$$z_{cls} = \sum_{i=0}^{s^2} \sum_{j=0}^B 1_{ij}^{obj} C_{ij} + l_{noobj} (1 - 1_{ij}^{obj}) \sum_{c \in C} (p_i(c) - \mu_i(c))^2$$

$$z = z_{loc} + z_{cls}$$

Trong đó:

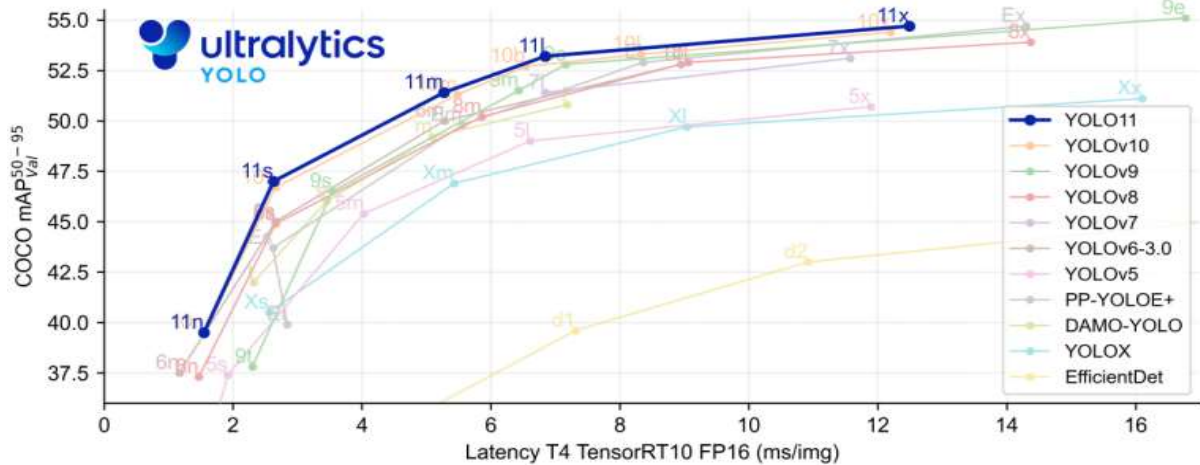
- $1_i^{obj}$ : Hàm indicator có giá trị 0, 1 nhằm xác định xem cell  $i$  có chứa vật thể hay không. Bằng 1 nếu chứa vật thể và 0 nếu không chứa.
- $1_{ij}^{obj}$ : Cho biết bounding box thứ  $j$  của cell  $i$  có phải là bounding box của vật thể được dự đoán hay không?
- $C_{ij}$ : Điểm tin cậy của ô  $i$ ,  $P(\text{contain object}) * \text{IoU}(\text{predict bbox}, \text{ground truth box})$
- $\mu_{ij}$ : Điểm tự tin dự đoán
- $C$ : Tập hợp tất cả các lớp
- $p_i(c)$ : Xác suất có điều kiện, có hay không ô  $i$  có chứa một đối tượng của lớp  $c \in C$
- $\mu_i(c)$ : Xác suất có điều kiện dự đoán.

Có thể ban đầu công thức trên khá khó hiểu với người bắt đầu. Chúng ta hãy hiểu đơn giản hóa mục đích của chúng:

- $z_{loc}$ : là hàm mất mát của bounding box dự báo so với thực tế
- $z_{cls}$ : Là là hàm mất mát của phân phối xác suất. Trong đó tổng đầu tiên là mất mát của dự đoán có vật thể trong cell hay không? Và tổng thứ 2 là mất mát của phân phối xác suất nếu có vật thể trong cell.

#### 5. So sánh các phiên bản Yolo

Ngoài ra để điều chỉnh phạt loss function trong trường hợp dự đoán sai bounding box ta thông qua hệ số điều chỉnh  $l_{coord}$  và ta muốn giảm nhẹ hàm loss function trong trường hợp cell không chứa vật thể bằng hệ số điều chỉnh  $l_{noobj}$



Hình 4: So sánh giữa các phiên bản YOLO

Mô hình YOLOv11 thể hiện những bước tiến đáng kể về cả tốc độ suy luận và độ chính xác so với các phiên bản trước. Trong phân tích benchmark, YOLOv11 được so sánh với nhiều thế hệ tiền nhiệm, từ YOLOv5 cho đến YOLOv10. Như trình bày trong Hình 4.10, YOLOv11 liên tục vượt trội hơn các mô hình này, đạt mAP cao hơn trên tập COCO trong khi vẫn duy trì tốc độ suy luận nhanh hơn. [9]

Biểu đồ so sánh hiệu năng hé lộ một số điểm chính:

- Các biến thể YOLOv11 (11n, 11s, 11m và 11x) tạo thành một “đường giới hạn hiệu năng” rõ rệt, với mỗi phiên bản đạt mAP50–95 trên COCO cao hơn tại ngưỡng độ trễ tương ứng.
- Đặc biệt, YOLOv11x đạt khoảng 54,5 % mAP50–95 ở độ trễ chỉ 13 ms, vượt trội so với tất cả các phiên bản YOLO trước đó.
- Các biến thể trung cấp, nhất là YOLOv11m, thể hiện hiệu quả vượt trội khi đạt độ chính xác tương đương các mô hình lớn thế hệ trước nhưng chỉ cần thời gian xử lý ít hơn đáng kể.

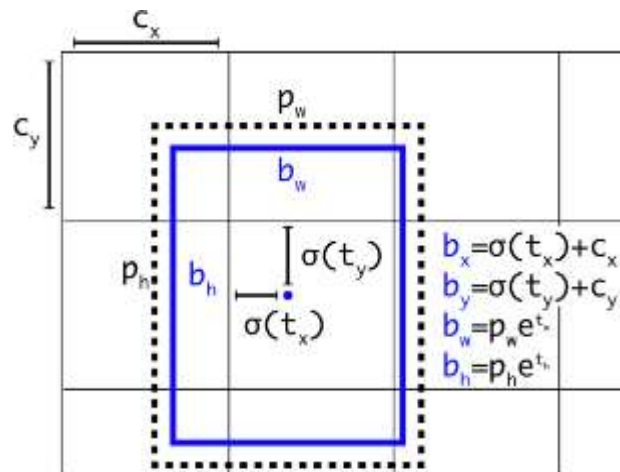
Một quan sát quan trọng khác là trong vùng độ trễ thấp (2–6 ms), YOLOv11s vẫn duy trì độ chính xác cao (khoảng 47 % mAP50–95) ở tốc độ mà trước đây chỉ đạt được bởi các mô hình kém chính xác hơn. Điều này đánh dấu bước tiến then chốt cho các ứng dụng thời gian thực, nơi cả tốc độ lẫn độ chính xác đều quan trọng.

Đường cong hiệu năng của YOLOv11 còn cho thấy khả năng mở rộng tốt giữa các biến thể, giúp người dùng dễ dàng lựa chọn cấu hình mô hình phù hợp với yêu cầu cân bằng giữa tốc độ và độ chính xác.

## 6. Dự báo bounding box

Như đã trình bày ở trên, để tìm bounding box ta cần quan tâm đến anchor box và cell.

Giả sử một anchor box có kích thước  $(p_w, p_h)$  tại cell trên feature map góc trên bên trái  $(c_x, c_y)$ , mô hình dự đoán 4 tham số  $(t_x, t_y, t_w, t_h)$ . Trong đó 2 tham số đầu là độ lệch so với góc trên bên trái của cell và 2 tham số sau là tỉ lệ với anchor box. Các tham số này giúp xác định được tọa độ bounding box có tâm  $(b_x, b_y)$  và kích thước  $(b_w, b_h)$ .



Hình 5. Công thức ước lượng bounding box từ anchor box