

ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN



NGUYỄN HỮU PHÚC

NGUYỄN HỮU NGHĨA

**PHÁT TRIỂN MÔ HÌNH CHĂM SÓC THÚ CỪNG BẰNG AI
VÀ IOT**

KHÓA LUẬN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC

TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2023.

ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN



NGUYỄN HỮU PHÚC

NGUYỄN HỮU NGHĨA

**PHÁT TRIỂN MÔ HÌNH CHĂM SÓC THÚ CỪNG BẰNG AI
VÀ IOT**

KHÓA LUẬN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC

NGƯỜI HƯỚNG DẪN: TS. TRẦN QUANG HUY

TP. HỒ CHÍ MINH THÁNG 6 NĂM 2023

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi, các số liệu và kết quả nghiên cứu nêu trong luận văn là trung thực, được các đồng tác giả cho phép sử dụng và chưa từng được công bố trong bất kì một công trình nào khác.

Tác giả luận văn

Nguyễn Hữu Phúc

Nguyễn Hữu Nghĩa

LỜI CẢM ƠN

Trước hết, nhóm em xin gửi lời cảm ơn chân thành và tri ân sâu sắc đến TS. Trần Quang Huy đã tận tình hướng dẫn để nhóm có thể hoàn thành các nghiên cứu của mình. Những ngày tháng dài cùng làm việc với thầy, nhóm đã trau dồi thêm nhiều kiến thức hữu ích và hiệu quả trong quá trình thực hiện đề tài.

Nhóm em cũng xin gửi lời cảm ơn đến các thầy cô trong khoa Công nghệ thông tin trường Đại học Sài Gòn đã có những góp ý chân thành cho nhóm nghiên cứu, đồng thời cung cấp nhiều kiến thức quý giá thông qua các môn học, các buổi chuyên đề để nhóm có thể hoàn thành tốt luận văn của mình.

Nhóm tác giả

Nguyễn Hữu Phúc

Nguyễn Hữu Nghĩa

MỤC LỤC

LỜI CAM ĐOAN	3
LỜI CẢM ƠN.....	4
MỤC LỤC	4
DANH MỤC CÁC CHỮ VIẾT TẮT.....	7
DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ.	8
MỞ ĐẦU	10
1. Tính cấp thiết của đề tài.....	10
2. Lý do chọn đề tài.	10
3. Mục tiêu đề tài.	10
4. Đối tượng nghiên cứu.....	10
5. Phạm vi nghiên cứu.....	11
6. Phương pháp nghiên cứu.....	11
7. Đóng góp của đề tài.....	11
8. Cấu trúc của khóa luận tốt nghiệp.....	11
CHƯƠNG 1. ĐẶT VẤN ĐỀ.....	12
1.1. Tổng quan bài toán chăm sóc thú cưng.....	12
1.1.1. Giới thiệu bài toán.....	12
1.1.2. Ứng dụng.....	13
1.1.3. Thách thức.....	13
1.2. Tổng quan các nghiên cứu nhận dạng động vật.....	13
1.3. Tổng quan các nghiên cứu phát hiện động vật trong nhà.....	14
1.4. Hướng giải quyết.....	15
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	17
2.1. Tổng quan về Thị Giác Máy Tính.....	17
2.1.1. Giới thiệu về Thị Giác Máy Tính.....	17
2.1.2. Các ứng dụng của Thị Giác Máy Tính.....	17
2.2. Tổng quan học sâu (Deep Learning).....	18
2.2.1. Giới thiệu về học sâu.....	18

2.2.2. Nguyên lý hoạt động của học sâu.....	19
2.2.3. Học sâu và xử lý ảnh.	21
2.3. Tổng quan Về YOLO (You Only Look One).	22
2.3.1. Giới thiệu về YOLO.....	22
2.3.2. Nguyên lý hoạt động của mạng YOLO.....	23
2.3.3. Cấu trúc mạng của YOLOv8.....	25
2.4. Tổng quan các thiết bị IoT sử dụng trong mô hình chăm sóc thú cưng.....	26
2.5. Sơ đồ hoạt động các thiết bị IoT.....	31
CHƯƠNG 3. XÂY DỰNG HỆ THỐNG NHẬN DẠNG THÚ CƯNG (MÈO) TRONG CÁC THIẾT BỊ IOT VÀ AI.....	34
3.1. Ứng dụng mô hình YOLOv8 vào mô hình chăm sóc thú cưng.	34
3.1.1 Chuẩn bị bộ dữ liệu.	34
3.1.2. Ứng dụng mô hình YOLOv8 đã huấn luyện.	35
3.2. Giao tiếp giữa thiết bị IoT với ứng dụng điện thoại.....	36
3.2.1. Tổng quan về React Native.	36
3.2.2. Tổng quan về Firebase.....	39
3.3.3. Xây dựng dữ liệu trên FireBase để kết nối ứng dụng điện thoại.....	40
3.3. Phát triển ứng dụng.....	41
3.4. Mô tả hoạt động tổng quát của mô hình.	45
3.4.1. Xây dựng mô hình 3D tự động hóa cho mèo.	45
3.4.2. Xây dựng mô hình tự hóa động hóa cho mèo.	46
3.5. Hoạt động của ứng dụng mobile.	48
3.6. Hoạt động của thiết bị IoT.....	49
3.7. Hoạt động của mô hình YOLO trong mô hình chăm sóc thú cưng.....	50
CHƯƠNG 4. KẾT QUẢ VÀ KHUYẾN NGHỊ.....	52
4.1. Kết quả huấn luyện mô hình YOLOv8.	52
4.2. Kết quả mô hình chăm sóc thú cưng.	55
4.3. Đánh giá kết quả.....	57
4.4. Hướng phát triển.....	57
KẾT LUẬN VÀ KHUYẾN NGHỊ.....	59

TÀI LIỆU THAM KHẢO	60
--------------------------	----

DANH MỤC CÁC CHỮ VIẾT TẮT

CNN	:	Convolutional neural network.
ML	:	Machine Learning
DL	:	Deep Learning
DNN	:	Deep Neural Network
IoT	:	Internet of Things
AI	:	Artificial Intelligence
YOLO	:	You Only Look Once
CSPNet	:	Cross Stage Partial Network
R-CNN	:	Region-based Convolutional Neural Network
FCOS Fully	:	Convolutional One-Stage Object Detection
HRNet32	:	High-Resolution Network
ResNet101	:	Residual Network
COCO	:	Common Objects in Context
SOTA	:	State-of-the-Art
IOU	:	Intersection over Union
C2F	:	Coarse-To-Fine
CSP	:	Cross-Stage-Partial
SPPF	:	Spatial Pyramid Pooling Fusion
PAN-FPN	:	Path Aggregation Network with Feature Pyramid Network
WebRTC	:	Web Real-Time Communications
RTSP	:	Real Time Streaming Protocol

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ.

Hình 1. Kiến trúc của mạng nơ ron [5].....	20
Hình 2. Kiến trúc tổng quan mạng nơron tích chập-CNN[6].....	21
Hình 3. Kiến trúc mạng YOLO[8].	23
Hình 4. Chia các ô lưới và cho đến khi nhận diện các đối tượng có trong bức hình[9].....	24
Hình 5. Espressif ESP32-DevKitC-32UE.	27
Hình 6. ESP32-CAM Development Board (with camera).	27
Hình 7. Step Motor Động Cơ Bước 42.....	28
Hình 8. Mạch nguồn cách ly AC-DC 12V 2A & Mạch Nguồn AC-DC 5V.....	28
Hình 9. Mạch điều khiển động cơ bước DRV8825 & Đế ra chân Driver động cơ bước DRV8825.....	28
Hình 10. Module 1 Relay Kích Mức Cao Thấp 5V.	29
Hình 11. Mạch Điều Khiển Động Cơ L298 DC Motor Driver.	29
Hình 12. Bơm Nước Mini DC - Bơm Chìm 3-5V.....	29
Hình 13. Mạch Ghi Phát Âm Thanh ISD1820 (20s) & Loa 5W 4R Tròn (size 52mm). ...	30
Hình 14. Mạch Khuếch Đại Âm Thanh LM386.	30
Hình 15. Cảm Biến Mức Nước.....	30
Hình 16. Cảm Biến Siêu Âm SRF05.....	31
Hình 17. Sơ đồ hoạt động của các thiết bị IoT.	32
Hình 18. Vẽ bounding box trên website roboflow.	34
Hình 19. Nhận diện mèo trên tập dữ liệu đã được huấn luyện.....	36
Hình 20. Dữ liệu được tạo trên FireBase.....	40
Hình 21. Trang đăng nhập, đăng ký.	42
Hình 22. Trang đặt lại mật khẩu.	42
Hình 23. Trang chính.....	43
Hình 24. Trang quan sát mèo.	43
Hình 25. Trang cài đặt.	44
Hình 26. Trang cá nhân người dùng.....	44
Hình 27. Tổng quan về mô hình 3D.....	45
Hình 28. Mặt Trước và Mặt Phải của Mô hình 3D.	45
Hình 29. Từ trên nhìn xuống của mô hình 3D.	46
Hình 30. Mặt trước và từ trên xuống của mô hình.	46
Hình 31. Mặt trái và phải mô hình.	47
Hình 32. Đồ đựng thức ăn và bơm nước uống cho mèo.	47
Hình 33. Chỗ dự trữ thức ăn cho mèo.	48
Hình 34. Sơ đồ lắp thiết bị IoT của mô hình cho ăn, uống.	48
Hình 35. Cách tính Intersection over Union (IOU) [19].	53
Hình 36. Cách tính Precision và Recall [20].	53

Hình 37. Kết quả bộ dữ liệu được huấn luyện trên mô hình YOLOv8.	55
Hình 38. Ảnh được chụp từ ESP32-Camera đã được nhận dạng.	56
Hình 39. Hình ảnh động cơ quay thức ăn và vòi nước cho mèo uống đang hoạt động.	56

MỞ ĐẦU

1. Tính cấp thiết của đề tài.

Trên thực tế ngày nay, thú cưng như chó, mèo được mọi người coi như “bạn thân”, hoặc một cách dễ thương. Những chú chó, mèo không đơn giản chỉ là những động vật, nó còn được coi là một thành viên trong gia đình. Vì vậy những người chủ rất quan tâm đến việc chăm sóc cho những thú cưng của mình. Họ không ngần ngại chi trả những số tiền không nhỏ để tìm người hay thuê dịch vụ “chăm sóc kỹ lưỡng cho thú cưng”. Vì vậy các trung tâm chăm sóc thú cưng ngày càng nhiều, nhưng lại ít có trung tâm nào hoạt động chuyên nghiệp, như là sử dụng sự thông minh của trí tuệ nhân tạo để chăm sóc và theo dõi thú cưng một cách dễ dàng và thuận tiện hơn.

2. Lý do chọn đề tài.

Dựa vào số liệu tình hình bệnh trên thú cưng nêu trên cho thấy mức độ nguy hiểm là rất cao, ảnh hưởng trực tiếp đến sức khỏe của thú cưng cũng như chủ nhân của thú cưng. Do đó để giảm mức ảnh hưởng và quan trọng nhất là giảm tỉ lệ ca tử vong, việc điều trị kịp thời và nhanh chóng cho các thú cưng là rất cần thiết. Mặc dù hiện nay, trên thế giới đã có nhiều nghiên cứu đưa ra các thuật toán tự động phát hiện bệnh ở thú cưng, tuy nhiên tại Việt Nam các nghiên cứu này chưa thực sự phát triển mạnh mẽ và rộng rãi, đặc biệt với nghiên cứu của sinh viên. Vì thế, đây là đề tài rất phù hợp để nghiên cứu đưa ra các phương án giải quyết tối ưu.

3. Mục tiêu đề tài.

- Mục tiêu của đề tài là đề xuất mô hình chăm sóc thú cưng.
- Cài đặt thực nghiệm phương pháp đề xuất.
- Đánh giá kết quả và đưa ra hướng phát triển cho tương lai.

4. Đối tượng nghiên cứu.

Đối tượng nghiên cứu của đề tài là mô hình chăm sóc thú cưng, thuật toán nhận diện đối tượng.

5. Phạm vi nghiên cứu.

- Thú nuôi trong nhà cụ thể là mèo.
- Các phương pháp tiếp cận trích xuất, phân loại, dự đoán bệnh thú nuôi và chăm sóc sức khỏe.

6. Phương pháp nghiên cứu.

- Tổng hợp, nghiên cứu và phân tích các bài báo, tài liệu liên quan đến đề tài.
- Thực nghiệm, nghiệm thu và đánh giá mô hình.
- Phân tích và đánh giá kết quả.

7. Đóng góp của đề tài.

- Đề xuất quy trình hoạt động của mô hình trên ảnh thời gian thực.
- Đề xuất sử dụng thuật toán YOLO trong Deep Learning để giải quyết bài toán nhận diện.
- Xây dựng mô hình và đánh giá kết quả.

8. Cấu trúc của khóa luận tốt nghiệp.

Cấu trúc của luận văn sẽ bao gồm 4 chương chính như sau:

- Chương 1: Đặt vấn đề.
- Chương 2: Cơ sở lý thuyết.
- Chương 3: Xây dựng hệ thống nhận dạng thú cưng (mèo) trong các thiết bị IoT và AI.
- Chương 4: Kết quả và khuyến nghị.

CHƯƠNG 1. ĐẶT VẤN ĐỀ.

1.1. Tổng quan bài toán chăm sóc thú cưng.

1.1.1. Giới thiệu bài toán.

Để quan sát thú cưng của mình mọi lúc và mọi nơi, cũng như quan sát hành động của chúng, thức ăn, nước uống để kịp thời nhận biết bệnh, sự khác lạ của thú cưng. Chúng em đã sử dụng module tự tạo bằng ngôn ngữ Python để thực hiện việc phát hiện, trích xuất hình ảnh. Tuy nhiên, công việc trên đòi hỏi rất nhiều thời gian kể cả khi áp dụng nhiều công nghệ.

Ngoài ra, trong đề tài còn có một vấn đề rất quan trọng chính là việc nhận dạng mèo so với trẻ em và người lớn khi ở xa. Đối và tinh thần sáng tạo trong việc áp dụng các công nghệ và giải pháp phù hợp. Bằng cách tận dụng các mô hình được huấn luyện sẵn, tối ưu hóa mô hình, sử dụng phần cứng tăng tốc và tinh chỉnh các tham số, chúng em có thể tạo ra một hệ thống nhận diện và quan sát thú cưng hiệu quả.

Qua đó, việc phát triển công nghệ nhận diện và quan sát thú cưng có thể tiếp tục đem lại nhiều tiện ích và phát triển hơn. Các công nghệ như thị giác máy tính, học sâu và Internet of Things (IoT) sẽ đóng vai trò quan trọng trong việc nâng cao khả năng nhận diện và quản lý thú cưng. Điều này sẽ mang lại lợi ích lớn cho chủ nuôi thú cưng, giúp họ theo dõi, chăm sóc và yêu thương thú cưng một cách tốt nhất.

Mặc dù có những thách thức về thời gian và hiệu suất, nhưng với sự sáng tạo và áp dụng các giải pháp phù hợp, chúng em có thể xây dựng một mô hình chăm sóc và nhận diện thú cưng hiệu quả. Việc này sẽ mang lại lợi ích lớn cho chủ nuôi thú cưng, giúp họ quản lý và chăm sóc thú cưng một cách tốt nhất trong mọi tình huống.

Từ những lý do trên, bài toán được hình thành để nghiên cứu các phương pháp trích xuất hình ảnh, tự động hóa thiết bị. Trong lĩnh vực thị giác máy tính, bài toán này có thể được gọi là bài toán phát hiện vật thể (Object Detection) .

1.1.2. Ứng dụng.

Việc xây dựng mô hình để quan sát thú cưng trong nhà mang nhiều ý nghĩa quan trọng:

- Giám sát và chăm sóc từ xa: Chủ nuôi sử dụng ứng dụng di động Họ có thể cho thú nuôi ăn, uống. Nếu phát hiện vấn đề sức khỏe hay hành vi bất thường, chủ nuôi có thể nhận được cảnh báo để xử lý kịp thời.
- Về nông nghiệp: có thể ứng dụng mô hình trên để giúp người nông dân trong việc quan sát nông trại lớn nhiều động vật. Dùng các công nghệ được áp dụng để thay thế công việc tay chân cần nhiều người và thời gian.
- Từ mô hình này, nhóm nghiên cứu có thể góp phần phát triển công cuộc sử dụng trí tuệ nhân tạo trong lĩnh vực nông nghiệp ngày nay.

1.1.3. Thách thức.

Bài toán chăm sóc thú cưng bằng AI và IoT cũng tồn tại những khó khăn và thách thức riêng:

- Ảnh chứa nhiều dung lượng, chưa tối ưu (hình ảnh đẹp dựa vào dung lượng mạng).
- Hình thể của mèo luôn biến đổi, không nhất quán.
- Độ chính xác và đáng tin cậy của dữ liệu: Mô hình này đòi hỏi dữ liệu chính xác và đáng tin cậy từ các cảm biến IoT. Đôi khi, dữ liệu có thể bị nhiễu hoặc bị ảnh hưởng bởi môi trường xung quanh, gây ra sai sót trong việc phân tích và đưa ra quyết định. Đảm bảo tính chính xác và đáng tin cậy của dữ liệu là một thách thức quan trọng.

1.2. Tổng quan các nghiên cứu nhận dạng động vật.

Với các thách thức nêu trên, các nhà nghiên cứu trên thế giới trong nhiều năm qua đã đề xuất nhiều phương pháp tiếp cận khác nhau trong những năm qua. Các nghiên cứu nhận biết động vật là một lĩnh vực nghiên cứu khoa học được quan tâm nhiều trong các lĩnh vực sinh học, tâm lý học và kỹ thuật. Mục đích của nghiên cứu nhận biết động vật là cung cấp cho

chúng em các cách tiếp cận và kỹ thuật để tìm hiểu cách động vật nhận biết và tương tác với môi trường của chúng.

Các nghiên cứu nhận biết động vật bao gồm nhiều phương pháp, từ các nghiên cứu hành vi tự nhiên cho đến các phép đo thần kinh và các phương pháp hình ảnh học. Các phương pháp này được sử dụng để nghiên cứu khả năng nhận thức của động vật, bao gồm khả năng học hỏi, nhận thức, ghi nhớ, suy luận và hiểu biết.

Các nghiên cứu nhận biết động vật được áp dụng trong nhiều lĩnh vực, từ nghiên cứu cơ sở về hành vi động vật, đến ứng dụng trong thiết kế robot và phát triển trí tuệ nhân tạo. Các kết quả nghiên cứu có thể cung cấp cho chúng em kiến thức về cách động vật nhận biết và tương tác với môi trường của chúng, từ đó có thể giúp chúng em cải thiện các kỹ thuật và sản phẩm liên quan đến động vật.

1.3. Tổng quan các nghiên cứu phát hiện động vật trong nhà.

Năm 2021, có 1 bài báo nói về một hệ thống giám sát thời gian thực cho thú cưng tại nhà sử dụng Raspberry pi được phát triển. Phương pháp đề xuất bao gồm một hệ thống nhận dạng YOLOv3-Tiny dựa trên Raspberry Pi để phát hiện nhanh chóng và dự đoán khung ranh giới tốt hơn cho hành vi của mèo. Dựa trên phương pháp YOLOv3-Tiny, các phương pháp được thực hiện như sau [1]:

Các hình ảnh thu thập bằng điện thoại di động được cắt đều thành 416 * 416 pixel.

Các hình ảnh được chụp trong các khoảng thời gian khác nhau và xoay ngẫu nhiên thành dương hoặc âm 20 độ để làm cho tập dữ liệu huấn luyện mạnh mẽ hơn.

Sử dụng 1400 bức ảnh về các chuyển động của mèo trong phòng để đánh dấu và huấn luyện. Tập dữ liệu cũng được sử dụng để huấn luyện mô hình YOLOv3. Kết quả đầu ra được phân loại thành sáu hành động của mèo. Chúng là ngủ, ăn, ngồi xuống, đi bộ, đi vệ sinh và tìm kiếm thùng rác. Độ chính xác trung bình của cả hai mô hình là 98%. Dựa trên các ràng buộc về tốc độ và tiêu thụ bộ nhớ, mô hình có khả năng nhận diện hình ảnh nhanh chóng.

Hệ thống Raspberry Pi liên tục chụp lại các hình ảnh và gửi tin nhắn đến điện thoại di động về các hành động đó.

1.4. Hướng giải quyết.

Qua các công trình nghiên cứu trên, chúng em đề xuất một phương pháp tiếp cận bao gồm hai giai đoạn. Giai đoạn đầu tiên, mô hình sẽ lấy hình ảnh từ camera để xử lý. Kết quả của giai đoạn một sẽ được sử dụng để loại bỏ phần nền (bao gồm các vật dụng trong nhà và những thứ không liên quan), sau đó mô hình sẽ chọn những vùng ứng viên từ dữ liệu đã xử lý để tiến hành xác định con vật. Trong mỗi giai đoạn, phương pháp chính để trích xuất các đối tượng là ứng dụng phương pháp học sâu: mạng tính chập đầy đủ (Fully Convolutional Network) với kiến trúc chính là YOLOv8.

Chương 1 đã đề cập đến đề tài phát triển mô hình chăm sóc thú cưng bằng sự kết hợp của trí tuệ nhân tạo (AI) và Internet of Things (IoT). Trong chương này, chúng em đã trình bày một tổng quan về bài toán chăm sóc thú cưng, ứng dụng của nó và các thách thức đang tồn tại. Chúng em giới thiệu bài toán chăm sóc thú cưng và nhấn mạnh tầm quan trọng của việc cung cấp sự chăm sóc tốt nhất cho thú cưng trong cuộc sống hiện đại. Chúng em đã xem xét các khía cạnh chính của bài toán, bao gồm việc đảm bảo dinh dưỡng, sức khỏe và sự an toàn cho thú cưng. Chúng em đã ứng dụng của AI và IoT trong lĩnh vực chăm sóc thú cưng. Chúng em đã tìm hiểu về các công nghệ nhận dạng động vật và phát hiện động vật trong nhà, nhằm theo dõi và quản lý hoạt động của thú cưng. Chúng em nhận thấy rằng AI và IoT có thể cung cấp những giải pháp thông minh và tự động hóa để giúp chủ nuôi quản lý và chăm sóc thú cưng một cách hiệu quả. Tuy nhiên, trong quá trình nghiên cứu, chúng em đã gặp phải những thách thức. Chúng em đã đề cập đến sự đa dạng của các loài động vật và khả năng phát hiện và nhận dạng chính xác. Cũng như các vấn đề về tích hợp và tương tác giữa các hệ thống AI và IoT trong mô hình chăm sóc thú cưng. Với những thách thức và ứng dụng đã đề cập, chương này cung cấp một cái nhìn tổng quan về hướng giải quyết của đề tài. Chúng em đã nhìn vào các nghiên cứu hiện có về nhận dạng và phát hiện động vật, và từ đó xác định các phương pháp và công nghệ phù hợp để phát triển mô hình chăm sóc thú cưng thông minh. Mục tiêu là tạo ra một hệ thống tự động, thông minh và liên tục để

theo dõi, chăm sóc và ghi lại hoạt động của thú cưng một cách chính xác và hiệu quả. Chương 1 đã đặt nền tảng cho việc tiếp tục nghiên cứu và phát triển trong đề tài này. Các nội dung đã trình bày đã làm rõ tầm quan trọng và tiềm năng của việc áp dụng AI và IoT trong chăm sóc thú cưng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.

2.1. Tổng quan về Thị Giác Máy Tính.

2.1.1. Giới thiệu về Thị Giác Máy Tính.

Một trong những công nghệ tiên tiến liên quan đến AI là thị giác máy tính. Nhờ những tiến bộ trong trí tuệ nhân tạo và những đổi mới trong học sâu và mạng nơ-ron, lĩnh vực này đã có thể có những bước tiến nhảy vọt trong những năm gần đây và có thể vượt qua con người trong một số nhiệm vụ liên quan đến phát hiện và ghi nhận đối tượng.

Computer Vision hay còn gọi là Thị giác máy tính là lĩnh vực khoa học máy tính tập trung vào việc tái tạo các phần phức tạp của hệ thống thị giác con người và cho phép máy tính xác định và xử lý các đối tượng trong hình ảnh và video giống như cách con người làm.

Một trong những yếu tố thúc đẩy sự phát triển của thị giác máy tính là lượng dữ liệu chúng em tạo ra ngày nay được sử dụng để đào tạo và làm cho thị giác máy tính tốt hơn.

Nó không chỉ thay thế con người làm những công việc tính toán đơn giản đến khó khăn, mà nó còn có khả năng bắt chước động vật và con người. Một trong số những khả năng đó là nhận biết được thế giới qua “mắt” của nó. Bằng việc kết hợp các mô hình khác như: máy học, mạng nơ-ron,...[2] giúp cho chúng dần tiến tới một hệ thống nhân tạo có những quyết định linh hoạt và chính xác hơn.

Thị giác máy tính bao gồm các lĩnh vực sau:

- Xử lý hình ảnh: Thuật toán xử lý ảnh như tăng/giảm chất lượng ảnh, lọc nhiễu...
- Nhận diện mẫu: Giải thích các kỹ thuật khác nhau để phân loại mẫu.

2.1.2. Các ứng dụng của Thị Giác Máy Tính.

Thị giác máy tính đã có ứng dụng đáng chú ý trong nhận dạng động vật dựa trên hình ảnh và video. Điều này mang lại nhiều lợi ích trong nghiên cứu, bảo tồn và quản lý động vật. Thị giác máy tính có khả năng nhận dạng loài động vật thông qua việc phân tích các đặc trưng hình ảnh như hình dạng, màu sắc và cấu trúc. Nó có thể phân loại và nhận biết loài

động vật khác nhau, từ chim, động vật có vú đến động vật hoang dã qua các phương pháp học sâu như YOLO v5, Cascade R-CNN trong bộ trích xuất tính năng HRNet32 (hai giai đoạn của Anchor) và FCOS trong bộ trích xuất tính năng ResNet50 và ResNet101 (một giai đoạn của Anchor-Free) [3]. Hơn nữa, thị giác máy tính cũng có khả năng nhận dạng và phân biệt cá thể động vật cá nhân dựa trên các đặc điểm đặc trưng riêng biệt như hình dạng mặt, màu lông và các đặc điểm khác. Điều này hỗ trợ trong việc theo dõi và nghiên cứu các cá thể động vật cụ thể trong tự nhiên, nâng cao hiểu biết về sinh thái học, hành vi và quản lý động vật. Với sự phát triển của thị giác máy tính, tiềm năng trong việc nhận dạng động vật sẽ tiếp tục mở ra nhiều cơ hội trong các lĩnh vực như nghiên cứu, bảo tồn môi trường và giáo dục môi trường.

Không chỉ thế, thị giác máy tính ứng dụng trong việc điều khiển tiến trình như các robot trong công nghiệp, hay các thiết bị, xe tự hành hay trong các quá trình kiểm tra trong môi trường công nghiệp, ...

Ngoài ra, thị giác máy tính còn có vai trò tương tác (đóng vai trò làm đầu vào cho thiết bị trong quá trình tương tác giữa người và máy).

Tuy nhiên, thị giác máy tính cũng đối mặt với nhiều thách thức và hạn chế, bao gồm khó khăn trong việc hiểu và định nghĩa các đặc trưng hình ảnh, sự bất ổn của các phương pháp học máy khi đối mặt với các tình huống mới và phức tạp, và sự thiếu chính xác trong việc phân loại và nhận diện đối tượng trong một số tình huống đặc biệt.

2.2. Tổng quan học sâu (Deep Learning).

2.2.1. Giới thiệu về học sâu.

Ngày nay, Artificial Intelligence (AI - trí tuệ nhân tạo) và Machine Learning (ML - máy học) hiện diện trong mọi lĩnh vực của đời sống con người, từ kinh tế, giáo dục, y khoa cho đến những công việc nhà, giải trí hoặc thậm chí là trong quân sự. Những ứng dụng nổi bật trong việc phát triển AI đến từ nhiều lĩnh vực để giải quyết nhiều vấn đề khác nhau. Nhưng những đột phá phần nhiều đến từ Deep Learning (DL - học sâu) - một mảng nhỏ đang mở rộng dần đến từng loại công việc, từ đơn giản đến phức tạp.

Tuy nhiên, không phải ai cũng biết được những nhà nghiên cứu trong ngành đã phải tốn bao nhiêu công sức, trí lực để đưa những khái niệm trừu tượng ra khỏi trang giấy, mô hình hóa nó thành những điều thú vị mà ai cũng sử dụng ngày nay. Công việc của họ cũng không phải chỉ ngắn gọn đến thế, cái mà họ cần làm là giải thích cái trừu tượng đó một cách tường minh, dựa trên những phân tích có được để xây dựng các model mới tốt hơn và quan trọng hơn hết là hiện thực hóa nó bằng ngôn ngữ lập trình.

Học sâu có thể được xem là một lĩnh vực con của máy học ở đó các máy tính sẽ học và cải thiện chính nó thông qua các thuật toán. Học sâu được xây dựng dựa trên các khái niệm phức tạp hơn rất nhiều, chủ yếu hoạt động với các mạng nơ-ron nhân tạo để bắt chước khả năng tư duy và suy nghĩ của bộ não con người.

Thật ra các khái niệm liên quan đến mạng nơ-ron nhân tạo và học sâu đã xuất hiện từ khoảng những năm 1960, tuy nhiên nó lại bị giới hạn bởi khả năng tính toán và số lượng dữ liệu lúc bấy giờ. Trong những năm gần đây, những tiến bộ trong phân tích dữ liệu lớn (Big Data) đã cho phép ta tận dụng được tối đa khả năng của mạng nơ-ron nhân tạo.

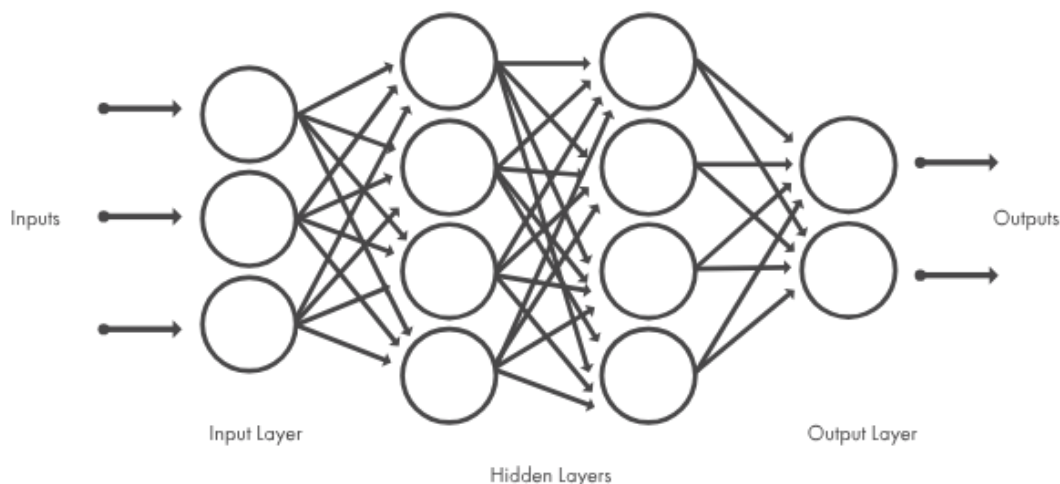
Mạng nơ-ron nhân tạo chính là động lực chính để phát triển học sâu. Các mạng nơ-ron sâu (DNN) bao gồm nhiều lớp nơ-ron khác nhau, có khả năng thực hiện các tính toán có độ phức tạp rất cao. Học sâu hiện đang phát triển rất nhanh và được xem là một trong những bước đột phá lớn nhất trong máy học [4].

2.2.2. Nguyên lý hoạt động của học sâu.

Học sâu là một phương pháp của máy học. Mạng nơ-ron nhân tạo trong học sâu được xây dựng để mô phỏng khả năng tư duy của bộ não con người.

Một mạng nơ-ron bao gồm nhiều lớp (layer) khác nhau, số lượng layer càng nhiều thì mạng sẽ càng “sâu”. Trong mỗi layer là các nút mạng (node) và được liên kết với những lớp liền kề khác. Mỗi kết nối giữa các node sẽ có một trọng số tương ứng, trọng số càng cao thì ảnh hưởng của kết nối này đến mạng nơ-ron càng lớn.

Mỗi nơ-ron sẽ có một hàm kích hoạt, về cơ bản thì có nhiệm vụ “chuẩn hoá” đầu ra từ nơ-ron này. Dữ liệu được người dùng đưa vào mạng nơ-ron sẽ đi qua tất cả layer và trả về kết quả ở layer cuối cùng, gọi là output layer[5].



Hình 1. Kiến trúc của mạng nơ ron [5].

Theo kiến trúc của mạng nơ ron trên có các thành phần sau:

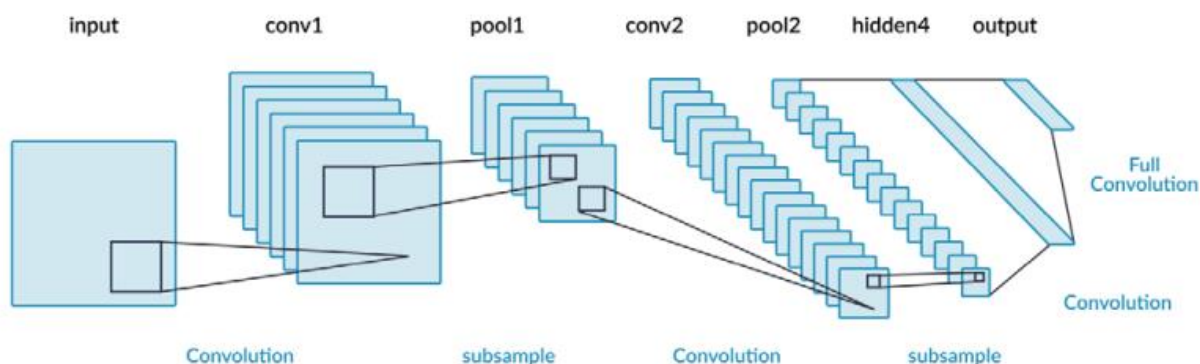
- Lớp đầu vào (Inputs Layer): Thông tin từ thế giới bên ngoài đi vào mạng nơ-ron nhân tạo ở lớp đầu vào. Các nút đầu vào xử lý dữ liệu, phân tích hoặc phân loại và sau đó truyền dữ liệu qua lớp khác.
- Lớp ẩn (Hidden Layers): Dữ liệu đưa vào lớp ẩn đến từ lớp đầu vào đến từ lớp ẩn tiếp theo. Mạng nơ-ron nhân tạo thông thường có một số lượng lớn lớp ẩn. Mỗi lớp ẩn phân tích dữ liệu đầu ra ở lớp trước và xử lý dữ liệu đầu vào nhanh hơn và lại gửi dữ liệu đến lớp sau.
- Lớp đầu ra (Output Layer): Lớp đầu ra cho biết kết quả cuối của toàn bộ dữ liệu được xử lý trong mạng nơ-ron nhân tạo. Lớp này có thể có một hoặc nhiều nút.

Trong quá trình huấn luyện mô hình mạng nơ-ron, các trọng số sẽ được thay đổi và nhiệm vụ của mô hình là tìm ra bộ giá trị của trọng số sao cho phán đoán là tốt nhất.

Deep learning đã được ứng dụng rộng rãi trong nhiều lĩnh vực như nhận dạng ảnh, giọng nói và văn bản, xử lý ngôn ngữ tự nhiên, và thậm chí trong lĩnh vực y tế, vật lý học, và tài chính. Các ứng dụng của deep learning đang được nghiên cứu và phát triển đầy tiềm năng, đặc biệt là khi kết hợp với các công nghệ mới như Internet of Things (IoT), blockchain, và trí tuệ nhân tạo (AI).

2.2.3. Học sâu và xử lý ảnh.

Mạng nơ-ron tích chập là một dạng của mạng nơ-ron đa tầng, mỗi tầng thuộc một trong 3 dạng: tích chập (convolution), lấy mẫu con (subsampling), kết nối đầy đủ (full connection)



Hình 2. Kiến trúc tổng quan mạng nơ-ron tích chập-CNN[6].

Mạng Neural Tích chập (CNN) sử dụng ảnh đầu vào như một tầng đầu vào, trong đó mỗi pixel trong ảnh tương ứng với một nơ-ron. Ảnh đầu vào này thường được gọi là feature map. Có thể coi feature map như một ảnh thông thường, trong đó mỗi pixel đại diện cho một nơ-ron.

Tầng tích chập (C - convolution) trong CNN hoạt động như một bộ trích xuất đặc trưng. Nó thực hiện tích chập của một hoặc nhiều feature map đầu vào với một hoặc nhiều kernel để tạo ra một hoặc nhiều feature map mới (kết quả). Quá trình tích chập này giúp tìm ra các đặc trưng quan trọng trong ảnh.

Tầng lấy mẫu con (S - subsampling) được sử dụng để giảm kích thước của feature map và giúp mạng neural chịu được các biến đổi như dịch chuyển, xoay, và nghiêng của dữ liệu. Quá trình lấy mẫu con thường thực hiện bằng cách chọn một vùng nhỏ trong feature map

và áp dụng một phép tính đơn giản (thường là lấy giá trị lớn nhất hoặc tính trung bình) để tạo ra một pixel trong feature map mới có kích thước nhỏ hơn.

Tầng kết nối đầy đủ (F - Full connection) thực hiện các công việc phân loại như trong mạng neural đa lớp thông thường. Tầng này kết nối tất cả các nơon từ tầng trước đến tầng đầu ra. Các tầng sau đó, mỗi tầng có một số feature map, mỗi feature map lại có một số kernel riêng biệt. Số lượng kernel bằng số lượng feature map trong tầng trước. Các kernel trong cùng một feature map có kích thước như nhau, và kích thước của kernel được xác định dựa trên yêu cầu của bài toán mạng neural. Các giá trị pixel trong một feature map được tính bằng cách tính tổng các tích chập của kernel tương ứng với các feature map trong tầng trước. Số lượng feature map trong tầng cuối cùng (tầng output) tương ứng với số lượng kết quả đầu ra mong muốn của bài toán [6].

2.3. Tổng quan Về YOLO (You Only Look One).

2.3.1. Giới thiệu về YOLO.

YOLO (You Only Look Once) là một mô hình được sử dụng để nhận diện và phân loại đối tượng trong ảnh hoặc video. Điểm nổi bật của YOLO là tốc độ nhanh và khả năng nhận diện đối tượng thời gian thực.

YOLO hoạt động bằng cách chia ảnh thành một lưới ô vuông và dự đoán các hộp giới hạn (bounding boxes) và gán nhãn (labels) cho các đối tượng trong từng ô vuông. Mỗi ô vuông sẽ được phân công để dự đoán các đối tượng nằm trong phạm vi của nó. Mỗi hộp giới hạn bao gồm các thông số như tọa độ (x, y), chiều rộng (width), chiều cao (height) và một xác suất (confidence) đại diện cho độ tin cậy của dự đoán.

Cấu trúc chính của YOLO bao gồm một mạng neural tích chập (convolutional neural network) để trích xuất đặc trưng và một mạng neural kết nối đầy đủ (fully connected network) để dự đoán hộp giới hạn và nhãn. Quá trình huấn luyện YOLO đòi hỏi một tập dữ liệu được gán nhãn với các hộp giới hạn và nhãn chính xác.

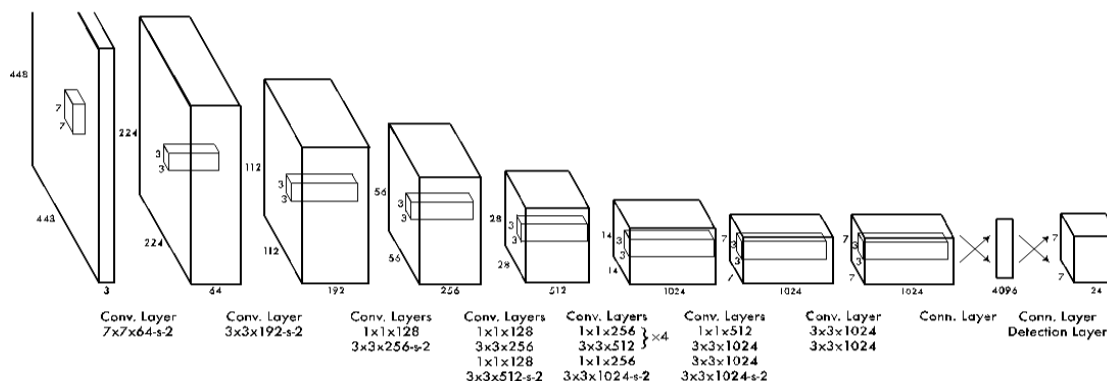
Có nhiều phiên bản YOLO được phát triển theo thời gian như YOLOv1, YOLOv2, YOLOv3, YOLOv4 và YOLOv5. Mỗi phiên bản có những cải tiến về độ chính xác và hiệu

suất. YOLOv4 và YOLOv5 là hai phiên bản đạt được kết quả ấn tượng trong các cuộc thi nhận diện đối tượng và thường được sử dụng trong các ứng dụng thực tế [7]. Những năm gần đây YOLO v6 được phát hành bởi Meituan vào năm 2022 và đang được sử dụng trong nhiều robot giao hàng tự động của công ty. YOLOv7 đã thêm các tác vụ bổ sung như ước tính vị trí trên tập dữ liệu COCO keypoints. YOLOv8 là phiên bản mới nhất của YOLO bởi Ultralytics. Là một mô hình tiên tiến, hiện đại (SOTA), YOLOv8 xây dựng trên thành công của các phiên bản trước đó.

YOLO được áp dụng trong nhiều lĩnh vực, bao gồm nhận diện và đếm đối tượng trong ảnh hoặc video, giám sát an ninh, tự động hóa xe tự hành, và nhiều lĩnh vực khác yêu cầu xử lý đối tượng thời gian thực.

2.3.2. Nguyên lý hoạt động của mạng YOLO.

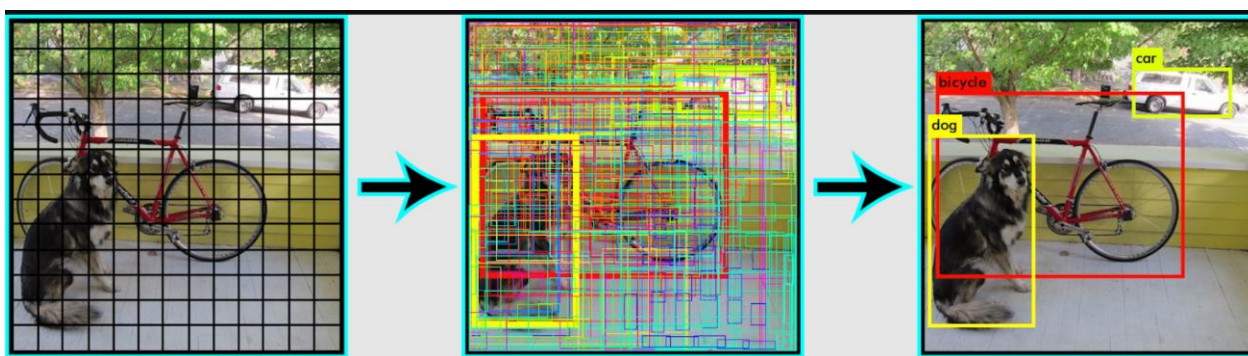
Mô hình YOLO bao gồm hai phần chính: Trích xuất đặc trưng (Feature Extractor) và Phát hiện đối tượng (Object Detection). Trong phần Trích xuất đặc trưng, các lớp tích chập (Convolutional layers) được sử dụng để tạo ra các bản đồ đặc trưng (Feature maps). Phần Phát hiện đối tượng bao gồm các lớp kết nối đầy đủ (Fully Connected layers), được sử dụng để phát hiện, dự đoán nhãn và tọa độ của các bounding box (hộp giới hạn) cho các đối tượng trên các bản đồ đặc trưng từ phần Trích xuất đặc trưng. Ảnh cần nhận dạng sẽ trải qua quá trình Trích xuất đặc trưng để tạo ra các bản đồ đặc trưng với các kích thước khác nhau. Sau đó, các bản đồ đặc trưng này được đưa qua phần Phát hiện đối tượng để dự đoán tên đối tượng, vị trí và kích thước của bounding box xung quanh đối tượng [8].



Hình 3. Kiến trúc mạng YOLO[8].

Kiến trúc mạng YOLO hoạt động theo các bước sau: Ảnh đầu vào được điều chỉnh kích thước thành 448x448 trước khi đi qua mạng tích chập. Sau đó, một lớp tích chập 1x1 được sử dụng để giảm số kênh đầu vào, tiếp theo là một lớp tích chập 3x3 để tạo ra đầu ra hình hộp lục giác. Hàm kích hoạt ReLU được áp dụng trong quá trình này, ngoại trừ lớp cuối cùng sử dụng hàm kích hoạt tuyến tính. Để tăng tính ổn định và khả năng tổng quát hóa của mô hình, các kỹ thuật bổ sung như chuẩn hóa batch và dropout được áp dụng. Những kỹ thuật này giúp điều chỉnh mô hình và ngăn chặn hiện tượng quá khớp. Với việc sử dụng các lớp tích chập và các kỹ thuật điều chỉnh này, mạng YOLO đạt được khả năng phát hiện và phân loại đối tượng chính xác và hiệu quả [8].

Nguyên lý hoạt động của mạng YOLO (You Only Look Once) dựa trên việc chia ảnh đầu vào thành một lưới ô vuông (grid) và áp dụng một mô hình dự đoán trên mỗi ô vuông đó. Mỗi ô vuông sẽ dự đoán xác suất các đối tượng xuất hiện trong ô đó và cung cấp thông tin về vị trí, kích thước và đặc điểm của đối tượng. Quá trình nhận diện đối tượng trong YOLO bắt đầu bằng việc xác định tâm của đối tượng trong mỗi ô vuông. Mạng sử dụng một mạng nơ-ron đơn giản để dự đoán các bounding box cho đối tượng trong mỗi ô và xác suất loại đối tượng đang được chứa bên trong. Điều này được thực hiện bằng cách sử dụng các feature map (bản đồ đặc trưng) từ các lớp tích chập trong mạng [9].



Hình 4. Chia các ô lưới và cho đến khi nhận diện các đối tượng có trong bức hình[9].

Đầu ra của mạng YOLO là một tensor có kích thước $S \times S \times B \times (5 + C)$. Đây có ý nghĩa như sau:

- S là số lượng ô lưới (grid) trong ảnh, tức số ô vuông được chia để dự đoán đối tượng.

- B là số lượng bounding box được dự đoán bởi mỗi ô lưới. Mỗi bounding box dự đoán sẽ chứa thông tin về vị trí, kích thước và đặc điểm của đối tượng.
- C là số lượng lớp đối tượng được dự đoán, tức số lượng các loại đối tượng mà mạng có khả năng nhận diện.
- Trong đó, 5 là số lượng thông tin gồm tọa độ tâm của bounding box, chiều rộng, chiều cao và xác suất của đối tượng xuất hiện trong ô lưới đó.

Tóm lại, tensor đầu ra của YOLO chứa thông tin về vị trí, kích thước và đặc điểm của các bounding box dự đoán cho từng ô lưới, cùng với xác suất của các lớp đối tượng được dự đoán.

Tổng quan, YOLO là một mô hình nhận diện đối tượng nhanh chóng và hiệu quả bằng cách chia ảnh thành lưới ô vuông và dự đoán các đối tượng trên từng ô vuông đó. Mạng YOLO có thể áp dụng cho nhiều tác vụ như nhận diện đối tượng, phân loại và định vị đối tượng trên ảnh và video với tốc độ xử lý nhanh.

2.3.3. Cấu trúc mạng của YOLOv8.

YOLOv8 là mô hình phát hiện đối tượng thời gian thực phổ biến nhất, với kiến trúc mạng nhẹ và phương pháp hợp nhất đặc trưng hiệu quả. Nó kết hợp những ưu điểm của nhiều mô hình phát hiện đối tượng thời gian thực và đạt kết quả phát hiện chính xác hơn.

YOLOv8 sử dụng phương pháp Anchor-Free thay vì Anchor-Based. YOLOv8 sử dụng TaskAlignedAssigner động cho chiến lược tìm kiếm phù hợp. Nó tính toán độ tương quan giữa Anchor-level cho mỗi đối tượng bằng phương trình (1), trong đó s là điểm phân loại, u là giá trị IOU, α và β là các siêu tham số trọng số. Nó chọn m anchors có giá trị lớn nhất (t) cho mỗi đối tượng làm mẫu dương, và chọn các anchor còn lại làm mẫu âm, sau đó tiến hành huấn luyện thông qua hàm mất mát. Sau những cải tiến trên, YOLOv8 chính xác hơn YOLO v5 1%, làm cho nó trở thành mô hình phát hiện chính xác nhất cho đến thời điểm hiện tại.

$$t = s^{\alpha} \times u^{\beta} \quad (1).$$

Phần Backbone là phần cốt lõi của mạng và giữ vai trò trong việc trích xuất đặc trưng. YOLOv8 sử dụng một phần backbone tương tự như YOLO v5, nhưng thay thế mô-đun C3 bằng mô-đun C2F dựa trên ý tưởng CSP [10]. Việc sử dụng mô-đun C2F giúp mạng thu thập được nhiều thông tin gradient hơn và đảm bảo tính nhẹ nhàng của mô hình. Tiếp theo là mô-đun SPPF (Spatial Pyramid Pooling Fusion) được sử dụng ở cuối phần Backbone. Mô-đun này gồm ba lớp MaxPool kích thước 5x5 được xếp liên tiếp, sau đó kết hợp các lớp này. Mục đích của mô-đun này là đảm bảo độ chính xác của các đối tượng ở các tỷ lệ khác nhau trong khi vẫn giữ tính nhẹ nhàng của mạng. Phần Neck tập trung vào việc hợp nhất đặc trưng từ các lớp khác nhau để tạo ra thông tin toàn cảnh. YOLOv8 sử dụng phương pháp hợp nhất đặc trưng PAN-FPN (Path Aggregation Network with Feature Pyramid Network) để tăng cường việc hợp nhất và sử dụng thông tin từ các lớp đặc trưng ở các tỷ lệ khác nhau. Decoupled Head kết hợp độ tin cậy và hộp regression để đạt được mức độ chính xác mới [11]. YOLOv8 hỗ trợ tất cả các phiên bản YOLO và có khả năng chuyển đổi giữa các phiên bản khác nhau. Nó cũng có khả năng chạy trên nhiều nền tảng phần cứng khác nhau (CPU-GPU), đem lại tính linh hoạt mạnh mẽ.

2.4. Tổng quan các thiết bị IoT sử dụng trong mô hình chăm sóc thú cưng.

Các thiết bị IoT trong mô hình chăm sóc thú cưng đều là các thiết bị mua ở cửa hàng điện tử trong thành phố Hồ Chí Minh.

Thông tin các thiết bị IoT:

ESP32 là một dòng vi xử lý tích hợp Wifi và Bluetooth, phát triển bởi công ty Espressif Systems. Nó được sử dụng rộng rãi trong các ứng dụng IoT (Internet of Things) và dự án nhúng. thiết bị này dùng để điều khiển các thiết bị điện tử khác thông qua code Arduino.



Hình 5. Espressif ESP32-DevKitC-32UE.

ESP32-CAM là phiên bản của ESP32 được tích hợp sẵn camera. Nó là một mô-đun nhỏ gọn có khả năng chụp ảnh và quay video, kết hợp với khả năng kết nối Wi-Fi và Bluetooth của ESP32. Camera được tích hợp với ESP32 là Sipeed Lichee DVP Camera OV2640 2MP là một module camera được sử dụng chủ yếu với các bo mạch phát triển nhúng như Lichee Tang hoặc Lichee Pi Zero. Module này được trang bị camera OV2640 có độ phân giải 2MP. Thiết bị này dùng để nhận diện thú cưng (mèo) đến mô hình cho ăn của mèo trong thời gian cho ăn của mèo.



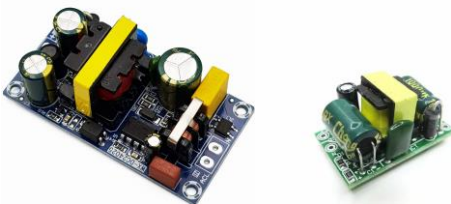
Hình 6. ESP32-CAM Development Board (with camera).

Động cơ bước 42 17 PM là một loại động cơ bước (stepper motor) có đường kính 42mm và sử dụng 4 dây để điều khiển. Động cơ này có 200 bước mỗi vòng xoay (full step) và sử dụng dòng điện xoay chiều (AC) hoặc dòng điện một chiều (DC) để hoạt động. Động cơ bước dùng để đưa thức ăn với mức ước lượng thức ăn đã cho trước trên ứng dụng mobile.



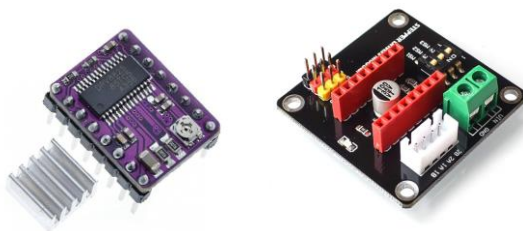
Hình 7. Step Motor Động Cơ Bước 42.

Mạch nguồn AC-DC (12V 2A và 5V 700mA) tích hợp bảo vệ nhiệt độ, quá dòng và ngắn mạch, cách ly điện áp cao và thấp, đầu vào điện áp rộng, đầu ra DC12V được điều chỉnh chính xác, kích thước nhỏ gọn, hiệu suất ổn định, hiệu suất cao.



Hình 8. Mạch nguồn cách ly AC-DC 12V 2A & Mạch Nguồn AC-DC 5V.

Driver động cơ bước DRV8825 với đầy đủ các tính năng của một driver chuyên nghiệp: điều chỉnh dòng giới hạn, vi bước (1/32 bước), bảo vệ quá dòng, quá nhiệt, v.v... Hoạt động ở dải điện áp cao từ 8.2V đến 45V và có thể đạt được xấp xỉ 1,8A trên mỗi pha mà không cần tản nhiệt.



Hình 9. Mạch điều khiển động cơ bước DRV8825 & Đế ra chân Driver động cơ bước DRV8825.

Module 1 relay kích mức cao-thấp 5V có thể kiểm soát các thiết bị gia dụng khác nhau, tùy chọn kích mức cao hoặc thấp với jumper. Sử dụng cách ly mang đến hiệu suất ổn định và an toàn.



Hình 10. Module 1 Relay Kích Mức Cao Thấp 5V.

Mạch điều khiển động cơ DC L298N có khả năng điều khiển 2 động cơ DC, dòng tối đa 2A mỗi động cơ, mạch tích hợp diode bảo vệ và IC nguồn 7805 giúp cấp nguồn 5VDC cho các module khác (chỉ sử dụng 5V này nếu nguồn cấp < 12VDC).



Hình 11. Mạch Điều Khiển Động Cơ L298 DC Motor Driver.

Động cơ bơm chìm 5VDC có kích thước rất nhỏ gọn, vì thuộc dạng bơm chìm nên động cơ có khả năng chống nước và hoạt động khi ngâm chìm trong nước .



Hình 12. Bơm Nước Mini DC - Bơm Chìm 3-5V.

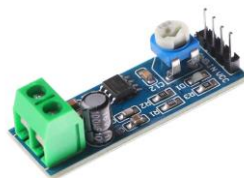
Mạch Thu Phát Âm Thanh ISD1820 - V2 được sử dụng để lưu và phát âm thanh do người dùng ghi vào. Mạch có tích hợp sẵn micro trên board nên có thể thu âm trực tiếp, dữ liệu

được lưu trữ đến 100 năm và có thể ghi - xóa đến 100.000 lần. Mạch có thể ghi âm được khoảng 10 giây.



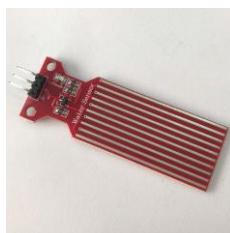
Hình 13. Mạch Ghi Phát Âm Thanh ISD1820 (20s) & Loa 5W 4R Tròn (size 52mm).

Mạch khuếch đại âm thanh đơn giản LM386 có thiết kế nhỏ gọn, đơn giản, dễ sử dụng, dải điện áp đầu vào rộng 4~12VDC với hệ số khuếch đại tối đa lên đến 200 lần, thích hợp cho các ứng dụng cần sự nhỏ gọn, đơn giản: Robot phát tiếng nói, ampli mini,...



Hình 14. Mạch Khuếch Đại Âm Thanh LM386.

Cảm biến mực nước phát hiện nhận dạng nước, tiết kiệm chi phí để xác định mực nước bằng cách đo lượng nước,giọt nước của nó bằng một loạt các đường mạch song song.



Hình 15. Cảm Biến Mực Nước.

Cảm biến siêu âm UltraSonic HY-SRF05 được sử dụng để nhận biết khoảng cách từ vật thể đến cảm biến nhờ sóng siêu âm, cảm biến có thời gian phản hồi nhanh, độ chính xác cao, phù hợp cho các ứng dụng phát hiện vật cản, đo khoảng cách bằng sóng siêu âm



Hình 16. Cảm Biến Siêu Âm SRF05.

2.5. Sơ đồ hoạt động các thiết bị IoT.

Sau khi các thu thập được các thiết bị IoT, chúng em bắt đầu kết nối với các thiết bị IoT và mắc dây điện với nguồn điện phù hợp với từng thiết bị.

ESP32 dùng để kết nối và điều khiển các thiết bị cảm biến và động cơ và các thiết bị này hoạt động khi nhận dữ liệu từ firebase.

ESP32-Camera dùng để lấy ảnh từ nơi đặt và lấy khung hình đưa lên website với đường dẫn <https://192.168.xx.x> và phía server sẽ lấy ảnh liên tục từ đường dẫn đó và nhận diện.

Động cơ bước 42 17PM và driver động cơ bước DRV8825 được sử dụng để thực hiện các chuyển động chính xác và điều khiển bằng cách gửi xung từ driver DRV8825. Điều này cho phép chúng em điều khiển vị trí và tốc độ của động cơ.

Mạch nguồn AC-DC (12V 2A và 5V 700mA) được sử dụng để cung cấp nguồn điện cho các thiết bị khác trong hệ thống. Nó chuyển đổi dòng điện xoay chiều (AC) thành dòng điện một chiều (DC) với mức điện áp và dòng điện phù hợp.

Module 1 relay kích mức cao-thấp 5V cho phép chúng em điều khiển các thiết bị ngoại vi khác thông qua tín hiệu kích từ thiết bị điều khiển khác.

Mạch điều khiển động cơ DC L298N: Mạch này được sử dụng để điều khiển động cơ DC. Nó có khả năng điều chỉnh chiều quay, tốc độ và hướng di chuyển của động cơ DC.

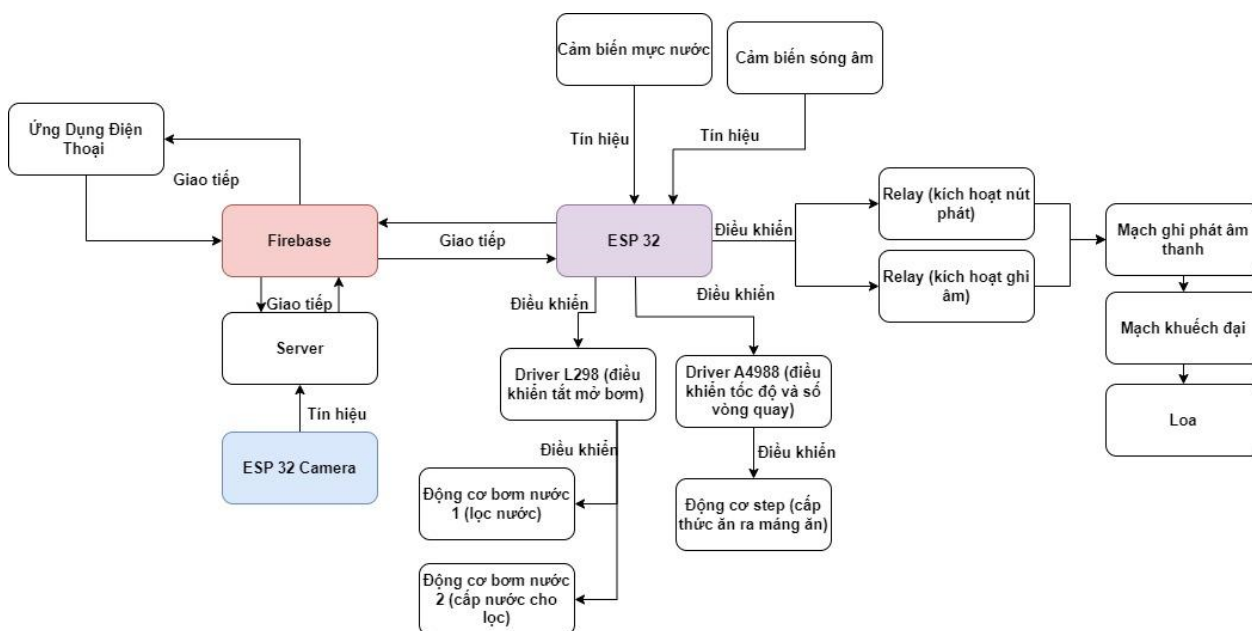
Động cơ bơm chìm 5VDC: Đây là một động cơ bơm dùng để bơm nước hoặc chất lỏng khác. Nó được cung cấp nguồn 5V để hoạt động và có thể được điều khiển bằng tín hiệu từ ESP32.

Mạch thu phát âm thanh ISD1820 - V2 và mạch khuếch đại âm thanh đơn giản LM386: Mạch thu phát âm thanh ISD1820 - V2 cho phép thu âm và phát lại âm thanh.

Mạch khuếch đại âm thanh LM386 được sử dụng để tăng cường và điều chỉnh âm thanh trước khi đưa ra loa hoặc thiết bị ra âm khác.

Cảm biến mực nước phát hiện nhận dạng nước dùng để phát hiện mức nước trong một hệ thống. Khi mức nước đạt đến một ngưỡng xác định, nó sẽ tạo ra tín hiệu để điều khiển các thiết bị khác hoặc cảnh báo.

Cảm biến siêu âm Ultrasonic HY-SRF05 sử dụng sóng siêu âm để đo khoảng cách dung tích khối lượng đồ ăn cho mèo.



Hình 17. Sơ đồ hoạt động của các thiết bị IoT.

Chương 2 đã trình bày cơ sở lý thuyết cho việc phát triển mô hình chăm sóc thú cưng bằng sự kết hợp của trí tuệ nhân tạo (AI) và Internet of Things (IoT). Các nội dung chính trong

chương này bao gồm tổng quan về thị giác máy tính, học sâu (Deep Learning), mạng YOLO và các thiết bị IoT sử dụng trong mô hình chăm sóc thú cưng.

Chương này tập trung vào các thiết bị IoT sử dụng trong mô hình chăm sóc thú cưng. Chúng em đã đề cập đến sự đa dạng của các thiết bị IoT, bao gồm cả cảm biến, thiết bị đo lường và thiết bị điều khiển. Chúng em đã trình bày các tính năng và chức năng của các thiết bị này trong việc cung cấp chăm sóc và giám sát thú cưng.

Cuối cùng, chúng em đã trình bày sơ đồ hoạt động của các thiết bị IoT trong mô hình chăm sóc thú cưng. Sơ đồ này cho thấy sự tương tác giữa các thiết bị IoT, cảm biến và hệ thống AI để thực hiện việc theo dõi, phát hiện và chăm sóc thú cưng một cách tự động và thông minh.

CHƯƠNG 3. XÂY DỰNG HỆ THỐNG NHẬN DẠNG THÚ CỪNG (MÈO) TRONG CÁC THIẾT BỊ IOT VÀ AI.

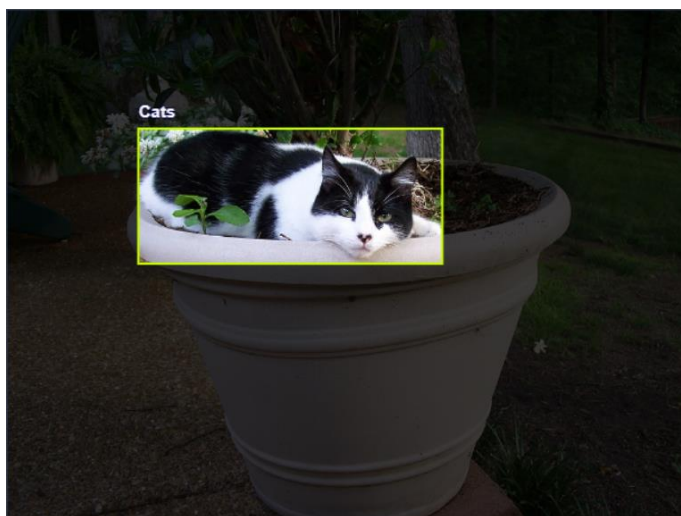
3.1. Ứng dụng mô hình YOLOv8 vào mô hình chăm sóc thú cưng.

3.1.1 Chuẩn bị bộ dữ liệu.

Chuẩn bị bộ dữ liệu để huấn luyện mô hình YOLOv8 là một quá trình quan trọng để đảm bảo hiệu suất và độ chính xác cao trong việc nhận diện đối tượng. Một cách phổ biến để tạo bộ dữ liệu là sử dụng công cụ Roboflow để tự vẽ bounding box trên ảnh.

Đầu tiên, để bắt đầu, chúng em cần có một tập hợp ảnh đại diện cho các đối tượng chúng em muốn nhận diện. Có thể thu thập ảnh từ các nguồn khác nhau như Kaggle(Cat Dataset) [12], Roboflow,... chọn lựa các tệp ảnh có chứa tổng thể của mèo.

Sau đó, chúng em sử dụng công cụ Roboflow để tạo bounding box trên ảnh. Roboflow cung cấp giao diện trực quan để vẽ bounding box xung quanh các đối tượng trong ảnh. Chúng em chỉ cần kéo và thả để tạo bounding box cho mỗi đối tượng trong ảnh. Roboflow sẽ tự động lưu trữ thông tin về vị trí và kích thước của bounding box.



Hình 18. Vẽ bounding box trên website roboflow.

Sau khi hoàn tất quá trình tạo bounding box cho tất cả các ảnh trong bộ dữ liệu, chúng em tiến hành huấn luyện mô hình YOLOv8 bằng bộ dữ liệu đã được chuẩn bị trước đó. Bộ dữ

liệu này bao gồm 2.700 ảnh cho quá trình huấn luyện, 764 ảnh để kiểm tra và 388 ảnh dùng cho việc xác nhận. Chia bộ dữ liệu thành ba phần: tập huấn luyện (train), tập kiểm tra (test) và tập xác nhận (valid). Quá trình này giúp đánh giá hiệu suất của mô hình trên dữ liệu không được sử dụng trong quá trình huấn luyện và kiểm tra tính tổng quát của mô hình.

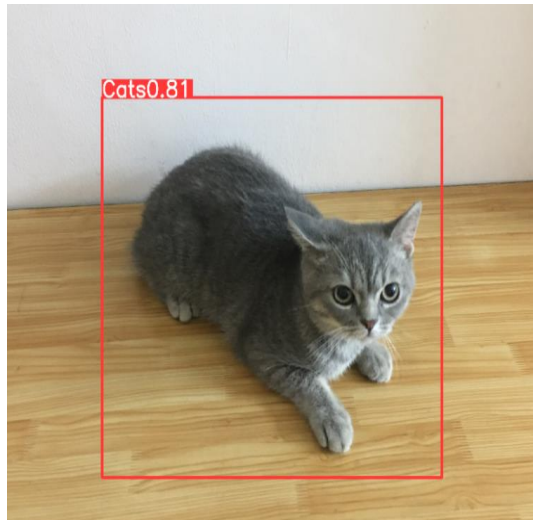
Cuối cùng, chúng em đã có bộ dữ liệu hoàn chỉnh và sẵn sàng để huấn luyện mô hình YOLO. Bộ dữ liệu này sẽ chứa ảnh đã được vẽ bounding box và các thông tin về vị trí, kích thước và đặc điểm của các đối tượng. Với bộ dữ liệu này, chúng em có thể tiến hành huấn luyện mô hình YOLO để nhận diện các đối tượng trong ảnh với độ chính xác cao.

3.1.2. Ứng dụng mô hình YOLOv8 đã huấn luyện.

Mô hình YOLOv8 đã được huấn luyện và có thể được ứng dụng vào các thiết bị IoT thông qua thiết bị trung gian để nhận diện thú cưng một cách hiệu quả. Điều này mang lại nhiều lợi ích trong việc quản lý và chăm sóc thú cưng.

Với YOLOv8, việc nhận diện và phân loại các loại thú cưng trở nên dễ dàng và tự động hơn. Mô hình có khả năng nhận diện rõ ràng thú cưng mèo và nhiều loại khác. Thông qua việc cho mô hình YOLOv8 nhận diện thú cưng thông qua hình ảnh của ESP32 camera đã gửi qua, chủ nuôi có thể nhận được thông tin chính xác về loại thú cưng và cảnh báo khi có những hoạt động bất thường xảy ra.

Ứng dụng này mang lại lợi ích vượt trội cho người nuôi thú cưng. Chủ nuôi có thể dễ dàng giám sát việc ăn uống của thú cưng mà không ở nhà, từ việc kiểm tra xem chó hay mèo có đến nơi cho ăn không để đánh giá sức khỏe. YOLOv8 giúp tạo ra một môi trường chăm sóc thú cưng tốt hơn và đáng tin cậy, giúp chủ nuôi có thể đáp ứng nhu cầu và quan tâm tốt nhất đến thú cưng của mình.



Hình 19. Nhận diện mèo trên tập dữ liệu đã được huấn luyện.

3.2. Giao tiếp giữa thiết bị IoT với ứng dụng điện thoại.

3.2.1. Tổng quan về React Native.

Để triển khai các ứng dụng phức tạp, React-Native sử dụng các dependency khác. Việc tạo ra ứng dụng di động thường là công việc khó khăn đối với các nhà phát triển, bởi vì họ phải tạo ra một ứng dụng cho nền tảng Android và một ứng dụng cho nền tảng iOS, đòi hỏi kiến thức về hai lĩnh vực khác nhau. Ví dụ như đối với ứng dụng di động Android, cần có kiến thức về IDE (Android Studio, ngôn ngữ lập trình: Kotlin hoặc Java), và đối với ứng dụng di động iOS (ngôn ngữ lập trình: Swift hoặc Objective-C và IDE: Xcode). Ngoài ra, một số framework ứng dụng di động hybrid phổ biến do các nhà phát triển tạo ra không thành công trong việc tạo ra trải nghiệm tương tự trên nhiều nền tảng native [13].

Để giải quyết vấn đề này, bài báo này sử dụng những ưu điểm của React-Native, một framework có khả năng tạo ra các ứng dụng di động hybrid cho các nền tảng khác nhau. React-Native được phát triển bởi các nhà phát triển của Facebook vào năm 2015, cho phép viết mã chỉ một lần bằng ngôn ngữ lập trình Javascript ES6 và xây dựng ứng dụng di động cho cả iOS và Android với mã nguồn duy nhất cho cả hai nền tảng native.

có tên là Redux hoặc Mobx, có khả năng quản lý trạng thái trong react native. Ngoài ra, bài báo này mô tả các dependency do các nhà phát triển khác của React-Native tạo ra đã

được sử dụng trong công việc nghiên cứu này, chẳng hạn như cơ sở dữ liệu và giao diện người dùng.

Mục tiêu của nghiên cứu này là:

- Đánh giá trải nghiệm người dùng của ứng dụng di động được tạo bởi framework React-Native có đáp ứng yêu cầu của người dùng hay không.
- Kiểm tra xem định dạng giao diện người dùng có dễ dàng so với các nền tảng native khác không.
- Phân biệt giữa việc tạo ra các ứng dụng đơn giản và phức tạp bằng React-Native.

React Native là một trong những framework phổ biến nhất cùng với Flutter, được sử dụng để phát triển ứng dụng di động hybrid. Trong một khảo sát của các thành viên trên Stack Overflow về framework được mong muốn nhất, React Native đứng đầu với tỷ lệ 13%, trong khi đối thủ là Flutter đứng thứ hai với tỷ lệ 6,7%, như được trình bày trong tài liệu [14]. Vì sự hỗ trợ của React Native hơn các framework khác, nên React Native được lựa chọn để làm ứng dụng trong bài báo này.

React Native là một framework mã nguồn mở cho phép bạn xây dựng ứng dụng di động sử dụng JavaScript và React. Nó được phát triển bởi Facebook và được ra mắt lần đầu tiên vào năm 2015.

React Native cho phép bạn sử dụng các thành phần (components) React để xây dựng giao diện người dùng cho ứng dụng di động. Tuy nhiên, khác với React thông thường, React Native không sử dụng HTML và CSS để tạo giao diện mà thay vào đó sử dụng các thành phần đã được xây dựng sẵn để tạo giao diện di động.

Một điểm mạnh của React Native là nó cho phép bạn tái sử dụng code giữa các nền tảng khác nhau. Bạn có thể sử dụng một phần lớn code của ứng dụng của mình cho cả iOS và Android, đồng thời cũng có thể tùy chỉnh code cho từng nền tảng một cách riêng biệt.

React Native cũng cung cấp nhiều tính năng mạnh mẽ để giúp xây dựng ứng dụng di động, bao gồm cập nhật dữ liệu trong thời gian thực, điều hướng và tương tác với các thiết bị phần cứng như máy ảnh và cảm biến.

Với React Native, bạn có thể xây dựng các ứng dụng di động chất lượng cao với tốc độ phát triển nhanh chóng và tiết kiệm thời gian và công sức so với việc phát triển ứng dụng di động truyền thống cho các nền tảng khác nhau.

Ngoài các điểm mạnh đã nêu, React Native còn có những tính năng khác sau:

- Thư viện component phong phú: React Native cung cấp một thư viện component phong phú và đa dạng, giúp cho việc xây dựng giao diện ứng dụng di động trở nên dễ dàng hơn.
- Hot Reloading: Hot Reloading cho phép các nhà phát triển xem trực tiếp sự thay đổi của mã trong quá trình phát triển. Khi bạn chỉnh sửa mã, ứng dụng của bạn sẽ tự động tải lại để hiển thị các thay đổi ngay lập tức.
- Hỗ trợ cho các thư viện của bên thứ ba: React Native hỗ trợ cho các thư viện của bên thứ ba như Redux, React Navigation, Firebase và nhiều thư viện khác nữa.
- Được cập nhật thường xuyên: React Native được cập nhật thường xuyên để cải thiện hiệu suất, tối ưu hóa mã và sửa các lỗi.
- Cộng đồng phát triển lớn: React Native có một cộng đồng phát triển lớn và nhiều tài liệu học tập có sẵn, giúp cho việc học tập và sử dụng nó trở nên dễ dàng hơn

Ứng dụng được xây dựng bằng ngôn ngữ Javascript sử dụng framework React Native và sử dụng nơi lưu trữ dữ liệu là Realtime DataBase của FireBase.

- Mục tiêu của ứng dụng: Xây dựng một ứng dụng có thể giúp giúp người dùng quan sát và chăm sóc cho thú cưng trong nhà khi họ đi vắng

- Đảm bảo giao diện được thiết kế để cung cấp trải nghiệm người dùng tốt. Thiết kế giao diện để tối ưu hóa trải nghiệm. Thiết kế giao diện đơn giản và trực quan, giúp người dùng dễ sử dụng và hiểu cách thao tác trên ứng dụng.
- Tương thích khi chạy ứng dụng trên Android lẫn IOS
- Kiểm tra và sửa lỗi: Kiểm tra giao diện trên nhiều thiết bị và môi trường khác nhau để phát hiện và sửa lỗi.

3.2.2. Tổng quan về Firebase.

Firebase là một cơ sở dữ liệu thời gian thực và cũng hoạt động như một Backend-as-a-Service (BaaS). Nó cho phép lưu trữ một danh sách các đối tượng. Google Firebase là một phần mềm phát triển ứng dụng được hỗ trợ bởi Google, cho phép nhà phát triển phát triển ứng dụng cho Android, iOS và Web. Firebase là một nhóm các dịch vụ của Google trên nền tảng đám mây, bao gồm tin nhắn trực tiếp, xác thực người dùng, cơ sở dữ liệu thời gian thực, lưu trữ, hosting, v.v. Firebase cung cấp cơ sở dữ liệu thời gian thực, xác thực, lưu trữ đám mây, các chức năng đám mây, v.v. Firebase thực hiện dữ liệu thời gian thực trong cơ sở dữ liệu. Firebase giúp việc trao đổi dữ liệu với cơ sở dữ liệu dễ dàng. Firebase cung cấp backend cho ứng dụng iOS, Android và Web. Các ứng dụng Firebase có thể được triển khai qua một kết nối bảo mật đến máy chủ Firebase. Firebase cung cấp một bảng điều khiển để quản lý dễ dàng. Nó cung cấp nhiều dịch vụ hữu ích. Firebase có tính bảo mật cao và cài đặt tối thiểu. Tính năng Cloud Functions là một trong những tính năng mới của nền tảng đám mây Firebase, cho phép nhà phát triển viết chương trình bằng ngôn ngữ JavaScript và đặt chúng trên nền tảng đám mây Firebase.[15] Sử dụng Firebase, mỗi đơn vị có thể truy cập trực tiếp vào cơ sở dữ liệu.[16] Firebase là một giải pháp đám mây đa dịch vụ do Google sở hữu, dành cho nhà phát triển di động và web[17]. Nhóm tính năng trong Firebase giúp tích hợp cơ sở dữ liệu đám mây tự động trong cả ứng dụng web và di động[18].

Lợi ích khi sử dụng Firebase.

- Xây dựng ứng dụng nhanh chóng mà không tốn thời gian, nhân lực để quản lý hệ thống và cơ sở hạ tầng phía sau: Firebase cung cấp cho bạn chức năng như phân tích,

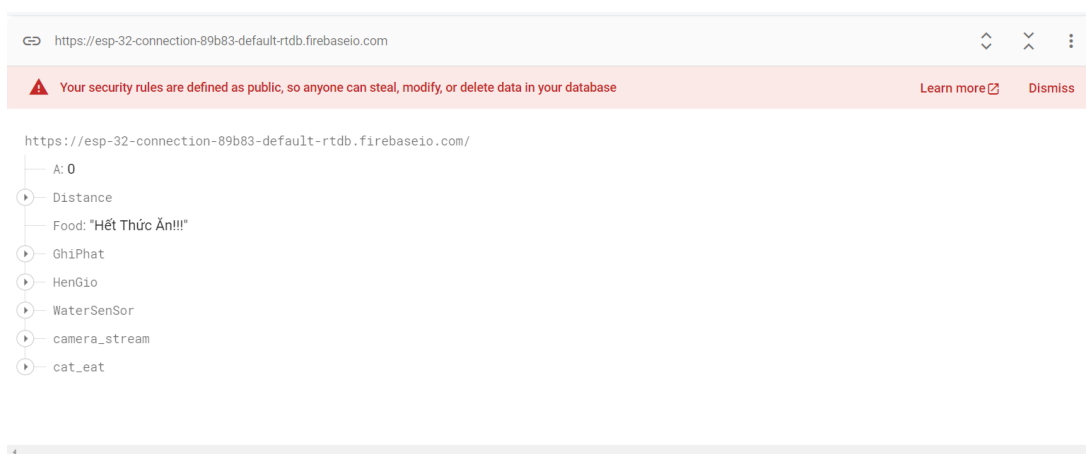
cơ sở dữ liệu, báo cáo hoạt động và báo cáo các sự cố lỗi để bạn có thể dễ dàng phát triển, định hướng ứng dụng của mình vào người sử dụng nhằm đem lại các trải nghiệm tốt nhất cho họ.

- Uy tín chất lượng đảm bảo từ Google: Firebase được google hỗ trợ và cung cấp trên nền tảng phần cứng với quy mô rộng khắp thế giới, được các tập đoàn lớn và các ứng dụng với triệu lượt sử dụng từ người dùng.
- Quản lý cấu hình và trải nghiệm các ứng dụng của Firebase tập trung trong một giao diện website đơn giản, các ứng dụng này hoạt động độc lập nhưng liên kết dữ liệu phân tích chặt chẽ.

Điểm yếu của Firebase.

- Điểm yếu duy nhất của “Firebase” chính là phần Realtime Database, mà đúng hơn chỉ là phần Database. Cơ sở dữ liệu của họ được tổ chức theo kiểu trees, parent-children, không phải là kiểu Table nên những ai đang quen với SQL có thể sẽ gặp khó khăn khá nhiều.

3.3.3. Xây dựng dữ liệu trên FireBase để kết nối ứng dụng điện thoại.

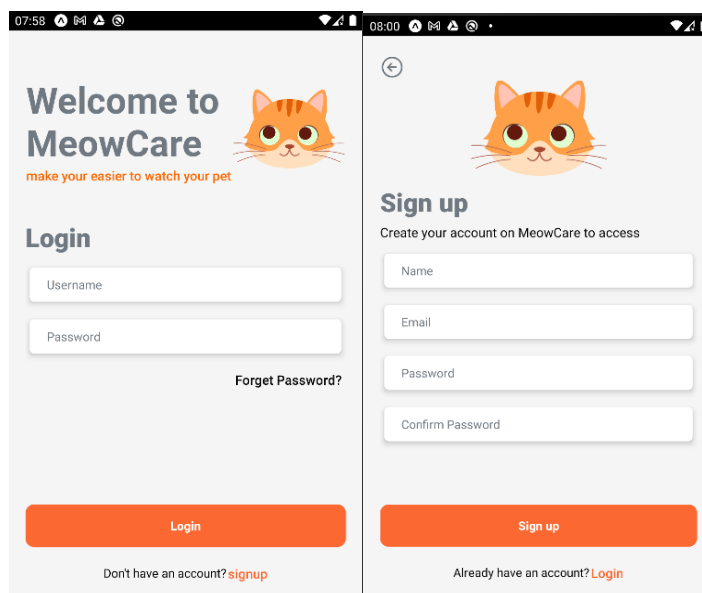


Hình 20. Dữ liệu được tạo trên FireBase.

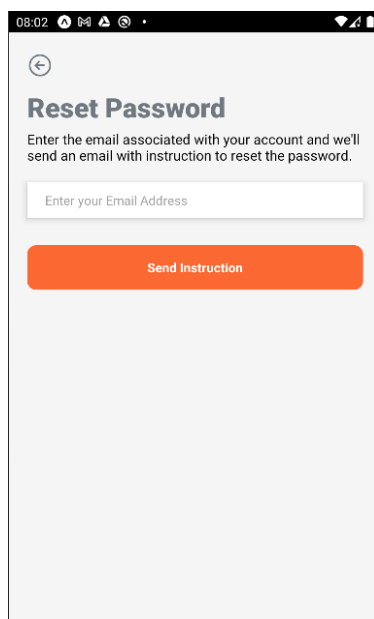
3.3. Phát triển ứng dụng.

Các hoạt động của ứng dụng React Native sử dụng Firebase lưu trữ dữ liệu để quan sát mèo từ xa và cho mèo ăn, uống:

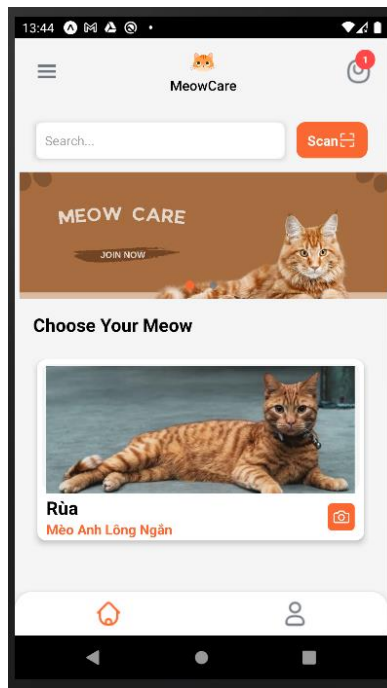
- **Thiết kế giao diện người dùng:** Xác định các thành phần giao diện cần thiết để người dùng có thể quan sát mèo từ xa. Bao gồm một màn hình hiển thị hình ảnh của mèo, các nút điều khiển cho việc chọn ăn và uống, và các thông báo liên quan đến trạng thái của mèo.
- **Thiết lập Firebase:** Tạo một dự án Firebase và cấu hình các dịch vụ Firebase như là Firebase Realtime Database để lưu trữ dữ liệu liên quan đến mèo và các hoạt động của nó.
- **Kết nối ứng dụng với Firebase:** Sử dụng Firebase SDK và các thư viện tương thích với React Native để thiết lập kết nối giữa ứng dụng của bạn và dịch vụ Firebase. Điều này bao gồm việc xác thực người dùng và truy cập vào cơ sở dữ liệu Firebase để đọc và ghi dữ liệu liên quan đến mèo.
- **Hiển thị hình ảnh/video mèo:** Sử dụng Esp32 Cam để up hình ảnh sang dạng Base64 và sau đó giải mã Base64 thành ảnh upload lên app. Các nút điều khiển có thể được thêm vào để cho phép người dùng cài đặt các chức năng.
- **Gửi và lưu trữ dữ liệu:** Khi người dùng chức năng, gửi các yêu cầu tương ứng đến cơ sở dữ liệu Firebase để lưu trữ thông tin và cập nhật trạng thái của mèo.
- **Xử lý thông báo:** Thông báo cho người dùng khi hết đồ ăn, nước uống và khi mèo không ăn đủ bữa.



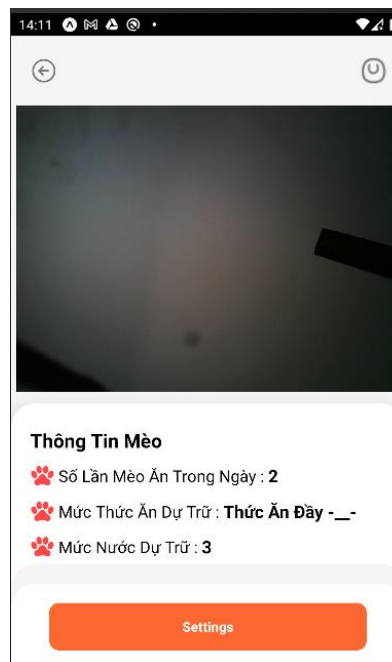
Hình 21. Trang đăng nhập, đăng ký.



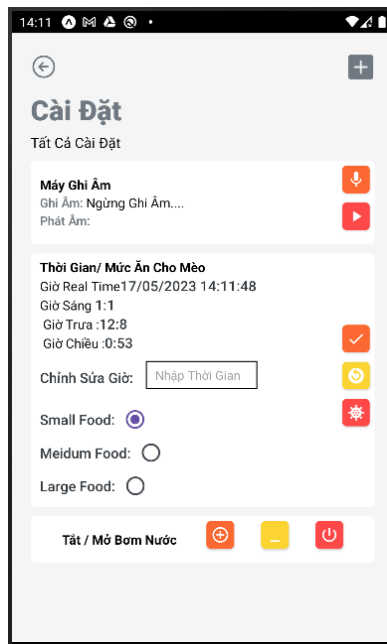
Hình 22. Trang đặt lại mật khẩu.



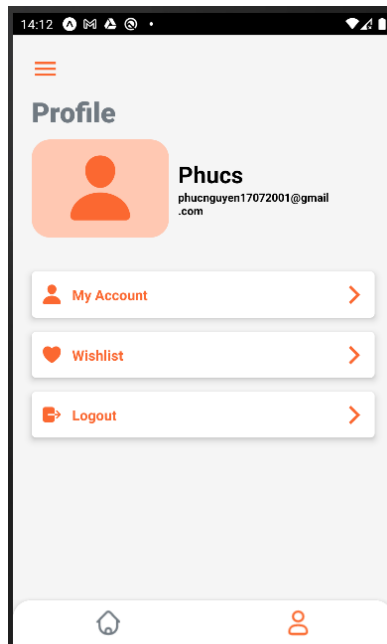
Hình 23. Trang chính.



Hình 24. Trang quan sát mèo.



Hình 25. Trang cài đặt.

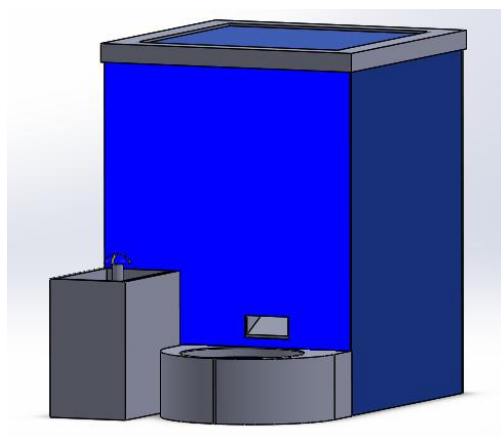


Hình 26. Trang cá nhân người dùng.

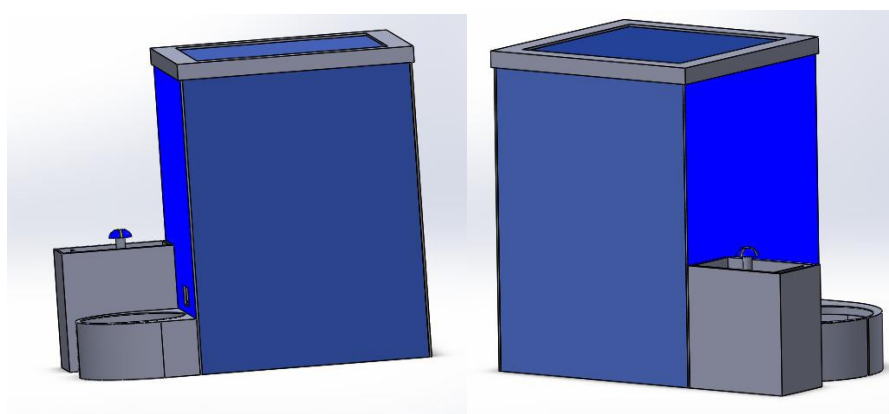
3.4. Mô tả hoạt động tổng quát của mô hình.

3.4.1. Xây dựng mô hình 3D tự động hóa cho mèo.

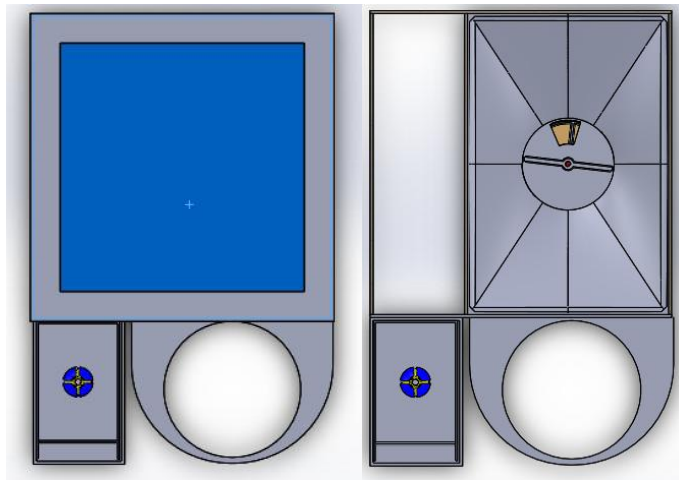
Để xây dựng được một mô hình cho ăn của mèo thì chúng em đã quan sát và đánh giá con mèo của chúng em để có thể vẽ ra một mô hình cho ăn, uống và nhận diện mèo khi đến ăn.



Hình 27. Tổng quan về mô hình 3D.

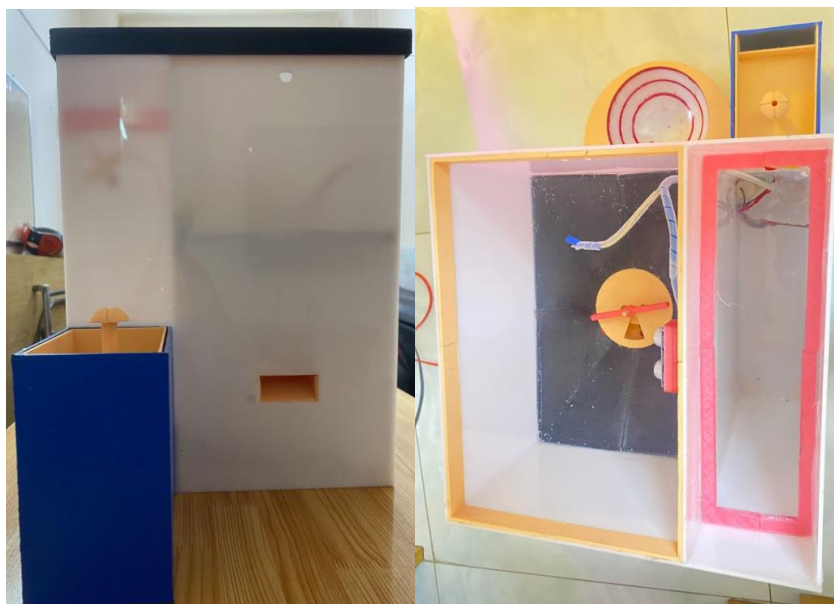


Hình 28. Mặt Trước và Mặt Phải của Mô hình 3D.



Hình 29. Từ trên nhìn xuống của mô hình 3D.

3.4.2. Xây dựng mô hình tự hóa động hóa cho mèo.



Hình 30. Mặt trước và từ trên xuống của mô hình.



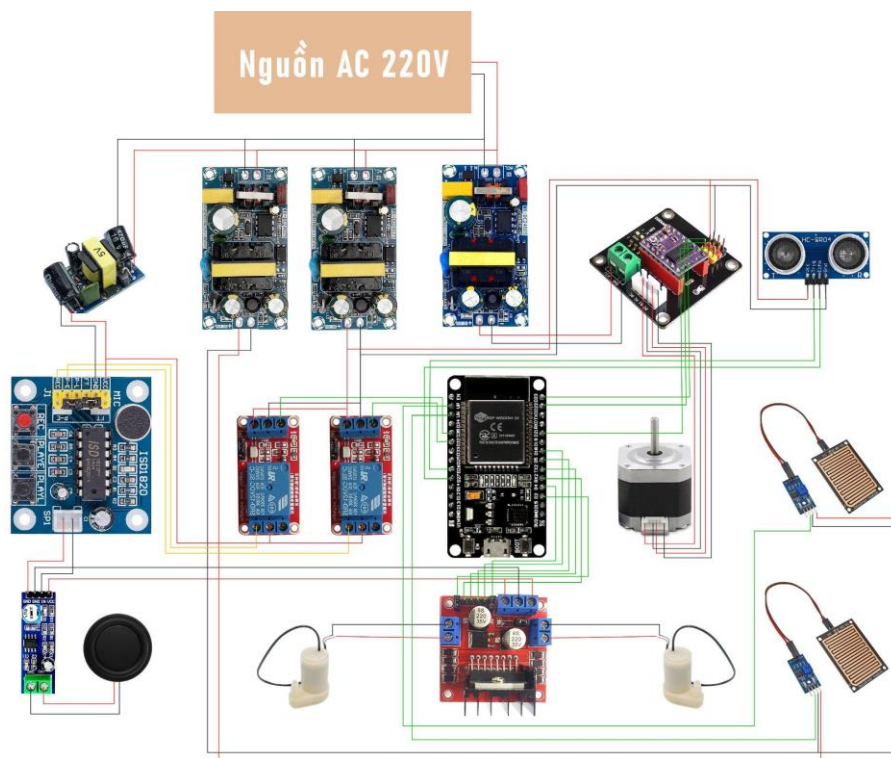
Hình 31. Mặt trái và phải mô hình.



Hình 32. Đồ đựng thức ăn và bơm nước uống cho mèo.



Hình 33. Chỗ dự trữ thức ăn cho mèo.



Hình 34. Sơ đồ lắp thiết bị IoT của mô hình cho ăn, uống.

3.5. Hoạt động của ứng dụng mobile.

Trang đăng nhập: thực hiện việc đăng nhập bằng username và password.

Trang đăng ký: thực hiện việc đăng ký bằng name , email và password.

Trang gửi lại mật khẩu: thực hiện việc gửi lại mật khẩu qua email.

Trang chính: tìm kiếm và hiển thị thông tin giảm giá cũng như camera của user.

Trang quan sát camera: hiển thị hình ảnh mèo ăn và số lần ăn của mèo trong 1 ngày , số lượng thức ăn và nước uống dự trữ nếu như gần hết sẽ thông báo cho user biết.

Trang cài đặt:

Cài đặt ghi/phát âm: nút ghi âm và phát âm để kêu mèo lại vào lúc giờ ăn.

Cài đặt giờ ăn: có ba nút.

Set thời gian ăn và lượng thức ăn.

Set lại ô trống ở ô thời gian để người dùng để sử dụng.

Để thức ăn tự động chạy ra khi nhấn nút (trong trường hợp cho ăn thêm ngoài 3 bữa 1 ngày).

Cài đặt máy bơm nước: Thực hiện tắt mở bơm nước để bơm nước cho mèo uống.

Trang thông tin cá nhân: Để người dùng có thể đăng xuất tài khoản , và hiển thị thông tin cá nhân (họ tên, gmail,...).

3.6. Hoạt động của thiết bị IoT.

Thu thập dữ liệu: Thiết bị IoT có thể thu thập dữ liệu từ các cảm biến hoặc nguồn thông tin khác để đo đạc, giám sát và ghi nhận các thông số như nhiệt độ, độ ẩm, ánh sáng, chất lượng không khí, áp suất, v.v.

Truyền dữ liệu: Thiết bị IoT có khả năng truyền dữ liệu qua mạng Internet. Dữ liệu thu thập từ thiết bị sẽ được truyền đến ứng dụng điện thoại để xử lý và phân tích.

Tương tác: Thiết bị IoT có thể tương tác với nhau để thực hiện các nhiệm vụ hoặc kích hoạt các hành động. Ví dụ, một hệ thống cho ăn tự động có thể cho phép bạn điều khiển thời gian cho ăn, bơm nước, thiết bị âm thanh và các thiết bị khác thông qua điện thoại di động.

Tự động hóa: Thiết bị IoT thực hiện các hoạt động tự động dựa trên dữ liệu thu thập từ Firebase. Ví dụ, hệ thống tự động chăm sóc mèo có thể dựa trên dữ liệu về mực nước để tự động điều chỉnh lượng nước cần thiết cho vật nuôi.

Theo dõi và quản lý từ xa: Thiết bị IoT cho phép người dùng theo dõi và quản lý từ xa các hoạt động và thiết bị. Bằng cách kết nối với mạng Internet, người dùng có thể giám sát và điều khiển các thiết bị IoT từ xa thông qua ứng dụng di động.

Phân tích dữ liệu: Thiết bị IoT thu thập dữ liệu có thể được phân tích để trích xuất thông tin hữu ích và đưa ra quyết định. Phân tích dữ liệu từ các thiết bị IoT có thể giúp cải thiện hiệu suất.

3.7. Hoạt động của mô hình YOLO trong mô hình chăm sóc thú cưng.

Mô hình YOLO (You Only Look Once) là một mô hình phát hiện đối tượng tiên tiến trong lĩnh vực thị giác máy tính. Nó được sử dụng rộng rãi trong nhiều ứng dụng như nhận diện đối tượng, theo dõi vật thể và cả chăm sóc thú cưng.

Trong mô hình chăm sóc thú cưng, YOLO có thể được áp dụng để nhận diện và theo dõi động vật trong hình ảnh hoặc video, nhằm cung cấp thông tin và quan sát cho việc chăm sóc và giám sát thú cưng. Dưới đây là một số cách mà YOLO có thể được sử dụng trong mô hình chăm sóc thú cưng:

- Phát hiện loại động vật: Mô hình YOLO có thể được huấn luyện để nhận diện và phân loại các loại động vật khác nhau như chó, mèo, nhiều loại động vật khác. Điều này giúp phát hiện sự hiện diện của động vật và nhận biết chúng trong các hình ảnh hoặc video.
- Theo dõi vật nuôi: Mô hình YOLO có khả năng theo dõi đối tượng qua các khung hình liên tiếp. Điều này cho phép chúng em theo dõi động vật trong thời gian thực trong video. Điều này hữu ích để giám sát hành vi của thú cưng, đảm bảo chúng an toàn và tìm hiểu thói quen của chúng.

- Phát hiện hành vi không bình thường: YOLO có thể được sử dụng để phát hiện các hành vi không bình thường của thú cưng, như chúng ăn không đủ bữa, biểu hiện sức khỏe kém.

Khi đã có một mô hình IoT thực tế, ứng dụng và mô hình YOLOv8 đã được huấn luyện trên tập dữ liệu của chúng em tìm được sẽ bắt đầu thực nghiệm với mô hình đó.

Chương 3 tập trung vào việc xây dựng hệ thống nhận dạng thú cưng (mèo) trong các thiết bị IoT và sử dụng trí tuệ nhân tạo (AI). Các nội dung chính trong chương này bao gồm ứng dụng mô hình YOLOv8 vào mô hình chăm sóc thú cưng, giao tiếp giữa thiết bị IoT với ứng dụng điện thoại, phát triển ứng dụng và mô tả hoạt động tổng quát của mô hình.

Đầu tiên, chúng em đã trình bày ứng dụng mô hình YOLOv8 vào mô hình chăm sóc thú cưng. Quá trình này bao gồm chuẩn bị bộ dữ liệu và ứng dụng mô hình YOLOv8 đã được huấn luyện để nhận dạng và phát hiện thú cưng, đặc biệt là mèo.

Tiếp theo, chúng em đã kết nối giữa thiết bị IoT và ứng dụng điện thoại để xây dựng giao diện và giao tiếp giữa các thiết bị IoT và ứng dụng điện thoại.

Sau đó, chúng em đã mô tả hoạt động tổng quát của mô hình, bao gồm xây dựng mô hình 3D tự động hóa cho mèo và mô hình tự động hóa động hóa cho mèo. Các hoạt động này đóng vai trò quan trọng trong việc theo dõi và chăm sóc thú cưng một cách tự động và hiệu quả.

Cuối cùng, chúng em đã xem xét hoạt động của ứng dụng di động và thiết bị IoT trong mô hình chăm sóc thú cưng. Ứng dụng di động cho phép người dùng tương tác và kiểm soát các thiết bị IoT để chăm sóc thú cưng, trong khi mô hình YOLO đóng vai trò quan trọng trong việc nhận dạng và phát hiện thú cưng trong thời gian thực.

Chương 3 đã xây dựng nền tảng cần thiết cho hệ thống nhận dạng thú cưng trong các thiết bị IoT và sử dụng AI. Các kết quả và công nghệ đều đóng góp vào phát triển mô hình chăm sóc thú cưng thông minh và tự động.

CHƯƠNG 4. KẾT QUẢ VÀ KHUYẾN NGHỊ.

4.1. Kết quả huấn luyện mô hình YOLOv8.

Trong quá trình huấn luyện, mạng nơ-ron tính toán từng ảnh trong tệp dữ liệu đầu vào và lặp lại các ảnh này nhiều lần để tối ưu hóa hàm mất mát. Mục tiêu là tìm ra bộ trọng số tốt nhất để đạt được độ chính xác cao nhất trong quá trình nhận dạng. Khi huấn luyện thành công, bộ trọng số của mô hình được lưu trữ trong một file kết quả có định dạng ".pt". YOLOv8 cho phép người dùng sử dụng bộ trọng số đã có để tiếp tục quá trình huấn luyện, giúp tiết kiệm thời gian so với việc phải huấn luyện từ đầu.

Quá trình huấn luyện mô hình yêu cầu một lượng lớn tài nguyên tính toán. Nếu máy tính không đủ mạnh, thời gian huấn luyện sẽ kéo dài. Để giảm thời gian huấn luyện và đảm bảo hiệu suất và chất lượng mô hình, người dùng có thể sử dụng Google Colab. Điều này cho phép huấn luyện trên môi trường điện toán đám mây và tận dụng tài nguyên mạnh mẽ của Colab. Số lượng ảnh tối đa trong mỗi batch phụ thuộc vào cấu hình phần cứng của máy tính.

Các tham số được sử dụng chính trong mô hình YOLOv8:

`epochs = 250`: Số lượt huấn luyện trên toàn bộ dữ liệu.

`batch = 16 - 128`: số lượng mỗi ảnh vào vòng huấn luyện (mặc định sẽ là 16).


`imgsz = 640`: kích thước ảnh đầu vào.

`data = "path/to/data.yaml"`: Đường dẫn đến thư mục data chuẩn YOLOv8.

`model = yolov8n.pt`: chọn kích thước model để huấn luyện bộ dữ liệu.

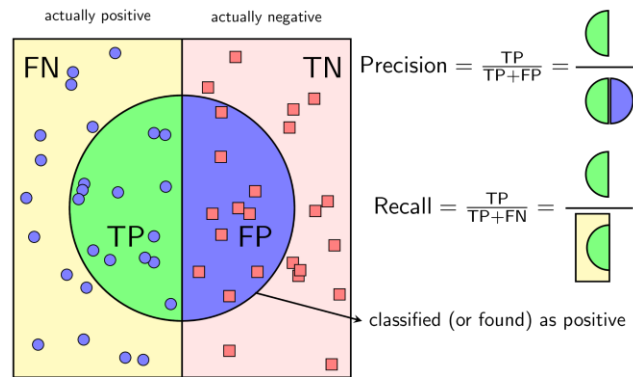
Bên cạnh đó còn có hàm `optimize`, `learning rate` và chỉ số `patient`, đó là những yếu tố quan trọng trong việc tinh chỉnh và đánh giá mô hình. Việc lựa chọn các tham số phù hợp với từng bài toán là một quá trình thực nghiệm và đòi hỏi thời gian.

Intersection over Union (IOU): Đây là một độ đo sử dụng để tính toán mức độ trùng hợp giữa bounding box dự đoán và bounding box thực tế của đối tượng. Nếu IOU vượt qua ngưỡng xác định, ta coi đối tượng đã được phát hiện chính xác[19].

$$\text{Intersection over Union (IOU) or Jaccard Index} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Hình 35. Cách tính Intersection over Union (IOU) [19].

Precision và Recall: Precision biểu thị độ chính xác trong việc dự đoán tên và vị trí của đối tượng, trong khi Recall biểu thị khả năng phát hiện đối tượng trong dữ liệu đầu vào. Hai giá trị này được tính dựa trên True Positive (TP), False Negative (FN), True Negative (TN) và False Positive (FP) nhằm đo lường độ chính xác của mô hình[20]:



Hình 36. Cách tính Precision và Recall [20].

Sau khi cho mô hình YOLOv8 dự đoán trên 760 ảnh của tập xác nhận (valid) thì mô hình dự đoán được 757, giá trị IOU = 0.7 (giá trị được đưa ra bởi YOLOv8) và tìm được các thông số sau:

True Position (TP) = 745.

False Position (FP) = 12.

False Negative (FN) = 34.

Dựa vào TP, FP và FN, chúng em có thể tính toán một số chỉ số đánh giá hiệu suất của mô hình:

$$Precision = \frac{TP}{(TP + FP)} = \frac{745}{(745 + 12)} \approx 0.984$$

$$Recall = \frac{TP}{(TP + FN)} = \frac{745}{(745 + 34)} \approx 0.956$$

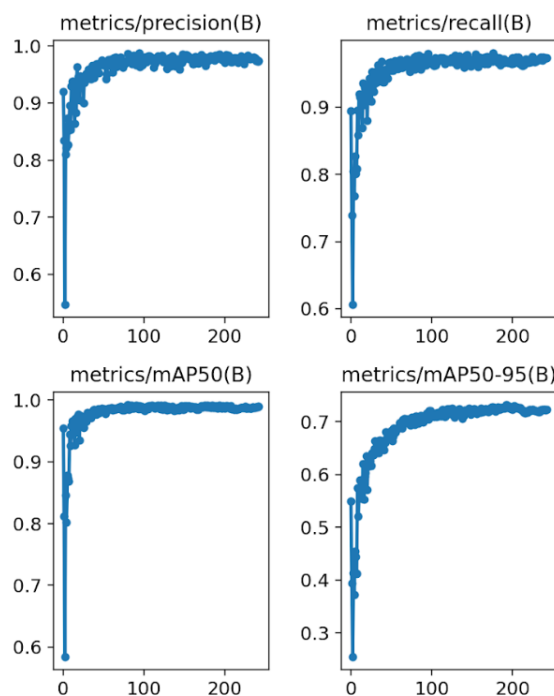
Precision là tỷ lệ tấm hình dự đoán đúng là positive trên tổng số tấm hình được dự đoán là positive. Với precision khoảng 0.984, mô hình có khả năng tránh sai dự đoán positive rất cao.

Recall là tỷ lệ tấm hình dự đoán đúng là positive trên tổng số tấm hình thực tế là positive. Với recall khoảng 0.956, mô hình bắt được hầu hết tấm hình positive, chỉ bỏ sót một số nhỏ.

Để đánh giá Precision và Recall như đã nêu, ta cần hiểu về khái niệm ngưỡng dự đoán IOU. Nếu giá trị IOU vượt qua ngưỡng xác định, ta coi rằng đối tượng đã được dự đoán chính xác. Trên hình ảnh kết quả của mô hình, chúng em có thể thấy các bounding box được vẽ để đại diện cho đối tượng được nhận dạng trong ảnh.

Qua các thông số Precision và Recall cho thấy mô hình YOLOv8 đã được huấn luyện với tập dữ liệu vẫn còn nhiều thiếu sót trong quá trình huấn luyện làm cho quá trình nhận diện vào thực tế vẫn còn nhiều thiếu sót. Vì vậy việc đánh giá kết quả và tinh chỉnh các tham số là quan trọng để đảm bảo mô hình nhận dạng đối tượng với độ chính xác cao và khả năng phát hiện tốt.

Sau đây là kết quả huấn luyện của mô hình YOLOv8.



Hình 37. Kết quả bộ dữ liệu được huấn luyện trên mô hình YOLOv8.

4.2. Kết quả mô hình chăm sóc thú cưng.

Sau khi đưa hệ thống vào hoạt động thực tế thì cho thấy mèo lúc đầu còn không quen chỗ ăn mới và chỗ uống nước nên còn rụt rè sau khi hoạt động một thời gian thì các mèo đã quen thuộc và bắt đầu sử dụng hệ thống này.

Từ hình chụp ta nhận thấy được kết quả tập dữ liệu, kết quả huấn luyện đạt kết quả tốt, dù hình ảnh đang trong điều kiện không rõ và không cụ thể con vật đang phân loại.

Thú cưng được giám sát chính xác hơn: Nhờ vào cảm biến IoT và công nghệ AI, mô hình có thể theo dõi thời gian ăn của thú cưng một cách tự động và liên tục. Chủ thú cưng sẽ nhận được thông tin chi tiết về thời gian ăn, mức thức ăn còn lại, mức nước. Điều này giúp chủ nuôi dễ dàng kiểm soát.

Chăm sóc tự động và linh hoạt: Mô hình cho phép chủ thú cưng thiết lập lịch trình chăm sóc tự động cho thú cưng thông qua ứng dụng di động. Hệ thống AI và IoT sẽ tự động cung cấp thức ăn, nước uống, theo lịch trình đã được thiết lập hoặc có thể tự thiết lập bằng tay.



Hình 38. Ảnh được chụp từ ESP32-Camera đã được nhận dạng.

Sau khi sử dụng ứng dụng để cài thời gian tự động là dữ liệu được đưa lên Firebase và ESP32 sẽ lấy thông tin từ đó và điều khiển các động cơ quay và máy bơm nước cho mèo ăn uống sẽ hoạt động khi tới thời gian đã cài và phần ghi âm từ loa sẽ phát ra âm thanh để mà người chủ đã ghi để kêu gọi con mèo của mình đến ăn.



Hình 39. Hình ảnh động cơ quay thức ăn và vòi nước cho mèo uống đang hoạt động.

Trong khi mèo ăn thì ESP32-Camera sẽ hoạt động để nhận diện là con mèo có đến chỗ ăn sẽ đưa ra thông báo trên ứng dụng hoặc người chủ có thể quan sát trên ứng dụng bằng các hình ảnh đã được nhận dạng và thay đổi liên tục từ Firebase.

Từ việc quan sát việc cho ăn uống của mèo mà từ đó người chủ có thể đưa ra các đánh giá về thú cưng của mình là đang gặp vấn đề về cơ thể cũng như tâm lý của mèo.

4.3. Đánh giá kết quả.

Sau khi chạy thực nghiệm với thú cưng của chúng em, đặc biệt là mèo, chúng em đã thấy mô hình YOLOv8 hoạt động hiệu quả với kết quả nhận diện tốt. Mức độ nhận dạng sai và phát hiện đối tượng không chính xác của mô hình này là khá thấp, cho thấy khả năng nhận dạng đáng kể. Điều này đảm bảo rằng mô hình có thể đáp ứng yêu cầu của chúng em trong việc nhận diện đúng thú cưng và loại bỏ các sai sót không mong muốn.

Bên cạnh đó, các thiết bị IoT cũng hoạt động tốt với chương trình đã được nạp vào thiết bị kết nối. Kết quả là, chúng em đã có thể giám sát việc thú cưng của mình ăn uống từ xa và nhận được thông tin quan trọng về hành vi và tình trạng sức khỏe của chúng.

Về phần ứng dụng di động, chúng em đã thành công trong việc lấy dữ liệu từ Firebase và cập nhật dữ liệu một cách trơn tru mà không gây ra bất kỳ lỗi nào trên nền tảng Firebase. Điều này cho thấy ứng dụng di động đã được phát triển với mức độ hoàn thiện cao và đã trải qua quá trình kiểm thử và sửa lỗi kỹ lưỡng.

Mặc dù các thiết bị IoT hoạt động tốt nhưng vẫn còn một số vấn đề như mèo khi mới tiếp cận với hệ thống cho ăn, uống. Thiết bị ESP32-CAM có góc chụp rất nhỏ nên việc quan sát con mèo của người chủ có nhiều khó khăn và mô hình YOLOv8 có thể nhận diện nhầm lẫn đưa ra thông báo sai khi người chủ không quan sát khi những lúc con mèo mình đang ăn uống.

Tổng kết lại, mô hình YOLOv8 đã chứng minh khả năng nhận diện của mô hình, các thiết bị IoT đã hoạt động một cách ổn định và ứng dụng di động đã hoàn thiện với việc lấy dữ liệu và giao tiếp một cách mượt mà. Những thành tựu này sẽ tiếp tục đóng góp vào việc quản lý và chăm sóc tốt hơn cho thú cưng.

4.4. Hướng phát triển.

Mặc dù đã có khả năng nhận diện thú cưng khi đến chỗ cho ăn, việc xác định thú cưng đã ăn thức ăn đó hay chưa vẫn còn chưa chính xác. Để cải thiện điều này, một hướng phát triển tiếp theo có thể là tăng cường hiệu suất của mô hình YOLOv8 bằng cách nghiên cứu và cải tiến. Bằng cách áp dụng trí tuệ nhân tạo nâng cao như học sâu hoặc học tăng cường, mô

hình có thể tự động học và cải thiện theo thời gian, nâng cao khả năng nhận diện và giảm thiểu sai sót.

Đồng thời, mở rộng ứng dụng mobile có thể mang lại thêm giá trị cho người dùng. Bằng cách tích hợp tính năng gửi cảnh báo khi thú cưng rời khỏi khu vực đã định sẵn hoặc tích hợp trò chơi và tính năng giải trí liên quan đến thú cưng, ứng dụng di động có thể trở nên thông minh và hữu ích hơn.

Mở rộng phạm vi áp dụng của mô hình cũng mang lại tiềm năng và cơ hội kinh doanh. Ngoài việc áp dụng cho chăm sóc thú cưng, mô hình có thể được sử dụng trong lĩnh vực an ninh, quản lý kho hàng hay nhận dạng đối tượng trong các lĩnh vực khác.

Để đánh giá hoạt động ăn uống của thú cưng một cách chính xác, có thể phát triển hệ thống theo dõi hoạt động ăn uống bằng cách sử dụng cân, cảm biến chuyển động hoặc hệ thống theo dõi mức độ tiêu thụ thức ăn. Kết hợp với thông tin từ mô hình nhận diện YOLOv8 và dữ liệu từ các cảm biến này, người dùng có thể theo dõi và đánh giá mức độ ăn uống của thú cưng một cách chính xác hơn.

Cuối cùng, để cung cấp khả năng xem video trực tiếp từ ESP32 Camera trên ứng dụng di động, cần triển khai các công nghệ phát video trực tiếp như WebRTC(Web Real-Time Communications) hoặc RTSP (Real Time Streaming Protocol). Bằng cách tích hợp các giao thức này vào ứng dụng di động, người dùng có thể xem video từ camera ESP32 một cách trực tiếp và tiện lợi, mở ra khả năng giám sát thú cưng và tương tác với chúng từ xa một cách linh hoạt hơn.

KẾT LUẬN VÀ KHUYẾN NGHỊ.

Tóm lại, trong báo cáo này chúng em đã xem xét và triển khai mô hình nhận diện YOLOv8 kết hợp với công nghệ IoT để phát triển ứng dụng chăm sóc thú cưng. Kết quả cho thấy mô hình nhận diện đã cho kết quả khả quan và có thể nhận diện thú cưng đến chỗ cho ăn. Tuy nhiên, việc xác định thú cưng có ăn đồ ăn đã đưa tới hay không vẫn còn là một thách thức.

Mở rộng ứng dụng mobile cũng là một hướng phát triển tiếp theo quan trọng. Chúng em có thể tích hợp tính năng gửi cảnh báo khi thú cưng rời khỏi khu vực đã định sẵn hoặc thêm các tính năng giải trí và tương tác liên quan đến thú cưng.

Ngoài ra, để đạt được mục tiêu xác nhận thú cưng có ăn đồ ăn đã đưa tới hay không, chúng em có thể phát triển hệ thống theo dõi hoạt động ăn uống sử dụng cân, cảm biến chuyển động hoặc hệ thống theo dõi mức độ tiêu thụ thức ăn. Kết hợp thông tin từ mô hình nhận diện và dữ liệu từ các cảm biến này, người dùng có thể theo dõi và đánh giá mức độ ăn uống của thú cưng một cách chính xác hơn.

Cuối cùng, chúng em cũng có thể mở rộng phạm vi áp dụng của ứng dụng này sang các lĩnh vực khác như an ninh, quản lý kho hàng hoặc nhận dạng đối tượng trong các lĩnh vực khác. Điều này sẽ tạo ra nhiều ứng dụng tiềm năng và mở rộng cơ hội kinh doanh.

Tổng kết lại, việc kết hợp mô hình nhận dạng YOLOv8 và công nghệ IoT đã mở ra nhiều cơ hội trong việc chăm sóc thú cưng. Tuy nhiên, còn rất nhiều công việc để nghiên cứu và phát triển tiếp theo để cải thiện hiệu suất, mở rộng tính năng và đạt được mục tiêu xác nhận thú cưng có ăn đồ ăn đã đưa tới hay không.

TÀI LIỆU THAM KHẢO

- [1] *Rung-Ching Chen, “Monitoring the behaviors of pet cat based on YOLO model and raspberry Pi”, January 2021 International Journal of Applied Science and Engineering 18(5):1-12.*
- [2] *Simplilearn (May 17, 2023), “What Is Computer Vision: Applications, Benefits and How to Learn It”(https://www.simplilearn.com/computer-vision-article).*
- [3] *M. Tan, W. Chao, J.-K. Cheng, M. Zhou, Y. Ma, X. Jiang, J. Ge, L. Yu and L. Feng, “Animal Detection and Classification from Camera Trap Images Using Different Mainstream Object Detection Architectures,” Animals, vol. 12, no. 15, p. 1976, Aug. 2022.*
- [4] *Simplilearn (Feb 21, 2023), “The Best Introduction to Deep Learning - A Step by Step Guide”. (https://www.simplilearn.com/tutorials/deep-learning-tutorial/introduction-to-deep-learning).*
- [5] *MathWorks,(7 May 2019), “What Is Deep Learning?3 things you need to know”, (https://www.edge-ai-vision.com/2019/05/what-is-deep-learning-three-things-you-need-to-know).*
- [6] *H. Bui and P. H. Phuong, “Image search using deep learning method” VNU Journal of Science: Computer Science and Communication Engineering, vol. 37, no. 2, pp. 1-10, Dec. 2021.*
- [7] *Gaurav Maindola (August 27, 2021), “A Brief History of YOLO Object Detection Models From YOLOv1 to YOLOv5” (https://machinelearningknowledge.ai/a-brief-history-of-yolo-object-detection-models).*

- [8] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (13 June 2019), *You Only Look Once: Unified, Real-Time Object Detection* (<https://papers.readthedocs.io/en/latest/imagedetection/yolo/>).
- [9] Jay, (06 May 2022), “Giới thiệu về YOLO: Phát hiện đối tượng thời gian thực”, (<https://hashdork.com/vi/c%C6%B0%E1%BB%9Di-l%E1%BB%9Bn/>).
- [10] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh. “CSPNet: A new backbone that can enhance learning capability of CNN”, 2020, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, vol. 2020-June, pp. 1571–1580.
- [11] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun. *YOLOX: Exceeding YOLO Series in 2021*. 2021, pp. 1–7.
- [12] Chris Crawford and 1 collaborator updated 5 years ago(2019), *Cat Dataset*, (<https://www.kaggle.com/datasets/crawford/cat-dataset>).
- [13] Danielsson, W., Froberg , A., & Berglund, E. (2016). *React Native Application Development- A comparison between native Android and ReactNative* ,(pp. 1 - 70).
- [14] J. Stangarone, “Mrc Productivity,” [Online]. Available: *The Mobile App Comparison Chart: Hybrid vs. Native vs. Mobile Web*.
- [15] Li WJ, Yen C, Lin YS, Tung SC, Huang S. *Just IoT Internet of Things based on the Firebase real-time database. In 2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE) 2018 Feb 8* (pp. 43-47).
- [16] Alsalemi A, Al Homsy Y, Al Disi M, Ahmed I, Bensaali F, Amira A, Alinier G. “Real-time communication network using firebase cloud IoT platform for ECMO

- simulation*". In 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) 2017 Jun 21 (pp. 178-182).
- [17] "Firebase | App success made simple," Firebase, [Online]. (<https://firebase.google.com>).
- [18] Tram, Mai. "Firebase." (<https://www.theseus.fi/handle/10024/263641>) (2019).
- [19] Adrian Rosebrock (November 7, 2016), "Intersection over Union (IoU) for object detection" (<https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection>).
- [20] Machine Learning cơ bản (Jan 3, 2018), "Bài 33: Các phương pháp đánh giá một hệ thống phân lớp", (<https://machinelearningcoban.com/2017/08/31/evaluation>).