

ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA ĐIỆN TỬ-VIỄN THÔNG

# ĐỒ ÁN TỐT NGHIỆP

NGÀNH: ĐIỆN TỬ-VIỄN THÔNG  
CHUYÊN NGÀNH: KỸ THUẬT MÁY TÍNH

ĐỀ TÀI:

## ỨNG DỤNG THIẾT BỊ NHÚNG TRONG NHẬN DẠNG CHỮ SỐ ĐỒNG HỒ NƯỚC

Người hướng dẫn: ThS. TRẦN VĂN LÍC

Sinh viên thực hiện: NGUYỄN TIẾN DŨNG – 106190148

LÊ ĐỨC QUỐC ANH-106190143

Lớp: 19DTCLC4

Đà Nẵng, 7/2023

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN TỬ-VIỄN THÔNG**

# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: ĐIỆN TỬ-VIỄN THÔNG**  
**CHUYÊN NGÀNH: KỸ THUẬT MÁY TÍNH**

**ĐỀ TÀI:**

## **ỨNG DỤNG THIẾT BỊ NHÚNG TRONG NHẬN DẠNG CHỮ SỐ ĐỒNG HỒ NƯỚC**

Người hướng dẫn: **ThS. TRẦN VĂN LÍC**

Sinh viên thực hiện: **NGUYỄN TIẾN DŨNG – 106190148**

**LÊ ĐỨC QUỐC ANH–106190143**

Lớp: **19DTCLC4**

**Đà Nẵng, 7/2023**

## TÓM TẮT

**Tên đề tài:** *Ứng dụng thiết bị nhúng trong nhận dạng chữ số đồng hồ nước*

Sinh viên: Nguyễn Tiến Dũng – mã sinh viên: 106190148 – Lớp: 19DTCLC4.

Sinh viên: Lê Đức Quốc Anh - mã sinh viên: 106190143 – Lớp: 19DTCLC4.

Ngày nay, thời đại công nghệ ngày càng phát triển, những bước tiến vượt trội về công nghệ đang giúp mọi thứ trong cuộc sống ngày càng trở nên dễ dàng hơn. Để giảm chi phí cho nguồn nhân công trong quá trình thống kê các con số khi sử dụng đồng hồ nước của mỗi hộ gia đình, và giúp người dân kiểm soát được các mức sử dụng nước nhằm tiết kiệm nước không bị hoang phí nên nhóm đã thực hiện một giải pháp giúp tiết kiệm về sức lao động và thời gian, tiền bạc - một thiết bị thông minh được lắp đặt ngay trong chính hộ gia đình “*Ứng dụng thiết bị nhúng trong nhận dạng chữ số đồng hồ nước*” là đề tài mà nhóm muốn thực hiện.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

TT	Họ tên sinh viên	Số thẻ SV	Lớp	Ngành
1	Nguyễn Tiến Dũng	106190148	19DTCLC4	Điện tử viễn thông
2	Lê Đức Quốc Anh	106190143	19DTCLC4	Điện tử viễn thông

1. Tên đề tài đồ án:  
    Ứng dụng thiết bị nhúng trong nhận dạng chữ số đồng hồ nước.
2. Đề tài thuộc diện: ☐ Đề tài được giao cho nhóm sinh viên thực hiện.
3. Các số liệu và dữ liệu ban đầu:  
    Thông tin về LoRaWAN  
    Thông tin về các linh kiện, thiết bị  
    Thông tin về Gateway
4. Nội dung các phần thuyết minh và tính toán.

TT	Họ tên sinh viên	Nội dung
1	Nguyễn Tiến Dũng	Nghiên cứu mô hình CNN Lập trình xây dựng mô hình CNN Xử lý ảnh và xây dựng thư viện xử lý ảnh Chụp ảnh bằng ESP32-CAM Nhúng mô hình đã huấn luyện
2	Lê Đức Quốc Anh	Nghiên cứu mô hình CNN Lập trình xây dựng mô hình CNN Xử lý ảnh và xây dựng thư viện xử lý ảnh Chụp ảnh bằng ESP32-CAM Nhúng mô hình đã huấn luyện

5. *Họ và tên người hướng dẫn.*

ThS. Trần Văn Líc

6. *Ngày giao nhiệm vụ đồ án:* 01/04/2023.

7. *Ngày hoàn thành đồ án:* 14/07/2023.

*Đà Nẵng, ngày tháng năm 2023*

**Trưởng Bộ môn.....**

**Người hướng dẫn**

## LỜI NÓI ĐẦU

Chúng em xin gửi lời cảm ơn chân thành đến khoa Điện tử - Viễn thông, trường Đại học Bách Khoa Đà Nẵng đã tạo điều kiện tốt nhất để chúng em thực hiện đề tài. Em xin đặc biệt cảm ơn thầy Trần Văn Líc, thầy Hồ Phước Tiến, thầy Phan Trần Đăng Khoa cùng những anh chị đi trước vì những ý kiến đóng góp quý giá và sự hỗ trợ nhiệt tình để giúp chúng em thực hiện đề tài này.

Những kiến thức được học trong quá trình rèn luyện khi ở trường là nguồn tài nguyên quý giá giúp chúng em tăng cường kiến thức để thực hiện đề tài cũng như là hành trang quý báu giúp nhóm chúng em đi những bước đầu tiên trên con đường theo đuổi sự nghiệp của mình.

Đề tài tuy sẽ còn những thiếu sót về mặt chuyên môn, còn nhiều mục tiêu đề ra mà chưa đáp ứng được nhưng chúng em đã cố gắng hoàn thành hết mình bằng khả năng của bản thân. Chúng em mong nhận được sự cảm thông cũng như chia sẻ góp ý quý báu từ thầy cô về mục tiêu đề tài để đề tài được hoàn thiện hơn trong tương lai.

Một lần nữa chúng em xin chân thành cảm ơn!

*Đà Nẵng, ngày tháng năm 2023*

Nhóm sinh viên thực hiện

Nguyễn Tiến Dũng – Lê Đức Quốc Anh

## **LỜI CAM ĐOAN**

**Kính gửi:** Hội đồng chấm đồ án tốt nghiệp Khoa Điện tử – Viễn thông, trường Đại học Bách Khoa – Đại học Đà Nẵng.

**Họ và tên sinh viên:** Nguyễn Tiến Dũng – 106190148

Lê Đức Quốc Anh – 106190143

**Lớp:** 19DTCLC4

Chúng em xin cam đoan đồ án tuân thủ tốt các quy định về liêm chính học thuật:

- Không sử dụng các hình thức gian dối trong việc trình bày thể hiện các hoạt động.
- Không bịa đặt, đưa ra các thông tin sai lệch so với nguồn gốc trích dẫn.
- Không ngụy tạo số liệu trong quá trình khảo sát, thí nghiệm, thực hành, thực tập.
- Không đạo văn, sử dụng từ ngữ, cách trích dẫn, diễn đạt của người khác như thể là của mình, trình bày, sao chép, dịch đoạn, hoặc nêu ý tưởng của người khác mà không có trích dẫn.
- Không đạo văn, sử dụng lại thông tin nghiên cứu của mình mà không có trích dẫn hoặc phân mảnh thông tin về kết quả nghiên cứu của mình để công bố trên nhiều ấn phẩm.

Sinh viên thực hiện 1

Sinh viên thực hiện 2

**Nguyễn Tiến Dũng**

**Lê Đức Quốc Anh**

## MỤC LỤC

LỜI NÓI ĐẦU .....	i
LỜI CAM ĐOAN .....	ii
MỤC LỤC.....	iii
DANH SÁCH CÁC BẢNG, HÌNH VẼ .....	vi
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT .....	ix
MỞ ĐẦU.....	1
CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI .....	3
1.1 Tổng quan về đề tài .....	3
1.2 Đề xuất giải pháp .....	10
1.3 Mục tiêu nghiên cứu .....	11
1.4 Đối tượng và phạm vi nghiên cứu.....	11
1.4.1 Đối tượng nghiên cứu .....	11
1.4.2 Phạm vi nghiên cứu.....	11
1.5 Các tiếp cận và phương pháp nghiên cứu .....	12
1.5.1 Cách tiếp cận.....	12
1.5.2 Phương pháp nghiên cứu .....	12
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT .....	13
2.1 Thiết bị nhúng ESP32-CAM.....	13
2.2 Mạng nơ - ron tích chập CNN .....	14
2.2.1 Giới thiệu về máy học .....	14
2.2.2 Khởi tạo dữ liệu huấn luyện trong máy học.....	18



2.2.3 Đánh giá chất lượng một mô hình máy học .....	20
2.3 Thư viện TensorFlow Lite cho Vi xử lý .....	25
2.3.1 OCR Tesseract .....	25
2.3.2 OpenCV .....	25
2.3.3 TensorFlow Lite .....	25
2.4 Mạng không dây LoRaWAN .....	29
2.4.1 Kỹ thuật điều chế tín hiệu LoRa .....	31
2.4.2 Giao thức LoRaWAN .....	33
2.4.3 Bảo mật trong mạng LoRaWAN .....	38
2.5 KẾT LUẬN .....	40
CHƯƠNG 3 THIẾT KẾ VÀ PHÂN TÍCH HỆ THỐNG .....	43
3.1 Tổng quan sơ đồ hệ thống .....	43
3.2 Thu thập dữ liệu .....	44
3.2.1 Giới thiệu giải thuật xử lý ảnh trên ESP32-CAM .....	44
3.2.2 Thu thập và xử lý dữ liệu máy học .....	46
3.3 Thực hiện mô hình máy học (model training) .....	53
3.4 Triển khai mô hình trên ESP32-CAM .....	54
3.4.1 Công cụ sử dụng triển khai trên ESP32-Cam .....	54
3.4.2 Các thư viện cần thiết và cài đặt các thông số trong chương trình .....	55
3.5 Truyền tin qua LoRaWAN .....	56
CHƯƠNG 4 KẾT QUẢ VÀ ĐÁNH GIÁ HỆ THỐNG .....	58
4.1 Kết quả .....	58
4.1.1 Hiệu năng của mô hình máy học .....	58
4.1.2 Kết quả thử nghiệm trên máy tính .....	60
4.1.3 Kết quả thử nghiệm thực tế .....	62

4.2 Đánh giá thiết kế phần cứng .....	65
4.3 Đánh giá quá trình triển khai thực tế .....	66
KẾT LUẬN.....	67
HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI .....	69
TÀI LIỆU THAM KHẢO.....	70

## DANH SÁCH CÁC BẢNG, HÌNH VẼ

Bảng 2.1 Ví dụ về ma trận lỗi.....	20
Bảng 2.2 Ví dụ về True/False Positive/Negative .....	21
Bảng 2.3 Ví dụ 2 về True/False Positive/Negative .....	22
Bảng 2.4 Ví dụ về Precision/Recall/F1 – score .....	23
Bảng 2.5 Ví dụ về Macro/weighted - averaged Precision/Recall/F1 – score.....	24
Bảng 2.6 So sánh LoRa và các công nghệ truyền thông IoT hoặc M2M khác .....	30
Bảng 2.7 So sánh kết nối OTAA và ABP .....	38
Bảng 3.1 Thiết kế mô hình máy học của nhóm.....	53
Bảng 3.2 Bảng cấu hình siêu tham số cho mô hình .....	54
Hình 1.1 Đồng hồ nước kết nối sóng Apator Powogaz.....	3
Hình 1.2 Đồng hồ đo nước siêu âm Apator Model Ultrimis W .....	4
Hình 1.3 Đồng hồ nước MBUS Apator Powogaz Model JS16-AT MBUS-02 .....	4
Hình 1.4 Đồng hồ nước thông minh model SWM015B .....	5
Hình 1.5 Hệ thống giám sát nước thông minh .....	6
Hình 1.6 Hệ thống thông minh có EIM bên ngoài .....	6
Hình 1.7 Đồng hồ nước STUF-280T sử dụng công nghệ đo lượng nước siêu âm .....	7
Hình 1.8 Hệ thống bao gồm modul WROOM32 TM và ArduCam OV 7670.....	8
Hình 1.9 Hệ thống nhận dạng dựa trên Raspberry Pi.....	8
Hình 1.10 Tổng thể về giải pháp SWaMM .....	9
Hình 2.1 Sơ đồ chân ESP32-CAM.....	14
Hình 2.2 Mối quan hệ giữa trí tuệ nhân tạo, máy học và học sâu [17] .....	14
Hình 2.3 So sánh máy học và học sâu [18] .....	15
Hình 2.4 Các nhánh máy học thường gặp [19].....	15
Hình 2.5 So sánh mạng thần kinh tự nhiên và nhân tạo .....	17
Hình 2.6 Quá trình lan truyền ngược.....	18

Hình 2.7 Minh hoạ các dạng dữ liệu [25].....	19
Hình 2.8 So sánh các framework trí tuệ nhân tạo [32].....	26
Hình 2.9 So sánh giữa TensorFlow Lite và STM32Cube.AI [33] .....	27
Hình 2.10 Vị trí của LoRa và LoRaWAN trong mô hình OSI.....	32
Hình 2.11 Một số ưu điểm của mạng sử dụng LoRaWAN [35] .....	34
Hình 2.12 Vị trí của LoRaWAN, LoRa trong mô hình mạng .....	35
Hình 2.13 Sơ đồ các thành phần trong mạng LoRaWAN .....	35
Hình 2.14 Mã hoá được tạo ra trong quá trình ủy nhiệm quyền kết nối .....	39
Hình 2.15 Thiết bị đầu cuối gửi gói tin yêu cầu kết nối.....	39
Hình 2.16 Máy chủ kết nối gửi trả về gói tin cho phép kết nối.....	40
Hình 3.1 Sơ đồ hệ thống.....	43
Hình 3.2 Module ESP32-CAM tích hợp camera OV2640 .....	44
Hình 3.3 Lưu đồ quá trình xử lý ảnh trực tiếp trên ESP32-CAM .....	45
Hình 3.4 Ảnh bao quanh 5 chữ số .....	45
Hình 3.5 Ảnh nhị phân của từng chữ số .....	46
Hình 3.6 Ảnh minh họa bộ dataset của tác giả Kucev Roman.....	46
Hình 3.7 Ảnh minh họa bộ dữ liệu UFPR-ADMR.....	47
Hình 3.8 Ảnh minh họa bộ dữ liệu Copel-AMR.....	48
Hình 3.9 Ảnh minh họa bộ dữ liệu của tác giả Sloan96.....	48
Hình 3.10 Ảnh sau khi zoom gần .....	49
Hình 3.11 Ảnh minh họa bộ dữ liệu MR-AMR .....	49
Hình 3.12 Một số hình ảnh bộ dữ liệu mà nhóm thu thập trực tiếp trên ESP32-CAM.....	50
Hình 3.13 Lưu đồ khối tiền xử lý ảnh và huấn luyện máy học .....	50
Hình 3.14 Thiết bị thu thập dữ liệu huấn luyện.....	51
Hình 3.15 Biểu đồ phân bố dữ liệu cho từng lớp .....	51
Hình 3.16 Ảnh minh họa một số trường hợp số quay ở giữa của dữ liệu .....	52
Hình 3.17 Espressif IDF trên Visual Studio Code .....	55

Hình 3.18 Các thư viện cần thiết khi làm việc trên Espressif IDF .....	55
Hình 3.19 Board LoRa RF Thing UCA.....	57
Hình 4.1 Biểu đồ quá trình huấn luyện, bảng phân phối độ chính xác .....	58
Hình 4.2 Biểu đồ quá trình huấn luyện, bảng phân phối độ chính xác với KFold.....	59
Hình 4.3 Giá trị Precision/Recall .....	59
Hình 4.4 Bảng Confusion Matrix .....	60
Hình 4.5 Ví dụ minh họa về trường hợp dự đoán đúng với ảnh tốt .....	61
Hình 4.6 Ví dụ minh họa về trường hợp dự đoán đúng với ảnh xấu.....	62
Hình 4.7 Hai định dạng ảnh được chụp từ Camera .....	62
Hình 4.8 Kết quả là các giá trị huấn luyện cuối cùng .....	63
Hình 4.9 Biểu đồ Accuracy và Loss đối với model được nhúng vào ESP32-Cam.....	63
Hình 4.10 Hai loại định dạng ảnh được chụp sau khi đã qua xử lý và lưu lại .....	63
Hình 4.11 Kết quả dự đoán trực tiếp trên ESP32-Cam .....	64
Hình 4.12 Phần hệ thống Camera đặt trên mặt đồng hồ nước .....	65

## DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Cụm từ viết tắt	Tên đầy đủ	Tiếng Việt
IoT	Internet Of Things	Internet vạn vật
AI	Artificial Intelligence	Trí tuệ nhân tạo
CPU	Central Processing Unit	Bộ xử lý trung tâm
GPU	Graphics Processing Unit	Bộ xử lý đồ họa
ANN	Artificial Neural Network	Mạng nơ-ron nhân tạo
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
LoRa	Long Range Radio	Truyền thông tầm xa
LoRaWan	Long Range Wireless Area Network	Mạng không dây diện rộng
LPWAN	Low-Power Wide Area Network	Mạng diện rộng năng lượng thấp
HSV	Hue Saturation Value	Không gian màu HSV
4G	Fourth Generation of broadband cellular network	Công nghệ truyền thông không dây thế hệ thứ 4
5G	Fifth Generation of broadband cellular network	Công nghệ truyền thông không dây thế hệ thứ 5
GCC	GNU Compiler Collection	Tập hợp các trình biên dịch GNU
MCU	Microcontroller Unit	Vi điều khiển
OCR	Optical Character Recognition	Nhận dạng kí tự
RAM	Random Access Memory	Bộ nhớ truy cập tạm thời
ROM	Read-Only Memory	Bộ nhớ chỉ đọc
SVM	Support Vector Machine	Máy vector hỗ trợ
WAN	Wide Area Network	Mạng diện rộng

## MỞ ĐẦU

### 1. Nhu cầu thực tế và tính cấp thiết của đề tài

Công nghệ đã và đang thúc đẩy sự phát triển trên toàn cầu nói chung và Việt Nam nói riêng, cùng với chính sách hỗ trợ của nước ta đang giúp cho sự phát triển của ngành công nghệ thông tin, đặc biệt là trong lĩnh vực trí tuệ nhân tạo, khoa học dữ liệu và Internet of Things (IoT). Một số thành tựu đáng kể trong nhiều lĩnh vực với ứng dụng hệ thống IoT như: camera giám sát thông minh, nhà thông minh, hệ thống quản lý và giám sát các thông số như nhiệt độ, độ ẩm, đèn chiếu sáng và tưới nước tự động, hệ thống quản lý năng lượng thông minh, giao thông thông minh và chăm sóc sức khỏe thông minh...

Tuy nhiên, đối với các lĩnh vực nêu trên vẫn còn khá mới đối với việc ứng dụng trong đời sống nhân dân cả nước. Tại mỗi khu vực hộ dân hàng tháng sẽ có nhân viên đến từng hộ, kiểm tra thông số nước thông qua đồng hồ nước hộ gia đình, sau đó thống kê lại và báo cáo với công ty hoặc đơn vị quản lý nước ở đó để họ nhập số liệu lên hệ thống, từ đó thống kê được lượng nước sử dụng của tất cả hộ dân trong một khu vực nhất định và gửi tin nhắn thu phí nước tiêu thụ đến từng nhà. Cách làm này lộ rõ khuyết điểm như tốn chi phí thuê nhân lực cho một công việc quá đơn giản, thủ tục rườm rà mất thời gian do phải đến từng hộ dân, trong trường hợp hộ dân đó đi vắng thì nhân viên lại phải tiếp tục đến vào hôm khác, chủ nhà có thể gian lận bằng cách tự viết chỉ số nước giả lên bảng gắn ở ngoài nhà... Tiếp nối xu thế phát triển các hạ tầng IoT trong cuộc cách mạng công nghiệp 4.0, nhóm đề xuất ra ý tưởng kết nối toàn bộ hạ tầng đồng hồ nước hiện nay của Việt Nam lại với nhau bằng dự án chế tạo thiết bị tự động nhận diện chỉ số nước ứng dụng trí tuệ nhân tạo và kết nối tới hạ tầng LoRaWAN.

### 2. Mục tiêu đề tài

Thiết kế một hệ thống giám sát khối lượng nước sử dụng hàng tháng bằng công nghệ AI kết hợp camera để tự động thu thập dữ liệu trên mặt đồng hồ nước.

### 3. Phương pháp nghiên cứu

Phương pháp nghiên cứu xuyên suốt đồ án tốt nghiệp là tìm hiểu các đề tài liên quan, bài báo quốc tế có nội dung về các công nghệ mạng không dây LoRa/LoRaWAN, các thuật toán nhận diện, phân loại thuộc về “AI on the Edge” – một lĩnh vực AI trên nền vi xử lý, ngoài ra, tìm hiểu các phương thức về thuật toán mạng nơ ron tích chập và so

sánh. Cũng cần phải tự đánh giá ưu khuyết điểm của các loại phương thức, thuật toán,...được ứng dụng nhiều trong AI. Từ đó thiết kế, thực hiện áp dụng vào thực tế.

#### **4. Cấu trúc đồ án tốt nghiệp**

Cấu trúc đồ án gồm có bốn chương như sau:

##### **Chương 1:** Tổng quan về đề tài

Trong chương này tìm hiểu tổng quan về đề tài *Ứng dụng thiết bị nhúng trong nhận dạng chữ số đồng hồ nước*.

##### **Chương 2:** Cơ sở lý thuyết

Trong chương này sẽ tìm hiểu tổng quan về các lý thuyết nền tảng được ứng dụng trong thiết bị nhúng nhận dạng chữ số đồng hồ nước.

##### **Chương 3:** Thiết kế hệ thống

Trong chương này sẽ thực hiện việc lựa chọn các mô-đun cấu tạo nên phần cứng, cách phát triển nên phần mềm của hệ thống.

##### **Chương 4:** Kết quả và đánh giá

Nội dung chương 4 trình bày về những kết quả đạt được của đề tài về việc đánh giá hiệu quả đề tài.

Tuy nhiên do thời gian có hạn nên báo cáo này còn chưa thể đề cập đầy đủ mọi vấn đề liên quan, và chắc chắn không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được nhiều ý kiến đóng góp để chúng em có thêm những kiến thức quý báu cho những công việc trong tương lai.



## **CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI**

### **1.1 Tổng quan về đề tài**

#### **Giới thiệu về các loại đồng hồ nước thông minh**

Hiện nay, Việt Nam đã sản xuất một số thiết bị đồng hồ nước thông minh giúp quản lý và sử dụng nước hiệu quả. Tuy nhiên việc sử dụng các thiết bị này đều phải thay thế toàn bộ đồng hồ nước gây ra khó khăn trong việc lắp đặt và chi phí lớn nếu triển khai trên diện rộng. Dưới đây là một số thiết bị đồng hồ nước đáng chú ý:

- Đồng hồ nước kết nối sóng radio Apator Powogaz của công ty cổ phần quốc tế Thiên Sinh Thái [1]:



Hình 1.1 Đồng hồ nước kết nối sóng Apator Powogaz

- Đồng hồ đo nước siêu âm Apator Model Ultrimis W của công ty cổ phần quốc tế Thiên Sinh Thái [2]:



Hình 1.2 Đồng hồ đo nước siêu âm Apator Model Ultrimis W

- Đồng hồ nước MBUS Apator Powogaz Model JS16-AT MBUS-02 của công ty cổ phần quốc tế Thiên Sinh Thái [3]:



Hình 1.3 Đồng hồ nước MBUS Apator Powogaz Model JS16-AT MBUS-02

- Đồng hồ nước thông minh model SWM015B của công ty Rynan Technologies [4]:



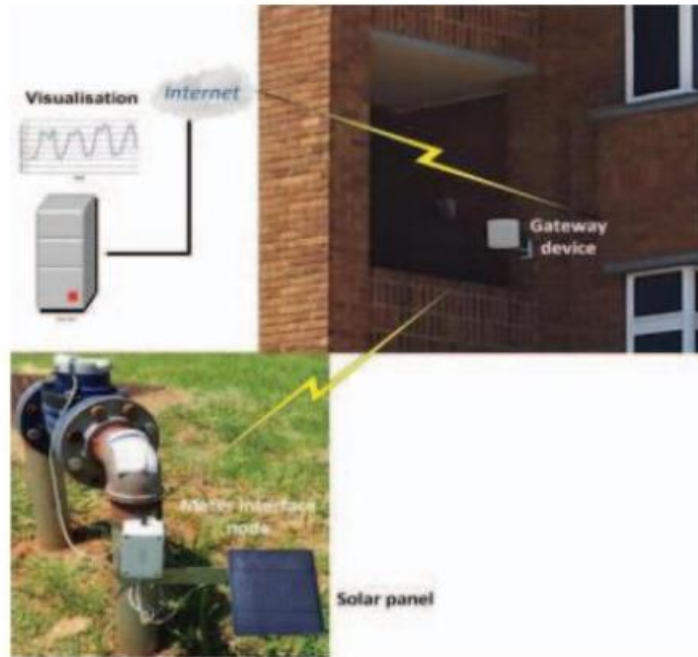
Hình 1.4 Đồng hồ nước thông minh model SWM015B

Nhìn chung các sản phẩm đồng hồ nước thông minh ở Việt Nam có ưu điểm là độ chính xác cao. Nhược điểm của thiết bị này là chi phí cao, muốn thay đổi phải xin phép cơ quan chức năng nhà nước mới cho phép thay mới toàn bộ đồng hồ nước trên diện rộng dẫn đến thủ tục rườm rà.

Ở nước ngoài hiện nay có một số đề tài nghiên cứu gần đây chế tạo đồng hồ nước với sự can thiệp vào cấu trúc của đồng hồ nước, một số đề tài không can thiệp vào cấu trúc nhưng yêu cầu phần cứng cao, có kết nối Internet và mức tiêu thụ năng lượng đáng kể nên chi phí phát sinh lớn và không phù hợp với nhu cầu thực tế ở Việt Nam. Dưới đây là một số đề tài mà nhóm đã tìm hiểu:

- Smart water meter system for user - centric consumption measurement [5]:

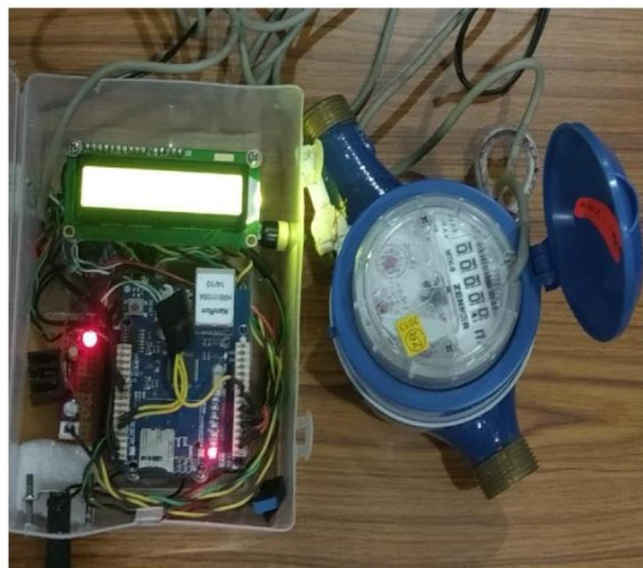
Hệ thống quản lý nước dựa trên cảm biến không dây (WSN) và được giám sát thông qua Pandora FMS. Dữ liệu dạng tín hiệu được lấy từ cảm biến và được kết nối với bộ xử lý STM32W108 và bộ thu phát RF. Bộ thu phát này sẽ truyền dữ liệu dạng số tới gateway gồm STM32W108 và Raspberry Pi, dữ liệu sau đó được gửi lên mạng.



Hình 1.5 Hệ thống giám sát nước thông minh

- A novel smart water - meter based on IoT and smartphone app for city distribution management [6]:

Hệ thống gắn cảm biến từ vào trong đồng hồ nước để đọc thông tin. Hệ thống bao gồm thành phần phần cứng gọi là modul giao diện điện tử (EIM) nằm cùng với đồng hồ nước. EIM được thiết kế xung quanh bo mạch CPU, đồng hồ thời gian thực, cổng để giao tiếp với mạng TCP/IP hoặc Wi-Fi hoặc Bộ định tuyến GSM/3G/4G, pin tích hợp và bộ sạc.



Hình 1.6 Hệ thống thông minh có EIM bên ngoài

- Architectural framework of smart water meter reading system in IoT environment [7]:

Hệ thống đồng hồ nước thông minh được chế tạo sẵn dạng kỹ thuật số, đọc dữ liệu nước bằng một cảm biến dòng chảy siêu âm và một cặp cảm biến nhiệt độ PT1000, sau đó gửi dữ liệu lên mạng.



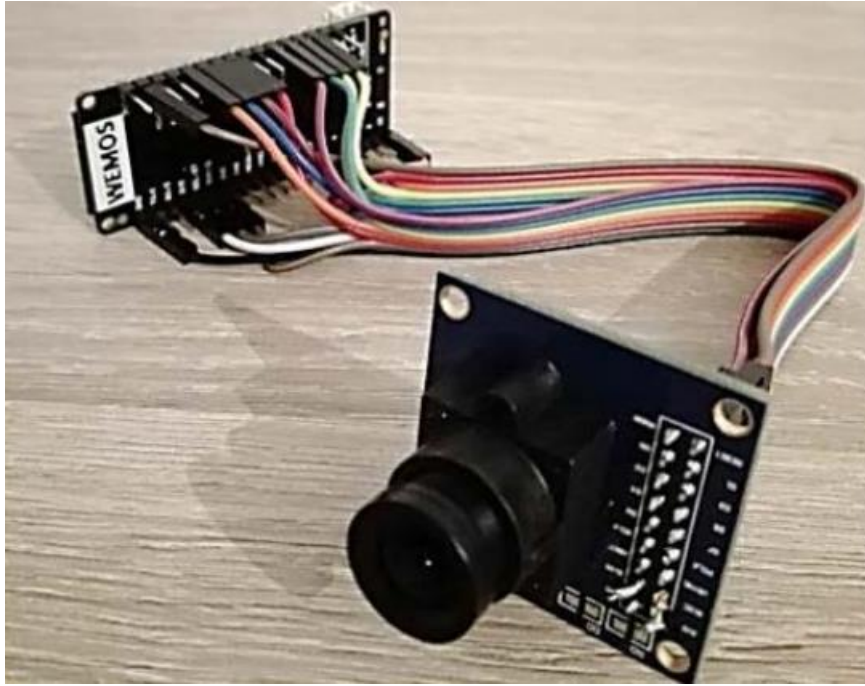
Hình 1.7 Đồng hồ nước STUF-280T sử dụng công nghệ đo lượng nước siêu âm

- Evaluation of Recognition of Water - meter Digits with Application Programs, APIs, and Machine Learning Algorithms [8]:

Đề tài không đề cập đến phần cứng sử dụng mà chỉ nghiên cứu một số thuật toán máy học (KNN, SVM, CNN) được sử dụng cho các chương trình/API mà đề tài đề cập đến là Anyline, Line, Google Vision, Microsoft Azure Computer Vision và Naver. Tập dataset được thu thập bằng camera điện thoại nên khó áp dụng cho các thiết bị ở tầng vật lý của mô hình OSI do không tương thích về camera, bộ nhớ,...

- OCR - based Solution for The Integration of Legacy And - Or Non ElectricCounters in Cloud Smart Grids [9]:

Ảnh được chụp từ vi điều khiển, nhưng không được xử lý tại vi điều khiển mà được truyền lên mạng và xử lý nhận diện ký tự bằng Google Cloud Vision API (Application Programming Interface).



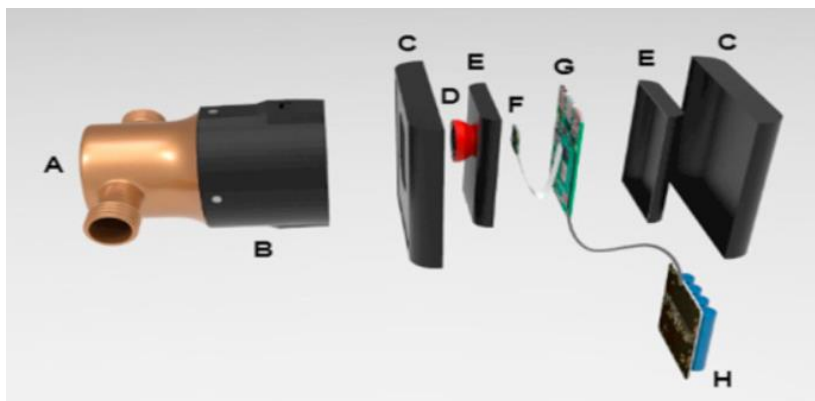
Hình 1.8 Hệ thống bao gồm modul WROOM32 TM và ArduCam OV 7670

- Application of Deep Residual Neural Network to Water Meter Reading Recognition [10]:

Đề tài sử dụng mô hình học sâu R - FCN để phát hiện vùng số và nhận diện chữ số trên phần cứng Nvidia GTX 1070Ti GPU (graphics processing unit).

- A fully AI - based system to automate water meter data collection in Morocco Country [11]:

Đề tài sử dụng phần cứng gồm camera kết nối với máy tính nhúng Raspberry Pi. Thiết bị sử dụng xử lý ảnh để phát hiện vùng số và mô hình học sâu để nhận diện chỉ số nước.

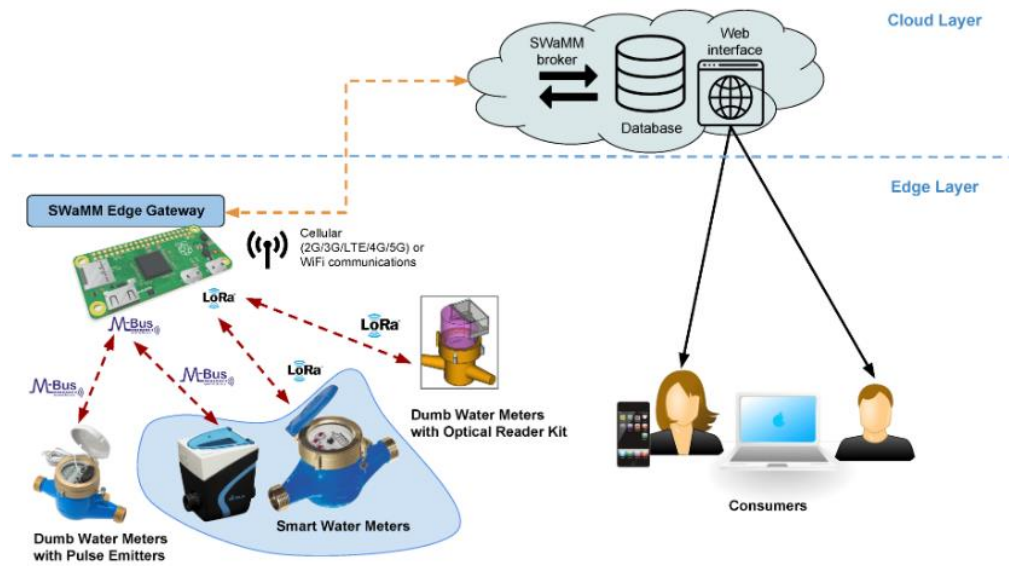


Hình 1.9 Hệ thống nhận dạng dựa trên Raspberry Pi

- Wireless Middleware Solutions for Smart Water Metering [12]:



Hệ thống gồm camera + module truyền thông LoRa gắn ở đồng hồ nước. Ảnh chụp được gửi tới LoRa gateway (sử dụng máy tính nhúng Raspberry Pi) và xử lý nhận diện kí tự số tại LoRa gateway, sau đó mới được truyền lên mạng.



Hình 1.10 Tổng thể về giải pháp SWaMM

- Integrating my Neptune Water Meter with HomeAssistant [13]:

Đề tài sử dụng phần cứng gồm một camera VC0706 (độ phân giải ảnh chụp tối đa 640 x 480, giá thành 27\$ ~ 600 000 VNĐ) và module Wemos D1 Mini lite (giá thành 2.95\$ ~ 60 000 VNĐ). Đề tài gắn thêm nhiều đèn led mới xử lý được chữ số khiến hệ thống trở nên công kênh. Đề tài không đề cập thiết kế phần cứng tối ưu về năng lượng.

- A Smart Water Metering System Based on Image Recognition and Narrowband Internet of Things [14]:

Đề tài sử dụng phần cứng gồm một IC STM32F103ZET6 (giá thành 2.51\$ ~ 50 000 VNĐ) và camera OV7725 (độ phân giải ảnh chụp tối đa 640 x 480, giá thành 9\$ ~ 180 000 VNĐ). Thiết bị sử dụng một model máy học đã được công bố để chạy dự đoán trên IC STM32F103. Đề tài không đề cập thiết kế tối ưu về năng lượng.

- Automatic reading of domestic electric meter: an intelligent device based on image processing and ZigBee/Ethernet communication [15]:

Đề tài thực hiện phát hiện vùng chữ số và tiền xử lí ảnh rất hiệu quả trên phần cứng cấu hình thấp, năng lượng tiêu thụ thấp với dòng ở chế độ ngủ là 0.9 uA và khi hoạt động có dòng cao nhất là 1.35A. Tuy nhiên bước nhận diện chữ số lại được thực hiện trên một mô hình máy học chạy trên máy chủ (PC server).

Đối với tình hình nghiên cứu và sản xuất thiết bị ở nước ngoài: Ưu điểm chung là có độ chính xác rất cao. Tuy nhiên nhược điểm đáng đề cập đến như: thay thế toàn bộ đồng

hồ nếu sử dụng đồng hồ dạng kỹ thuật số hoặc can thiệp vào cấu trúc bên trong đồng hồ, cần kết nối Internet, yêu cầu phần cứng cấu hình cao, không tối ưu về chi phí và năng lượng.

## 1.2 Đề xuất giải pháp

Xu thế AI at the Edge giải quyết được rất nhiều những vấn đề tồn đọng so với việc triển khai ứng dụng trí tuệ nhân tạo trên các máy chủ đám mây như cách truyền thống trước đây:

- Tồn thời gian và chi phí xử lý khi phải chạy các thuật toán trí tuệ nhân tạo trên máy chủ đám mây.
- Độ tin cậy thấp do phụ thuộc vào tốc độ đường truyền Internet và sự khả dụng của đường truyền (khi mất kết nối Internet sẽ không xử lý các thuật toán trí tuệ nhân tạo được).
- Không tiết kiệm năng lượng do phải liên tục duy trì một đường truyền Internet bằng thông ổn định.
- Dữ liệu được gửi lên Internet tiềm ẩn rủi ro về riêng tư do dữ liệu có thể bị rò rỉ từ các lỗ hổng bảo mật trên máy chủ đám mây hay các dịch vụ chạy trên nền tảng đám mây.

Để đón đầu xu thế nhóm đã nghiên cứu chế tạo một thiết bị đọc chỉ số đồng hồ nước thông minh có tích hợp thuật toán xử lý ảnh và mô hình học sâu (Deep Learning) chạy trực tiếp trên thiết bị mà nhóm sử dụng.

Ngày nay, các sản phẩm đồng hồ nước trên thị trường Việt Nam đã có rất nhiều loại, có nhiều mẫu mã mới thuộc loại đồng hồ nước thông minh có tích hợp sẵn các module cảm biến đo lưu lượng nước, về cơ bản thì đây là một loại đồng hồ nước mới với nhiều ưu điểm như về độ chính xác cao, không phải thay đồng hồ nước mỗi khi hết số, các thông số tiêu dùng nước hàng tháng tránh bị sai lệch nhưng nhược điểm của nó lại là giá thành đắt đỏ hơn thế nữa thường nó chỉ được sử dụng nhiều ở các khu công nghiệp lớn, nhà máy và xí nghiệp nhưng vẫn cần tới sức người do các nhân công đọc số điện nước ngoài ra nếu muốn sử dụng còn phải được sự chấp thuận từ các cơ quan chức năng và ban ngành có liên quan nên việc áp dụng nó một cách đầy đủ trên toàn bộ các hộ gia đình là khó có thể triển khai ngay lập tức vậy nên nhóm đã có một ý tưởng về một cơ hội thay đổi cách mà các hộ gia đình đang sử dụng nước và chi trả cho các khoản phí nước tiêu dùng đó là xây dựng hệ thống đồng hồ nước tích hợp mô hình AI để lấy được



các giá trị số trên mặt đồng hồ nước khi đó chúng ta sẽ phần nào giải quyết được về vấn đề ngân sách phải trả cho nhân công và giá thành lại rẻ.

### 1.3 Mục tiêu nghiên cứu

Mục tiêu nghiên cứu của đề tài là phát triển một hệ thống nhúng bao gồm một hệ thống phần cứng sử dụng các module vi xử lý nhằm mục đích chụp hoặc lấy hình từ mặt trên của đồng hồ nước và một hệ thống máy học hoặc học sâu để đưa ra dự đoán, nhận diện các chữ số trên đồng hồ nước từ đó gửi giá trị dự đoán về hạ tầng LoRaWAN chung (mục tiêu lâu dài sẽ là đưa hệ thống sử dụng trên toàn phổ và xa hơn là cả nước). Từ đó sẽ giảm thiểu sự sai sót về mặt đáng tin cậy trong việc thu chi cho khoản tiền nước tiêu dùng của người dân, không những thế việc sử dụng hệ thống tự động có thể giúp giảm tiền trả hàng tháng cho nguồn nhân công đọc số nước mỗi tháng.

### 1.4 Đối tượng và phạm vi nghiên cứu

#### 1.4.1 Đối tượng nghiên cứu

Xử lý ảnh – đọc thông tin ảnh chụp từ Camera, triển khai các thuật toán tiền xử lý hình ảnh chụp từ ESP32-CAM và chuẩn bị một cơ sở dữ liệu cho mô hình học sâu. Ngoài ra còn cần phải tối ưu hoá thuật toán xử lý ảnh cũng như phân bổ quản lý bộ nhớ để đảm bảo hiệu suất và hoạt động ổn định của thiết bị.

Xây dựng mô hình huấn luyện – để nhận diện được chữ số trên mặt đồng hồ nước thì cần phải có một mô hình học máy hoặc học sâu nhằm giúp đưa ra kết quả đánh giá nhận diện cho các ảnh được chụp, thêm vào đó cũng cần phải chú ý tài nguyên bộ nhớ của hệ vi xử lý.

Tìm hiểu các loại thư viện thu nhỏ nhằm giảm thiểu kích thước của mô hình và hỗ trợ chạy trên một hệ vi xử lý độc lập mà không cần kết nối trực tiếp trên máy tính.

Tìm hiểu mạng không dây về các, các giao thức truyền gói tin đường dài và sử dụng ít năng lượng cũng như không gian bộ nhớ, đảm bảo độ riêng tư không bị trùng lặp với các mạng không dây thông thường.

#### 1.4.2 Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài tập trung vào việc áp dụng hệ thống mô hình AI được nhúng trên ESP32-CAM để thực hiện nhận diện chữ số trên đồng hồ nước và quá trình giao tiếp giữa việc truyền gói tin qua mạng không dây.

## **1.5 Các tiếp cận và phương pháp nghiên cứu**

### **1.5.1 Cách tiếp cận**

Sự tăng trưởng của công nghệ 4.0 đã giúp cho công nghệ AI trở nên vượt trội hơn, thông qua đó việc nhận diện số trên mặt đồng hồ nước mà sử dụng trên một hệ vi xử lý nhỏ là rất khó, trước hết là về yếu tố vi xử lý thì đối với một hệ thống nhỏ như module ESP32-CAM sẽ không đủ bộ nhớ để xử lý hoặc nếu có cũng sẽ rất khó khăn và cần nhiều thời gian để có thể đưa ra một đánh giá, nhưng ngày nay với những bước tiến mới về công nghệ AI, các thư viện mới đã hỗ trợ rất nhiều cho việc xây dựng một mô hình AI trên hệ thống vi xử lý và điều này đã giúp nhóm có thể tiếp cận với đề tài.

### **1.5.2 Phương pháp nghiên cứu**

Phương pháp nghiên cứu xuyên suốt đồ án tốt nghiệp này là tìm hiểu các đề tài, bài báo quốc tế tin cậy có nội dung liên quan đến công nghệ không dây Lora, các mô hình mô phỏng, AI on the Edge, các thuật toán mạng nơ ron tích chập và so sánh, đánh giá ưu khuyết điểm của các mô hình, thuật toán,... Từ đó thiết kế, thực hiện áp dụng vào thực tế.

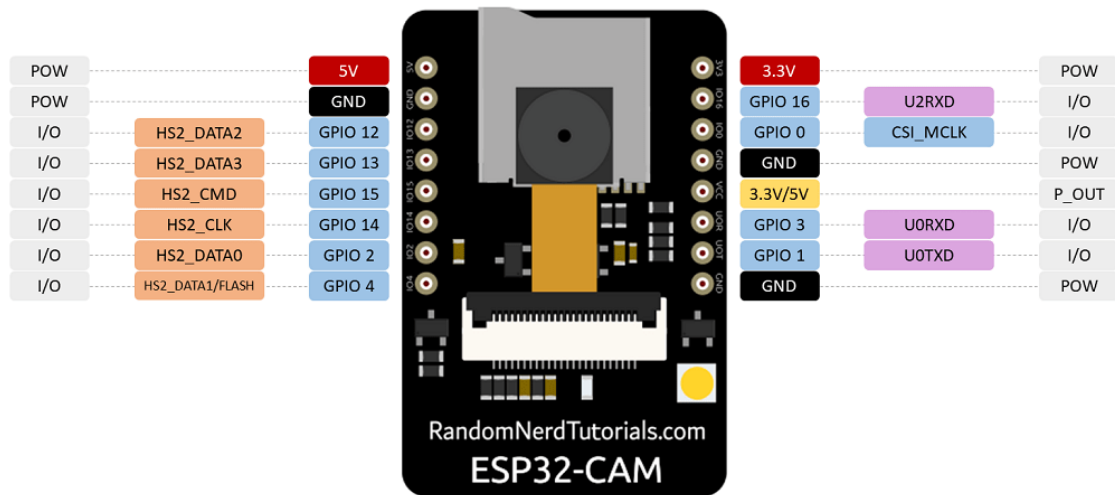
## CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

### 2.1 Thiết bị nhúng ESP32-CAM

ESP32-CAM là một module dựa trên SoC ESP32 mạnh mẽ, được thiết kế để có thể gắn Module Camera để thực hiện các dự án với thu thập hình ảnh, stream video chất lượng trung bình, phù hợp cho một số dự án IoT về giám sát và theo dõi. ESP32-CAM có một camera kích thước nhỏ, rất cạnh tranh trong ngành, giống như module chính, module này có thể được xử lý công việc độc lập, module có kích thước nhỏ gọn chỉ 40 x 27 x 12 mm, dòng nghỉ chỉ 6mA.

Thông số cơ bản của Esp-32 [16]:

- IC chính: ESP32-S (AI-Thinker)
- Mô-đun Wi-Fi BT SoC 802.11 b/g/n/e/i
- CPU 32-bit công suất thấp, cũng có thể phục vụ bộ xử lý ứng dụng
- Tốc độ đồng hồ lên đến 160MHz, sức mạnh tính toán lên đến 600 DMIPS
- Tích hợp 520 KB SRAM, 4MPSRAM bên ngoài
- Dải tần số: 1421 ~ 2484 Mhz
- Bluetooth: 4.2 BR/EDR BLE
- Hỗ trợ UART / SPI / I2C / PWM / ADC / DAC
- Hỗ trợ máy ảnh OV2640 và OV7670, đèn flash tích hợp
- Hỗ trợ tải lên WiFi hình ảnh
- Hỗ trợ thẻ TF
- Hỗ trợ nhiều chế độ ngủ
- Nhúng Lwip và FreeRTOS
- Hỗ trợ chế độ hoạt động STA / AP / STA + AP
- Hỗ trợ cấu hình thông minh / công nghệ AirKiss
- Hỗ trợ nâng cấp cục bộ và từ xa cho công nối tiếp (FOTA)

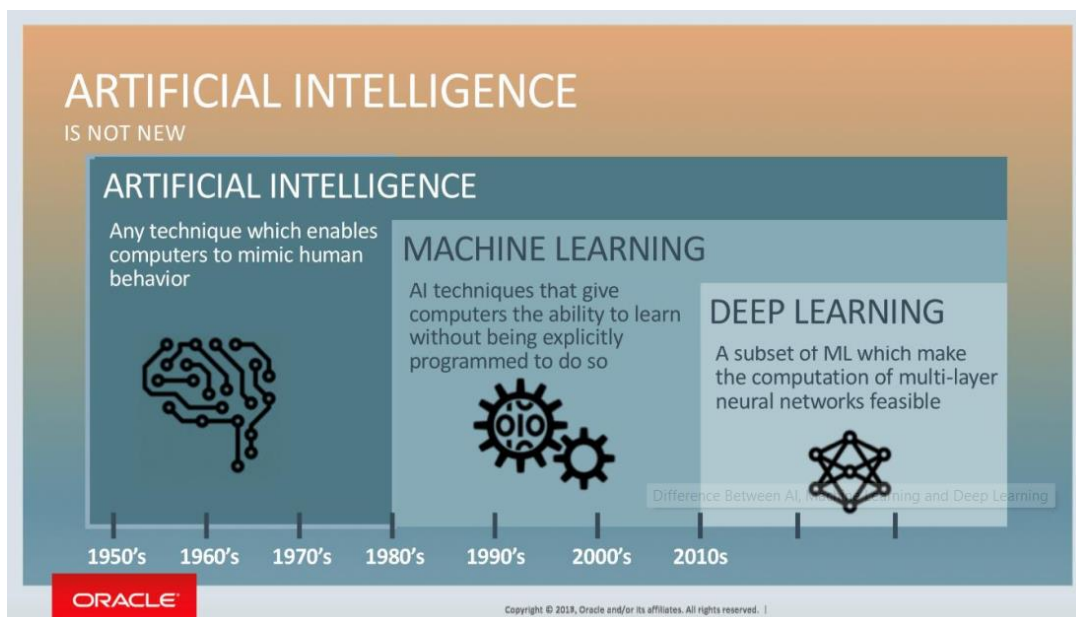


Hình 2.1 Sơ đồ chân ESP32-CAM

## 2.2 Mạng nơ-ron tích chập CNN

### 2.2.1 Giới thiệu về máy học

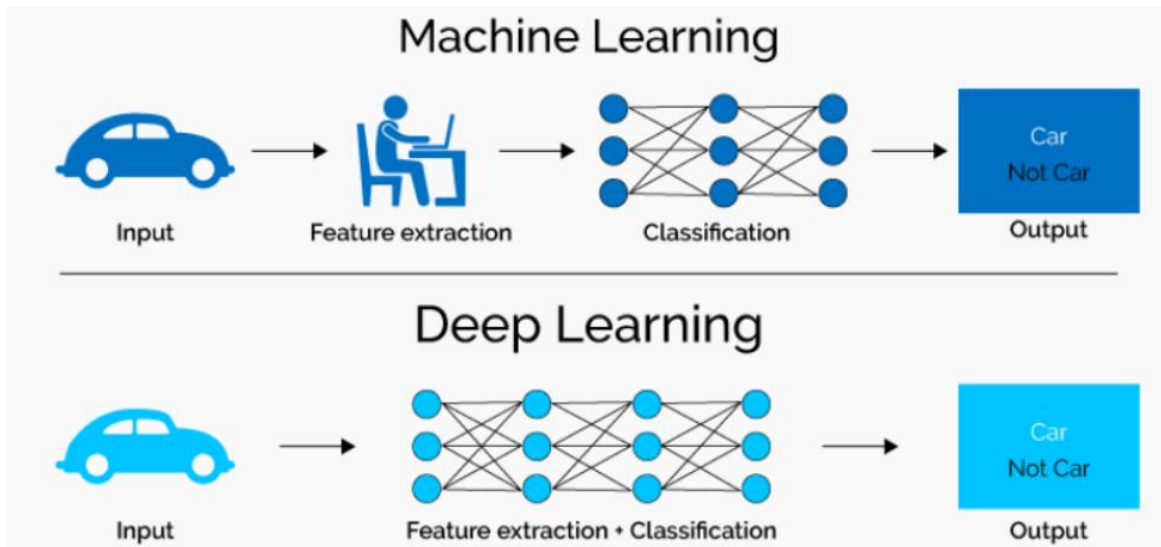
Trí tuệ nhân tạo là một khái niệm rộng, bên trong nó bao hàm máy học và học sâu.



Hình 2.2 Mối quan hệ giữa trí tuệ nhân tạo, máy học và học sâu [17]

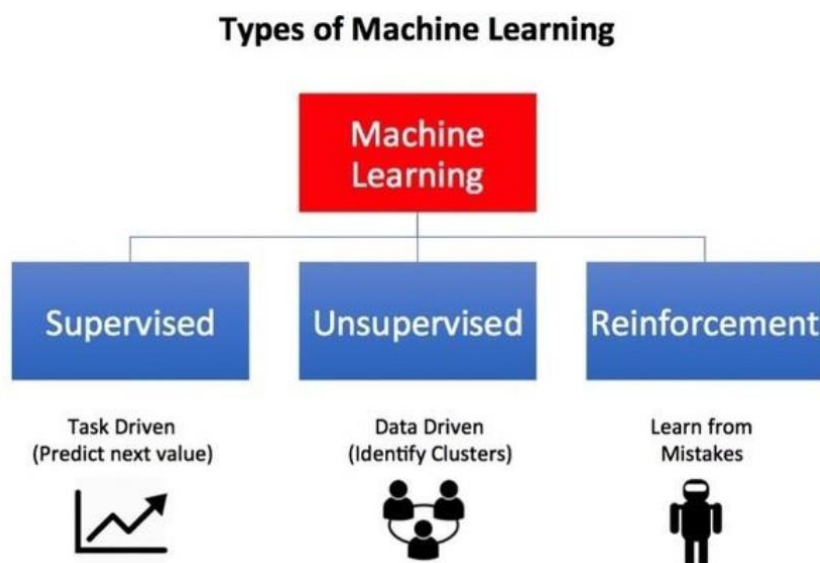
**Trí tuệ nhân tạo:** Là tất cả những kỹ thuật giúp cho máy tính có thể bắt chước được hành vi của con người.

- **Máy học:** Cho máy tính có khả năng học dựa vào dữ liệu cho trước mà không cần phải lập trình. Máy tính sẽ tự động thực hiện một nhiệm vụ trên dữ liệu mới thông qua các giải thuật đã được học.
- **Học sâu:** Là một nhánh của máy học, trong đó nó chứa mạng thần kinh nhân tạo và có khả năng tự động trích xuất ra đặc trưng của một nhóm dữ liệu như văn bản, âm thanh, hình ảnh,...



Hình 2.3 So sánh máy học và học sâu [18]

Trong máy học thì có ba nhánh thường gặp nhất là Supervised, Unsupervised và Reinforcement.

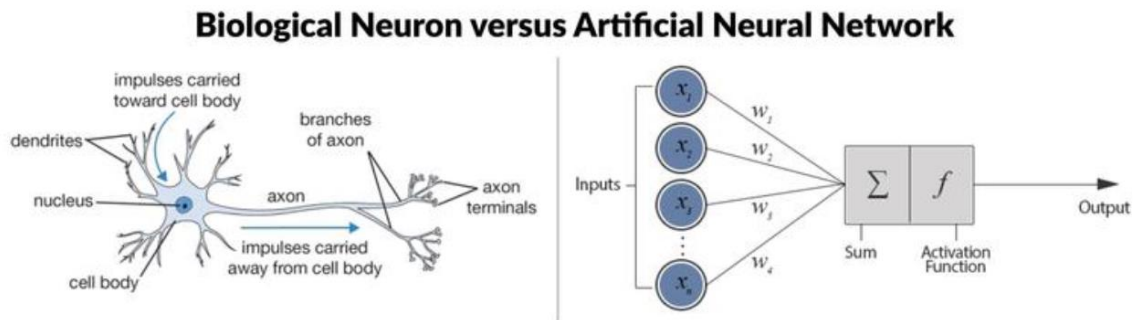


Hình 2.4 Các nhánh máy học thường gặp [19]

- **Unsupervised (học không giám sát):** Là xây dựng mô hình tự động phân cụm dữ liệu đầu vào (input) chưa được gán nhãn thành các nhóm nhỏ với đặc trưng riêng thông qua việc xem xét các điểm tương đồng giữa chúng [20].
- **Reinforcement (học tăng cường):** Là dạy cho máy tính thực hiện tốt một công việc thông qua sự tương tác với môi trường xung quanh và nhận được phần thưởng cho lần thực hiện công việc đúng.
- **Supervised (học có giám sát):** Là xây dựng mô hình dự đoán dựa vào cặp dữ liệu đầu vào và đầu ra (output) đã biết từ trước, dữ liệu input được gán nhãn trước khi huấn luyện. Mô hình ở đây không có nghĩa là thuật toán mà một mô hình sẽ sử dụng một hoặc nhiều thuật toán để tìm ra bộ tham số phù hợp cho mô hình từ đó tối ưu kết quả dự đoán [21]. Trong Supervised được chia thành nhiều nhánh nhỏ [22], trong đó phổ biến nhất có thể chia thành hai bài toán sau:
  - **Regression:** Đối với mỗi dữ liệu đầu vào sẽ đi qua một hàm xử lý cho ra output dự đoán. Input và output của bài toán là những giá trị liên tục ở dạng số thực và giá trị output không phải luôn nằm trong danh sách các giá trị cho trước. Một số bài toán có thể kể đến như dự đoán giá nhà, giá vàng, kích thước của mạch tuần tự,...
  - **Classification:** Là bài toán xác định nhãn của một đối tượng như văn bản, hình ảnh,... Classification được chia thành hai loại là Binary Classification và Multiclass Classification. Binary Classification dùng để chỉ những bài toán trong đó chỉ phân biệt hai lớp dữ liệu, ví dụ như phân biệt đúng và sai, mạch tổ hợp hoặc không phải mạch tổ hợp,... Trong Multiclass Classification thì có số phân lớp nhiều hơn hai và output là kết quả dự đoán cho biết input thuộc một trong các nhãn nào được cung cấp từ trước. Đề tài của nhóm thuộc phạm vi nhánh Multiclass Classification.

Mạng học sâu được chia thành hai loại là học sâu có giám sát (Supervised Deep Learning) và học sâu không có giám sát. Trong đó học sâu có giám sát gồm ba loại là mạng thần kinh tích chập (Convolutional Neural Network – CNN), mạng thần kinh nhân tạo (Artificial Neural Network – ANN) và mạng thần kinh hồi quy (Recurrent neural network – RNN) [23].

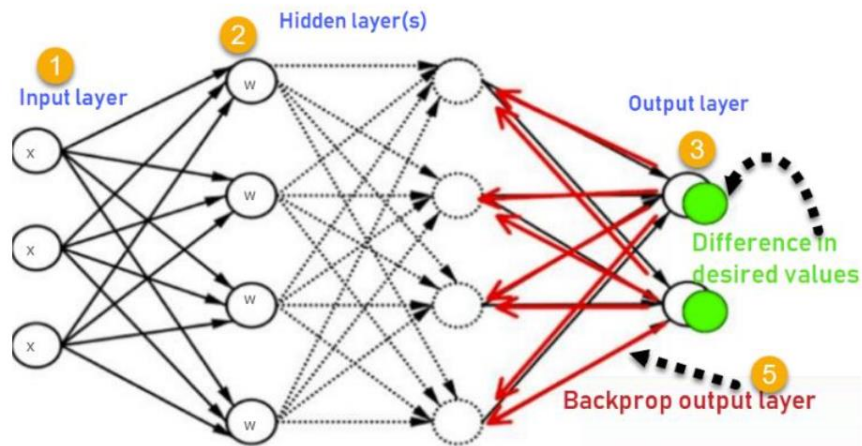
Mạng thần kinh nhân tạo là một hệ thống học tập có giám sát được xây dựng từ một số lượng lớn các phần tử đơn giản, được gọi là neuron hoặc perceptron. Mỗi tế bào thần kinh có thể đưa ra các quyết định đơn giản và cung cấp các quyết định đó cho các tế bào thần kinh khác, được tổ chức thành các lớp liên kết với nhau [24].



Hình 2.5 So sánh mạng thần kinh tự nhiên và nhân tạo

Trong mạng thần kinh nhân tạo, hàm kích hoạt của một nút xác định đầu ra của nút đó với một đầu vào hoặc tập hợp các đầu vào. Một IC (vi mạch số) có thể được coi là một mạng lưới kết nối các hàm kích hoạt lại với nhau để có thể “BẬT” (1) hoặc “TẮT” (0), tùy thuộc vào đầu vào. Điều này tương tự như hoạt động của perceptron tuyến tính trong mạng thần kinh. Nếu không áp dụng hàm kích hoạt thì tín hiệu đầu ra sẽ chỉ là một hàm tuyến tính đơn giản. Hàm tuyến tính chỉ là một đa thức một bậc. Hiện nay có các loại hàm kích hoạt được dùng phổ biến như: ReLu, SoftMax...

Trong quá trình lan truyền ngược (back propagation), các thông số của mô hình (weights) liên quan đến hàm mất mát (loss) được cập nhật. Mục tiêu của 1 mô hình học sâu là giảm thiểu giá trị độ loss của mô hình. Thông qua việc tính gradient của hàm loss từ các thông số của mô hình sẽ từng bước giảm thiểu gradient này. Khi gradient được giảm sẽ đồng thời cập nhật lại các thông số của mô hình, quá trình này được gọi là lan truyền ngược.



Hình 2.6 Quá trình lan truyền ngược

Một mạng thần kinh tích chập trong học sâu thường chứa bốn lớp sau đối với dạng dữ liệu ảnh:

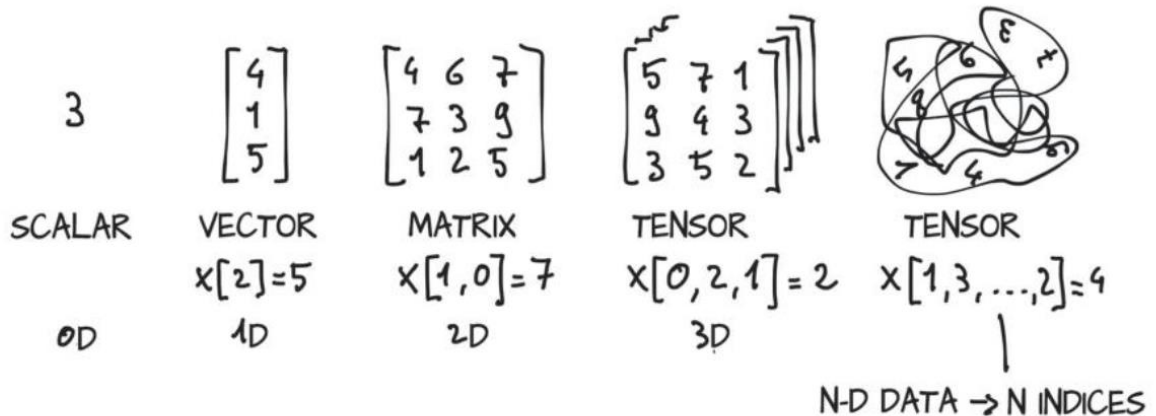
- **Convolution layer:** Tích chập ảnh đầu vào với một kernel/filter để cho ra một ảnh ở dạng feature map.
- **Pooling layer:** Giảm kích thước của feature map, giúp giảm số phép tính trong một mạng học sâu nhưng vẫn giữ lại được những đặc trưng quan trọng.
- **Fully connected layer:** Kết hợp các feature map lại với nhau để cho ra output của bài toán.
- **Activation layer:** Chứa hàm kích hoạt giúp khử nhiễu cho các feature map bằng việc chọn lọc chỉ lấy các giá trị feature cụ thể. Điểm khác biệt lớn nhất giữa mạng ANN và mạng CNN đó là ở mạng CNN thì dữ liệu đầu vào được trải qua quá trình tự động trích xuất đặc trưng thông qua lớp Convolution, rồi sau đó mới đến quá trình tính toán phía sau gần tương tự như ANN.

### 2.2.2 Khởi tạo dữ liệu huấn luyện trong máy học

Trong lập trình thì có các loại dữ liệu như mảng một chiều, mảng hai chiều... Dù mảng là một chiều hay nhiều chiều thì bản chất khi lưu vào bộ nhớ, nó vẫn là một vùng nhớ gồm những ô nhớ liên tiếp nhau. Mảng một chiều là một tập hợp các giá trị trong một danh sách, do đó có thể xem mảng một chiều như một vector có  $Y$  phần tử. Một mảng hai chiều có thể được coi như một ma trận  $A$  có kích thước  $Y$  (dòng)  $\times$   $X$  (cột). Nếu xem mỗi cột  $X$  của ma trận  $A$  là một vector có  $Y$  dòng thì  $A$  lúc này gồm  $X$  vector



có Y chiều ghép lại. Trong lĩnh vực máy học thì A được gọi là một tensor. Tensor một chiều là một vector, tensor hai chiều là một ma trận, tensor ba chiều là một khối lập phương (cube)....



Hình 2.7 Minh họa các dạng dữ liệu [25]

Các bộ dữ liệu dùng cho học sâu thường có kích thước rất lớn. Trong quá trình huấn luyện các mô hình học sâu thì không thể truyền toàn bộ dữ liệu vào mô hình cùng một lúc bởi dữ liệu thường có kích thước lớn hơn RAM máy tính. Xuất phát từ lý do này, các bộ thư viện (framework) học sâu đều hỗ trợ các hàm huấn luyện mô hình với dữ liệu được xử lý theo dạng generator. Dữ liệu sẽ không được khởi tạo ngay toàn bộ từ đầu mà sẽ huấn luyện đến đâu sẽ được khởi tạo đến đó theo từng phần nhỏ gọi là batch. Kích cỡ của mỗi batch là một siêu tham số (hyperparameter) dùng để tối ưu quá trình huấn luyện một hệ thống học sâu. Tùy theo định dạng dữ liệu là văn bản, hình ảnh, mảng,... mà những bộ tạo dữ liệu huấn luyện khác nhau sẽ được sử dụng. Có hai hình thức khởi tạo dữ liệu trong TensorFlow như sau:

- **In memory Dataset:** Khởi tạo các dataset ngay từ đầu và dữ liệu được lưu trữ trên memory. Tốc độ huấn luyện nhanh nhưng dễ xảy ra tình trạng tràn bộ nhớ (out of memory).
- **Generator Dataset:** Dữ liệu được sinh ra theo từng batch nhỏ và xen kẽ với quá trình huấn luyện từ các hàm khởi tạo generator. Tốc độ chậm hơn nhưng có thể tiền xử lý dữ liệu trước khi đưa dữ liệu vào huấn luyện. Đây là phương pháp nhóm lựa chọn sử dụng do khả năng tiền xử lý dữ liệu trước khi đưa vào huấn luyện.

### 2.2.3 Đánh giá chất lượng một mô hình máy học

Các thông số quan trọng dùng để đánh giá chất lượng một mô hình máy học [26]:

- Độ chính xác:

Độ chính xác là một trong những thông số dùng để đánh giá chất lượng một mô hình phân lớp máy học. Công thức tính độ chính xác:

$$\text{Độ chính xác} = \frac{\text{số lần dự đoán đúng}}{\text{Tổng số lần dự đoán đúng}} \quad (2.1)$$

Độ chính xác được đánh giá bằng tỉ lệ giữa số lần dự đoán đúng trên tổng số lần chạy dự đoán. Ví dụ cho một mô hình phân lớp có 10 nhãn (10 lớp), chạy dự đoán 10 ảnh dữ liệu đầu vào thì mô hình dự đoán đúng được 9 trên 10 ảnh, nghĩa là độ chính xác bằng  $9/10 = 90\%$ . Cách đánh giá này có nhược điểm là chỉ cho biết bao nhiêu phần trăm dữ liệu đầu vào được dự đoán đúng mà không chỉ ra được cụ thể mỗi dữ liệu đó được mô hình dự đoán và phân loại như thế nào, nhãn nào được phân loại đúng nhiều nhất, và dữ liệu thuộc nhãn nào thường bị phân loại nhầm vào nhãn khác,.. Các giá trị này có thể được đánh giá bằng việc sử dụng một ma trận được gọi là ma trận lỗi (confusion matrix).

- Ma trận lỗi:

Ma trận lỗi là một bảng dùng để đánh giá mức độ dự đoán thành công của một mô hình phân lớp máy học. Một trục của bảng hiển thị giá trị nhãn thực tế và trục còn lại hiển thị giá trị nhãn mà mô hình dự đoán được. Xét ví dụ sau:

Dự đoán    Thực tế	Mạch tổ hợp	Mạch tuần tự
Mạch tổ hợp	18	6
Mạch tuần tự	1	452

Bảng 2.1 Ví dụ về ma trận lỗi

Theo như bảng ma trận lỗi trên thì dữ liệu đầu vào thực tế có tổng cộng  $18 + 1 = 19$  schematic là thiết kế của mạch tổ hợp, nhưng mô hình phân lớp lại dự đoán chỉ có 18 schematic là mạch tổ hợp và dự đoán nhầm 1 schematic thành mạch tuần tự. Tương tự,

có tất cả  $452 + 6 = 458$  mạch tuần tự nhưng mô hình lại dự đoán chỉ có 452 schematic mạch tuần tự, còn lại nhầm sang mạch tổ hợp.

Đối với các mô hình phân lớp cho nhiều lớp khác nhau như mô hình nhận diện các chữ số từ 0 đến 9 thì bảng ma trận lỗi này có thể giúp nhận ra được xu thế dự đoán sai của mô hình, chẳng hạn như có bao nhiêu số 6 thường được dự đoán sai thành số 9.

- True/False - Positive/Negative

Xét ví dụ sau:

Dự đoán    Thực tế	Mèo (Positive)	Cá (Negative)
Mèo (Positive)	4	6
Cá (Negative)	1	2

Bảng 2.2 Ví dụ về True/False Positive/Negative

Trong những bài toán phân lớp có hai lớp dữ liệu (Binary Classification) thì lớp dữ liệu quan trọng hơn thường định nghĩa là lớp Positive (P - dương tính), lớp còn lại được gọi là Negative (N - âm tính). False Positive Rate còn được gọi là False Alarm Rate (tỉ lệ báo động nhầm) và False Negative Rate được gọi là Miss Detection Rate (tỉ lệ bỏ sót). Ví dụ trong bài toán dò mìn thì có thể chấp nhận việc báo động nhầm hơn là bỏ sót, tức là có thể chấp nhận False Alarm Rate cao để đạt được Miss Detection Rate thấp. Do bài toán mà đề tài của nhóm giải quyết không thuộc Binary Classification nên không cần quan tâm hoàn toàn đến ý nghĩa của khái niệm này trong các nội dung sẽ được trình bày sau đây. Khái niệm này được trình bày với mục tiêu chỉ để bổ sung cho ý nghĩa của thông số Precision/Recall/F1 - score ở các mục sau.

Nhìn vào ma trận ở bảng 2.2 trên, giả sử đặt nhãn Mèo là Positive, nhãn Cá là Negative, thì lúc này True Positive (TP) nghĩa là khi mô hình dự đoán chính xác có 4 ảnh mèo trong tổng cộng  $4 + 1 = 5$  ảnh mèo thực tế. Tương tự True Negative (TN) nghĩa là khi mô hình dự đoán chính xác có 2 ảnh cá trong tổng cộng  $6 + 2 = 8$  ảnh cá thực tế. Các trường hợp sai còn lại là dự đoán nhầm 6 ảnh cá thành mèo (False Positive - FP) và dự đoán nhầm 1 ảnh mèo thành ảnh cá (False Negative - FN). Positive/Negative ở đây được hiểu như là tên gọi của các nhãn và True hay False là tùy vào mô hình dự đoán

đúng hay sai. Từ đây có công thức tính độ chính xác dựa vào True/False Positive/Negative, công thức này cho ra giá trị giống như công thức 2.1 nhưng được tách thành bốn đại lượng:

$$\text{Độ chính xác} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.2)$$

Nếu như giá trị của bốn đại lượng TP, TN, FP, FN gần bằng nhau thì sẽ không thấy được vấn đề xảy ra. Giả sử số lượng mẫu test được thay đổi như sau:

Dự đoán    Thực tế	Mạch tổ hợp	Mạch tuần tự
Mạch tổ hợp	1	1
Mạch tuần tự	8	90

Bảng 2.3 Ví dụ 2 về True/False Positive/Negative

$$\text{Độ chính xác} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{1 + 90}{1 + 90 + 1 + 8} = 91\%$$

Con số 91% có thật sự cho thấy rằng mô hình dự đoán được rất chính xác giữa mèo và gà không? Ví dụ có tổng số ảnh test là  $1 + 8 + 1 + 90 = 100$  ảnh (gồm cả ảnh mèo và cá), trong đó thực tế có tổng cộng 9 ảnh mèo và 91 ảnh cá. Trong số 91 ảnh cá để test thì mô hình dự đoán được đúng 90/91 ảnh, đây là con số ấn tượng. Tuy nhiên trong số 9 ảnh mèo thì mô hình chỉ dự đoán đúng được 1/9 ảnh mèo. Điều này cho thấy con số độ chính xác 91% là không hợp lý vì thực tế mô hình dự đoán sai rất nhiều cho ảnh mèo. Giả sử mô hình không dự đoán đúng được ảnh mèo nào, thì độ chính xác bằng công thức trên vẫn là  $\frac{0+90}{0+90+1+9} = 90\%$ . Sự không hợp lý này được giải quyết thông qua khái niệm Precision và Recall sẽ được trình bày ở mục sau.

- Precision, Recall và F1 – score:

Precision, Recall và F1 - score là ba đại lượng giúp đánh giá chất lượng của mô hình dựa trên chất lượng của từng nhãn riêng.

Precision là tỉ lệ giữa số ảnh dự đoán đúng nhãn chia cho tổng số ảnh được dự đoán thuộc nhãn này.

Recall là tỉ lệ giữa số ảnh dự đoán đúng nhãn chia cho tổng số ảnh thực tế của nhãn đó.

Xét ví dụ sau:

Dự đoán    Thực tế	Mèo	Cá	Gà
Mèo	4	6	3
Cá	1	2	0
Gà	1	2	6

Bảng 2.4 Ví dụ về Precision/Recall/F1 – score

$$\text{Precision của nhãn Mèo} = \frac{4}{4 + 6 + 3} = 30,8\%$$

$$\text{Recall của nhãn Mèo} = \frac{4}{1+1+4} = 66,7\%$$

Công thức tính F1 – score:

$$F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (2.3)$$

Thông qua F1 - score, một nhãn có Precision = 60% và Recall = 100% sẽ cho F1 (được xem giả sử tương ứng với độ chính xác) = 0,75 < F1 của một nhãn có Precision = Recall = 80%.

- Macro/weighted – averaged precision/recall/F1 – score:

Chất lượng của một mô hình phân lớp máy học được đánh giá một cách khách quan hơn bằng việc tính trung bình cộng sáu đại lượng Macro/Weighted-averaged/precision/ recall/ F11 – score trên từng nhãn.

Class	Precision	Recall	F1 - score
-------	-----------	--------	------------

<b>Mèo</b>	30.8%	66.7%	42.1%
<b>Cá</b>	66.7%	20.0%	30.8%
<b>Gà</b>	66.7%	66.7%	66.7%

Bảng 2.5 Ví dụ về Macro/weighted - averaged Precision/Recall/F1 – score

Macro - averaged precision =  $(31\% + 67\% + 67\%) / 3 = 54.7\%$

Macro - averaged recall =  $(67\% + 20\% + 67\%) / 3 = 51.1\%$

Macro - averaged F1 - score =  $(42.1\% + 30.8\% + 66.7\%) / 3 = 46.5\%$

Weighted - averaged precision =  $(6 \times 30.8\% + 10 \times 66.7\% + 9 \times 66.7\%) / 25 = 58.1\%$

Weighted - averaged recall =  $(6 \times 66.7\% + 10 \times 20.0\% + 9 \times 66.7\%) / 25 = 48.0\%$

Weighted - averaged F1 - score =  $(6 \times 42.1\% + 10 \times 30.8\% + 9 \times 66.7\%) / 25 = 46.4\%$

Một mô hình phân lớp tốt là mô hình mà bốn đại lượng trên đều cao (ngoại trừ đại lượng F1 - score), tức càng gần một (100%) càng tốt.

- Cross – validation:

Trong lĩnh vực thống kê hay khai phá dữ liệu thì một trong những công việc chính đó là huấn luyện một mô hình máy học trên tập dữ liệu sẵn có. Một mô hình có thể đưa ra dự đoán khá chính xác trên tập dữ liệu huấn luyện, tuy nhiên lại có thể dự đoán không hoàn toàn chính xác đối với các dữ liệu mới trong tương lai. Cross - validation giúp đánh giá được hiệu năng trung bình của mô hình máy học trong những trường hợp này. Toàn bộ dữ liệu được chia thành hai phần là tập train và tập validation [27]. Cách chia, số lần chia và cách tổng hợp kết quả tùy thuộc vào các dạng cross - validation khác nhau được sử dụng như K - fold cross - validation, Stratified cross - validation...

Cross - validation có thể được sử dụng cho tập dữ liệu cân bằng hoặc mất cân bằng. Đối với các mô hình máy học có số lượng tập dữ liệu huấn luyện mất cân bằng giữa các lớp trong một mô hình phân lớp, để đánh giá mô hình một cách khách quan, thay vì chia tập dữ liệu một lần thành hai tập train và tập validation với lượng dữ liệu cho mỗi lớp trong từng tập được phân chia ngẫu nhiên, cross - validation chia tập dữ liệu qua nhiều lần, với mỗi lần sẽ thực hiện đánh giá độ chính xác riêng trên từng lần chia tập dữ liệu. Giả sử có 100 dữ liệu trải đều cho 10 lớp và đây là tập dữ liệu bị mất cân bằng. Sau khi chia ngẫu nhiên tập dữ liệu này ở lần đầu từ 100 dữ liệu ra tập validation thì sẽ có thể

xảy ra tình huống có lớp chỉ có rất ít dữ liệu (từ 4 đến 5 mẫu), trong khi các lớp khác có tới hơn 10 mẫu. Đánh giá độ chính xác với số lượng mẫu rất ít sẽ dẫn đến không phản ánh chính xác được hiệu năng của mô hình như đã trình bày ở các mục cách đánh giá chất lượng trên. Sử dụng cross - validation giúp đảm bảo có được một thông số độ chính xác của mô hình một cách khách quan và chuẩn xác hơn.

## **2.3 Thư viện TensorFlow Lite cho Vi xử lý**

### **2.3.1 OCR Tesseract**

Nhắc đến việc nhận diện ký tự chữ hoặc số thì hiện nay không thể không kể đến bộ thư viện OCR Tesseract này [28]. Tesseract là một bộ thư viện nhận diện ký tự đồ sộ, với khả năng nhận diện nhiều loại chữ, số ở nhiều hình dạng khác nhau như viết tay, viết in... từ nhiều ngôn ngữ khác nhau trên toàn thế giới. Đề tài của nhóm chỉ thực hiện nhận diện mười chữ số với hình dạng gần như cố định và giống nhau trên các loại đồng hồ nước ở Việt Nam, do đó việc sử dụng Tesseract sẽ gây lãng phí tài nguyên một cách vô ích. Bên cạnh đó do nhiều tính năng nên bộ thư viện này sử dụng các bộ compiler kết hợp nhiều thư viện C++ nên có kích thước lớn, yêu cầu tài nguyên phần cứng cao về ROM và RAM, do đó không phù hợp và rất khó để triển khai trong một thời gian ngắn trên một thiết bị nhúng kích thước nhỏ, tiết kiệm năng lượng với bộ nhớ chỉ vài trăm, vài ngàn KB.

### **2.3.2 OpenCV**

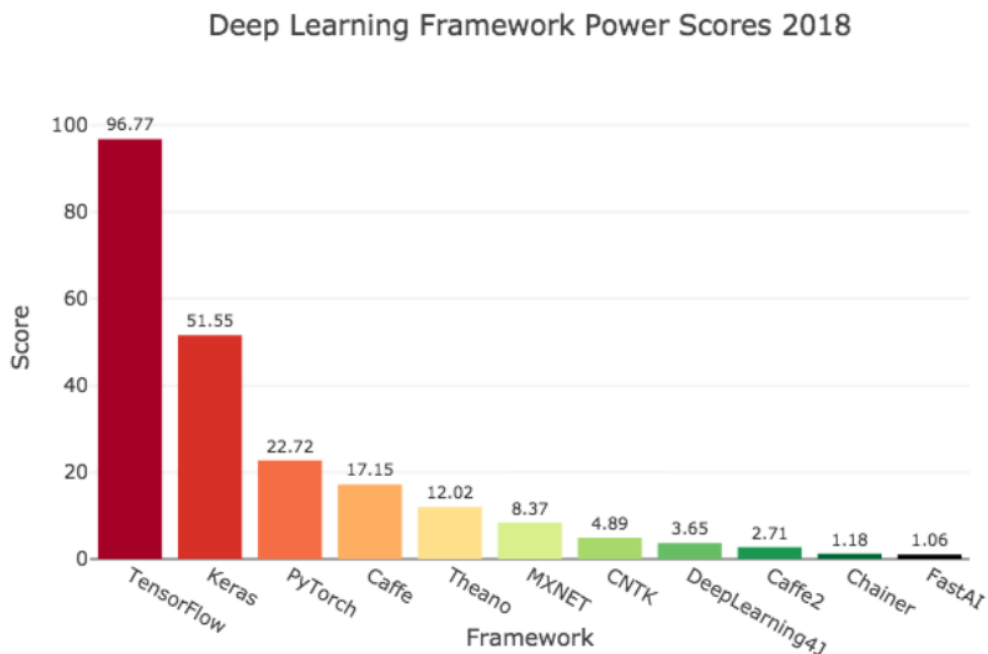
OpenCV hỗ trợ một số công cụ xử lý ảnh và thực hiện một số tác vụ trên ảnh như nhận diện chữ số sử dụng SVM, Template Matching, so khớp đặc trưng SIFT, SURF... [29]. Cũng tương tự như Tesseract, OpenCV là một bộ thư viện có rất nhiều tính năng, kích thước thư viện lớn nhưng sẽ dễ dàng hơn khi xây dựng mô hình huấn luyện thuật toán hoặc xử lý ảnh với ngôn ngữ lập trình Python còn về build thư viện theo ngôn ngữ lập trình C để phù hợp cho việc nạp hay biên dịch thuật toán lên các cấu trúc vi xử lý của ESP32-CAM là khá phức tạp nên không phù hợp để triển khai trong một thời gian ngắn vậy nên lựa chọn thư viện này là không khả thi.

### **2.3.3 TensorFlow Lite**

TensorFlow là một bộ thư viện phần mềm mã nguồn mở miễn phí được phát triển bởi Google dành cho máy học trong nhiều loại hình tác vụ như nhận thức và hiểu ngôn

ngữ. Nó hiện được sử dụng cho cả nghiên cứu lẫn sản xuất thông qua việc hợp tác với nhiều đội ngũ tại Google để ứng dụng vào các sản phẩm như Google dịch, Gmail, Google Photo, Google Search [30]. TensorFlow được phát hành theo giấy phép mã nguồn mở Apache 2.0 vào năm 2015 [31].

Hiện nay lĩnh vực trí tuệ nhân tạo có các framework – là các bộ công cụ hỗ trợ cho lập trình viên, các nhà phát triển ứng dụng dễ dàng tạo ra ứng dụng trí tuệ nhân tạo một cách nhanh nhất. Theo như thống kê ở hình 2.8 thì TensorFlow hoàn toàn vượt trội hơn so với các framework khác về ba tiêu chí mức độ phổ biến, độ ưu thích và hiệu quả sử dụng.



Hình 2.8 So sánh các framework trí tuệ nhân tạo [32]

Tháng 5/2017, Google công bố dự án TensorFlow Lite như là một nhánh con của bộ thư viện TensorFlow. TensorFlow Lite được thiết kế tối ưu hơn cho các ứng dụng trí tuệ nhân tạo triển khai trên các thiết bị nhúng có cấu hình thấp. TensorFlow Lite chủ yếu được thiết kế cho tác vụ Chạy dự đoán, người dùng được khuyến nghị chạy tác vụ Huấn luyện trên các thiết bị cấu hình cao vì một thiết bị nhúng cấu hình thấp không thể thực hiện huấn luyện các bộ dữ liệu lớn lên tới hàng nghìn mẫu.

Tháng 1/2019, Google lần đầu công bố phiên bản thử nghiệm của dự án TensorFlow Lite hỗ trợ chạy dự đoán trên thiết bị Android và iOS.



Tháng 5/2019, Google thông báo rằng dự án TensorFlow Lite dành cho vi điều khiển của họ sẽ được kết hợp với dự án uTensor của hãng thiết kế vi xử lý ARM.

Hiện nay trên thế giới có hai bộ thư viện trí tuệ nhân tạo cho các thiết bị nhúng cấu hình thấp là TensorFlow Lite và STM32Cube.AI. Hai nền tảng này có các điểm khác nhau như sau:



Hình 2.9 So sánh giữa TensorFlow Lite và STM32Cube.AI [33]

Nền tảng STM32Cube.AI ít tốn bộ nhớ flash hơn và thời gian chạy dự đoán thấp hơn, tuy nhiên nhược điểm của STM32Cube.AI là tiêu tốn nhiều Ram hơn và đây là bộ thư viện mã nguồn đóng, chỉ hỗ trợ cho các dòng vi xử lý STM32. Hơn nữa hiện nay các bo mạch nhúng chạy chip xử lý của hãng STM có giá thành cao hơn mặt bằng của dòng bo mạch nhúng ESP và đối với một bộ thư viện mã nguồn đóng thì hoàn toàn có khả năng chi phí triển khai ứng dụng trí tuệ nhân tạo sử dụng STM32Cube.AI sẽ ngày càng gia tăng trong tương lai.

TensorFlow Lite tốn nhiều bộ nhớ flash hơn và thời gian chạy dự đoán lâu hơn. Tuy nhiên nền tảng này có một ưu điểm rõ rệt là tiêu thụ ít Ram hơn và là nền tảng mã nguồn mở, có thể chỉnh sửa để chạy được ở rất nhiều phần cứng khác nhau và cả những phần cứng chi phí thấp như ESP32. Đối với các ứng dụng xử lý trí tuệ nhân tạo không quá phức tạp như bài toán nhận diện chữ số của nhóm, không yêu cầu thời gian xử lý nhanh do chỉ chạy dự đoán một tháng một lần, yêu cầu chi phí thấp thì TensorFlow Lite là một lựa chọn đúng đắn mà đề tài hướng tới.

TensorFlow Lite gồm có hai thành phần chính:

- **Trình thông dịch TensorFlow Lite:** Từ bộ tham số sau khi huấn luyện trên các thiết bị cấu hình cao sẽ được nạp vào những thiết bị nhúng cấu hình thấp và chạy dự đoán trực tiếp trên các thiết bị này mà không cần kết nối đến internet.
- **Bộ chuyển đổi định dạng TensorFlow Lite:** Từ bộ tham số được huấn luyện trên thiết bị cấu hình cao sẽ được bộ chuyển đổi về định dạng mà thiết bị nhúng như vi điều khiển, điện thoại thông minh... có thể hiểu được.

TensorFlow Lite được thiết kế để giúp dễ dàng thực hiện các ứng dụng máy học trên các thiết bị ở tầng vật lý của mô hình mạng OSI (edge - processing), thay vì gửi dữ liệu đến những máy chủ đám mây cấu hình cao để xử lý. Đối với các nhà phát triển, thực hiện ứng dụng máy học trực tiếp ngay trên thiết bị giúp cải thiện:

- **Độ trễ:** Không cần phải tốn thời gian gửi đến máy chủ và chờ máy chủ phản hồi.
- **Quyền riêng tư:** Dữ liệu đầu vào (input) được lưu trữ và xử lý dự đoán ngay trên thiết bị.
- **Tiết kiệm điện năng:** Do không cần phải cố gắng gửi dữ liệu lên Internet hay trang bị phần cứng hỗ trợ kết nối Internet.

Hiện tại TensorFlow Lite chỉ hỗ trợ nền tảng phần cứng Arm Cortex - M và ESP32, với một số board được hỗ trợ chính thức sau:

- Arduino Nano 33 BLE Sense
- SparkFun Edge: sử dụng vi điều khiển Apollo3 Blue tích hợp bộ xử lý ARM Cortex - M4F 48MHz.
- STM32F746 Discovery kit
- Adafruit EdgeBadge: sử dụng vi điều khiển ATSAMD51J19.
- Adafruit TensorFlow Lite for Microcontrollers Kit: sử dụng board PyBadge tích hợp bộ xử lý SAMD51 Cortex M4F.
- Adafruit Circuit Playground Bluefruit: sử dụng board PyBadge/EdgeBadge.
- Espressif ESP32 - DevKitC: Phần cứng này tương tự board ESP32 - Cam mà nhóm sử dụng.

- Espressif ESP – EYE
- Wio Terminal: ATSAMD51
- Himax WE - I Plus EVB Endpoint AI Development Board: sử dụng bộ xử lý WE - I Plus ASIC (HX6537 - A) tích hợp Synopsys ARC EM9D DSP.
- Synopsys DesignWare ARC EM Software Development Platform: sử dụng bộ xử lý DesignWare ARC EM.

Trong số những phần cứng kể trên thì board ESP32 - Cam có giá thành thấp nhất (chỉ 160 000 VNĐ) và nhỏ gọn phù hợp để đặt lên mặt đồng hồ. Hơn nữa board ESP32 - Cam được tích hợp sẵn camera, một số phần cứng kể trên không tích hợp camera. Đó là lí do vì sao nhóm sử dụng ESP32 - Cam cho đề tài này.

## 2.4 Mạng không dây LoRaWAN

Mạng di động (Cellular Networks): Sử dụng mạng di động như 4G LTE, 5G và các tiêu chuẩn truyền thông tương tự, các thiết bị IoT có thể kết nối và truyền dữ liệu từ xa thông qua các mạng di động. Mạng di động cung cấp phạm vi rộng và tốc độ truyền dữ liệu cao, cho phép kết nối IoT ở xa với hạ tầng mạng di động hiện có.

LPWAN (Low-Power Wide Area Network): LPWAN là một loại mạng không dây được thiết kế đặc biệt cho các ứng dụng IoT có yêu cầu thấp về năng lượng và phạm vi truyền dẫn rộng. Các công nghệ LPWAN phổ biến bao gồm LoRaWAN (Long Range Wide Area Network), NB-IoT (Narrowband IoT) và Sigfox. LPWAN cho phép truyền thông tầm xa, thậm chí đến hàng trăm hoặc hàng nghìn mét, mà không yêu cầu năng lượng lớn.

Satellite Communication (Truyền thông vệ tinh): Truyền thông qua vệ tinh cho phép truyền dữ liệu IoT đến các vùng xa hoặc cùng với khả năng liên lạc liên tục. Hệ thống truyền thông vệ tinh cho phép kết nối tầm xa trên toàn cầu, không bị hạn chế bởi phạm vi địa lý hay cơ sở hạ tầng mạng cục bộ. Điều này hữu ích trong các ứng dụng như theo dõi vị trí, nông nghiệp thông minh, và giám sát môi trường.

Wi-Fi: Wi-Fi không chỉ được sử dụng trong phạm vi cục bộ, mà cũng có thể được mở rộng để truyền thông tầm xa hơn. Sử dụng công nghệ điểm truy cập không dây từ xa (Wireless Access Point), có thể mở rộng mạng Wi-Fi để kết nối và truyền dữ liệu giữa các thiết bị IoT từ xa.

Ethernet và VPN (Virtual Private Network): Truyền thông tầm xa cũng có thể được thực hiện thông qua kết nối Ethernet dựa trên cáp mạng. Kết hợp với VPN, các thiết bị IoT có thể kết nối với mạng cục bộ thông qua internet và truyền dữ liệu từ xa một cách bảo mật.

Bluetooth Low Energy (BLE): Công nghệ truyền thông không dây được sử dụng rộng rãi trong lĩnh vực IoT. Mặc dù Bluetooth thường được sử dụng trong các ứng dụng truyền thông cục bộ ngắn, như kết nối thiết bị trong phạm vi gần nhau, nhưng có một phiên bản Bluetooth đặc biệt được phát triển cho các ứng dụng truyền thông tầm xa trong IoT, đó là Bluetooth Low Energy (BLE) hoặc Bluetooth Smart.

<p>Mạng di động (Cellular Networks)</p> <ul style="list-style-type: none"> <li>-Long Range</li> <li>-High Data Rates</li> <li>-Low Battery Life</li> <li>-High Cost</li> </ul>	<p>LoRaWAN (Long Range)</p> <ul style="list-style-type: none"> <li>-Long Range</li> <li>-Low Data Rates</li> <li>-Low Battery Life</li> <li>-Low Cost</li> <li>-High Capacity Potential</li> </ul>	<p>NB-IoT (Narrowband IoT)</p> <ul style="list-style-type: none"> <li>-Stationary Devices</li> <li>-Short Range</li> <li>-Low Data Rates</li> <li>-Good Battery Life</li> <li>-Low Cost</li> </ul>	<p>Cat-M1 (Category M1)</p> <ul style="list-style-type: none"> <li>-Long Range</li> <li>-High Data Rates</li> <li>-Low Battery Life</li> <li>-Medium Cost</li> </ul>
<p>Satellite Communication (Truyền thông vệ tinh)</p> <ul style="list-style-type: none"> <li>-Long Range</li> <li>-High Data Rates</li> <li>-High Cost</li> <li>-Low Battery Life</li> </ul>	<p>Wi-Fi (Local Area Network)</p> <ul style="list-style-type: none"> <li>-Short Range</li> <li>-High Data Rates</li> <li>-Low Battery Life</li> <li>-Medium Cost</li> </ul>	<p>Ethernet và VPN (Virtual Private Network)</p> <ul style="list-style-type: none"> <li>-Short Range</li> <li>-Medium Data Rates</li> <li>-Good Battery Life</li> <li>-Low Cost</li> </ul>	<p>Bluetooth Low Energy (BLE)</p> <ul style="list-style-type: none"> <li>-Very Short Range</li> <li>-Low Data Rates</li> <li>-Good Battery Life</li> <li>-Low Cost</li> </ul>

Bảng 2.6 So sánh LoRa và các công nghệ truyền thông IoT hoặc M2M khác

Trong nhóm các công nghệ truyền thông mạng không dây chính như mạng điện thoại (Cellular Networks 3G, 4G...), mạng Lan cục bộ như sóng Wifi, Ethernet, VPN...,

mạng truyền thông toàn cầu Satellite Communication, mạng điện thoại dùng cho các ứng dụng IoT (Cat - M1), mạng điện thoại băng thông hẹp năng lượng thấp dùng trong IoT (NB - IoT), mạng nội bộ cá nhân như sóng Bluetooth và mạng LPWAN sử dụng công nghệ LoRa. Trong số các mạng truyền thông không dây thì LoRa có ưu điểm đặc biệt vượt trội hơn các mạng khác như khoảng cách truyền xa, năng lượng tiêu thụ thấp, chi phí thấp và có khả năng chịu tải cao. LoRa là giải pháp liên lạc qua sóng vô tuyến tiết kiệm năng lượng ổn định và khoảng cách truyền xa nhất hiện tại. Đối với đề tài của nhóm chỉ truyền đi một gói tin gồm 5 byte dữ liệu với chu kỳ một tháng một lần thì sử dụng mạng LoRa là một giải pháp tối ưu, dù nhược điểm lớn nhất của LoRa là tốc độ truyền tải dữ liệu không cao.

#### **2.4.1 Kỹ thuật điều chế tín hiệu LoRa**

LoRa là một kỹ thuật điều chế tín hiệu được phát triển bởi Semtech. Semtech là một tập đoàn tại Mỹ cung cấp các vi mạch bán dẫn xử lý tín hiệu và các giải pháp về thuật toán nâng cao cho khách hàng cá nhân và doanh nghiệp. LoRa được sử dụng trong các hạ tầng mạng WAN yêu cầu năng lượng thấp (LPWAN). LoRa có các ưu điểm như khoảng cách truyền xa hơn hẳn các giao thức Bluetooth, Wifi nhưng vẫn tiết kiệm năng lượng. Dưới nguồn năng lượng pin, các thiết bị tích hợp LoRa có thể hoạt động lên tới 10 năm hoặc hơn và cho khoảng cách truyền xa lên đến 10 - 15 km ở những khu vực vắng vẻ và vài km ở khu vực thành thị (tùy vào vị trí đặt thiết bị đầu cuối sâu cỡ nào trong các tòa nhà). Khoảng cách truyền xa nhất mà LoRa ghi nhận được là hơn 700 km.

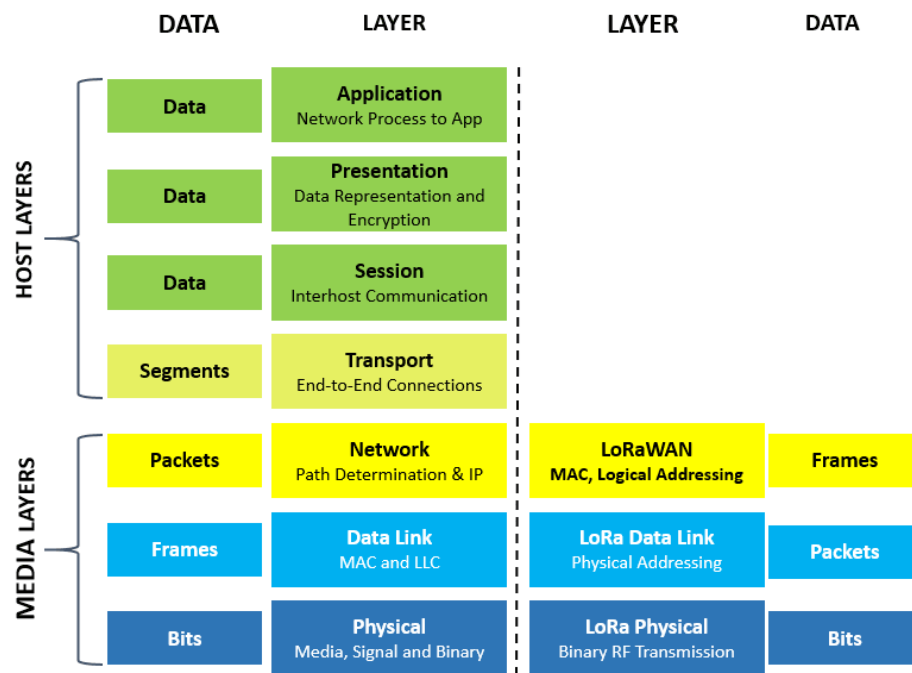
Các thiết bị tích hợp LoRa có thể truyền và nhận dữ liệu qua sóng vô tuyến thông qua kỹ thuật trải phổ sử dụng các xung chirp (CSS) trên các băng tần được quốc gia sở tại cho phép. Ưu điểm của việc sử dụng CSS là nó có khả năng hoạt động trong những môi trường nhiễu nhiều điện từ.

Semtech là nhà cung cấp các vi mạch tích hợp LoRa trong khi tổ chức phi lợi nhuận LoRa Alliance cung cấp giao thức mạng bắt đồng bộ LoRaWAN triển khai trên vi mạch tích hợp LoRa. Sự kết hợp này giúp xây dựng lên các hạ tầng mạng IoT hình sao quy mô lớn nhưng vẫn đảm bảo chi phí thấp và tiết kiệm năng lượng.

LoRa được phát triển từ công nghệ trải phổ CSS, được cải tiến lại với sự đánh đổi giữa độ nhảy của tín hiệu và tốc độ truyền (datarate). LoRa hoạt động ở băng tần 125

KHz hoặc 500 KHz cho kênh tải lên (Uplink) và 500 KHz cho kênh tải xuống (Downlink) [34]. LoRa sử dụng kỹ thuật trải phổ tần số của sóng mang theo trực đứng. Điều này giúp tiết kiệm năng lượng cho các thiết bị đầu cuối kết nối trong mạng bằng cách tối ưu hóa mức năng lượng tiêu thụ khi truyền và tốc độ truyền (datarate). Ví dụ như một thiết bị đầu cuối ở gần LoRa Gateway nên truyền gói tin ở độ trải phổ (SF - Spreading Factor) thấp. Một thiết bị đầu cuối đặt ở xa Gateway nên truyền với SF cao. SF giúp tăng độ nhạy của tín hiệu nhưng bù lại tốc độ truyền sẽ thấp tương ứng thời gian truyền tăng.

Trong mô hình mạng bảy lớp OSI thì LoRa nằm ở tầng vật lý. Một bộ thu phát tín hiệu sẽ truyền sóng vô tuyến được điều chế bằng kỹ thuật điều chế tín hiệu LoRa đến một bộ thu phát tín hiệu LoRa khác được tích hợp bên trong một LoRa Gateway và ngược lại.



Hình 2.10 Vị trí của LoRa và LoRaWAN trong mô hình OSI

### Một số thông số quan trọng trong LoRa:

- **Signal to Noise Ratio (SNR)**

SNR – tỉ lệ tín hiệu trên nhiễu - là tỉ lệ giữa bit rate và chip rate. Đây là tỉ số giữa năng lượng của tín hiệu trên nhiễu tại một thời điểm nhất định. SNR thường được biểu diễn dưới dạng decibel (dB).

- **Spreading Factor (SF)**

SF càng lớn thì càng làm tăng độ nhạy tín hiệu và khoảng cách truyền, nhưng đồng thời cũng làm tăng thời gian truyền. Chỉ số SF càng lớn thì lượng thông tin truyền đi trong một bit càng nhiều. Các giá trị của SF hỗ trợ để lập trình là: 6, 7, 8, 9, 10, 11, 12.

- **Coding Rate (CR)**

CR – tỉ lệ mã hóa - đại diện cho độ an toàn cho dữ liệu truyền đi hay là tỉ lệ đảm bảo của gói tin. Các giá trị có thể dùng trong lập trình là 4/5, 4/6, 4/7 hoặc 4/8. CR càng tăng thì gói tỉ lệ rút gói tin càng thấp, nhưng cũng đồng thời tăng thời gian truyền.

- **RSSI**

RSSI là đặc trưng cho độ nhạy của tín hiệu.

- **Signal Bandwidth (BW)**

BW hay băng thông là độ rộng của tần số trong dãy truyền. BW càng lớn thì tốc độ truyền dữ liệu càng lớn nhưng độ nhạy tín hiệu sẽ bị giảm (do SNR). Với mỗi chỉ số SF, BW càng hẹp thì càng tăng độ nhạy nhận được, tuy nhiên sẽ tăng thời gian truyền trong không khí. Băng thông có thể sử dụng trong khoảng từ 7.8 kHz đến 500 kHz. Các giá trị của băng thông có thể dùng lập trình được: 500 kHz, 250 kHz hoặc 125 kHz.

**Mối quan hệ giữa các thông số trong LoRa:**

- **Thời gian truyền**

$$T_s = \frac{2^{SF}}{BW} \quad (2.1)$$

- T: thời gian truyền trong không gian.
- SF: Spread Factor hệ số lan truyền trong không gian.
- BW: Bandwidth băng thông truyền tin trong không gian.

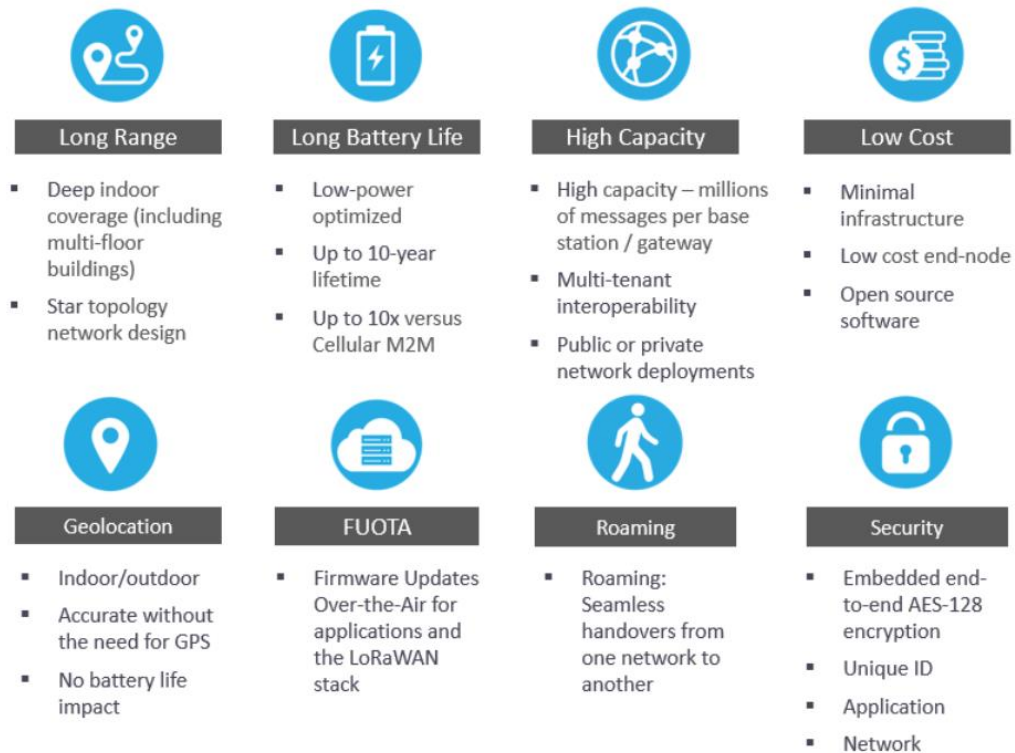
- **Tốc độ truyền bit trong đơn vị thời gian**

$$R_b = SF * \frac{BW}{2^{SF}} * CR \quad (2.2)$$

- R: khối lượng bit trên giây.
- CR: Code Rate chỉ số môi trường số lượng bit kiểm tra.

## **2.4.2 Giao thức LoRaWAN**

LoRaWAN là một giao thức mạng hình sao với ưu điểm cho khoảng cách truyền xa và tiết kiệm năng lượng nhờ vào việc tích hợp kỹ thuật điều chế tín hiệu LoRa.



Hình 2.11 Một số ưu điểm của mạng sử dụng LoRaWAN [35]

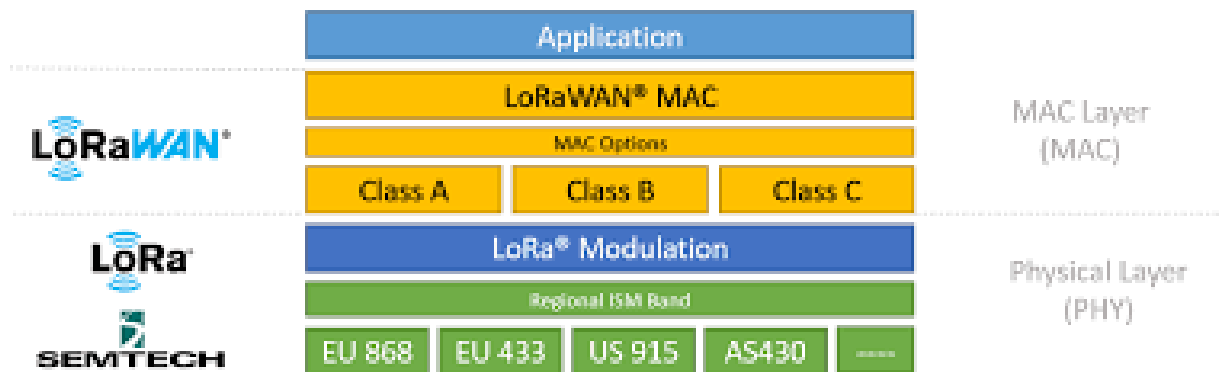
Xét về yếu tố tiết kiệm pin thì năng lượng cần để truyền một gói tin đi rất thấp, với điều kiện rằng kích thước gói tin phải thấp và chỉ truyền vài lần trong ngày. Khi thiết bị đầu cuối không hoạt động, nó có thể đi vào chế độ ngủ với năng lượng tiêu thụ thấp (cỡ mW) giúp cho thiết bị có thể kéo dài thời gian hoạt động liên tục đến nhiều năm.

Một mạng LoRaWAN có thể chịu tải hàng triệu gói tin. Tuy nhiên số lượng gói tin xử lý được phụ thuộc vào số lượng LoRa Gateway lắp đặt trong một khu vực. Một LoRa Gateway tiêu chuẩn hỗ trợ tám kênh truyền có thể xử lý hàng trăm ngàn gói tin trong vòng 24 giờ. Nếu mỗi thiết bị đầu cuối tích hợp LoRa gửi mười gói tin một ngày thì chỉ cần một LoRa Gateway có thể xử lý tới 10 000 thiết bị đầu cuối. Hay nói cách khác một LoRa Gateway có thể thu thập được chỉ số nước từ hơn mười ngàn hộ dân trong vòng bán kính 5 đến 15 km, xét trường hợp xấu nhất là một hộ dân gửi lại dữ liệu 10 lần trong ngày. Nếu một khu vực mạng có mười Gateway thì mạng có thể chịu tải tới một trăm ngàn thiết bị đầu cuối và một triệu gói tin. Cần lưu ý rằng không có một hướng đi cụ thể nào từ một thiết bị đầu cuối đến một Gateway và ngược lại, nghĩa là tất



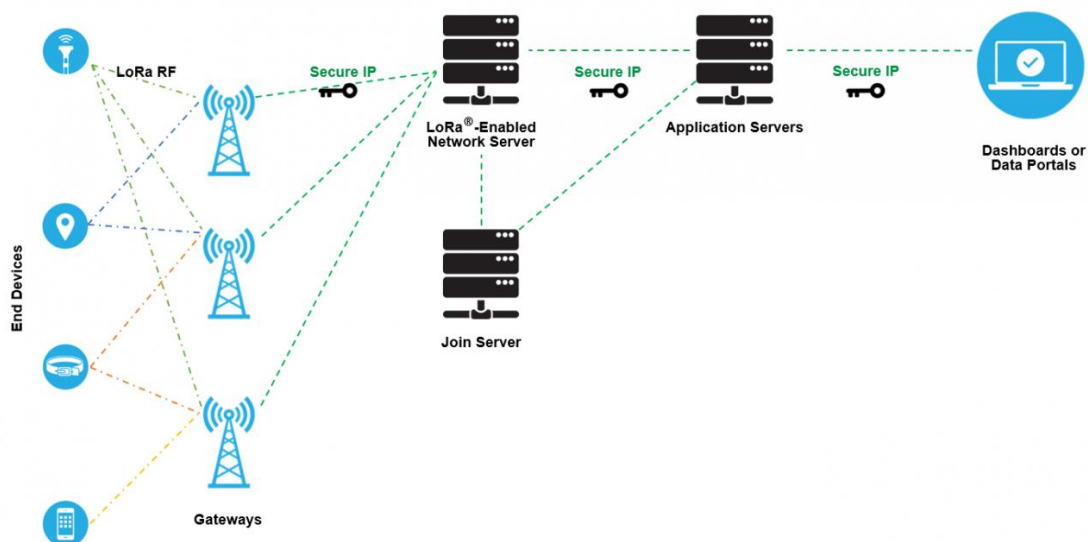
cả những gói tin gửi đến và gửi đi giữa thiết bị đầu cuối và Gateway được bắt đầu gửi hoặc được nhận bởi tất cả Gateway trong khu vực, sự trùng lặp gói tin được giải quyết ở trên máy chủ của LoRa Gateway. Nếu muốn tăng số lượng gói tin hay quy mô xử lý thì chỉ cần tăng số Gateway.

Trong một mạng sử dụng giao thức LoRaWAN thì LoRa nằm ở tầng vật lý với vai trò như một kỹ thuật điều chế tín hiệu không dây tầm xa. LoRaWAN là một giao thức mạng mở, truyền hai chiều có bảo mật, di động (mobility) và có tính nội địa hóa được chuẩn hóa và quản lý bởi tổ chức LoRa Alliance [34].



Hình 2.12 Vị trí của LoRaWAN, LoRa trong mô hình mạng

### 2.1.3. Các thành phần trong mạng LoRaWAN



Hình 2.13 Sơ đồ các thành phần trong mạng LoRaWAN

**End Device (thiết bị đầu cuối):** Là một cảm biến hoặc cơ cấu chấp hành được kết nối không dây với mạng LoRaWAN thông qua các LoRa Gateway (LoRaWAN Gateway) sử dụng kỹ thuật điều chế LoRa.

**LoRa - Enabled Network Server (máy chủ mạng) (LNS):** Máy chủ mạng quản lý toàn bộ mạng, điều khiển một cách tự động các thông số mạng để hệ thống có thể thích nghi với các điều kiện mạng luôn thay đổi. Mã hóa bảo mật theo chuẩn AES 128 bits được sử dụng cho đường truyền dữ liệu từ thiết bị đầu cuối đến tầng ứng dụng người dùng trên máy chủ đám mây, và cho đường truyền qua lại giữa các thiết bị đầu cuối và máy chủ LNS. Hệ thống máy chủ này nhằm đảm bảo tính xác thực của mọi thiết bị đầu cuối và tính toàn vẹn của gói tin. Máy chủ này cũng không thể đọc hoặc truy cập vào dữ liệu được gửi lên nó.

Những đặc điểm của máy chủ LNS:

- Kiểm tra địa chỉ của thiết bị
- Xác thực, đếm và quản lý gói tin.
- Xác nhận những gói tin nhận được.
- Tự động điều chỉnh tốc độ truyền gói tin sử dụng giao thức ADR.
- Trả lời tất cả những yêu cầu từ lớp địa chỉ MAC của thiết bị
- Chuyển tiếp tất cả những gói tin tải lên đến các máy chủ tầng ứng dụng phù hợp.
- Sắp xếp những gói tin tải xuống từ bất kỳ một máy chủ tầng ứng dụng nào đến những thiết bị đang kết nối vào mạng.
- Chuyển tiếp những gói tin yêu cầu kết nối (Join - request) và cho phép kết nối (Join - accept) giữa các thiết bị và máy chủ kết nối (Join server).

**LoRaWAN Gateway:** Nhận các gói tin được điều chế tín hiệu LoRa trong phạm vi phủ sóng tín hiệu của nó và chuyển các gói tin này đến máy chủ mạng LNS đang được kết nối với mạng lõi bằng giao thức IP. Trong LoRaWAN, mỗi gói tin tải lên của một thiết bị đầu cuối sẽ được nhận bởi tất cả các LoRaWAN Gateway trong phạm vi phủ sóng của nó. Đặc điểm này giúp giảm thiểu một cách đáng kể lỗi gói tin vì nếu Gateway này không nhận được thì sẽ có một Gateway khác nhận được gói tin hoàn chỉnh, giúp giảm tiêu thụ năng lượng trên những thiết bị đầu cuối và thêm khả năng định vị vị trí (giả sử mỗi Gateway được khai báo vị trí GPS).

LoRaWAN Gateway có thể kết nối đến một máy chủ (cloud server) thông qua đường mạng dây Ethernet, Wifi, mạng điện thoại 3G, 4G... LoRaWAN Gateway hoạt động hoàn toàn ở tầng vật lý của mô hình mạng và đóng vai trò không gì hơn là một thiết bị dùng để chuyển hướng gói tin LoRa lên mạng. LoRaWAN Gateway chỉ có một vai trò là kiểm tra tính toàn vẹn của gói tin LoRa nhận được thông qua giải thuật CRC. CRC (Cyclic Redundancy Check) là một giải thuật dùng để kiểm tra lỗi của gói tin. Nếu gói tin thông qua CRC được phát hiện có lỗi thì gói tin đó sẽ bị bỏ qua không nhận bởi Gateway. Nếu gói tin không có lỗi thì nó sẽ được gửi đến máy chủ mạng LoRa (LNS) cùng với một số siêu dữ liệu (metadata) như là giá trị RSSI của gói tin, thời gian nhận gói tin. Khi một LoRa Gateway thực hiện việc truyền gói tin trên kênh tải xuống, nó nhận lệnh truyền từ LNS mà không cần xử lý biên dịch gói tin trước. Khi một gói tin LoRa được gửi đi từ một thiết bị đầu cuối, tất cả LoRa Gateway sẽ nhận cùng một gói tin giống nhau và gửi đến LoRa LNS. Tại LNS sẽ thực hiện việc xóa gói tin trùng lặp do được nhận bởi nhiều Gateway. Thông qua giá trị RSSI của những gói tin giống nhau này, máy chủ LNS sẽ chọn gói tin nào được gửi từ Gateway mà có giá trị RSSI tốt nhất bởi vì điều này nghĩa là Gateway đó đang ở gần thiết bị cuối gửi gói tin LoRa đó nhất. Máy chủ LNS sẽ chọn Gateway đó để tải gói tin xuống thiết bị đầu cuối đó. Chính đặc điểm này giúp có nhiều sự lựa chọn loại Gateway khác nhau tùy vào mục đích sử dụng. Sử dụng LoRa Gateway 8 kênh cho hiệu năng nhận và xử lý gói tin thấp nhất hơn nhưng giá thành lại rẻ. LoRa Gateway 8 - 16 kênh có thể được dùng trong nhà hoặc ngoài trời. Gateway 64 kênh thường chỉ được dùng cho các hệ thống như các cột thu sóng ăng - ten trên cao hay các trạm tín hiệu đặt trên những tòa nhà chọc trời.

**Application Sever (máy chủ tầng ứng dụng):** Tầng này chịu trách nhiệm quản lý và thông dịch dữ liệu đến từ các thiết bị đầu cuối. Máy chủ này cũng tạo ra những gói tin tải xuống thuộc tầng ứng dụng đến các thiết bị đầu cuối.

**Join Sever (máy chủ kết nối):** Quản lý quá trình cho phép kết nối vào mạng LoRaWAN của các thiết bị đầu cuối. Máy chủ này chứa những thông tin cần thiết để xử lý những gói tin tải lên yêu cầu kết nối và tải xuống những gói tin cho phép kết nối. Máy chủ này báo hiệu cho máy chủ mạng biết rằng máy chủ tầng ứng dụng nào nên được kết nối đến thiết bị đầu cuối và trong phiên làm việc đó sẽ thực hiện việc phân giải ra mã khóa mạng (network session key) và mã khóa tầng ứng dụng (application session

key). Máy chủ kết nối sẽ liên lạc đến máy chủ mạng và máy chủ tầng ứng dụng tương ứng thông qua mã khóa mạng và mã khóa tầng ứng dụng. Chính vì lí do trên mà mỗi máy chủ kết nối đều chứa những thông tin sau của từng thiết bị đầu cuối mà nó quản lý:

- Mã DevEUI: Mã số định danh riêng cho từng thiết bị đầu cuối.
- Mã Appkey: Mã khóa tầng ứng dụng.
- Mã NwkSKey: Mã khóa tầng mạng.
- Thông tin định danh máy chủ tầng ứng dụng.
- Hồ sơ thông tin về dịch vụ tại thiết bị đầu cuối.

Nhằm đảm bảo tính bảo mật của hệ thống, chất lượng dịch vụ mạng và một số yêu cầu khác thì mỗi thiết bị trong mạng phải được ủy nhiệm và kích hoạt quyền kết nối vào mạng. Có hai phương pháp để kích hoạt quyền kết nối vào mạng:

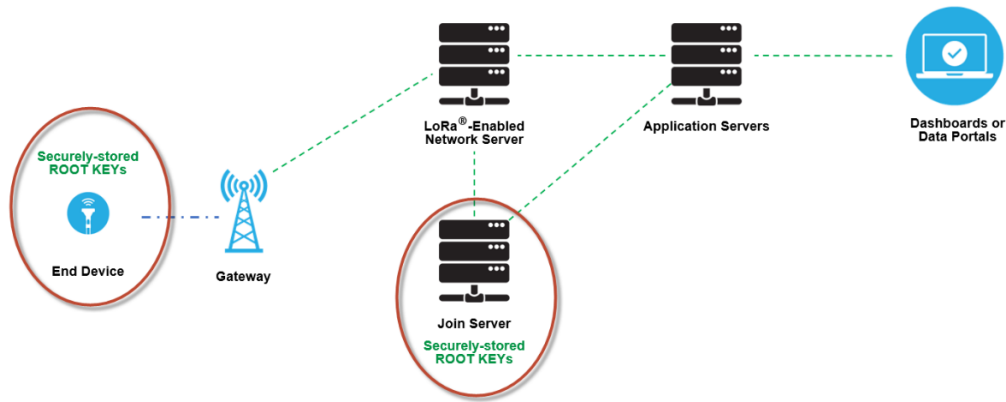
OTAA (Over – the - Air Activation)	ABP (Activation by Personalization)
<ul style="list-style-type: none"> <li>• Nhà sản xuất thiết bị tự động tạo ra các thông số cần thiết (như là định danh, mã khóa, vị trí máy chủ...)</li> <li>• Mã khóa bảo mật có thể được làm mới thường xuyên.</li> <li>• Mỗi thiết bị có thể chứa nhiều định danh để có thể tự động chuyển đổi một cách an toàn giữa các mạng và nhà vận hành trong suốt thời gian hoạt động của nó.</li> <li>• Chứa một số tính năng bảo mật cao cấp như chống giả mạo...</li> </ul>	<ul style="list-style-type: none"> <li>• Là một phương pháp ủy nhiệm quyền kết nối vào mạng được đơn giản hóa nhưng ít an toàn hơn.</li> <li>• Mã định danh và mã khóa được tạo ra riêng ở quá trình chế tạo.</li> <li>• Thiết bị có thể kết nối vào mạng ngay lập tức ngay khi khởi động, vì quá trình xin cho phép kết nối được bỏ qua.</li> <li>• Thiết bị chỉ có thể hoạt động trong những mạng/dịch vụ cụ thể.</li> </ul>

Bảng 2.7 So sánh kết nối OTAA và ABP

### 2.4.3 Bảo mật trong mạng LoRaWAN

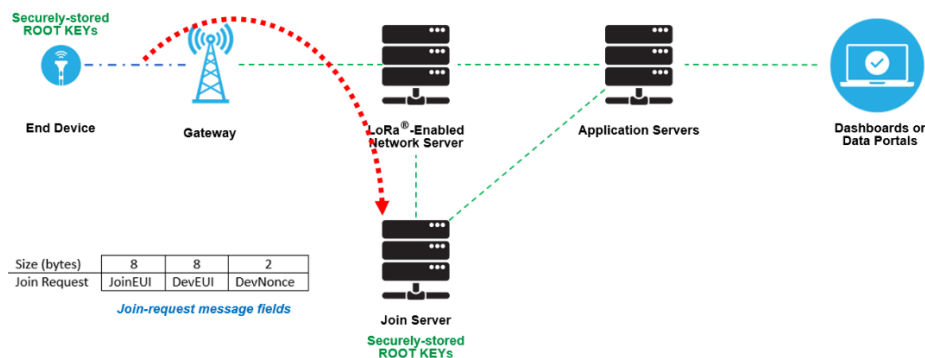
Tính bảo mật trong một mạng LoRaWAN được xác định dựa vào hai yếu tố chính: phương pháp ủy nhiệm kết nối và tính xác thực của gói tin. Chỉ có những thiết bị được ủy nhiệm mới có thể kết nối vào mạng. Địa chỉ MAC của mạng LoRaWAN và những

gói tin trong mạng được xác thực về nguồn gốc, được đảm bảo tính toàn vẹn của gói tin và được mã hóa. Những đặc điểm này nhằm đảm bảo rằng đường truyền dữ liệu trong mạng không bị can thiệp làm thay đổi, không bị nghe lén, bắt gói tin thay phát lại gói tin vì chỉ có những thiết bị hợp pháp được ủy quyền mới có thể kết nối vào mạng.



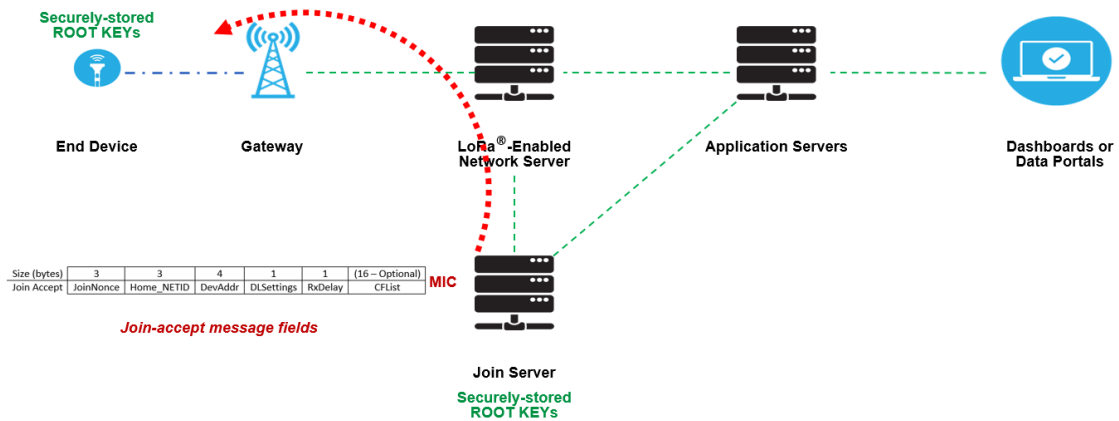
Hình 2.14 Mã hoá được tạo ra trong quá trình ủy nhiệm quyền kết nối

Mỗi mã hoá riêng được tạo ra ở mỗi thiết bị đầu cuối, và những mã này được lưu trữ tương ứng trên máy chủ kết nối. Sau đó mỗi thiết bị đầu cuối gửi một gói tin yêu cầu kết nối đến máy chủ kết nối như được minh họa trên hình 2.13.



Hình 2.15 Thiết bị đầu cuối gửi gói tin yêu cầu kết nối

Nếu mã hoá khớp nhau, máy chủ kết nối chấp thuận cho thiết bị đầu cuối được kết nối và gửi về một gói tin cho phép kết nối như được minh họa trên hình 2.14.



Hình 2.16 Máy chủ kết nối gửi trả về gói tin cho phép kết nối

## 2.5 KẾT LUẬN

Mục đích của đề tài là thiết kế một thiết bị nhỏ gọn, truyền xa và tiết kiệm năng lượng. Đặc điểm của thông tin truyền đi có kích thước rất nhỏ, chỉ 5 bytes cho 5 chữ số ở chế độ dự đoán toàn phần, và dữ liệu chỉ truyền đi mỗi tháng một lần tới trạm thu nên không cần tốc độ truyền và thời gian truyền nhanh. Nếu sử dụng các công nghệ truyền thông khác như Wifi, 3G, 4G, Zigbee... truyền đến trạm cơ sở sẽ tốn hao chi phí lớn, lãng phí tài nguyên do không tận dụng hết khả năng của các giao thức này (băng thông lớn, tốc độ truyền nhanh...). Mặt khác đặc điểm của giao thức truyền thông này là phải có được sự phủ sóng của trạm thu nên khi thiết bị rời khỏi vùng phủ sóng của trạm thu hay chuyển vùng trạm thu sẽ gây ra rắc rối cho việc quản lý, truyền dữ liệu. Sau đó để có thể truy cập dữ liệu từ các trạm thu này ở bất kì đâu thì phải gửi dữ liệu lên các dịch vụ máy chủ đám mây trả phí hoặc bị giới hạn rất nhiều về chức năng. Bên cạnh đó, đặc điểm của thiết bị yêu cầu sử dụng pin kích thước nhỏ, hoạt động lâu dài cho mỗi lần sạc pin. Vì vậy việc ứng dụng công nghệ LoRa và hạ tầng LoRaWAN là giải pháp phù hợp cho việc đáp ứng các yêu cầu bài toán đặt ra cho đề tài là thiết bị nhỏ gọn, truyền xa, mức sử dụng năng lượng thấp và tiết kiệm chi phí điều này sẽ mang lại hiệu quả tối ưu nhất cho dự án

Bản chất của hai bộ thư viện OpenCV và Tesseract là hàng trăm ngàn, hàng triệu dòng mã nguồn C/C++ được xây dựng bởi rất nhiều lập trình viên qua nhiều năm, do đó dung lượng thư viện lớn và đòi hỏi phần cứng cao. Bên trong hai bộ thư viện này chứa các module xử lý ảnh (thao tác với ảnh ở dạng các ma trận pixel bản chất là các mảng

hai chiều, ba chiều...) và module các thuật toán trí tuệ nhân tạo (như SVM trong OpenCV...). Sự đa dạng của hai khối thư viện này nhóm đã thử nghiệm quá trình huấn luyện và xử lý ảnh thông thường trên máy tính và tìm cách nạp chương trình xử lý ảnh đầy vào module ESP32-CAM nhưng không thành công với OCR Tesseract thì các nó yêu cầu lượng tài nguyên phần cứng và bộ nhớ khá lớn vậy nên các tài nguyên vốn có của ESP32-CAM có thể khiến cho việc chạy Tesseract bị cản trở hoặc thậm chí là tràn bộ nhớ PSRAM hơn nữa việc cấu hình cho Tesseract trên ESP32-CAM là khá khó khăn và tốn thời gian còn về phía thư viện OpenCV nhóm đã tìm hiểu và làm việc với Cmake (một platform hỗ trợ OpenCV chạy với C – bởi vì module ESP32-CAM không thực sự hỗ trợ Python) nhưng kết quả cũng không mấy khả quan, các thuật toán OpenCV chạy với C quá rườm rà việc đó cũng sẽ lại gặp vấn đề giống với thư viện Tesseract. Nhìn nhận bản chất vấn đề nhóm tiến hành tham khảo hai bộ thư viện mã nguồn mở này, chất lọc và tự viết lại thư viện theo ý mình với ít tính năng để cấp phát bộ nhớ phù hợp cho phần cứng hạn chế, giảm kích thước thư viện, mục đích mà nhóm muốn là có thể xử lý ảnh bằng cách can thiệp trực tiếp vào tệp thô của ảnh được chụp (bitmap). Phần mềm trên thiết bị của nhóm có hai module chính dùng để nhận diện chữ số như sau:

- Module xử lý ảnh:

Nhóm đã tự viết lại một số hàm xử lý ảnh như Cắt ảnh, Chuyển ảnh xám, Cân bằng sáng bằng Hist, Tăng độ sáng - tương phản, Lấy ngưỡng để chuyển ảnh đầu ra thành ảnh nhị phân bằng ngôn ngữ C chỉ sử dụng các bộ thư viện cơ bản của GCC compiler (stdio.h...) mà không sử dụng các thư viện của bên thứ ba như OpenCV, Tesseract. Kết quả đạt được là chương trình đủ tối ưu để chạy được trên board ESP32 - Cam.

- Module trí tuệ nhân tạo:

Trí tuệ nhân tạo là một lĩnh vực khó và nó đòi hỏi số lượng dòng code cùng quá trình nghiên cứu phát triển khó hơn rất nhiều so với module xử lý ảnh. Nhóm nhận thấy tự viết lại hoặc sử dụng các thuật toán máy học có sẵn là khả thi nhưng không tối ưu về chi phí và thời gian thực hiện, do Google từ năm 2017 đã đề xuất ra bộ thư viện TensorFlow Lite. Sau nhiều năm cùng với đội ngũ kỹ sư hàng đầu thế giới thì đến nay cơ bản bộ thư viện này đã hoàn thiện ở một mức độ mà nhóm nhận thấy có thể tận dụng khả năng thư viện của họ để thực hiện đề tài này một cách tối ưu nhất về thời gian và chi phí thực hiện. Hơn nữa xét riêng ở bài toán phân lớp ảnh (Image Classification) thì

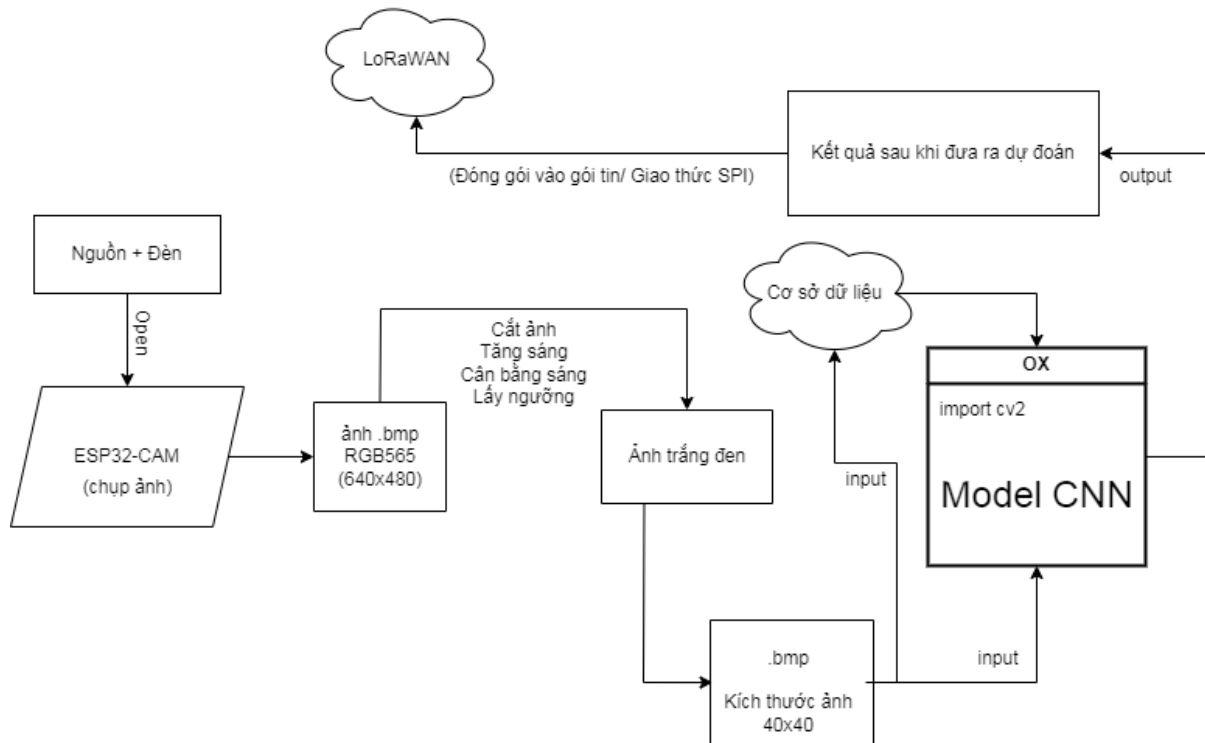
việc sử dụng mô hình học sâu cho thấy hiệu quả hơn hẳn phương pháp sử dụng các giải thuật máy học truyền thống vốn ưu điểm hơn ở các bài toán nhận diện âm thanh, văn bản...

Với các lí do kể trên, nhóm lựa chọn tự viết lại module xử lý ảnh và sử dụng thư viện máy học TensorFlow Lite.



## CHƯƠNG 3 THIẾT KẾ VÀ PHÂN TÍCH HỆ THỐNG

### 3.1 Tổng quan sơ đồ hệ thống



Hình 3.1 Sơ đồ hệ thống

Đầu tiên khi thiết bị được khởi động bằng công tắc, dòng vào sẽ giúp bật đèn Led thiết kế trên mạch và ESP32-CAM. ESP32-CAM chạy chương trình được nạp trên nó và lấy ảnh bằng cách chụp ảnh được cài đặt sẵn ở định dạng RGB565.

Sau khi ảnh được lấy sẽ đưa qua hàm chạy xử lý ảnh ví dụ như cắt ảnh, tăng độ sáng, cân bằng sáng và lấy ngưỡng ảnh để cho ảnh đầu ra là ảnh trắng đen. Khi đó chúng ta sẽ nhận được ảnh sau khi xử lý ở định dạng là ảnh xám với kích thước là 40x40 để đưa vào model huấn luyện được nhúng vào ESP32-CAM.

Sau khi ảnh được xử lý trước thềm chạy đưa ra dự đoán ảnh sẽ được làm input đầu vào cho ESP32-CAM chạy suy luận và nhận được kết quả dự đoán với ảnh đầu vào. Sau đó kết quả dự đoán sẽ được gửi về một nơi khác xa hơn qua LoRa.

## 3.2 Thu thập dữ liệu

### 3.2.1 Giới thiệu giải thuật xử lý ảnh trên ESP32-CAM

ESP32-CAM là một phiên bản phổ biến của module ESP32 được phát triển bởi Espressif Systems. Nó hỗ trợ Wifi, Bluetooth và các giao tiếp như UART, SPI, I2C, GPIO và PWM. ESP32-CAM thường tích hợp một camera OV2640 với độ phân giải cao và có khả năng điều chỉnh các thông số như màu sắc, độ sáng, độ tương phản và tỷ lệ khung hình... Ngoài ra ESP32-CAM hỗ trợ các loại thẻ nhớ như MicroSD để lưu trữ ảnh và video.

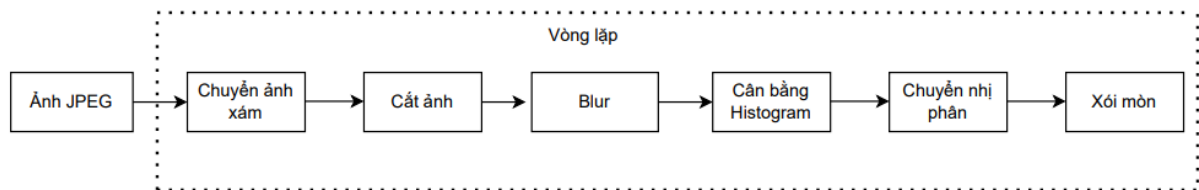
Với một module nhiều chức năng, mạnh mẽ và linh hoạt như vậy nhưng giá thành của ESP32-CAM lại thấp nhất hiện tại so với các dòng camera khác.



Hình 3.2 Module ESP32-CAM tích hợp camera OV2640

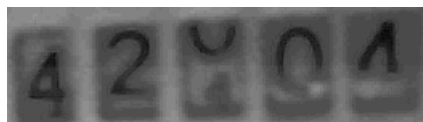
Camera trên ESP32-CAM có một nhược điểm lớn nhất đó là không có khả năng lấy nét. Sử dụng các giải pháp tăng độ nét bằng phần mềm cũng không khả thi vì ảnh chụp đã bị mờ và mất đi đặc trưng ảnh ngay từ ban đầu. Bên cạnh đó việc sử dụng các bước xử lý ảnh hoặc sử dụng các giải thuật cải tiến hơn như chỉnh ảnh ở vùng màu HSV sẽ gây tốn thêm thời gian chạy chương trình, gây hao phí năng lượng. Vì vậy nhóm chỉ sử dụng các giải thuật cơ bản nhất để hạn chế tối thiểu thời gian và năng lượng phần cứng để chạy chương trình.

Các ảnh chụp quá trình tiền xử lý và dữ liệu thu thập để huấn luyện mô hình học máy. Đối với bài toán nhận diện chữ số thì đặc trưng rõ nhất để phân biệt chúng chính là hình dạng. Đặc trưng này của chữ số được giữ lại thông qua quá trình tiền xử lý ảnh về dạng trắng đen rồi ảnh nhị phân, mỗi chữ số sẽ có khu vực xuất hiện giá trị nhị phân “1”, “0” khác nhau, từ đó đưa vào mô hình máy học để huấn luyện. Sau đây là lưu đồ quá trình xử lý ảnh:



Hình 3.3 Lưu đồ quá trình xử lý ảnh trực tiếp trên ESP32-CAM

Đầu tiên ảnh chụp ban đầu ở dạng ảnh màu JPEG kích thước 640x480 sẽ được lưu trong bộ nhớ PRAM của ESP32-CAM và được chuyển thành ảnh xám. Mặc dù ESP32-CAM và OV2640 có hỗ trợ kích thước lên tới 1600x1200 nhưng do nhóm chụp ảnh JPEG để xử lý ảnh bằng bitmap nên kích thước ảnh lớn không đủ bộ nhớ để lưu trữ. Vì lý do đó nên nhóm đã hạ kích thước phù hợp với bộ nhớ của ESP32-CAM. Ảnh có kích thước rất lớn đầu tiên sẽ đi qua bước cắt ảnh để cắt ra khu vực bao quanh tất cả 5 chữ số.



Hình 3.4 Ảnh bao quanh 5 chữ số

Thông qua các giá trị tọa độ cắt được cài đặt trong mã nguồn của thiết bị, ảnh sẽ được cắt nhỏ dần ra 5 ảnh chữ số riêng biệt để tiến hành tiền xử lý trên từng chữ số. Sau đó chương trình sẽ thực hiện việc tiền xử lý cho từng chữ số với các bước như ở trong phần vòng lặp ở hình 3.3. Một trong những bước tiền xử lý quan trọng nhất trong đề tài này là bước Histogram Equalization. Camera OV2640 trên ESP32 - Cam có nhược điểm là không có khả năng lấy nét, do đó ảnh bị mờ. Độ tương phản của ảnh rất thấp, các vùng màu xám chồng lấn lên nhau gây ra cảm giác nhìn không rõ chữ số, nếu không xử lý thì ảnh nhị phân sẽ bị mất đi đặc trưng hình dạng sau khi được chuyển đổi từ ảnh xám

này. Nhóm sử dụng giải thuật Histogram Equalization, ảnh xám được tối ưu về độ tương phản sau đó chuyển sang ảnh trắng đen sử dụng một giá trị ngưỡng do nhóm cài đặt. Thông qua bước này nhóm mới có thể tách ra được hình dạng nhất định của chữ số khi chuyển về dạng trắng đen.



Hình 3.5 Ảnh nhị phân của từng chữ số

### 3.2.2 Thu thập và xử lý dữ liệu máy học

Hiện nay có các bộ dữ liệu đồng hồ nước trên mạng mà nhóm đã tìm hiểu như:

- Water Meters Dataset của tác giả Kutsev Roman [36]:

Đây là bộ dữ liệu chứa các hình ảnh đồng hồ nước đa dạng và nhãn OCR để đọc số liệu của đồng hồ nước



Hình 3.6 Ảnh minh họa bộ dataset của tác giả Kucev Roman

Bộ dữ liệu này chụp cả ảnh xung quanh đồng hồ với các hình thái và góc chụp đa dạng từ xa đến gần, quay ngang quay dọc... Do đó bộ dữ liệu này sẽ phù hợp với các bài toán kết hợp giữa nhận diện chữ số trên ảnh và không cần đặt vị trí cắt ảnh chữ số thủ công như đề tài của nhóm. Nếu nhóm sử dụng bộ dữ liệu này thì sẽ không phù hợp vì

mục tiêu ban đầu của nhóm là sử dụng camera chụp ảnh ngay trên đồng hồ nước chứ không phải nhiều hình thái gần xa không cần thiết, thêm nữa việc sử dụng bộ dữ liệu để lấy hình ảnh chữ số phải thông qua các bước xử lý ảnh còn phức tạp và mất nhiều thời gian hơn việc nhóm tự thu thập dữ liệu.

- UFPR-ADMR Dataset từ công ty năng lượng Paraná (Copel) [37]:

Bộ dữ liệu này chứa 2000 hình ảnh đồng hồ quay số do các nhân viên của Công ty Năng lượng Paraná (Copel) được chụp từ bên trong nhà kho với định dạng JPG và độ phân giải 320x640 hoặc 640x320.



Hình 3.7 Ảnh minh họa bộ dữ liệu UFPR-ADMR

- UFPR-AMR Dataset từ phòng thí nghiệm thị giác, người máy và hình ảnh của Đại học Parana [38]:

Bộ dữ liệu bao gồm 2000 hình ảnh được chụp từ bên trong nhà kho của Công ty Năng lượng Paraná (Copel) với định dạng JPG và độ phân giải từ 2340x4160 đến 3120x4160.

- Copel - AMR dataset [39]:

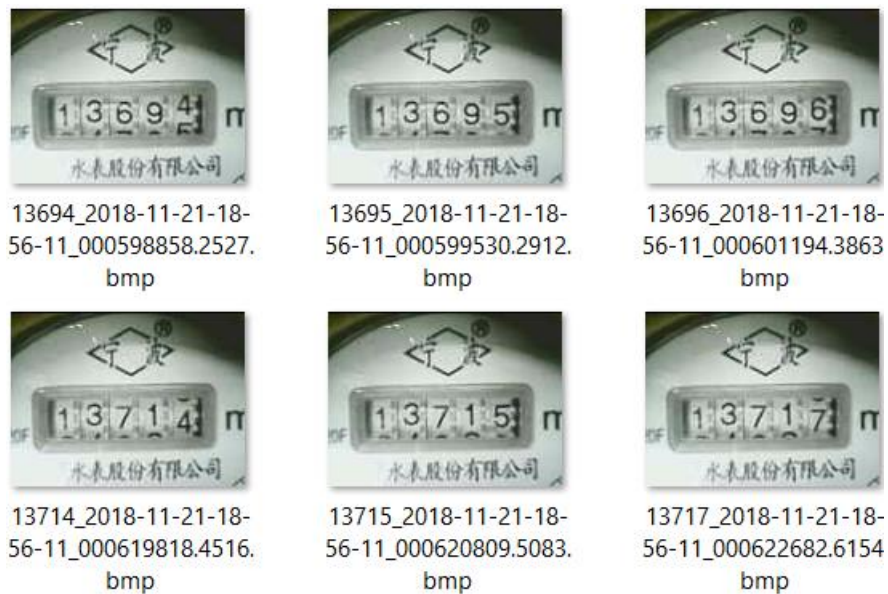
Bộ dữ liệu chứa 12500 hình ảnh được thu thập tại hiện trường bởi các nhân viên của Công ty Năng lượng Paraná (Copel) với các hình thái và góc chụp khác nhau như mờ, bụi bẩn, thay đổi tỷ lệ, xoay trong mặt phẳng và ngoài mặt phẳng, phản xạ, bóng và che khuất. Hình ảnh có độ phân giải 480x640 hoặc 640x480 tùy thuộc vào hướng mà chúng được chụp.





Hình 3.8 Ảnh minh họa bộ dữ liệu Copel-AMR

- Water - meter - recognition của tác giả Sloan96 [40]:



Hình 3.9 Ảnh minh họa bộ dữ liệu của tác giả Sloan96

Tập dữ liệu này được chụp giữ nguyên vị trí cố định. Tuy nhiên khoảng cách chụp khá xa và khi zoom ảnh lại gần thì sẽ gây nhiễu làm ảnh hưởng đến đặc trưng của các chữ số:



Hình 3.10 Ảnh sau khi zoom gần

Có thể thấy chữ số cuối cùng bên phải đã bị mờ nên khi chạy tiền xử lý tập dữ liệu này để huấn luyện là khả thi nhưng không đảm bảo đặc trưng sẽ không bị mất đi. Hơn nữa số lượng dữ liệu này cũng rất ít (364 ảnh) sẽ không đủ lượng dữ liệu để huấn luyện cho mô hình học sâu.

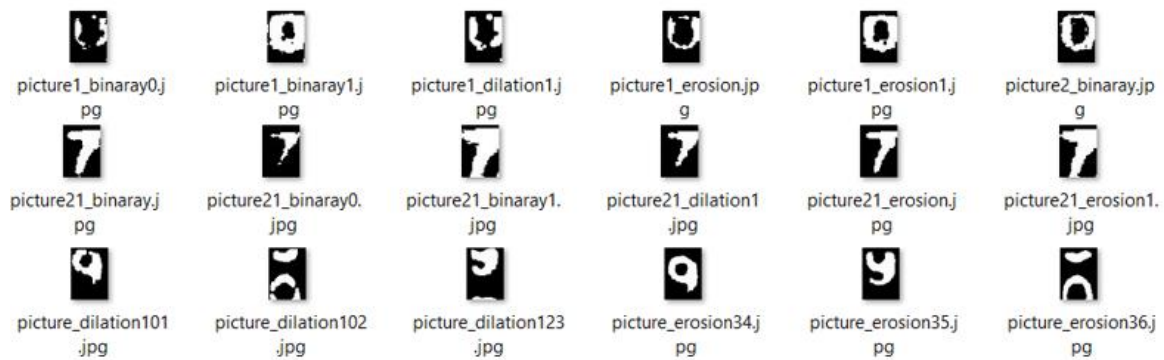
- MR - AMR Dataset của tác giả Ayman Naim, Abdessadek Aaroud, Dr Khalid, Chouaib El Hachimi [41]:



Hình 3.11 Ảnh minh họa bộ dữ liệu MR-AMR

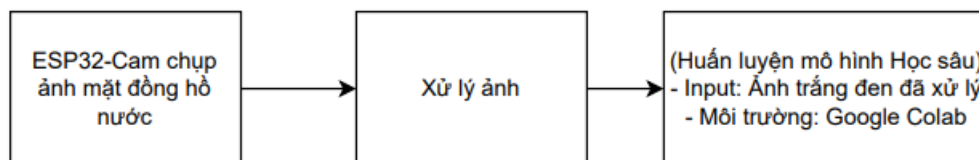
MR-AMR là một bộ dữ liệu công khai bao gồm 140000 hình ảnh của các chữ số ở trạng thái đầy đủ và số quay một nửa. Nó bao gồm các loại đồng hồ nước khác nhau. Ảnh được lưu dưới dạng nhị phân kích thước 28x28.

Bộ dữ liệu này thích hợp huấn luyện cho các camera có độ nét cao còn đối với camera ESP32-CAM đưa dữ liệu vào để huấn luyện sẽ làm giảm độ chính xác của mô hình vì ảnh chụp được từ ESP32-CAM khi chụp sẽ mất đi một phần đặc trưng của ảnh. Điều đó có thể nhận thấy được thông qua việc so sánh dữ liệu chụp từ chính camera trên ESP32-CAM đã qua tiền xử lý về ảnh nhị phân:



Hình 3.12 Một số hình ảnh bộ dữ liệu mà nhóm thu thập trực tiếp trên ESP32-CAM

Có thể thấy ảnh chụp bởi ESP32-CAM đã mất đi một phần đặc trưng một cách ngẫu nhiên do camera chất lượng thấp. Nếu tiền xử lý ảnh trên tập MR-AMR sao cho giống với ảnh chụp ESP32-CAM thì cũng không đảm bảo độ chính xác cao được. Hơn nữa tập dữ liệu này lại lưu ảnh ở kích thước 28x28, trong khi mô hình học của nhóm nhận input đầu vào lớn (40x40), nếu resize ảnh tăng kích thước sẽ khiến ảnh bị mờ và không còn giữ đúng hình dạng chữ số như trên tập dữ liệu gốc. Một trong những mục tiêu của đề tài này là tạo ra một thiết bị tối ưu cả về chi phí, do đó nhóm đề ra giải pháp tự thu nhập dữ liệu huấn luyện với mục tiêu có thể chạy dự đoán với phần cứng chất lượng thấp, giá thành thấp nhưng vẫn đạt độ chính xác cao với bộ dữ liệu tự thu thập.



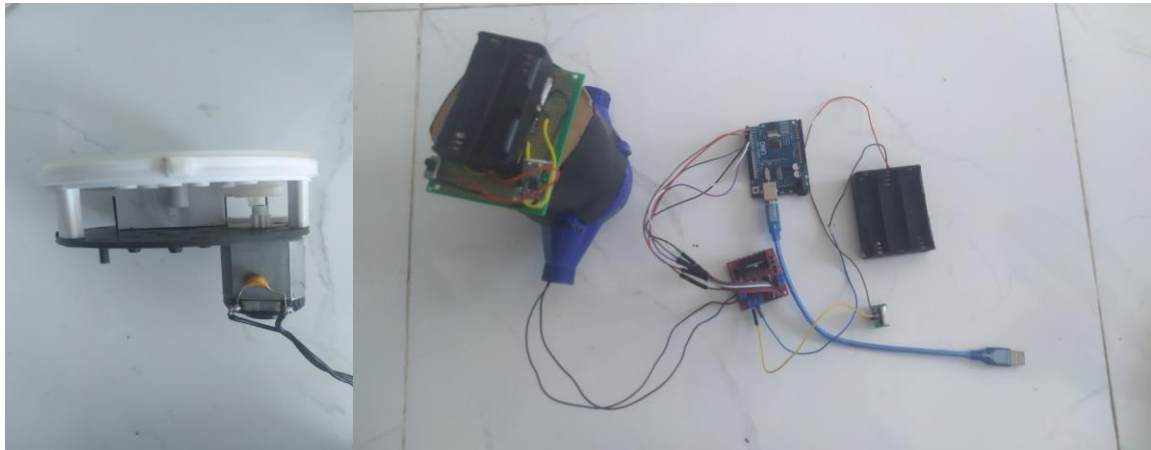
Hình 3.13 Lưu đồ khối tiền xử lý ảnh và huấn luyện máy học

Theo lưu đồ này thì ESP32-CAM chụp ảnh mặt đồng hồ nước, trải qua các bước tiền xử lý ảnh ở hình 3.3 sẽ cho ra ảnh nhị phân từng chữ số. Một bộ dữ liệu dataset được thu thập thông qua quá trình vừa chụp ảnh vừa sử dụng động cơ quay được nhóm gắn vào bên trong đồng hồ nước. Bộ dữ liệu này được huấn luyện trên một mô hình học sâu do nhóm tự xây dựng và chạy huấn luyện trên Google Colab. Sau khi huấn luyện xong sẽ chuyển đổi về dạng file .cc để nạp mô hình chạy dự đoán ngay trên ESP32-CAM mà không cần kết nối Internet.

Dữ liệu được thu thập để huấn luyện bằng cách gắn một cơ cấu động cơ do nhóm tự thiết kế vào bên trong đồng hồ nước. Motor được gắn với bánh răng sử dụng một

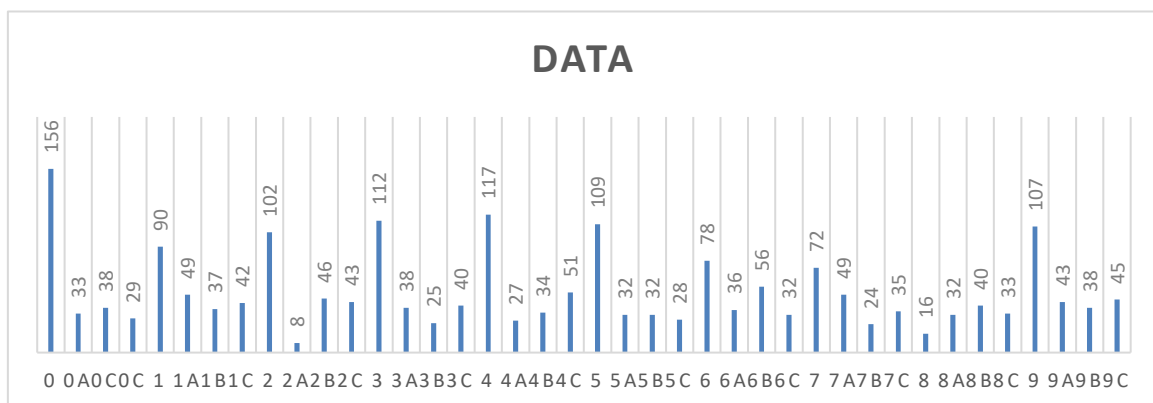


chân Analog của Arduino Uno tạo ra xung Hash để điều khiển một motor thông qua mạch cầu H L298N, cứ sau một ảnh chụp sẽ điều khiển động cơ nhích rất chậm làm cho số bên trong hộp số trên mặt đồng hồ quay rồi dừng lại, giúp nhóm lấy mẫu được ở các trường hợp số chỉ quay một phần đồng thời ESP32-CAM sẽ thực hiện chức năng chụp. Ảnh được lưu vào thẻ nhớ MicroSD đang kết nối đến ESP32 - Cam.



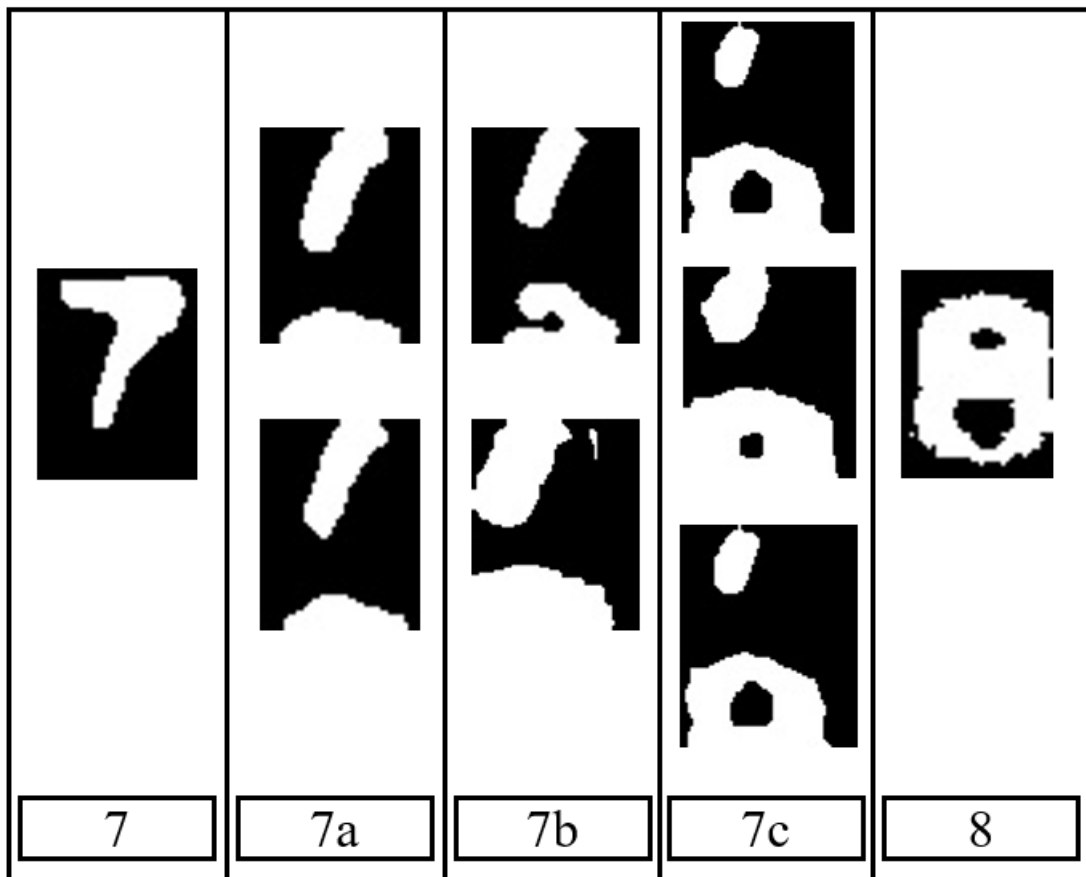
Hình 3.14 Thiết bị thu thập dữ liệu huấn luyện

Số lượng dữ liệu sau khi thu thập là 2164 mẫu, dưới đây là biểu đồ phân bố dữ liệu:



Hình 3.15 Biểu đồ phân bố dữ liệu cho từng lớp

Mô hình máy học của đề tài thực hiện bài toán phân lớp với số lượng nhãn là 40, tương ứng cho 10 trường hợp các chữ số nguyên 0 đến 9 và 30 trường hợp số quay ở giữa các chữ số nguyên này.



Hình 3.16 Ảnh minh họa một số trường hợp số quay ở giữa của dữ liệu

Để tăng tính rõ ràng trong việc nhận diện số trên mặt đồng hồ thì 1 số sẽ được chia thành 4 nhãn bao gồm số 7 là nhãn của ảnh được chụp đầy đủ nằm ở giữa khung hình và 3 nhãn còn lại là các khoảng giữa số 7 và số tiếp theo là số 8, tùy thuộc vào độ nhiều hay ít của trên ảnh nhóm sẽ chia ra làm 3 nhãn gần với số 7 đó là 7a, 7b và 7c. Có thể thấy 7a sẽ có một ít phần đầu của số 8 xuất hiện còn 7b sẽ chỉ còn một nửa số 7 xuất hiện và cuối cùng là số 8 sẽ xuất hiện phần nhiều so với phần còn lại của số 7.

Tập cơ sở dữ liệu nhóm chuẩn bị sẽ được chia thành 2 tập để bao gồm tập train và validation theo tỉ lệ 85% - 15%. Đầu tiên 85% được lấy từ mỗi 40 lớp để tổng hợp lại tạo thành tập train và số được làm tròn lên (ví dụ  $198 \times 85\% = 160.65$ , làm tròn lên nghĩa là lấy 161 mẫu trong 198 mẫu của nhãn 7, thực hiện tương tự cho các lớp còn lại). Bộ dữ liệu này bị mất cân bằng do các trường hợp số đầy đủ như nhãn 7 sẽ dễ dàng được thu thập hơn trong quá trình nhóm thu thập mẫu. Để khắc phục điều này thì nhóm đã thực hiện tăng số ảnh mỗi lớp (class) trên tập train này bằng phương pháp biến đổi hình thái ảnh (Augmentation) như xói mòn ảnh (erosion), giãn nở ảnh (dilation) và phương

pháp nhân bản (duplicate) ngẫu nhiên một số ảnh để tránh việc huấn luyện trên một bộ dataset bị mất cân bằng.

### 3.3 Thực hiện mô hình máy học (model training)

Loại lớp	Số Kernel	Size/Stride/Padding	Hàm kích hoạt
Input: 40 x 40 x 3			
Convolution 1	4	3 x 3 / 1 / valid	relu
Dropout (rate = 0.1)			
Max Pooling		2x2 / None / valid	
Convolution 2	8	3 x 3 / 1 / valid	relu
Max Pooling		2x2 / None / valid	
Convolution 3	16	3 x 3 / 1 / valid	relu
Dropout (rate = 0.2)			
Max Pooling		2x2 / None / valid	
Flatten			
Dense (units = 80, hàm kích hoạt: relu)			
Dense (units = 40, hàm kích hoạt softmax)			

Bảng 3.1 Thiết kế mô hình máy học của nhóm

Nhóm đã thử nghiệm với mô hình sử dụng phương thức như Yolo và thuật toán của Yolo là quá lớn nên không thể nạp vào bộ nhớ ESP32-CAM. Các mô hình học sâu hiện tại đều rất lớn chứa nhiều lớp xử lý bên trong mô hình và được thiết kế để chạy trên phần cứng có bộ nhớ và vi xử lý tương đối cao và lớn hơn ESP32-CAM. Hơn nữa nhiều mô hình có cấu trúc được thiết kế khác biệt như dữ liệu đầu vào là ảnh màu, hay kích thước ảnh đầu vào khác với ảnh nhị phân của nhóm... Nếu thực hiện sửa đổi các mô hình đã được công bố là khả thi nhưng không đảm bảo duy trì được hiệu năng như các mô hình này đã công bố, và sẽ tốn kém thời gian để nhóm nghiên cứu, chỉnh sửa và thực

nghiệm. Hơn nữa việc ứng dụng các mô hình phức tạp vào dữ liệu chỉ ở dạng ảnh nhị phân đơn giản là không tối ưu về chi phí tính toán của mô hình. Do đó nhóm thực hiện tự thiết kế mô hình ở bảng 3.1 trên và xét thấy mô hình đạt được hiệu năng mong muốn nên không tiếp tục thử nghiệm các mô hình học sâu đã được công bố khác. Trước khi quá trình huấn luyện được diễn ra thì cần thiết phải cài đặt các siêu tham số (hyperparameter) và cấu hình cho phù hợp vì chúng ảnh hưởng trực tiếp đến quá trình huấn luyện.

STT	Tên siêu tham số / Cấu hình	Giá trị
1	Learning rate	0.001
2	Batch size	64
3	Loss optimizer	Adam
4	Loss function	SparseCategoricalCrossentropy

Bảng 3.2 Bảng cấu hình siêu tham số cho mô hình

### 3.4 Triển khai mô hình trên ESP32-CAM

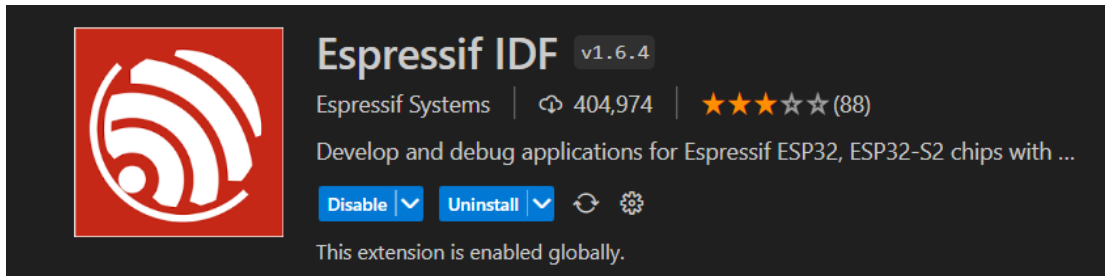
Sau khi thiết kế các thông số cho các lớp Kernel và cấu hình các siêu tham số cho mô hình huấn luyện, nhóm đã triển khai mô hình chạy trên ESP32-Cam bằng cách nhúng mô hình đã huấn luyện lên bảng mạch vi xử lý này, điều này nghĩa là board ESP32-Cam sẽ tự chụp ảnh, sau đó lấy thông số từ mô hình huấn luyện và thực hiện biên dịch cho mô hình và cuối cùng là sẽ chạy suy luận và đưa ra dự đoán.

Như đã nêu lên từ trước đó, bản thân của một board mạch vi xử lý như ESP32-Cam với chỉ một lượng nhỏ bộ nhớ là không thể nào đưa toàn bộ mô hình được huấn luyện vào, vậy nên nhóm đã thực hiện điều đó dựa trên một loại thư viện mã nguồn mở là TensorFlow Lite – là một thư viện mở rộng được phát triển từ Tensorflow nhằm giúp người dùng có thể triển khai các mô hình học máy và học sâu lên trên một hệ thống vi xử lý.

#### 3.4.1 Công cụ sử dụng triển khai trên ESP32-Cam

Như chúng ta đã biết, có nhiều loại công cụ hỗ trợ cho các loại vi xử lý khác nhau, một số rất phổ biến cho từng board như Arduino Uno, Mega,...thì có Arduino IDE, ESP

thì có Espressif IDF hay với STM32 thì có KeilC và CubeMX. Về bản thân ESP32-Cam mà nhóm sử dụng có thể lập trình trên cả Arduino IDE và PlatformIO chứ không chỉ riêng Espressif IDF, nhưng trong quá trình làm việc thì nhóm nhận thấy việc cấu hình trên Espressif IDF sẽ được hỗ trợ tối đa nhất.



Hình 3.17 Espressif IDF trên Visual Studio Code

### 3.4.2 Các thư viện cần thiết và cài đặt các thông số trong chương trình

Sau khi huấn luyện mô hình, mô hình sẽ được chuyển đổi qua định dạng file .tflite bằng thư viện TensorFlowLite khi đó mô hình huấn luyện sẽ được nén lại từ đó nhóm có thể chuyển file .tflite thành dạng Hex bằng lệnh “ `xxd -i converted_model.tflite > number_recognition_model_data.cc` ” khi đó model có thể được nhúng vào ESP32-Cam.

Tiếp theo là các thư viện cần thiết cho việc xây dựng chương trình cho việc nhúng tập mô hình đã chuyển đổi vào ESP32-Cam

bus	7/11/2023 4:28 PM	File folder
esp32-camera	7/11/2023 4:28 PM	File folder
esp-nn	7/11/2023 4:28 PM	File folder
fatfs	7/11/2023 8:19 PM	File folder
fb_gfx	7/11/2023 4:28 PM	File folder
freertos	7/11/2023 8:34 PM	File folder
nvs_flash	7/11/2023 4:28 PM	File folder
screen	7/11/2023 4:28 PM	File folder
sdmmc	7/11/2023 8:19 PM	File folder
tflite-lib	7/11/2023 4:28 PM	File folder

Hình 3.18 Các thư viện cần thiết khi làm việc trên Espressif IDF

Để có thể làm các công việc như biên dịch mô hình huấn luyện và các tham số đầu vào đầu ra của TensorFlowLite thì chúng ta cần phải có các thư viện hỗ trợ cho việc biên dịch các thuật toán về mạng thần kinh, ngoài ra còn cần các thư viện hỗ trợ việc chụp ảnh và lưu ảnh vào thẻ SD

Để xây dựng một chương trình có thể biên dịch được mô hình huấn luyện và chạy suy luận cũng như đưa ra các kết quả dự đoán, nhóm đã sử dụng các thành viên trong thư viện TFLite:

- Tflite::GetModel – để lấy được các thông số từ model.
- Tflite::MicroMutableOpResolver – để định nghĩa và quản lý các phép toán sử dụng trong mô hình.
- Tflite::Interpreter – để biên dịch các thành phần đầu vào như model, các phép toán được khai báo và các vùng bộ nhớ được cấp phát cho model.

Công việc sau khi khởi tạo các dữ liệu cho việc chạy suy luận thì chúng ta có thể gọi các biến con trỏ từ thành viên của thư viện Tflite nhằm đưa các dữ liệu đầu vào cho mô hình để khởi tạo lệnh chạy suy luận và đưa kết quả vào đầu ra bằng các biến thành viên như

- AllocateTensors – để cấp phát bộ nhớ cho đối tượng được biên dịch
- Input – Để lấy dữ liệu đầu vào và gán vào các biến con trỏ được gọi cho đầu vào của mô hình
- Invoke – Để thực hiện chạy suy luận và đưa ra dự đoán
- Output – Sẽ nhận kết quả dự đoán từ Invoke và lấy làm dữ liệu đầu ra.

Có thể thấy các biến thành viên thuộc thư viện Tflite đã được sử dụng lần lượt để gọi mô hình, định nghĩa các phép toán, biên dịch mô hình và các phép toán trong mô hình, lấy dữ liệu đầu vào, đưa ra dự đoán và cuối cùng là lấy kết quả đầu ra.

### 3.5 Truyền tin qua LoRaWAN

Theo như mục tiêu đề ra, sau khi hiện thực mô hình và chạy thử nghiệm nhóm muốn lưu trữ các dữ liệu bằng cách truyền dữ liệu đi qua giao thức mạng không dây như LoRa. Để kết nối với module LoRa nhóm sử dụng các chân TX-RX để đưa dữ liệu qua module và truyền đi.



Hình 3.19 Board LoRa RF Thing UCA

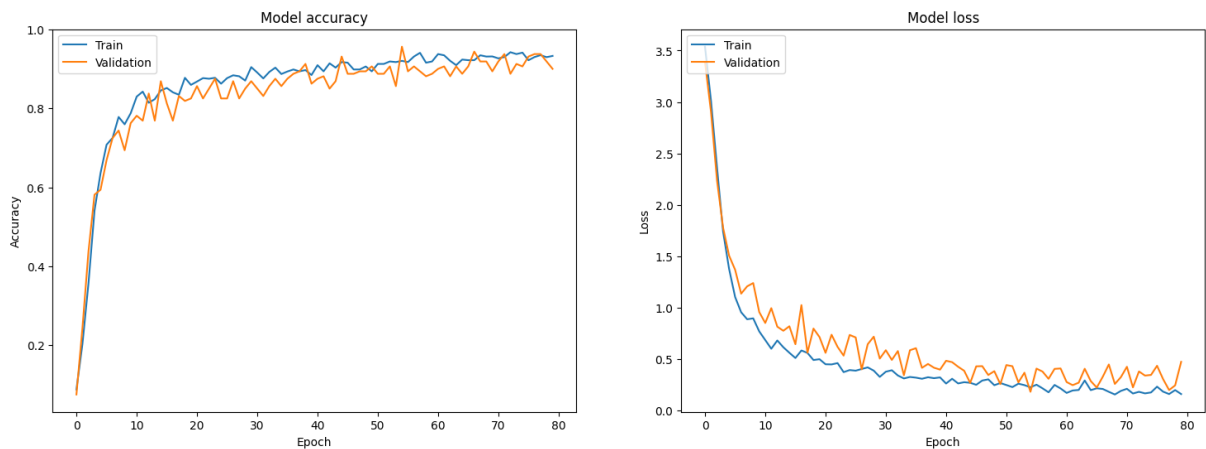
Đây là module LoRa mà nhóm đã sử dụng, các tham số đầu vào được gửi qua LoRa bằng kết nối giao thức TX – RX, module LoRa sẽ được nạp code sẵn để gửi lên GateWAY chung.

## CHƯƠNG 4 KẾT QUẢ VÀ ĐÁNH GIÁ HỆ THỐNG

### 4.1 Kết quả

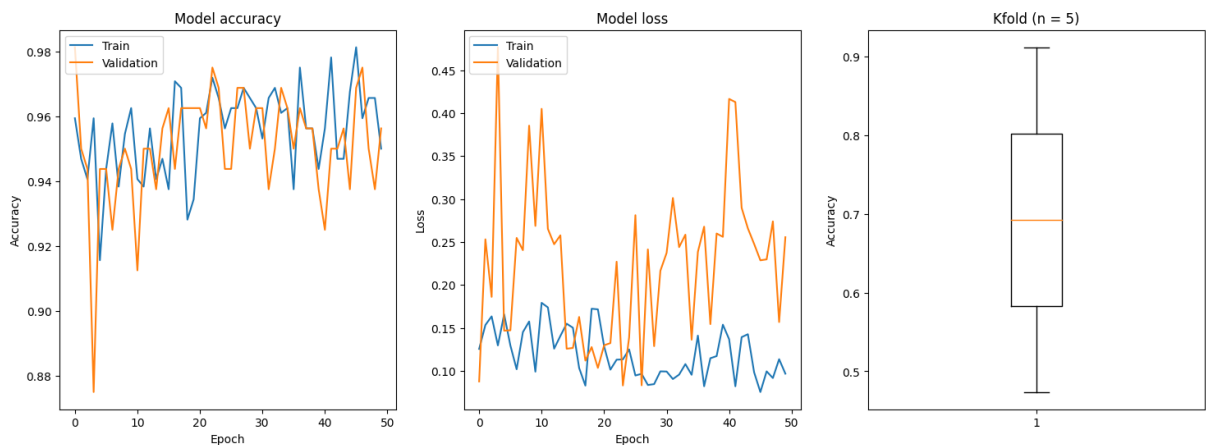
#### 4.1.1 Hiệu năng của mô hình máy học

Sau khi thực hiện train mô hình với 2164 mẫu. Kết quả về hiệu năng của mô hình được trình bày ở các hình dưới:



Hình 4.1 Biểu đồ quá trình huấn luyện, bảng phân phối độ chính xác

Biểu đồ huấn luyện cho thấy giá trị Loss và Validation gần như trùng với nhau cho thấy các giá trị về huấn luyện trên tập train và tập valid tương ứng với độ chính xác lên đến 0.94. Mô hình tạm thời đạt kết quả tốt, không bị quá underfit cũng như overfit, nhưng giá trị Accuracy còn chưa tuyệt đối chỉ đạt 0.94 cho nên các đưa ra dự đoán còn một vài dự đoán bị sai.





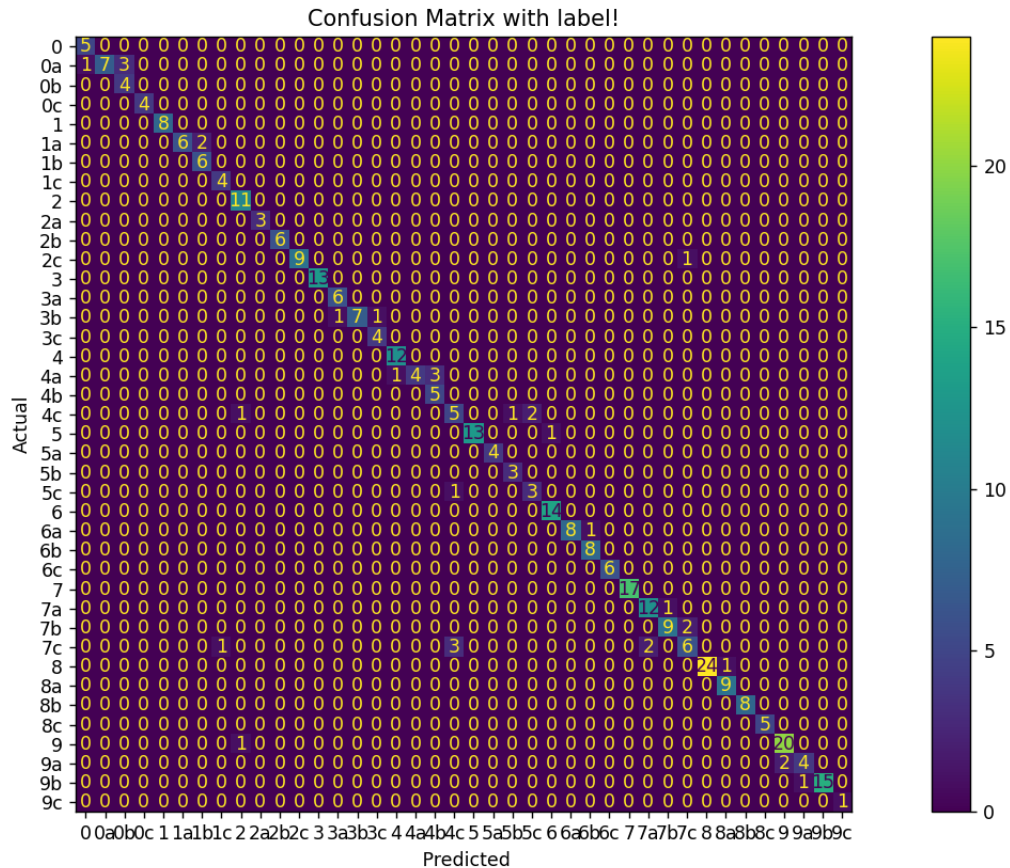
Hình 4.2 Biểu đồ quá trình huấn luyện, bảng phân phối độ chính xác với KFold.

Đối với KFold cho tập Test, tập Train và tập Validation được thực hiện với  $n = 5$  thì giá trị dự đoán chính xác bị giảm xuống còn 0.7 do quá trình huấn luyện nhóm đã sử dụng số lượng lớp Convolution lên tới 3 lớp và số lượng các lớp filter nhằm lấy đặc trưng trên các lớp kernel lần lượt là 4, 16 và 32 cùng với số cơ sở dữ liệu được sử dụng là 2164 ảnh cho tập train vậy mô hình là tương đối ổn định và chỉ cần train 1 lần, khi cho chạy 5 lần bằng Kfold khiến cho mô hình huấn luyện bị lặp đi lặp lại dẫn tới quá trình bị overfitting vậy nên chỉ cần huấn luyện 1 lần đối với mô hình.

	precision	recall	f1-score	support
0	0.83	1	0.91	5
0a	1	0.63	0.77	11
0b	0.57	0.64	0.60	4
0c	1	1	1.00	4
1	1	1	1.00	8
1a	0.87	1	0.93	8
1b	1	0.86	0.92	6
1c	0.8	1	0.89	4
2	0.85	1	0.92	11
2a	1	1	1.00	3
2b	1	1	1.00	6
2c	0.9	1	0.95	10
3	1	1	1.00	13
3a	0.86	1	0.92	6
3b	1	0.78	0.88	9
3c	0.8	1	0.89	4
4	0.92	1	0.96	12
4a	1	0.5	0.67	8
4b	0.625	1	0.77	5
4c	0.56	0.56	0.56	9
5	1	0.93	0.96	14

5	1	0.93	0.96	14
5a	1	1	1.00	4
5b	0.75	1	0.86	3
5c	0.6	0.75	0.67	4
6	0.93	1	0.96	14
6a	1	0.89	0.94	9
6b	0.89	1	0.94	8
6c	1	1	1.00	6
7	1	0.95	0.97	17
7a	0.86	0.92	0.89	13
7b	0.9	0.82	0.86	11
7c	0.75	0.5	0.60	12
8	1	0.96	0.98	25
8a	0.9	1	0.95	9
8b	1	1	1.00	8
8c	1	1	1.00	5
9	0.91	0.95	0.93	21
9a	0.8	0.67	0.73	6
9b	1	0.88	0.94	8
9c	1	1	1.00	9
Accuracy				352
Macro avg				0.90
				0.90
				0.89
				352

Hình 4.3 Giá trị Preision/Recall



Hình 4.4 Bảng Confusion Matrix

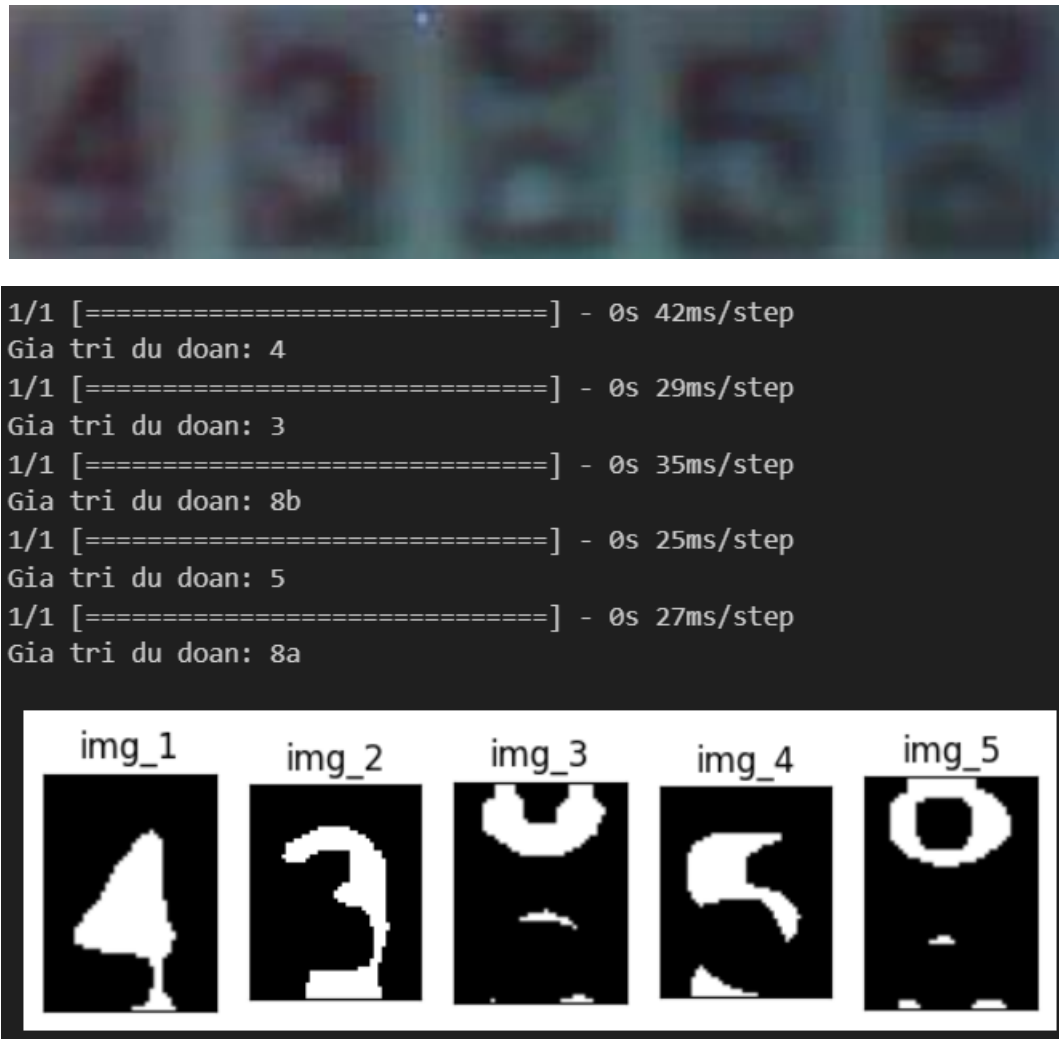
Nhóm đã thực hiện đánh giá mô hình bằng cách sử dụng tập test với Metric Function được phân chia sẵn trong lúc lấy cơ sở dữ liệu. Kết quả thu được thông qua tập Test cho thấy với một lượng dữ liệu ngẫu nhiên không được huấn luyện trước đó sẽ nhận được độ chính xác trung bình Macro-Avg là 94% với độ lệch chuẩn là 0.9. Điều này nghĩa là độ chính xác có thể tăng lên hoặc giảm đi 0.9% so với các giá trị được dự đoán là đúng.

Mô hình máy học sau khi huấn luyện được chuyển đổi về dữ liệu dạng mảng trong ngôn ngữ C để nạp vào bộ nhớ của ESP32 - Cam. Mô hình chiếm dung lượng 154 KB trong bộ nhớ của ESP32 - Cam.

#### 4.1.2 Kết quả thử nghiệm trên máy tính

Nhóm thực hiện quá trình đánh giá trên máy tính sau khi huấn luyện mô hình, nhóm đã chia thành 3 tập khi huấn luyện bao gồm 2164 ảnh cho train – 403 ảnh cho tập validation và 351 ảnh cho tập test và huấn luyện mô hình thu được kết quả mô hình như

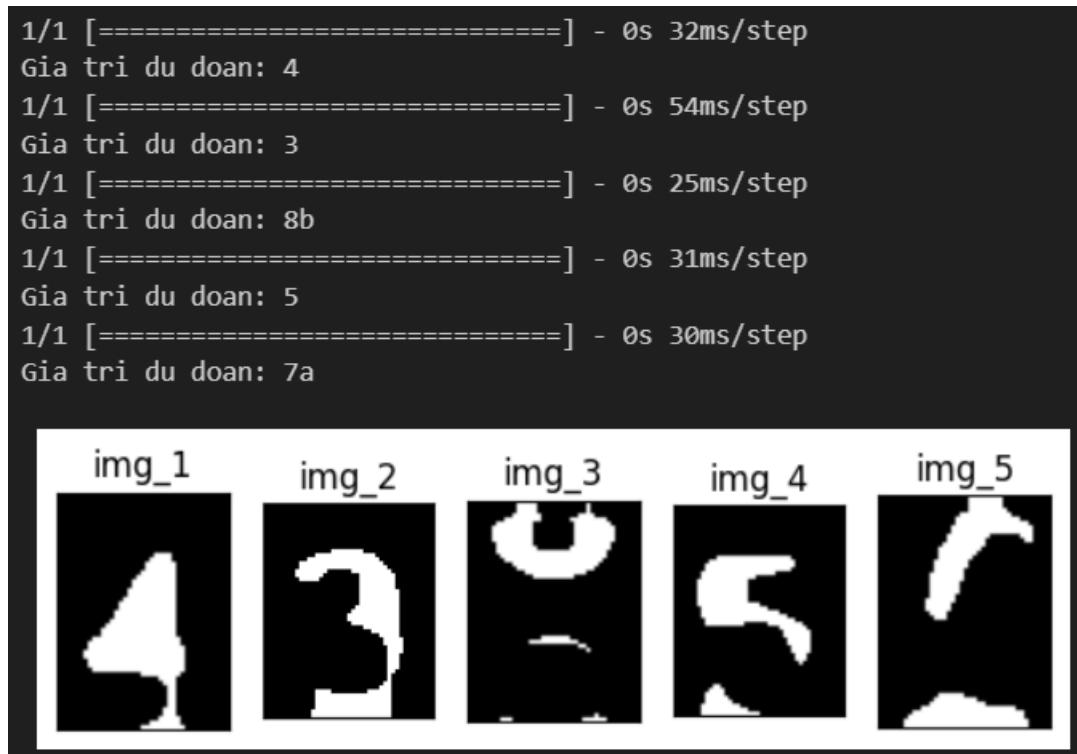
ở hình 4.1. Sau đó nhóm tiếp tục thử nghiệm quá trình chạy dự đoán trên máy tính bằng cách chụp ảnh từ module ESP32-CAM rồi sau đó lưu vào thẻ SD rồi lấy ảnh từ thẻ SD đưa vào máy tính thông qua đầu đọc thẻ SD, từ đó lấy làm ảnh đầu vào cho mô hình nhận diện nhằm kiểm tra mức độ chính xác của mô hình được huấn luyện từ trước đó và kết quả nhóm nhận được như sau:



Hình 4.5 Ví dụ minh họa về trường hợp dự đoán đúng với ảnh tốt

Trường hợp ảnh xấu hơn và sai về các bit màu trong quá trình chụp ảnh:





Hình 4.6 Ví dụ minh họa về trường hợp dự đoán đúng với ảnh xấu

Mô hình được huấn luyện với độ chính xác cao lên tới 94% đã đưa ra được những kết quả khá tích cực, có thể nhận diện được những ảnh mờ và những ảnh bị thiếu nét do bóng từ ánh sáng.

#### 4.1.3 Kết quả thử nghiệm thực tế

Sau khi thử nghiệm hiệu quả của mô hình trên máy tính, khi đã thu được kết quả khả quan từ mô hình, nhóm đã thực hiện chạy hệ thống trên thực tế bằng cách nhúng mô hình vào module ESP32-Cam. Nhóm đã thực hiện với một ảnh chụp ở cả hai định dạng ảnh là .bmp và .jpg được chụp và đưa cho đầu vào chạy dự đoán trên mô hình.

 PIC2.BMP	BMP File	5 KB
 PIC2.JPG	JPG File	2 KB

Hình 4.7 Hai định dạng ảnh được chụp từ Camera

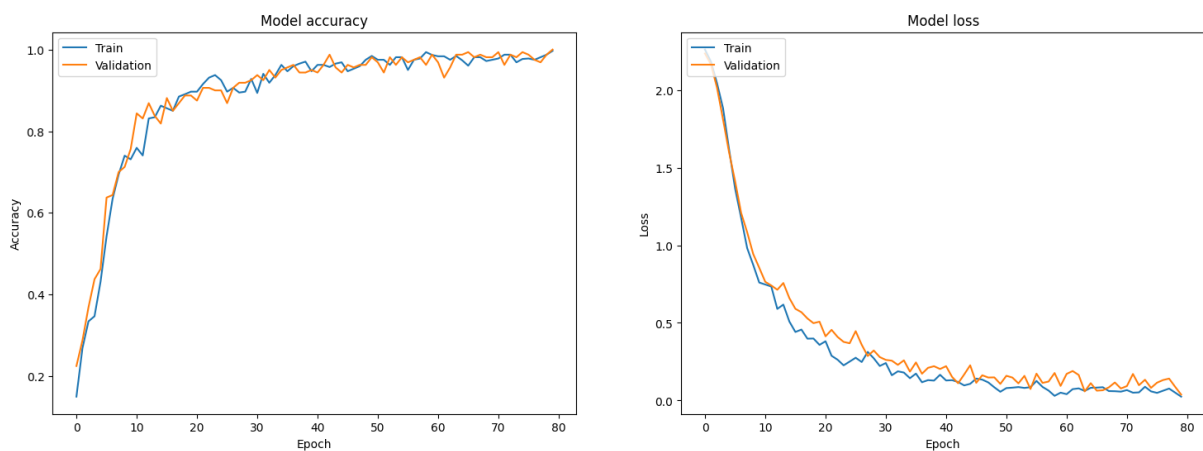
Bởi vì không gian bộ nhớ trên module ESP32-Cam là có hạn nên nhóm đã sử dụng mô hình sao cho kích thước không quá lớn, vậy nên nhóm sẽ giảm số nhãn xuống còn 10 nhãn chứ không chia rõ làm 40 nhãn, nhưng vẫn sẽ giữ các trường hợp ảnh ở giữa số hiện tại và số kế tiếp, khi đó mô hình mà nhóm thực hiện có kích thước là 325 KB, nhóm

đã thực hiện chạy thử nghiệm trước khi chuyển đổi mô hình trên máy tính và kết quả khả quan hơn so với 40 nhãn.

```
Epoch 79/80
10/10 [=====] - 1s 48ms/step - loss: 0.0508 - acc: 0.9875 - val_loss: 0.0884 - val_acc: 0.9875
Epoch 80/80
10/10 [=====] - 1s 54ms/step - loss: 0.0250 - acc: 0.9969 - val_loss: 0.0381 - val_acc: 1.0000
```

Hình 4.8 Kết quả là các giá trị huấn luyện cuối cùng

Accuracy ở những lần huấn luyện cuối cùng gần như đạt tới ngưỡng 100% và mô hình giữa tập train và tập valid là gần sát nhau không gặp phải hiện tượng overfitting hay underfitting.



Hình 4.9 Biểu đồ Accuracy và Loss đối với model được nhúng vào ESP32-Cam

Sau khi nhúng mô hình đã được chuyển đổi vào hệ vi xử lý, nhóm đã thực hiện chụp ảnh từ ESP32-Cam cùng với các bước xử lý ảnh để làm dữ liệu đầu vào cho mô hình, khi qua các bước xử lý ảnh nhóm đã thực hiện lưu ảnh vào SD để kiểm tra độ chính xác mà khung hình đã cắt.



Hình 4.10 Hai loại định dạng ảnh được chụp sau khi đã qua xử lý và lưu lại

Sau khi có dữ liệu đầu vào nhóm đã thực hiện chụp ảnh liên tục và chạy dự đoán liên tục 40 lần thì kết quả thực nghiệm cho thấy là chỉ với 1 lần được dự đoán là chính

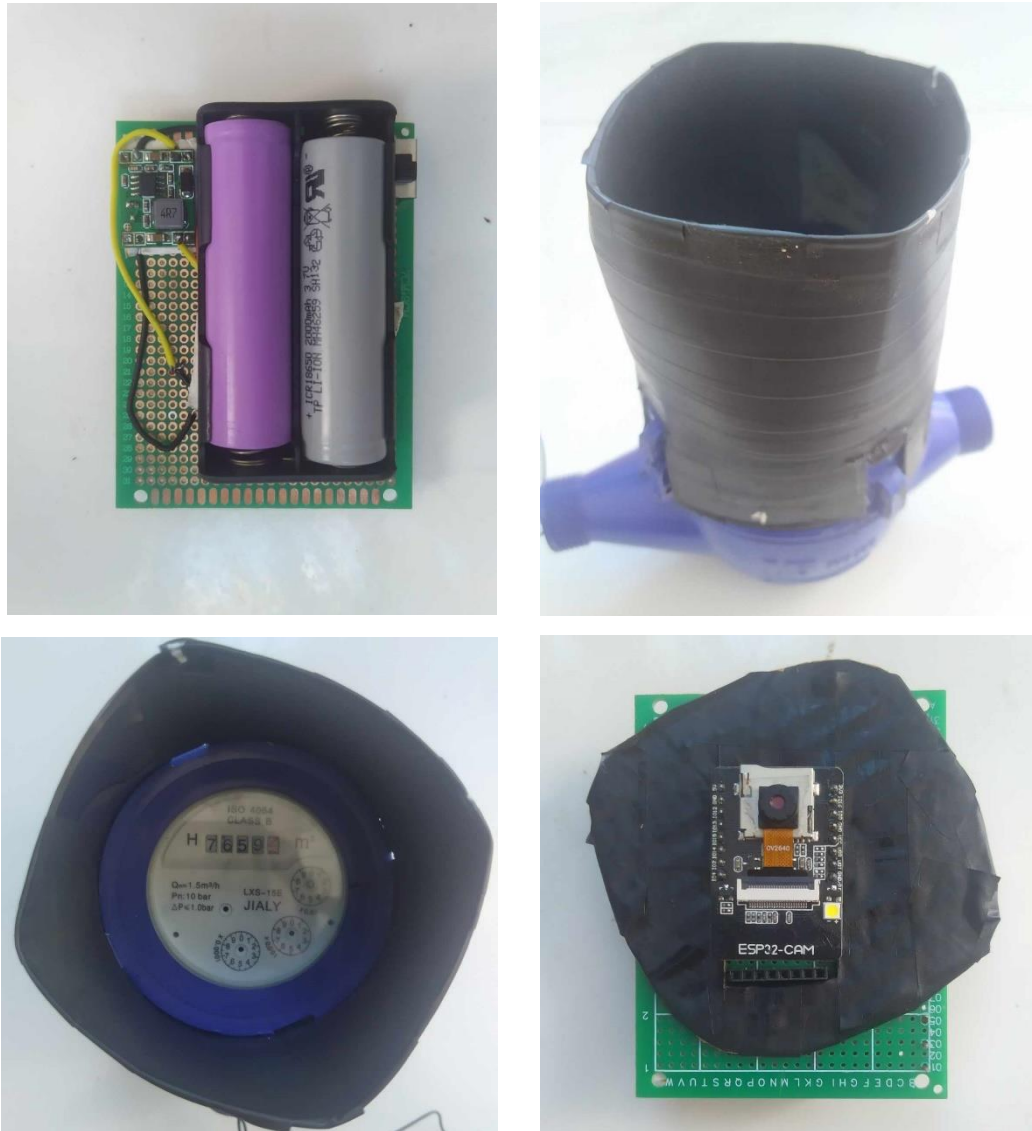
xác với số % được nhận định là số 2 cao nhất, còn có các lần còn lại được dự đoán gần đúng và bị nhầm ra các chữ số có nét cong như số 6, số 8, số 0 và số 9, hầu như không bị nhận nhầm là các số như 1, 5, 7 và 4.

```
Start Write Information -->
Predict Label || Label=0, Prob=0.351349 || Date=2086-01-01 Time=00:03:12
Predict Label || Label=1, Prob=0.000107 || Date=2086-01-01 Time=00:03:12
Predict Label || Label=2, Prob=0.354730 || Date=2086-01-01 Time=00:03:12
Predict Label || Label=3, Prob=0.000487 || Date=2086-01-01 Time=00:03:12
Predict Label || Label=4, Prob=0.002348 || Date=2086-01-01 Time=00:03:12
Predict Label || Label=5, Prob=0.015502 || Date=2086-01-01 Time=00:03:12
Predict Label || Label=6, Prob=0.093984 || Date=2086-01-01 Time=00:03:12
Predict Label || Label=7, Prob=0.002574 || Date=2086-01-01 Time=00:03:12
Predict Label || Label=8, Prob=0.130134 || Date=2086-01-01 Time=00:03:12
Predict Label || Label=9, Prob=0.048786 || Date=2086-01-01 Time=00:03:12
```

Hình 4.11 Kết quả dự đoán trực tiếp trên ESP32-Cam

Các kết quả dự đoán này sẽ được đóng vào gói tin và gửi tới một nơi để lưu trữ dữ liệu thông qua LoRa.

## 4.2 Đánh giá thiết kế phần cứng



Hình 4.12 Phần hệ thống Camera đặt trên mặt đồng hồ nước

Nhìn chung, các thiết bị và đề tài nghiên cứu mà nhóm đã phân tích ở chương một đa dạng về cách thiết kế hệ thống và thường có ưu điểm chung là độ chính xác cao. Tuy nhiên đa số lại chưa được tích hợp đầy đủ các yếu tố như dễ dàng triển khai trên diện rộng nhưng mà lại đảm bảo phù hợp với điều kiện kinh tế và môi trường ở nhiều vùng miền. Một số dự án khác cũng chưa được tối ưu về khả năng làm việc tự động cũng như việc đảm bảo mức tiêu thụ năng lượng vừa phải trong quá trình sử dụng, thêm vào đó là vấn đề truyền đi xa một nhằm giúp người dùng có thể gửi tới các đơn vị phụ trách cấp nước dùng cho các hộ dân kiểm soát. Board mạch nhúng ESP32 - Cam được tích hợp sẵn vi điều khiển, camera với độ phân giải trên lý thuyết có thể đạt 1600 x 1200 và một



số ngoại vi mà giá thành chỉ 160 000 VNĐ. Đề tài của nhóm sử dụng giải pháp phần cứng COTS (Commercial Off - The - Shelf) chi phí thấp, sử dụng thư viện máy học mã nguồn mở, được thiết kế để hoạt động lâu dài với mức điện năng tiêu thụ thấp (~1uA) và độ chính xác khá cao (> 90%). ESP32 - Cam là phần cứng mở (Open hardware) nên giá thành thậm chí có thể giảm xuống thấp hơn nếu không sử dụng board có sẵn dạng COTS. Bên cạnh đó thiết bị của nhóm còn có khả năng phân lớp cho nhiều khoảng giá trị số quay ở giữa và tích hợp công nghệ LoRa cho khoảng cách truyền xa, dễ triển khai thực tế trên diện rộng. Tuy nhiên nhược điểm của đề tài là chưa có khả năng phát hiện vùng chữ số (vị trí để xác định từng chữ số được cài đặt thủ công trong mã nguồn) và độ chính xác nhận diện chỉ số nước ở mức tương đối so với một số đề tài mà nhóm đã phân tích ở chương một.

### 4.3 Đánh giá quá trình triển khai thực tế

Nhìn chung quá trình triển khai thực tế của nhóm, nhóm đã thực hiện được mục tiêu đề ra – Xây dựng một mô hình mà có thể chụp ảnh từ mặt đồng hồ nước xử lý ảnh trên một tập bitmap thô, nhúng mô hình huấn luyện AI và một hệ vi xử lý, chạy suy luận để đưa ra dự đoán và cuối cùng là gửi dữ liệu qua LoRa. Nhìn chung thì mô hình đã đưa ra được dự đoán nhưng còn quá thấp. Kết quả dự đoán chưa được tốt có thể dẫn đến những kết quả sai lệch không đáng có trong quá trình chạy dự đoán. Các dự đoán sai có thể xảy ra bởi một vài nguyên nhân, ví dụ như mô hình cần cải thiện về mặt cơ sở dữ liệu, hiện tại cơ sở dữ liệu nhóm sử dụng còn tương đối ít, điều này có thể dẫn tới sự nhầm lẫn. Ngoài ra việc chạy dự đoán trên một hệ vi xử lý là khá nhỏ, việc bộ nhớ không thể hoàn toàn đáp ứng cũng có thể là nguyên nhân khiến cho việc dự đoán yếu hơn so với việc chạy trên thực tế. Nhóm đã cố gắng khắc phục trong một thời gian dài, tuy nhiên vì một vài bất cập về mặt thời gian, khối lượng công việc phải xử lý còn lớn vậy nên nhóm hiện chưa thể hoàn thiện được đề tài một cách hoàn hảo cho tới thời điểm viết báo cáo này.



## KẾT LUẬN

Bước đầu đề tài đã tìm hiểu được vấn đề cần giải quyết thông qua việc tìm hiểu thực trạng hạ tầng đồng hồ nước truyền thống đã quá cũ ở Việt Nam với nhiều bất cập như tốn kém chi phí cho một công việc quá đơn giản là thu thập chỉ số nước. Đề tài đã tham khảo nhiều thiết bị đã được thương mại hóa và đề tài nghiên cứu khác trên thế giới, từ đó đúc kết được những ưu điểm và hạn chế của các nghiên cứu này, giúp đề ra phương hướng giải quyết vấn đề và mục tiêu của thiết bị mà nhóm hướng tới. Mục tiêu chính của đề tài không chỉ là làm sao tạo ra một thiết bị chỉ đơn thuần là chạy được, mà nó còn phải thấp về chi phí, năng lượng và phải sử dụng các công nghệ cho phép dễ dàng triển khai trên diện rộng ở Việt Nam, ở cả những khu vực hiện đại như thành thị hay vùng nông thôn còn khó khăn, không có kết nối Internet.

Về lí thuyết đề tài đã tiến hành nghiên cứu nhiều công nghệ không chỉ là có thể sử dụng được, mà phải làm sao cho tối ưu về chi phí và thời gian thực hiện đề tài, bởi để có thể triển khai trên diện rộng ở Việt Nam thì thiết bị phải có giá thành thấp, ưu tiên các phần mềm mã nguồn mở, phần cứng mở và công nghệ truyền thông tầm xa phải đáp ứng tốt khả năng dễ dàng triển khai trên diện rộng. Một số công cụ và công nghệ có thể kể đến như LoRa, The Things Network, TensorFlow, KiCAD, ESP32 - Cam... đã được trình bày ở các phần trước trong báo cáo này.

Về việc hiện thực ý tưởng từ mục tiêu mà đề tài đặt ra, nhóm đã tiến hành xây dựng giải thuật tiền xử lý ảnh cho thấy có hiệu năng tốt ở nhiều môi trường ánh sáng khác nhau. Một kiến trúc mạng học sâu cũng đã được nhóm đề xuất và đã đạt được mục tiêu đủ nhẹ về dung lượng và độ phức tạp tính toán để có thể chạy được trên phần cứng nhúng chi phí thấp, nhưng vẫn duy trì được hiệu năng ở mức tốt. Một bộ dữ liệu huấn luyện cũng đã được nhóm tự thu thập cho đề tài. Phần cứng cũng bao gồm khả năng tự động chạy theo chu kỳ thu thập chỉ số nước ở Việt Nam và có thể truyền dữ liệu đi ở khoảng cách xa. Phần mềm của thiết bị cũng đã được điều chỉnh để có thể truyền phát LoRa ở tần số 923 Mhz là tần số cho phép tại Việt Nam. Bên cạnh đó một giải pháp theo dõi kết quả thu thập chỉ số nước cũng được đề xuất và phải đảm bảo không chỉ đơn thuần

là có thể chạy được, mà nó còn phải thuận tiện đến hết mức có thể cho phía người dùng hay cơ quan quản lý.

Nhìn chung, đề tài sử dụng linh kiện COTS chi phí thấp, năng lượng thấp cùng thư viện trí tuệ nhân tạo mã nguồn mở tạo tiền đề cho các ứng dụng trí tuệ nhân tạo sau này được phát triển, ví dụ như nhận diện khuôn mặt, nhận diện mã vạch trên sản phẩm hay các dự án về Quốc phòng - An ninh quốc gia như hệ thống tự động đánh chặn tên lửa xâm nhập vào không phận đất liền hoặc biển đảo thông qua việc nhận diện hình ảnh tên lửa, hệ thống tự động phát hiện người vượt biên trái phép được đặt ở các khu vực cửa ngõ biên giới hẻo lánh không có kết nối Internet hay hệ thống dẫn đường du hành đến các hành tinh xa hơn trong Hệ Mặt Trời cho vệ tinh nhân tạo, tàu không gian hoạt động trong môi trường vũ trụ không có kết nối Internet và sử dụng pin năng lượng Mặt Trời công suất thấp,...

Đề tài có hạn chế là chưa thực hiện bước phát hiện vùng số trên đồng hồ mà phải cài đặt vị trí chữ số một cách thủ công trong mã nguồn. Bề mặt của đồng hồ cũng phải tương đối sạch và không có nhiều động, vật cản. Mô hình máy học của đề tài hiện tại chỉ được huấn luyện với số mẫu dữ liệu còn hạn chế. Các phần mềm mã nguồn mở được sử dụng miễn phí nhưng nhược điểm là không ổn định bằng các phần mềm mã nguồn đóng hoặc được thương mại hóa. Thiết kế cơ khí của hộp đựng thiết bị cũng còn nhiều bất cập như không thể tùy ý thay đổi góc chụp của camera hay chưa có giải pháp chống lại việc ngưng tụ hơi nước bên ngoài hộp khiến cho việc nhận diện chỉ số nước bị ảnh hưởng.

## **HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI**

Với kết quả mà nhóm đạt được, nhìn chung dự án là một hướng đi mới vậy nên còn nhiều trở ngại đối với việc xây dựng mô hình AI trên một hệ vi xử lý vậy nên trong tương lai nhóm sẽ cố gắng khắc phục các nhược điểm tồn đọng. Trước hết cần nâng cấp giải thuật cho quá trình xử lý ảnh, cùng với đó là mô hình học sâu cũng rất cần thiết để đưa ra dự đoán tốt hơn, chính xác hơn và nhanh hơn nữa.

Về module truyền phát không dây LoRa còn gặp trở ngại khi truyền nếu gặp vật cản, khoảng cách có thể sẽ không được như mong đợi nếu truyền ở vị trí xa và mức năng lượng khi sử dụng trong quá trình truyền cũng có thể cải thiện vậy nên có thể nhóm sẽ thay thế hoặc nâng cấp loại module sử dụng trong dự án.

Tiếp tục thu thập thêm dữ liệu để làm cơ sở dữ liệu huấn luyện, giúp cho mô hình học máy có thể thu được trong nhiều loại đồng hồ nước khác nhau hơn nữa. Nghiên cứu thiết kế hộp nhựa chuẩn IP65 vào bất kỳ đồng hồ nào, ở bất kỳ đâu và có thể tùy chỉnh góc đặt hộp mà không phụ thuộc vào mặt vỏ hộp thiết kế 3D (chỉ dùng được cho một mẫu đồng hồ/hộp) như hiện nay. Ngoài ra cũng cần nghiên cứu vấn đề nước bốc hơi xung quanh hộp khiến cho đọng lại những giọt nước trên mặt đồng hồ làm cho quá trình chụp bị sai lệch ảnh, có thể thiết kế thêm cần gạt nước sử dụng năng lượng mặt trời để đỡ tốn năng lượng hơn.

## **TÀI LIỆU THAM KHẢO**

- [1] "Đồng hồ nước kết nối sóng radio Apator Powogaz," [Online]. Available: <https://rynantech.vn/products/product-category/dong-ho-do-luu-luong-nuoc-sinh-hoat-swm015b/>.
- [2] "Đồng hồ đo nước siêu âm Apator Model Ultrimis W," [Online]. Available: <https://eco-zenergy.com/product/dong-ho-do-nuoc-sieu-am-apator-ultrimis-w/>.
- [3] "Đồng hồ nước MBUS Apator Powogaz Model JS," [Online]. Available: <https://eco-zenergy.com/product/dong-ho-nuoc-apator-powogaz-js16-mbus-02/>.
- [4] "Đồng hồ nước thông minh model SWM015B," [Online]. Available: <https://rynantech.vn/products/product-category/dong-ho-do-luu-luong-nuoc-sinh-hoat-swm015b/>.
- [5] Mduduzi John Mudumbe, Adnan M. Abu-Mahfouz, "Smart water meter system for user-centric consumption measurement," In 2015 IEEE 13th International Conference on Industrial Informatics (INDIN).
- [6] M Suresh, U. Muthukumar, Jacob Chandapillai, "A novel smart water-meter based on IoT and smartphone app for city distribution management," In 2017 IEEE Region 10 Symposium (TENSYP).
- [7] Neeharika Cherukutota, Shraddha Jadhav, "Architectural framework of smart water meter reading system in IoT environment," In 2016 International Conference on Communication and Signal Processing (ICCSP).
- [8] Kwanchai Eurviriyakul, Kriatsanga Phiewluang, Sirisak Yawichai, Sirilak, Chaichana, "Evaluation of Recognition of Water-meter Digits with Application Programs, APIs, and Machine Learning Algorithms," In 2020 8th International Electrical Engineering Congress (iEECON).

- [9] Vlad Paul FERNOAGA, George-Alex STELEA, Alexandra BALAN, Florin SANDU, "OCR-based Solution for The Integration of Legacy And-Or Non-Electric Counters in Cloud Smart Grids," In 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME).
- [10] Yaoben Peng, Zhuang Chen, "Application of Deep Residual Neural Network to Water Meter Reading Recognition," In 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA).
- [11] Ayman Naim, Abdessadek Aaroud, Khalid Akodadi, Chouaib El Hachimi, "A fully AI-based system to automate water meter data collection in Morocco country," In Array, volume 10, July 2021.
- [12] Stefano Alvisi, Francesco Casellato, et al, "Wireless Middleware Solutions for Smart Water Metering," In Sensors 2019. [Online]. Available: <https://doi.org/10.3390/s19081853>.
- [13] Bickar, Keilin, "Integrating my Neptune Water Meter with HomeAssistant," 4/9/2019. [Online]. Available: <https://medium.com/@trumpetgod/integrating-myneptune-water-meter-with-home-assistant-896712a8c893>.
- [14] Guoxing Jin, Kai-Lung Bai, Han He, "A Smart Water Metering System Based on Image Recognition and Narrowband Internet of Things," In 2019 Rev. d'Intelligence Artif. [Online]. Available: <https://doi.org/10.18280/ria.330405>.
- [15] Yunzhou Zhang, Shanbao Yang, Xiaolin Su, Enyi Shi & Handuo Zhang, "Automatic reading of domestic electric meter: an intelligent device based on image processing and ZigBee/Ethernet communication," in 2016 Journal of Real-Time Image Processing volume 12. [Online]. Available: <https://doi.org/10.1007/s11554-013-0361-2>.
- [16] "Giới thiệu esp32-cam," [Online]. Available: <https://nshopvn.com/blog/gioi-thieu-mach-thu-phat-wifi-ble-esp32-cam-ai-thinker-huong-dan-su-dung-voi-arduino-thuc-hanh-lam-bo-mo-khoa-cua-nhan-dien-khuon-mat-bang-esp32-cam/>.

- [17] "What's the Difference Between AI, Machine Learning, and Deep Learning," 12/7/2018. [Online]. Available: <https://blogs.oracle.com/bigdata/post/whatx27s-the-difference-between-ai-machine-learning-and-deep-learning>.
- [18] "Insights Of The Machine Learning And The Deep Learning," 2/7/2018. [Online]. Available: <https://blog.thinkwik.com/insights-of-the-machine-learning-and-the-deep-learning/>.
- [19] "What are the types of machine learning," 5/12/2018. [Online]. Available: <https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f>.
- [20] K. R. Dalal, "“Analysing the Role of Supervised and Unsupervised Machine Learning in IoT”," In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC).
- [21] "Difference Between Algorithm and Model in Machine Learning," [Online]. Available: <https://machinelearningmastery.com/difference-between-algorithm-and-model-in-machine-learning>.
- [22] R. Saravanan, Pothula Sujatha, "A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification," In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS).
- [23] "Introduction to Supervised Deep Learning Algorithms!," [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/05/introduction-to-supervised-deep-learning-algorithms>.
- [24] "Deep Learning with TensorFlow 2.0 Tutorial – Building Your First ANN with TensorFlow 2.0," 27/8/2020. [Online]. Available: <https://kgptalkie.com/deep-learning-with-tensorflow-2-0-tutorial-building-your-first-ann-with-tensorflow-2-0/> [lastaccessed 5/2021].
- [25] "Deep Learning with PyTorch," [Online].

- [26] [Online]. Available: <https://developers.google.com/machine-learning/crash-course/classification/accuracy>,<https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2>,  
<https://towardsdatascience.com/multi-class-metrics-made-simple-pa>.
- [27] Sanjay Yadav, Sanyam Shukla, "Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification," in 2016 IEEE 6th International Conference on Advanced Computing (IACC).
- [28] Smith, R., "An Overview of the Tesseract OCR Engine," in Ninth International Conference on Document Analysis and Recognition (ICDAR 2007).
- [29] Zhang Shuaishuai, Peng Chen, "Research on License Plate Recognition Algorithm Based on OpenCV," In 2019 Chinese Automation Congress (CAC).
- [30] "TensorFlow: Open source machine learning," [Online]. Available: [https://www.youtube.com/watch?v=oZikw5k\\_2FM](https://www.youtube.com/watch?v=oZikw5k_2FM).
- [31] "Google Just Open Sourced TensorFlow, Its Artificial Intelligence Engine," [Online]. Available: <https://www.wired.com/2015/11/google-open-sources-its-artificial-intelligence-engine>.
- [32] "Deep Learning Framework Power Scores 2018," [Online]. Available: <https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a>.
- [33] "TinyML: Getting Started with STM32 X-CUBE-AI," [Online]. Available: <https://www.digikey.com/en/maker/projects/tinyml-getting-started-with-stm32-x-cu>.
- [34] "What are LoRa® and LoRaWAN," [Online]. Available: <https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>.
- [35] "What is LoRa," [Online]. Available: <https://lora-developers.semtech.com/get-started/what-is-lora>.

- [36] [Online]. Available: <https://github.com/jiarenf/Water-Meter-Number-DataSet>, <https://github.com/HCIILAB/Water-Meter-Number-DataSet>.
- [37] "Water Meters Dataset," [Online]. Available: <https://www.kaggle.com/tapakah68/yandextoloka-water-meters-dataset>.
- [38] "UFPR-AMR Dataset License Agreement," [Online]. Available: <https://web.inf.ufpr.br/vri/databases/ufpr-amr/license-agreement/>.
- [39] "Copel-AMR Dataset," [Online]. Available: <https://web.inf.ufpr.br/vri/databases/copel-amr/>.
- [40] [Online]. Available: [https://github.com/sloan96/water-meter-recognition/tree/master/opencv\\_version/data](https://github.com/sloan96/water-meter-recognition/tree/master/opencv_version/data).
- [41] Ayman, NAIM; Abdessadek, AAROUD; Khalid, AKODADI; Chouaib, El Hachimi, "MR-AMR Dataset," 2020.