

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN-ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN TỐT NGHIỆP
BẢN TÓM TẮT

**ỨNG DỤNG HỌC SÂU CHO VIỆC GIÁM SÁT
VÀ QUẢN LÝ KIỂM KÊ HÀNG HÓA TRONG
NGÀNH BÁN LẺ**

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Sinh viên: **NGUYỄN DUY HUÂN**
MSSV: 19119182

TP. HỒ CHÍ MINH – 05/2023

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐIỆN-ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN TỐT NGHIỆP
BẢN TÓM TẮT

**ỨNG DỤNG HỌC SÂU CHO VIỆC GIÁM SÁT
VÀ QUẢN LÝ KIỂM KÊ HÀNG HÓA TRONG
NGÀNH BÁN LẺ**

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Sinh viên: **NGUYỄN DUY HUÂN**
MSSV: 19119182

Hướng dẫn: **TS. HUỖNH THẾ THIỆN**

TP. HỒ CHÍ MINH – 05/2023

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin sinh viên

Họ và tên: Nguyễn Duy Huân
Tel: 0866078421

MSSV: 19119182
Email: huan2931@gmail.com

2. Thông tin đề tài

Tên của đề tài: Ứng dụng học sâu cho việc giám sát và quản lý kiểm kê hàng hóa trong ngành bán lẻ

Mục đích của đề tài: Thiết kế hệ thống kiểm kê hàng hóa dựa trên mạng học sâu

Đồ án tốt nghiệp được thực hiện tại: Bộ môn Kỹ thuật máy tính – Viễn thông, Khoa Điện - Điện Tử, Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh.

Thời gian thực hiện: Từ ngày 12/02/2023 đến 24/05/2023.

3. Các nhiệm vụ cụ thể của đề tài

- Tìm hiểu cấu trúc mạng YOLOv4-tiny.
- Thiết kế tối ưu kiến trúc mạng YOLOv4-tiny-SPP và YOLOv4-tiny-G3l dựa trên mạng YOLOv4-tiny.
- Đánh giá độ hiệu quả của mạng học sâu mới được thiết kế
- Thiết kế hệ thống xử lý dựa trên jetson nano và dữ liệu thời gian thực firebase

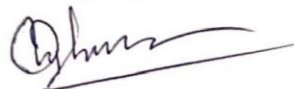
4. Lời cam đoan của sinh viên

Tôi Nguyễn Duy Huân cam đoan ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của tiến sĩ Huỳnh Thế Thiện.

Các kết quả công bố trong ĐATN là trung thực và không sao chép từ bất kỳ công trình nào khác.

Tp.HCM, ngày 24 tháng 05 năm 2023

SV thực hiện đồ án



Nguyễn Duy Huân

Giáo viên hướng dẫn xác nhận về mức độ hoàn thành và cho phép được bảo vệ:

Đã hoàn thành và cho phép được bảo vệ

Xác nhận của Bộ Môn

Tp.HCM, ngày tháng năm 2023

Giáo viên hướng dẫn

(Ký ghi rõ họ tên và học hàm học vị)



T.S. Huỳnh Thế Thiện

BẢN NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP
(Dành cho giảng viên hướng dẫn)

Đề tài: Ứng dụng học sâu cho việc giám sát và quản lý kiểm kê hàng hóa trong ngành bán lẻ

Sinh viên: Nguyễn Duy Huân

MSSV: 19119182

Hướng dẫn: TS. Huỳnh Thế Thiện

Nhận xét bao gồm các nội dung sau đây:

1. Tính hợp lý trong cách đặt vấn đề và giải quyết vấn đề; ý nghĩa khoa học và thực tiễn:

Đặt vấn đề rõ ràng, mục tiêu cụ thể; đề tài có tính mới, cấp thiết; đề tài có khả năng ứng dụng, tính sáng tạo.

Đặt vấn đề rõ ràng, mục tiêu cụ thể. Có khả năng ứng dụng và tính sáng tạo.

2. Phương pháp thực hiện/ phân tích/ thiết kế:

Phương pháp hợp lý và tập trung dựa trên cơ sở lý thuyết; có phân tích và đánh giá phù hợp; có tính mới và tính sáng tạo.

Có cơ sở về mặt thuật toán (kiến trúc mạng học sâu).

3. Kết quả thực hiện/ phân tích và đánh giá kết quả/ kiểm định thiết kế:

Phù hợp với mục tiêu đề tài; phân tích và đánh giá/ kiểm thử thiết kế hợp lý; có tính sáng tạo/ kiểm định chặt chẽ và đảm bảo độ tin cậy.

Kết quả thực hiện phù hợp với mục tiêu đề tài, là độ tin cậy cao.

4. Kết luận và đề xuất:

Kết luận phù hợp với cách đặt vấn đề, đề xuất mang tính cải tiến và thực tiễn; kết luận có đóng góp mới mẻ, đề xuất sáng tạo và thuyết phục.

Kết luận và đề xuất mang lại hiệu quả về mặt học năng (độ chính xác).

5. Hình thức trình bày và bố cục báo cáo:

Văn phong nhất quán, bố cục hợp lý, cấu trúc rõ ràng, đúng định dạng mẫu; có tính hấp dẫn, thể hiện năng lực tốt, văn bản trau chuốt.

Đúng định dạng mẫu và trình bày rõ ràng.

6. Kỹ năng chuyên nghiệp và tính sáng tạo:

Thể hiện các kỹ năng giao tiếp, kỹ năng làm việc nhóm, và các kỹ năng chuyên nghiệp khác trong việc thực hiện đề tài.

Tài phong chuyên nghiệp và có tính chủ động.

7. Tài liệu trích dẫn

Tính trung thực trong việc trích dẫn tài liệu tham khảo; tính phù hợp của các tài liệu trích dẫn; trích dẫn theo đúng chỉ dẫn APA.

Đúng và có cập nhật tài liệu tham khảo gần đây.

8. Đánh giá về sự trùng lặp của đề tài

Cần khẳng định đề tài có trùng lặp hay không? Nếu có, đề nghị ghi rõ mức độ, tên đề tài, nơi công bố, năm công bố của đề tài đã công bố.

Đề tài không trùng lặp. Có cơ sở về mặt thuật toán và kiến trúc mạng.

9. Những nhược điểm và thiếu sót, những điểm cần được bổ sung và chỉnh sửa*

*Cần chỉnh sửa tên nhiều tập data set thực tế hơn
thêm về cân bằng tập.*

10. Nhận xét tinh thần, thái độ học tập, nghiên cứu của sinh viên

Tốt

Đề nghị của giảng viên hướng dẫn

Ghi rõ: "Báo cáo đạt/ không đạt yêu cầu của một khóa luận tốt nghiệp kỹ sư, và được phép/ không được phép bảo vệ khóa luận tốt nghiệp"

Báo cáo đạt và được phép bảo vệ Khóa luận tốt nghiệp

Tp. HCM, ngày 11 tháng 05, năm 2023

Người nhận xét
(Ký và ghi rõ họ tên)

TS. Huỳnh Thế Thiện

LỜI CẢM ƠN

Trước tiên, xin được thể hiện lòng biết ơn sâu sắc đến TS. Huỳnh Thế Thiện, giảng viên tại trường đại học Sư Phạm Kỹ Thuật thành phố Hồ Chí Minh, vì những đề xuất và hỗ trợ quý báu của thầy đã giúp tôi có đủ điều kiện để thực hiện đề tài này. Hơn nữa, những kinh nghiệm và nghiên cứu khoa học của thầy trong lĩnh vực học máy và trí tuệ nhân tạo cung cấp cho tôi những ý tưởng và cơ hội để hoàn thành tốt đề tài này.

Xin chân thành cảm ơn các quý thầy cô giảng viên tại trường đại học Sư Phạm Kỹ Thuật thành phố Hồ Chí Minh, đặc biệt là các thầy cô trong ngành Công Nghệ Kỹ Thuật Máy Tính, khoa Điện - Điện Tử, đã giảng dạy và hỗ trợ tôi trong suốt quá trình học tập tại trường. Những đóng góp quý báu của các thầy cô đã góp phần quan trọng vào sự thành công của đề tài này.

Người thực hiện

Nguyễn Duy Huân

TÓM TẮT

Quản lý tồn kho và duy trì số lượng hàng hóa ở mức tối ưu là một yếu tố cực kỳ quan trọng trong ngành bán lẻ. Tồn kho được xác định là số lượng hàng hóa được lưu trữ trong kho, và việc giữ mức tồn kho cao vì sản phẩm không được tiêu thụ đúng như dự kiến hoặc nhập hàng vượt quá mức tiêu thụ có thể dẫn đến những hậu quả nghiêm trọng, gây tổn hại cho doanh nghiệp. Trong số các hậu quả này, tình trạng hết hàng trên các kệ tại siêu thị hoặc cửa hàng tiện lợi được xem là một trong những yếu tố cơ bản quyết định sự suy giảm doanh thu của doanh nghiệp.

Để giải quyết vấn đề đã đề xuất trong đề tài này, một giải pháp được đưa ra là sử dụng mô hình học sâu để giám sát và phát hiện vị trí hết hàng trên kệ hàng thông qua camera, và xử lý trên nền tảng phần cứng Jetson Nano. Mục đích của giải pháp là gửi thông tin về tình trạng hàng hóa trên kệ đến các nhân viên tại cửa hàng, giúp họ kịp thời xử lý và giảm thiểu các hậu quả, nhằm tối đa hóa lợi nhuận bán hàng cho các doanh nghiệp hoạt động trong ngành bán lẻ.

MỤC LỤC

DANH MỤC HÌNH	IX
DANH MỤC BẢNG	X
CÁC TỪ VIẾT TẮT.....	XI
CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI	1
1.1 ĐẶT VẤN ĐỀ	1
1.2 MỤC TIÊU ĐỀ TÀI	1
1.3 PHƯƠNG PHÁP NGHIÊN CỨU	2
1.4 BỐ CỤC TRÌNH BÀY.....	2
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	4
2.1 HỌC SÂU TRONG THỊ GIÁC MÁY TÍNH	4
2.2 MẠNG HỌC SÂU YOLO	5
2.2.1 Giới thiệu về mạng học sâu YOLO	5
2.2.2 Một số thuật ngữ cơ bản.....	6
2.2.3 Giới thiệu kiến trúc mạng YOLOv4-tiny	9
2.2.4 Lớp Phân bố chồng không gian (Spatial pyramid pooling)	10
2.2.5 Nhóm các lớp tích chập (Group convolutions)	11
CHƯƠNG 3 MÔ HÌNH ỨNG DỤNG	13
3.1 THIẾT KẾ MÔ HÌNH ỨNG DỤNG.....	13
3.1.1 Tổng quan mô hình.....	13
3.1.2 Nguyên lý hoạt động	14
3.2 THIẾT KẾ SƠ ĐỒ THUẬT TOÁN XỬ LÝ TRÊN JESON NANO	14
3.2.1 Mô hình kết hợp xử lý phát hiện phân loại OOS và IS	14
3.2.2 Đề xuất kiến trúc YOLOv4-tiny-SPP	15
3.2.3 Đề xuất kiến trúc YOLOv4-tiny-G3I.....	17
3.2.4 Sơ đồ thuật toán.....	20
3.3 THIẾT KẾ SƠ ĐỒ THUẬT TOÁN XỬ LÝ TRÊN GIAO DIỆN	20

3.3.1	<i>Sơ đồ thuật toán.....</i>	20
3.3.2	<i>Sơ đồ khối giao diện hiển thị</i>	22
CHƯƠNG 4 THỰC HIỆN MÔ HÌNH VÀ ĐÁNH GIÁ.....		23
4.1	HUẤN LUYỆN MÔ HÌNH HỌC SÂU DỰA TRÊN KIẾN TRÚC MẠNG YOLO	23
4.1.1	<i>Định dạng dữ liệu huấn luyện.....</i>	23
4.1.2	<i>Đánh giá kết quả huấn luyện.....</i>	24
4.2	LẬP TRÌNH HỆ THỐNG	28
4.3	ĐÁNH GIÁ KẾT QUẢ	30
CHƯƠNG 5 KẾT LUẬN.....		32
5.1	TỔNG KẾT LẠI QUÁ TRÌNH THỰC HIỆN ĐỀ TÀI.....	32
5.1.1	<i>Nghiệm thu kết quả</i>	32
5.1.2	<i>Tổng kết lại quá trình thực hiện đề tài</i>	32
5.2	HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI.....	33
TÀI LIỆU THAM KHẢO		34
PHỤ LỤC.....		36

DANH MỤC HÌNH

Hình 2. 1: Hình ảnh minh họa về Deep learning (nguồn Internet)	4
Hình 2. 2: So sánh FPS mạng YOLO (nguồn Internet)	5
Hình 2. 3: Kiến trúc chi tiết mạng YOLOv4-tiny	9
Hình 2. 4: Spatial pyramid pooling	10
Hình 2. 5: Group convolutions (nguồn Internet)	11
Hình 3. 1: Mô hình hệ thống giám sát OOS và IS	13
Hình 3. 2: Sơ đồ trình tự hoạt động hệ thống.....	14
Hình 3. 3: Sơ đồ khối mô hình xử lý phát hiện vị trí OOS và IS.....	15
Hình 3. 4: Kiến trúc YOLOv4-tiny-SPP	16
Hình 3. 5: kiến trúc YOLOv4-tiny-G3l	18
Hình 3. 6: Nhóm các lớp tích chập (nguồn Internet)	18
Hình 3. 7: Sơ đồ thuật toán xử lý trên Jetson Nano	20
Hình 3. 8: Sơ đồ thuật toán trên giao diện hiển thị	21
Hình 3. 9: Sơ đồ khối giao diện hiển thị	22
Hình 4. 1: Chuẩn bị dữ liệu huấn luyện	23
Hình 4. 2: YOLOv4-tiny và YOLOv4-tiny-SPP trên tập ảnh OSS	25
Hình 4. 3: YOLOv4-tiny và YOLOv4-tiny-G3l trên tập ảnh IS	27
Hình 4. 4: Hình ảnh giao diện hiển thị	29
Hình 4. 5: Giao diện hiển thị biểu đồ kiểm kê hàng hóa	30
Hình 4. 6: Kết quả chạy thử mô hình.....	30
Hình 4. 7: Giao diện kiểm kê hàng hóa trong ngày	31
Hình 5. 1:Mô hình áp dụng tại các siêu thị lớn.....	33

DANH MỤC BẢNG

Bảng 3. 1: So sánh BFLOPS giữa YOLOv4-tiny và YOLOv4-tiny-SPP	17
Bảng 3. 2: So sánh BFLOPS giữa YOLOv4-tiny và YOLOv4-tiny-G3l.....	19
Bảng 4. 1: Tập obj.data.txt và obj.names.txt	23
Bảng 4. 2: So sánh mAP giữa các mô hình trên tập ảnh phát hiện OOS.....	26
Bảng 4. 3: So sánh mAP giữa các mô hình trên tập ảnh phát hiện IS	28

CÁC TỪ VIẾT TẮT

OOS	Out Of Stock
IS	In Stock
ML	Machine Learning
DL	Deep Learning
AI	Artificial Intelligence
CNN	Convolutional Neural Network
YOLO	You Only Look Once
BFLOPS	Billion floating-point operations
SPP	Spatial pyramid pooling

CHƯƠNG 1

TỔNG QUAN ĐỀ TÀI

1.1 ĐẶT VẤN ĐỀ

Trong ngành bán lẻ, quản lý hàng tồn kho và duy trì lượng hàng hóa ở mức cần thiết là một yếu tố cực kỳ quan trọng. Tồn kho được định nghĩa là số lượng hàng hóa đang được lưu trữ trong kho. Nếu mức độ tồn kho cao là do sản phẩm không được tiêu thụ mạnh và hàng hóa nhập về nhiều hơn mức tiêu thụ, điều này sẽ dẫn đến rất nhiều hệ lụy gây tổn hại nặng nề cho doanh nghiệp. Trong việc quản lý hàng tồn kho, việc tránh tình trạng hết hàng trên kệ tại các địa điểm bán lẻ là một trong những yếu tố quan trọng. Việc này được giám sát bởi một số nhân viên tại các cửa hàng tiện lợi và các siêu thị lớn để đảm bảo lấp đầy hàng hóa vào các kệ hàng một cách liên tục và tránh việc hết hàng (OOS - Out Of Stock). Việc dẫn đến tình trạng OOS là hệ quả đến từ nhiều yếu tố khác nhau, từ phân bổ chi phí cho nguồn lao động và nhằm đáp ứng tốt sự hài lòng của khách hàng. Đối với các địa điểm bán lẻ các số liệu về lượng hàng có sẵn trên kệ IS (In-Stock) là một trong những yêu cầu tối thiểu cần phải có cho bất kì chuỗi cung ứng nào.

1.2 MỤC TIÊU ĐỀ TÀI

Để giải quyết vấn đề về OOS trong đề tài này sẽ tập trung vào việc phát triển ứng dụng nhằm mục đích theo dõi và đánh giá tình trạng của kệ hàng tại các cửa hàng bán lẻ hoặc trong các siêu thị lớn dựa trên học máy và thị giác máy tính nhằm nhận diện các vị trí OOS trên kệ hàng đồng thời kiểm kê các số liệu về lượng hàng hóa đang có trên kệ IS và gửi dữ liệu về cho

các nhân viên kịp thời lấp đầy các vị trí. Đề tài này sẽ tập trung vào nghiên cứu để giải quyết 2 mục tiêu chính.

Thứ nhất là phát triển và huấn luyện mô hình nhận diện các điểm OOS và các vị trí đang có hàng IS trên kệ hàng tại các siêu thị hoặc các cửa hàng tiện lợi dựa trên kiến trúc mạng học sâu YOLO.

Thứ hai nhúng mô hình nhận diện OOS và IS lên các nền tảng thiết bị nhúng và xử lý dữ liệu thu được sau đó đưa ra cảnh báo đến các nhân viên tại cửa hàng về tình trạng OOS của kệ hàng dựa trên phần cứng Jetson Nano và thư viện OPENCV DNN.

1.3 PHƯƠNG PHÁP NGHIÊN CỨU

Mô hình ứng dụng mạng học sâu nhằm nhận diện các vị trí OOS và IS trên kệ hàng sẽ được nghiên cứu và xây dựng dựa trên các vấn đề như sau:

- Nghiên cứu và ứng dụng kiến trúc mạng YOLO phù hợp để huấn luyện mô hình nhận diện với độ chính xác cao.
- Nghiên cứu về các nền tảng thiết bị nhúng chuyên dụng cho xử lý ảnh và AI cụ thể là phần cứng Jetson Nano nhằm ứng dụng kết quả huấn luyện từ YOLO và xử lý trên phần cứng Jetson Nano sử dụng Python và thư viện OPENCV DNN.

1.4 BỐ CỤC TRÌNH BÀY

Chương 1 nêu ra vấn đề thực tế về tình trạng OOS tại các cửa hàng và hậu quả của tình trạng này đem lại cho các doanh nghiệp kinh doanh hàng hóa bán lẻ và đưa ra giải pháp khắc phục dựa trên mô hình học sâu.

Chương 2 trình bày về mô hình kiến trúc mạng YOLO, tổng quan về phần cứng Jetson Nano và thư viện OPENCV DNN dùng để triển khai mô hình huấn luyện từ mạng YOLO lên nền tảng phần cứng của Jetson Nano.

Chương 3 trình bày về thiết kế mô hình nhận diện các vị trí OOS và IS trên kệ hàng tại các siêu thị hoặc cửa hàng tiện lợi, quy trình về các bước để chuẩn bị để huấn luyện mô hình học sâu dựa trên kiến trúc mạng YOLO và thiết lập cài đặt hệ điều hành và môi trường cần thiết trên phần cứng Jetson Nano.

Chương 4 trình bày về việc triển khai mô hình và thực nghiệm đánh giá lại kết quả thu được.

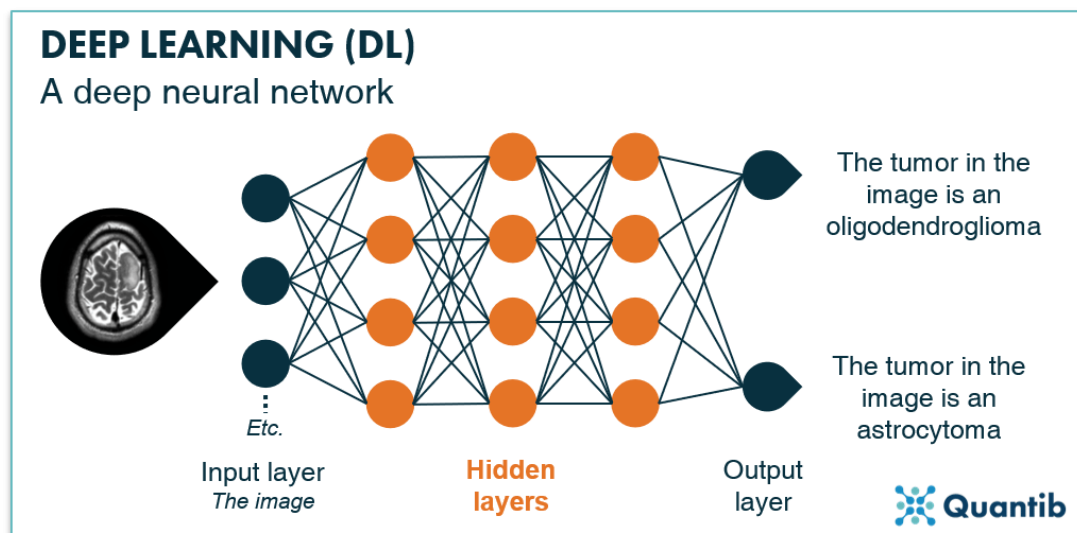
Chương 5 tổng kết và đề xuất định hướng phát triển của đề tài.

CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

2.1 HỌC SÂU TRONG THỊ GIÁC MÁY TÍNH

Trong thời đại hiện nay, khả năng trí tuệ nhân tạo được cung cấp bởi các hệ thống thông minh dựa trên máy học (ML - Machine Learning) và học sâu (DL - Deep Learning) đang ngày càng phổ biến. Máy học miêu tả khả năng của các hệ thống học tập từ lượng dữ liệu đào tạo để tự động hóa các quá trình phân tích mô hình và giải quyết các công việc liên quan [8]. Học sâu là một khái niệm dựa trên mạng Nơ-Ron nhân tạo và cho phép mô hình hóa các ứng dụng có yêu cầu cao hơn bằng cách sử dụng các cấu trúc mạng Nơ-Ron phức tạp hơn so với các mô hình học máy thông thường.



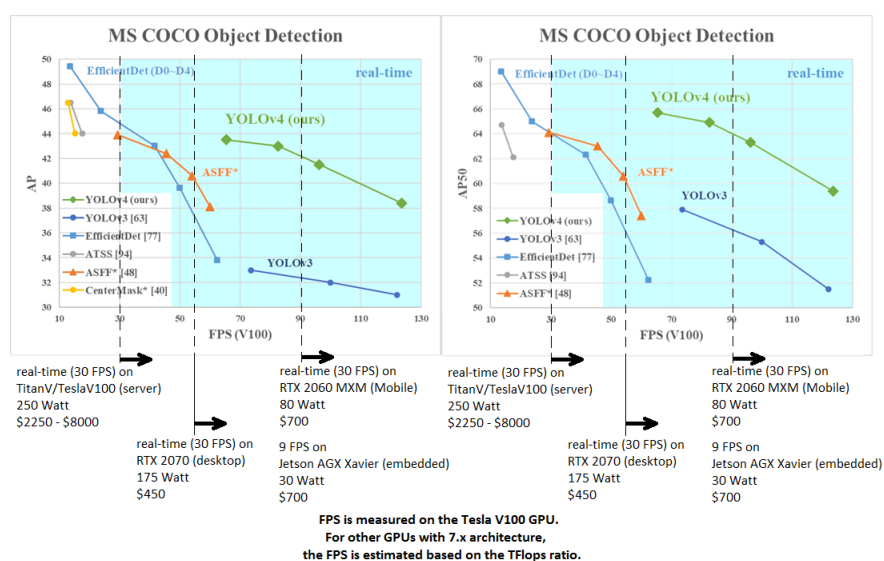
Hình 2. 1: Hình ảnh minh họa về Deep learning (nguồn Internet)

Học sâu là một phương pháp thuộc trong lĩnh vực học máy, tuy nhiên điểm khác biệt quan trọng của học sâu so với các phương pháp khác là sử dụng mạng Nơ-Ron nhân tạo để tự động hóa việc đào tạo. Bằng cách sử dụng số lượng lớn dữ liệu đầu vào, học sâu có khả năng đào tạo mô hình tự động và tạo ra kết quả chính xác hơn so với phương pháp truyền thống của học máy.

2.2 MẠNG HỌC SÂU YOLO

2.2.1 Giới thiệu về mạng học sâu YOLO

Khi chúng ta nhìn vào một bức ảnh hay xem một đoạn video, chúng ta có thể dễ dàng nhận ra các đối tượng có trong đó. Để phát triển các ứng dụng trong các lĩnh vực như giám sát an ninh, truy xuất hình ảnh, lái xe tự động, chúng ta cần áp dụng trí thông minh nhân tạo vào việc nhận diện các đối tượng trong hình ảnh. Nền tảng YOLO (You Only Look Once) [6] đã được phát triển để giải quyết các vấn đề này và đã được kiểm chứng trong nhiều năm. Mô hình YOLO cung cấp các kết quả dự đoán chính xác và hiện có nhiều phiên bản khác nhau phù hợp với các ứng dụng khác nhau.



Hình 2. 2: So sánh FPS mạng YOLO (nguồn Internet)

2.2.2 Một số thuật ngữ cơ bản

Để đánh giá được chất lượng của một mô hình mạng YOLO được huấn luyện cụ thể cần dựa trên các thông số để đánh giá và so sánh. Dưới đây sẽ trình bày về một số thông số cơ bản và quan trọng trong việc đánh giá mô hình huấn luyện mạng YOLO.

Hàm tính IOU (IOU selection threshold)

Hàm tính IOU sẽ tính toán tỉ lệ giữa diện tích phần giao chia cho diện tích phần hợp giữa hộp giới hạn được mô hình dự đoán ra và hộp giới hạn đã được khoanh vùng trước đó.

$$IOU = \frac{S_{A \cap B}}{S_{A \cup B}}$$

Công thức trên minh họa cho việc tính tỉ số IOU giữa hai hộp giới hạn, hộp giới hạn được gắn nhãn sẵn và hộp giới hạn được phát hiện bởi mô hình.

Hàm lỗi (Loss function)

Hàm lỗi (Loss function) trong YOLO tính thông qua việc dựa trên việc dự đoán của mô hình và các nhãn được gán để đánh giá.

- **Hàm lỗi vị trí (Localization loss):** Được dùng để tính toán giá trị lỗi của vị trí trung tâm của hộp giới hạn được dự đoán so với vị trí trung tâm của hộp giới hạn được đặt trước của đối tượng. Hàm lỗi vị trí dự đoán tọa độ trung tâm so với nhãn dự đoán thật được tính theo công thức sau:

$$L_{coord} = \sum_{i=0}^{S^2} \sum_{j=0}^B \Pi_{ij}^{obj} (x_i - x'_i)^2 + (y_i - y'_i)^2$$

Hàm lỗi giá trị dự đoán chiều dài và chiều rộng của hộp giới hạn được tính theo công thức sau:

$$L_{coord} = \sum_{i=0}^{S^2} \sum_{j=0}^B \Pi_{ij}^{obj} (\sqrt{w_i} - \sqrt{w'_i})^2 + (\sqrt{h_i} - \sqrt{h'_i})^2$$

-Hàm lỗi dự đoán (Confidence loss): Dùng để tính toán độ lỗi dự đoán của hộp giới hạn có chứa đối tượng được dự đoán so với nhãn được gán thực tế. Hàm lỗi dự đoán được tính theo công thức như sau:

$$L_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^B \Pi_{ij}^{obj} (C_i - C'_i)^2 + \lambda_{noobject} \sum_{i=0}^{S^2} \sum_{j=0}^B \Pi_{ij}^{obj} (C_i - C'_i)^2$$

Trong đó $\lambda_{noobject}$ là trọng số thành phần.

-Hàm lỗi phân loại (Classification loss): Dùng để tính toán độ lỗi khi dự đoán loại nhãn của đối tượng. Hàm lỗi phân loại được tính dựa theo công thức sau:

$$L_{classification} = \sum_{i=0}^{S^2} \Pi_i^{obj} \sum_{C \in class} (P_i(c) - P'_i(C))^2$$

Trong đó Π_i^{obj} sẽ bằng 1 nếu hộp giới hạn đang xét có chứa đối tượng cần phát hiện. $P_i(c)$ đại diện cho xác suất của lớp đối tượng C tại hộp giới hạn mà mô hình dự đoán được.

-Hàm lỗi tổng hợp (Total loss): Hàm lỗi tổng hợp được tính bằng tổng của 3 hàm lỗi dự đoán, hàm lỗi vị trí và hàm lỗi phân loại.

$$L_{total} = L_{classification} + L_{localization} + L_{confidence}$$

Tỉ lệ dự đoán (Precision)

Tỉ lệ dự đoán (Precision) được định nghĩa bằng tổng số nhãn dự đoán đúng trên cho tổng số nhãn dự đoán.

$$precision = \frac{TF}{TF + FP}$$

Trong đó TF (true positive) là tổng số nhãn mà mô hình dự đoán đúng là tích cực, FP (false positive) là tổng số nhãn mà mô hình dự đoán sai là tích cực. Tỷ lệ dự đoán (precision) sẽ thường được lựa chọn ưu tiên đối với các ứng dụng khi mà việc nhận diện nhầm các điểm FP sẽ mang lại ảnh hưởng lớn đến ứng dụng.

Recall

Tỷ lệ này cũng tương tự như tỷ lệ dự đoán (precision), recall cho thấy được tỷ lệ bỏ sót các điểm TP (true positive). Công thức của tỷ lệ recall được tính như sau:

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

Trong đó TP (true positive) là số dự đoán nhãn đúng tích cực, FN (false negative) là số dự đoán nhãn sai là tiêu cực.

F1-Score

Tỷ lệ F1-Score được lựa chọn khi mô hình cần cả hai Precision và Recall cao vì vậy tỷ số F1-Score sẽ giúp cân bằng giữa hai tham số này khi lựa chọn và đánh giá mô hình.

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Độ chính xác trung bình mAP (mean Average Precision)

Độ chính xác trung bình mAP (mean Average Precision) dùng để đánh giá trung bình độ chính xác của các lớp đối tượng. mAP sẽ được tính bằng tổng trung bình tỷ lệ dự đoán AP của các lớp đối tượng mà mô hình phát hiện.

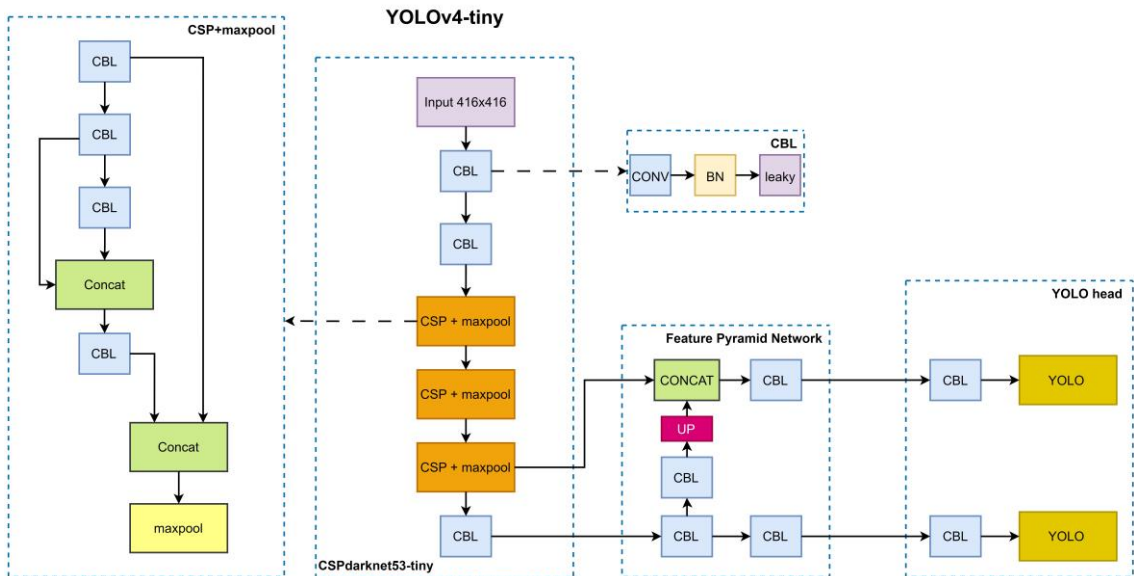
$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

Trong đó AP là tỷ lệ dự đoán chính xác đối với một lớp đối tượng, N là tổng số lớp đối tượng được dự đoán.

2.2.3 Giới thiệu kiến trúc mạng YOLOv4-tiny

YOLOv4-tiny là kiến trúc mạng được tối ưu hóa từ kiến trúc YOLOv4 trước đó. YOLOv4 sử dụng lớp backbone được kế thừa từ kiến trúc mạng CSP-darknet53 và từ đó YOLOv4-tiny đã rút gọn kiến trúc lớp backbone này lại và có tên là CSP-darknet53-tiny. Lớp backbone sẽ đảm nhận nhiệm vụ trích xuất các đặc trưng của hình ảnh đầu vào bằng các lớp tích chập 3×3 và 1×1 . Tại lớp neck của YOLOv4-tiny sử dụng kiến trúc FPN (Feature Pyramid Network) là một lớp dùng để trích xuất các đặc trưng của ảnh nhằm mục đích đạt được độ chính xác, FPN có tốc độ được cải thiện hơn so với các mạng khác như Faster R-CNN tuy nhiên độ chính xác không bằng so với mạng Faster R-CNN.

YOLOv4-tiny với lớp backbone CSP-darknet53-tiny với ảnh đầu vào 416×416 và được đưa về hai tỉ lệ 13×13 và 26×26 đi ra lớp YOLO để dự đoán. Độ phức tạp tính toán của mạng YOLOv4-tiny rơi vào khoảng 6700 BFLOPS.

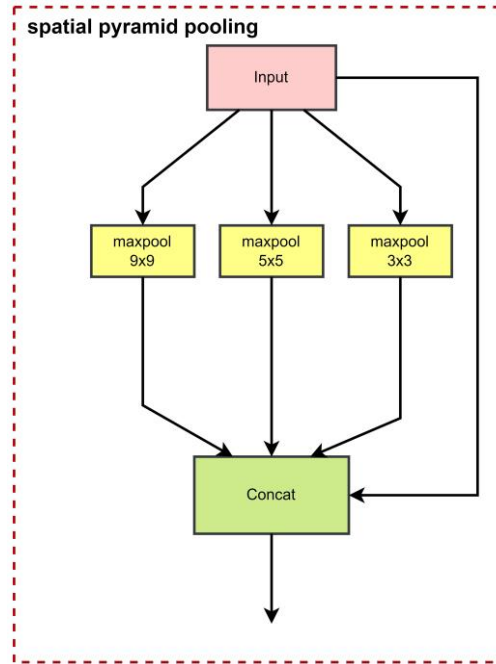


Hình 2. 3: Kiến trúc chi tiết mạng YOLOv4-tiny

Mạng YOLOv4-tiny sử dụng lớp CBL với hàm kích hoạt là leaky tại lớp backbone. Kiến trúc lớp backbone sử dụng kiến trúc CSP-darknet53-

tiny bao gồm ba lớp tích chập và ba lớp CSP đi kèm với lớp *maxpool*. Lớp neck sử dụng kiến trúc của FPN và đưa ra hai tỉ lệ hình 13×13 và 26×26 tới lớp head để dự đoán.

2.2.4 Lớp Phân bổ chồng không gian (Spatial pyramid pooling)



Hình 2. 4: Spatial pyramid pooling

Các lớp tích chập (convolution) sẽ có thể chấp nhận kích thước ảnh đầu vào với bất kỳ kích cỡ nào. Tuy nhiên lớp phân loại như SVM/softmax hoặc là lớp kết nối đầy đủ (fully connected) đều yêu cầu kích thước ảnh đầu vào qua các lớp này phải là một kích thước véc tơ cố định kích cỡ [1]. Các véc tơ như vậy có thể được tạo ra bằng phương pháp Bow (Bag-of-words) để cho các đặc trưng của ảnh có thể kết hợp lại với nhau [1]. Lớp phân bổ chồng không gian (Spatial pyramid pooling) có thể được áp dụng để tăng cường Bow bằng cách sử dụng ba lớp *maxpool* trích xuất đặc trưng của ảnh đầu vào với ba kích thước mặt nạ khác nhau bao gồm 9×9 , 5×5 , 3×3 . Sau đó các đặc trưng sẽ được tổng hợp lại thông qua lớp nối liền (concatenate), ví dụ với kích thước ảnh đầu vào có $H \times W \times C$ với H là độ cao, W là độ rộng và C là số đặc trưng của ảnh khi đi qua lớp SPP ta sẽ

được kết quả đầu ra với $H \times W \times 4C$. Dưới đây là biểu thức biểu thị cho đầu vào và đầu ra của lớp SPP.

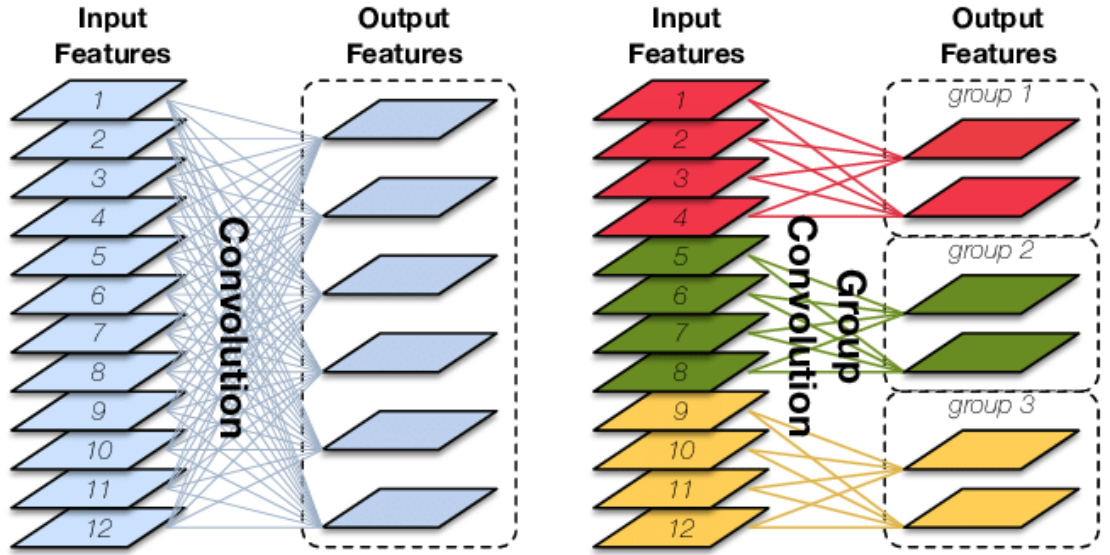
$$F_{mp9 \times 9} = F_{\max pool 9 \times 9}(F_{in})$$

$$F_{mp5 \times 5} = F_{\max pool 5 \times 5}(F_{in})$$

$$F_{mp3 \times 3} = F_{\max pool 3 \times 3}(F_{in})$$

$$F_{out} = F_{concat}(F_{mp9 \times 9}, F_{mp5 \times 5}, F_{mp3 \times 3}, F_{in})$$

2.2.5 Nhóm các lớp tích chập (Group convolutions)



Hình 2. 5: Group convolutions (nguồn Internet)

Hiệu quả của nhóm các lớp tích chập cho phép nén mô hình xuống gấp đôi hoặc nhiều hơn nữa bằng việc chia nhỏ các đặc trưng thành các nhóm nhỏ từ đó có thể giảm tối thiểu được độ phức tạp của mô hình mà vẫn giữ được độ chính xác của mô hình ở mức chấp nhận được. Với việc chia việc tính toán cho từng lớp tích chập thành các nhóm nhỏ sẽ giúp giảm đi số lượng phép tính BFLOPs đáng kể trong mô hình mạng. Công thức tính số lượng tham số của nhóm các lớp tích chập được tính thông độ dài, độ rộng và độ sâu của lớp tích chập. Với G là số nhóm các lớp tích chập được chia

ra, có thể thấy số lượng tham số sẽ giảm G lần so với lớp tích chập thông thường.

$$PARAM_{conv} = H \times W \times Din \times Dout$$

$$PARAM_{GroupConv} = H \times W \times \frac{Din}{G} \times \frac{Dout}{G} \times G$$

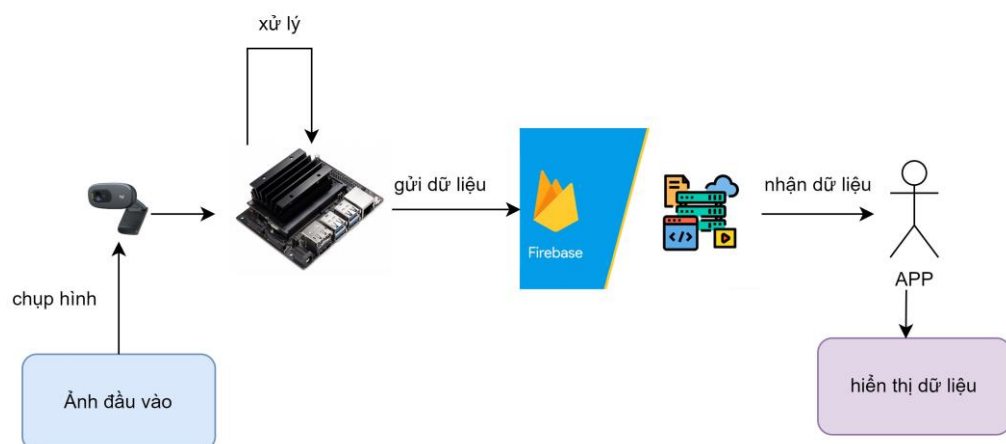
CHƯƠNG 3

MÔ HÌNH ỨNG DỤNG

3.1 THIẾT KẾ MÔ HÌNH ỨNG DỤNG

3.1.1 Tổng quan mô hình

Mô hình ứng dụng học sâu dựa trên mô hình mạng YOLO để phát hiện các vị trí OOS và các vị trí IS sẽ thực hiện 2 nhiệm vụ chính, một là hệ thống sẽ nhận dữ liệu đầu vào từ máy ảnh được đặt tại các siêu thị hoặc các cửa hàng tạp hóa quy mô lớn, hai là hệ thống sẽ tiến hành xử lý và phát hiện ra các vị trí OOS và IS trên hình ảnh đầu vào và gửi dữ liệu lên dịch vụ lưu trữ đám mây. Đồng thời xây dựng giao diện hiển thị kết nối với dịch vụ lưu trữ đám mây để hiển thị các thông số như số vị trí OOS, số vị trí IS trên kệ đồng thời đưa ra dữ liệu về tình trạng của kệ hàng như tỉ lệ hàng hiện có trên kệ.

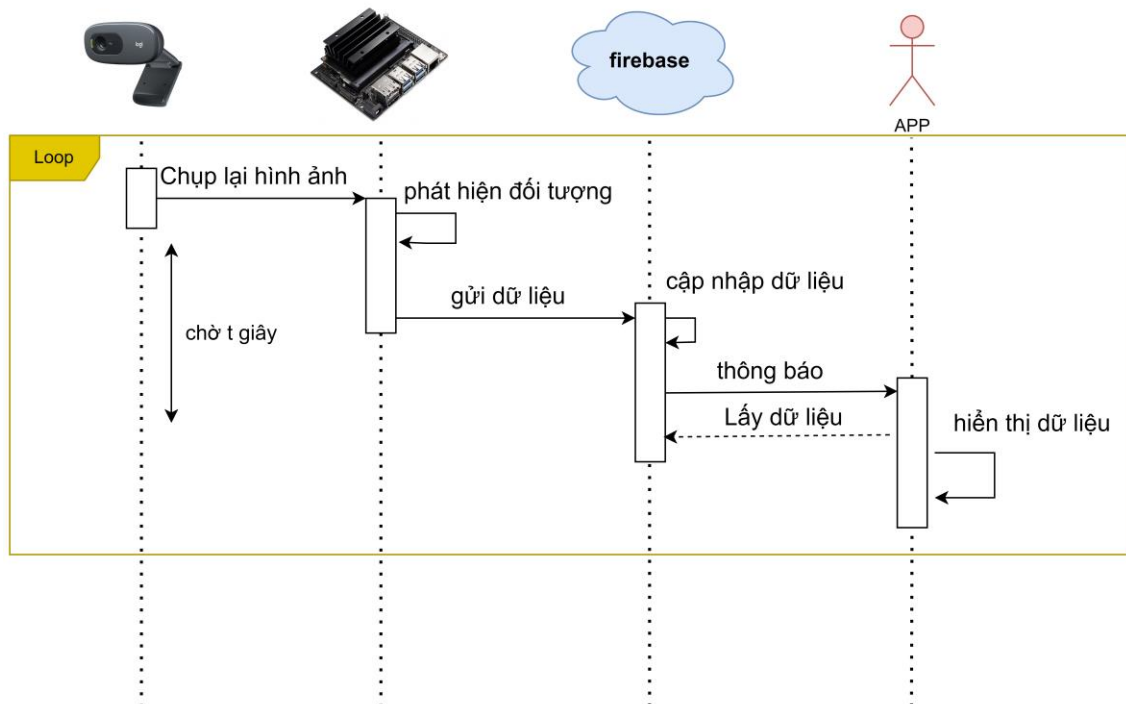


Hình 3. 1: Mô hình hệ thống giám sát OOS và IS

3.1.2 Nguyên lý hoạt động

Nguyên lý hoạt động của mô hình hệ thống ứng dụng học sâu dựa trên mô hình mạng YOLO để phát hiện các vị trí OOS và các vị trí IS có trình tự hoạt động dựa trên bốn thành phần chính bao gồm:

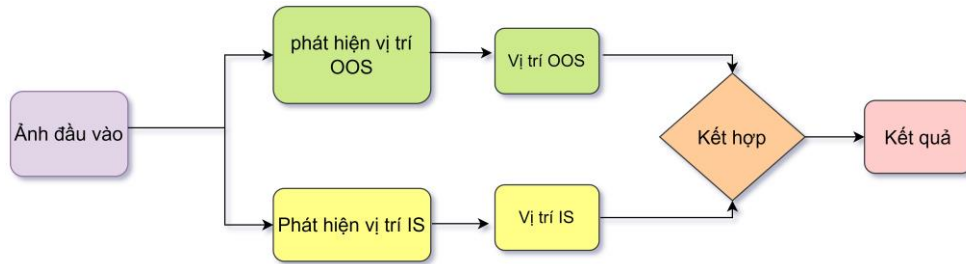
- Thành phần thu thập dữ liệu (Máy ảnh)
- Thành phần xử lý dữ liệu (Máy tính nhúng Jetson Nano)
- Thành phần lưu trữ (Dịch vụ lưu trữ đám mây thời gian thực firebase)
- Thành phần hiển thị (Giao diện ứng dụng)



Hình 3. 2: Sơ đồ trình tự hoạt động hệ thống

3.2 THIẾT KẾ SƠ ĐỒ THUẬT TOÁN XỬ LÝ TRÊN JESON NANO

3.2.1 Mô hình kết hợp xử lý phát hiện phân loại OOS và IS



Hình 3. 3: Sơ đồ khối mô hình xử lý phát hiện vị trí OOS và IS

Hình ảnh đầu vào được chụp từ máy ảnh sẽ được đưa vào độc lập lần lượt hai mô hình phát hiện OOS và IS được huấn luyện dựa trên kiến trúc mạng YOLO và đưa ra hai kết quả lần lượt là tọa độ các vị trí OOS và IS trên ảnh. Từ đó có thể phân loại và nhận biết được vị trí OOS và vị trí IS trên ảnh. Sau đó tiến hành kết hợp hai kết quả này lại và vẽ các hộp giới hạn lên hình ảnh đầu vào thu được từ trước và phân loại dựa theo kết quả thu được từ hai mô hình phát hiện trên sẽ cho ra được kết quả hình ảnh có đánh dấu các vị trí OOS và IS trên cùng một ảnh.

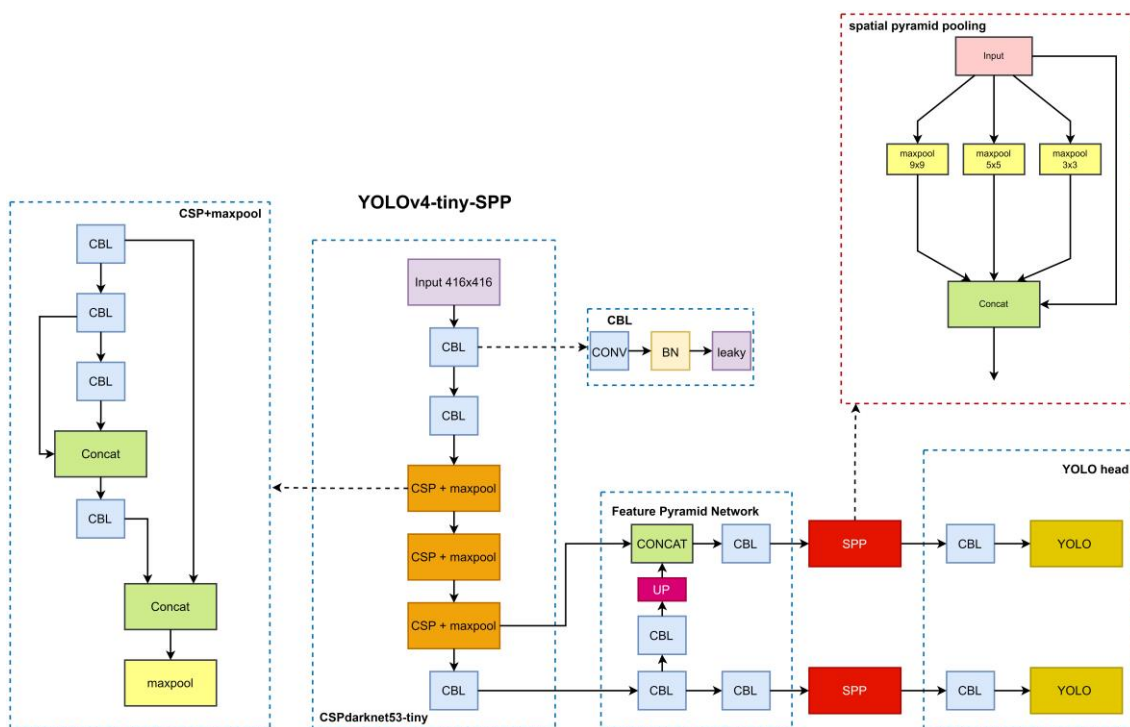
3.2.2 Đề xuất kiến trúc YOLOv4-tiny-SPP

Kiến trúc hệ thống được thiết kế trong đề tài này hiện tại không yêu cầu quá nhiều về tốc độ xử lý, mặc khác lại yêu cầu về độ chính xác khi nhận diện vật thể sẽ được chú trọng. Mặc dù kiến trúc mô hình YOLOv4-tiny đã được cộng đồng người dùng kiểm chứng với độ chính xác cao và tốc độ tính toán được tối ưu, nhưng nó chưa được tối ưu cho mỗi ứng dụng cụ thể và các tệp dữ liệu huấn luyện có đặc điểm riêng biệt. Để tối ưu hóa các thông số mong muốn khi huấn luyện mô hình, đề tài này đề xuất mô hình YOLOv4-tiny-SPP dành cho việc phát hiện vị trí OOS. bao gồm 30.000 ảnh huấn luyện đã được áp dụng các kỹ thuật tăng cường dữ liệu để làm tăng mức độ phát hiện toàn diện của mô hình.

Mô hình YOLOv4-tiny-SPP sẽ được tăng cường dựa trên kiến trúc của mô hình YOLOv4-tiny với việc tăng cường trước các lớp phát hiện vật

thể bằng các kiến trúc SPP (Spatial Pyramid Pooling) sẽ giúp tăng cường các đặc trưng của hình ảnh đầu vào trước các lớp kết nối đầy đủ trong mô hình YOLOv4-tiny.

Lớp SPP sẽ giúp cho mô hình trích xuất được nhiều đặc trưng của đối tượng thông qua ba lớp *maxpool* với ba kích thước khác nhau bao gồm 9×9 , 5×5 và 3×3 trước khi đi vào lớp kết nối đầy đủ (fully connected). Với ảnh đầu vào có kích thước $H \times W \times C$ sau khi đi qua 3 lớp *maxpool* kết nối song song và được tổng hợp lại ở lớp kết nối *Concatenation* như hình 3.4 thì đầu ra tương ứng sẽ là $H \times W \times 4C$. Như vậy có thể thấy số lượng tính chất đặc trưng của ảnh đầu vào đã được tăng lên 4 lần khi đi qua lớp SPP trước khi đi vào lớp kết nối đầy đủ.



Hình 3. 4: Kiến trúc YOLOv4-tiny-SPP

Kiến trúc YOLOv4-tiny-SPP được đề xuất sẽ tăng cường độ chính xác dựa trên kiến trúc YOLOv4-tiny với việc thêm vào 2 lớp SPP tại trước hai lớp *YOLO head*. Tăng cường với hai lớp SPP sẽ mang lại độ nhận diện chính xác của mô hình cao hơn so với kiến trúc YOLOv4-tiny nhờ vào kiến

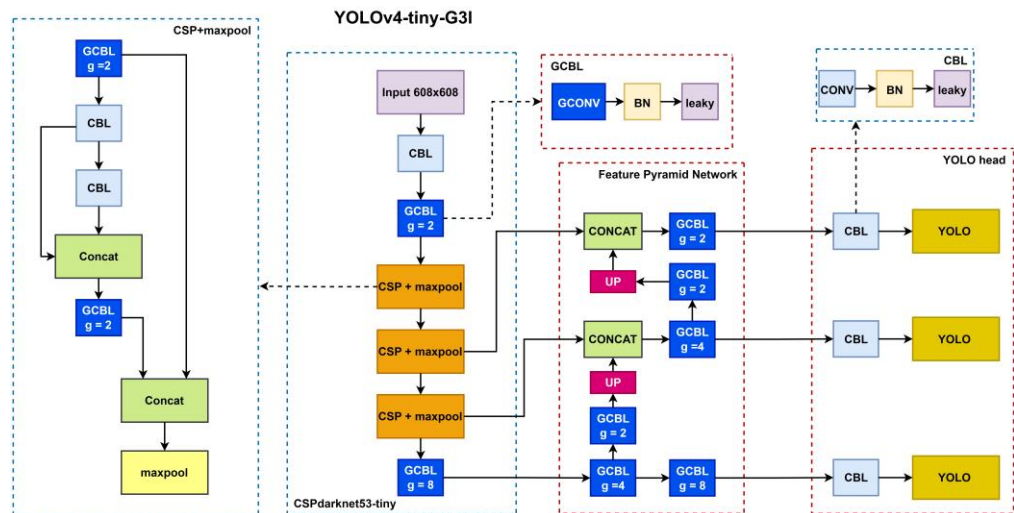
trúc SPP giúp trích xuất thêm nhiều đặc trưng ảnh hơn trước khi đi qua lớp nhận diện *YOLO head*. Tuy nhiên kiến trúc YOLOv4-tiny-SPP sẽ có số lượng phép tính lớn hơn so với YOLOv4-tiny nhưng điều này cũng không đáng kể và ảnh hưởng đến độ ổn định của hệ thống. Bảng dưới đây sẽ đưa ra so sánh giữa YOLOv4-tiny và YOLOv4-tiny-SPP.

Bảng 3. 1: So sánh BFLOPS giữa YOLOv4-tiny và YOLOv4-tiny-SPP

	BFLOPS
YOLOv4-tiny	6.787
YOLOv4-tiny-SPP	6.856

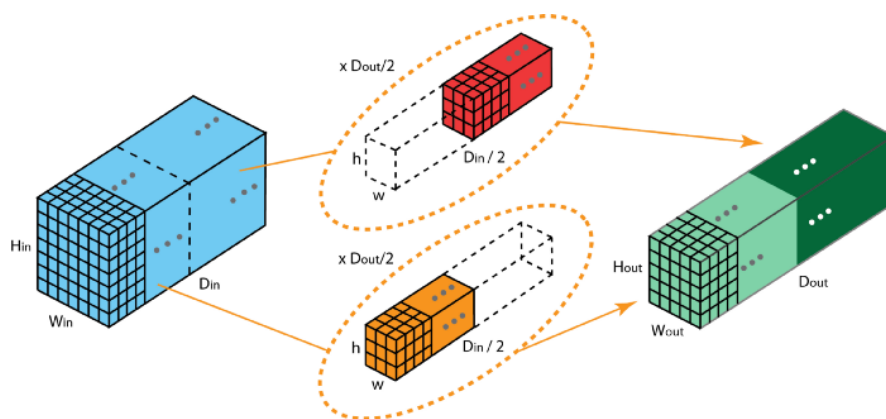
3.2.3 Đề xuất kiến trúc YOLOv4-tiny-G3l

Trong bài toán tìm kiếm vật thể độ sắc nét của hình ảnh đầu vào sẽ ảnh hưởng rất lớn tới độ chính xác của mô hình đặc biệt là đối với các vật thể nhỏ, việc tăng độ phân giải hình ảnh đầu vào sẽ đem lại được hiệu quả đáng kể về độ chính xác của mô hình. Tuy nhiên tăng kích thước hình ảnh đầu vào phải đánh đổi với lượng tham số của mô hình tăng lên đáng kể. Trong đề tài này sẽ đề xuất kiến trúc YOLOv4-tiny-G3l dựa trên kiến trúc YOLOv4-tiny với kích thước ảnh đầu vào tăng từ 416×416 lên 608×608 . Kiến trúc YOLOv4-tiny-G3l sẽ thêm một lớp *YOLO head* với mục đích tăng cường lớp phát hiện ở nhiều độ phân giải ảnh khác nhau.



Hình 3. 5: kiến trúc YOLOv4-tiny-G3l

Kiến trúc này sẽ áp dụng để huấn luyện cho tập dữ liệu bao gồm 9.000 ảnh phát hiện các mặt hàng trên kệ với đặc điểm chung là các vật thể cần phát hiện có kích thước nhỏ vì vậy với kiến trúc YOLOv4-tiny-G3l sẽ cho ra được độ chính xác của mô hình cao hơn so với YOLOv4-tiny. Tuy nhiên việc tăng kích thước hình ảnh đầu vào thì số lượng phép tính mà mô hình cần tính toán tăng lên đáng kể, do đó để tối ưu hóa được kích thước của mô hình, kiến trúc YOLOv4-tiny-G3l sẽ thay thế các lớp CNN thông thường bằng các nhóm CNN để giảm tối đa số lượng phép tính BFLOPs mà vẫn giữ được độ chính xác của mô hình ở mức kỳ vọng.



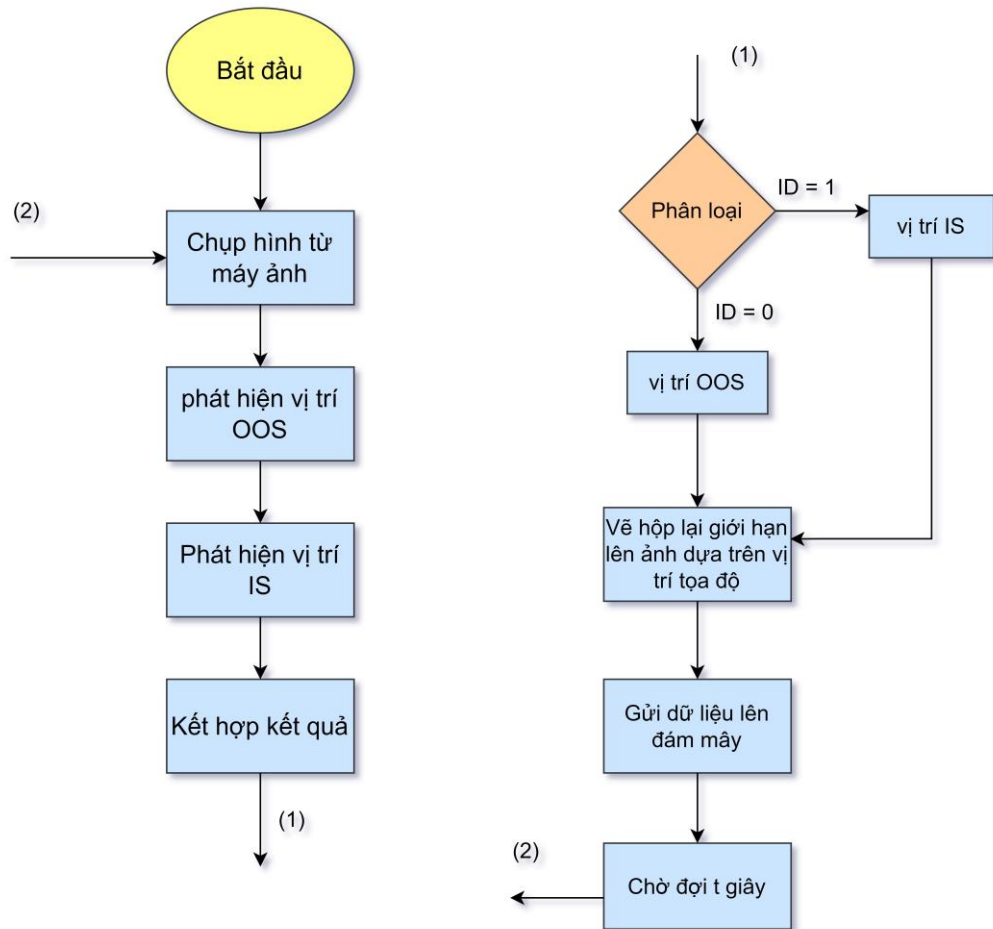
Hình 3. 6: Nhóm các lớp tích chập (nguồn Internet)

Nhóm các lớp tích chập (group convolutions) có ý tưởng thay vì nhân toàn bộ mặt nạ với toàn bộ các kênh như lớp tích chập thông thường thì Nhóm các lớp tích chập sẽ chia tách các đặc trưng ra thành nhiều nhóm nhỏ và nhân tích chập với từng lớp tích chập của từng nhóm sau đó đi qua lớp tổng hợp (Concatenate) để tổng hợp các đặc trưng lại với nhau. Điều này giúp cho mô hình sẽ giảm đi đáng kể số lượng phép tính BFLOPs. Đối với lớp tích chập truyền thống số lượng tham số sẽ được tính thông qua $H \times W \times Din \times Dout$, tuy nhiên đối với nhóm các lớp tích chập thì được tính bằng $H \times W \times \frac{Din}{G} \times \frac{Dout}{G} \times G$. Như vậy có thể thấy khi áp dụng nhóm các lớp tích chập vào kiến trúc YOLOv4-tiny-G3l sẽ giảm tham số đi G lần so với dùng lớp tích chập truyền thống.

Bảng 3. 2: So sánh BFLOPS giữa YOLOv4-tiny và YOLOv4-tiny-G3l

	BFLOPS
YOLOv4-tiny	6.787
YOLOv4-tiny-G3l	7.001

3.2.4 Sơ đồ thuật toán



Hình 3. 7: Sơ đồ thuật toán xử lý trên Jetson Nano

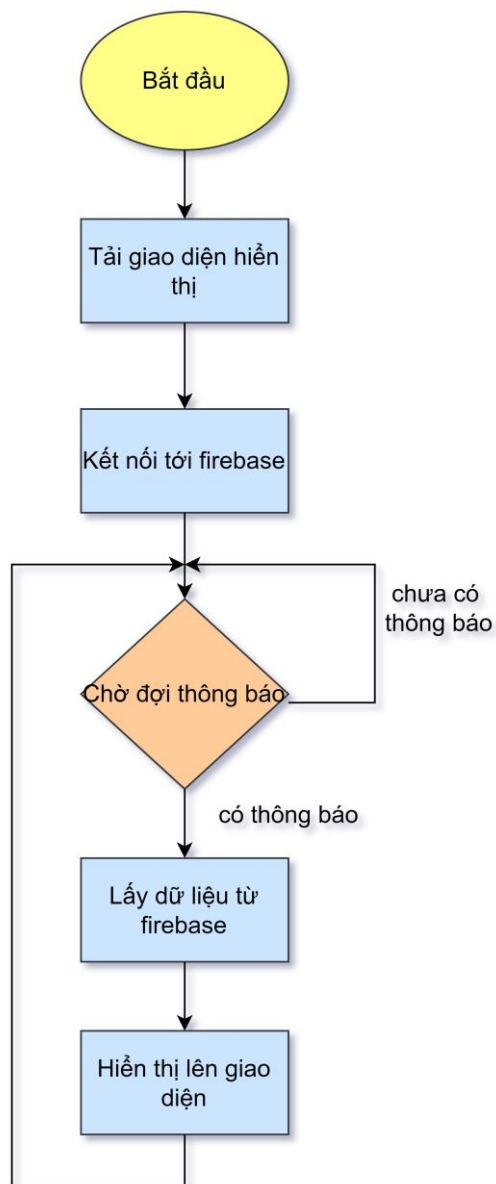
Tại thời điểm bắt đầu thành phần xử lý (máy tính nhúng Jetson Nano) sẽ nhận hình ảnh đầu vào từ thành phần thu thập dữ liệu (máy ảnh) và tạm thời lưu lại hình ảnh để xử lý

3.3 THIẾT KẾ SƠ ĐỒ THUẬT TOÁN XỬ LÝ TRÊN GIAO DIỆN

3.3.1 Sơ đồ thuật toán

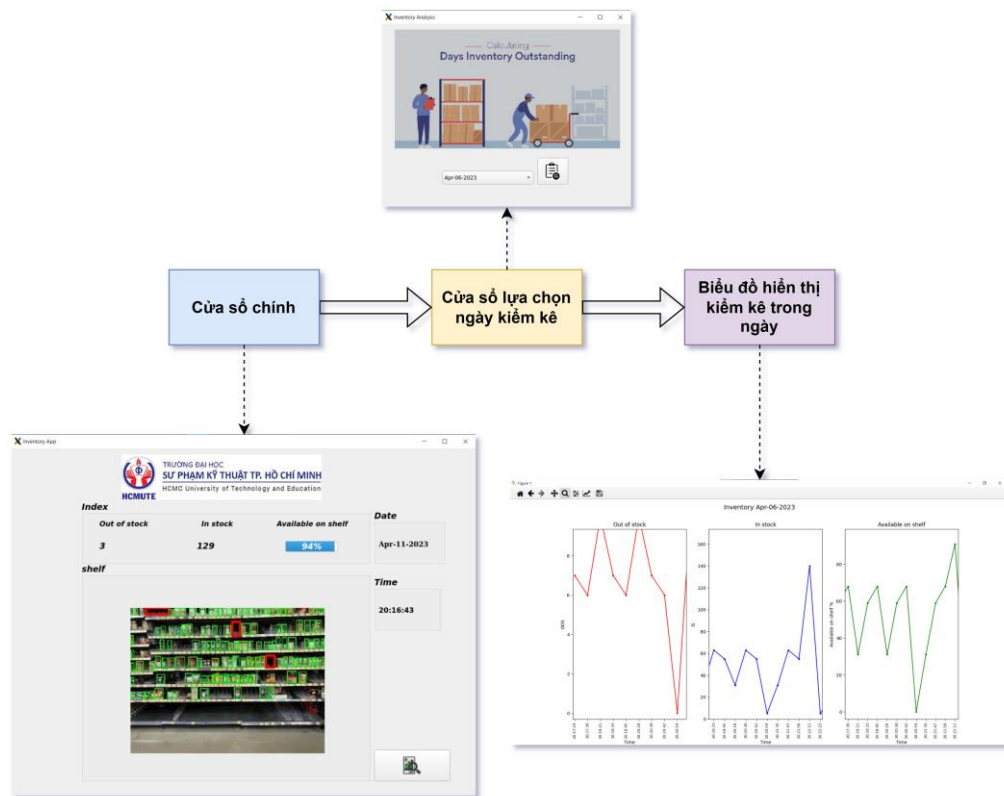
Trong đề tài này sẽ sử dụng phần mềm thiết kế giao diện QT để triển khai thiết kế giao diện hiển thị các thông số OOS, IS và thông số tỉ lệ hàng có sẵn trên kệ dựa trên dữ liệu được lưu trữ tại dịch vụ lưu trữ đám mây thời gian thực cùng với hình ảnh đã được vẽ các hộp giới hạn chỉ định phân

biệt hai vị trí OOS và IS để hiển thị lên giao diện giúp cho các nhân viên tại các cửa hàng tiện lợi hoặc các siêu thị có thể giám sát tình hình hàng hóa trên kệ hàng từ xa và chủ động giải quyết vấn đề OOS một cách sớm nhất có thể.



Hình 3. 8: Sơ đồ thuật toán trên giao diện hiển thị

3.3.2 Sơ đồ khối giao diện hiển thị



Hình 3. 9: Sơ đồ khối giao diện hiển thị

Giao diện hiển thị trong đề tài sẽ được thiết kế để giám sát các kệ hàng cũng như các chỉ số về OOS và IS. Sơ đồ giao diện hiển thị sẽ gồm một giao diện chính và hai giao diện phụ dùng để kiểm kê và vẽ biểu đồ mối liên hệ giữa các chỉ số OOS, IS và tỉ lệ phần trăm hàng còn trên kệ theo từng ngày.

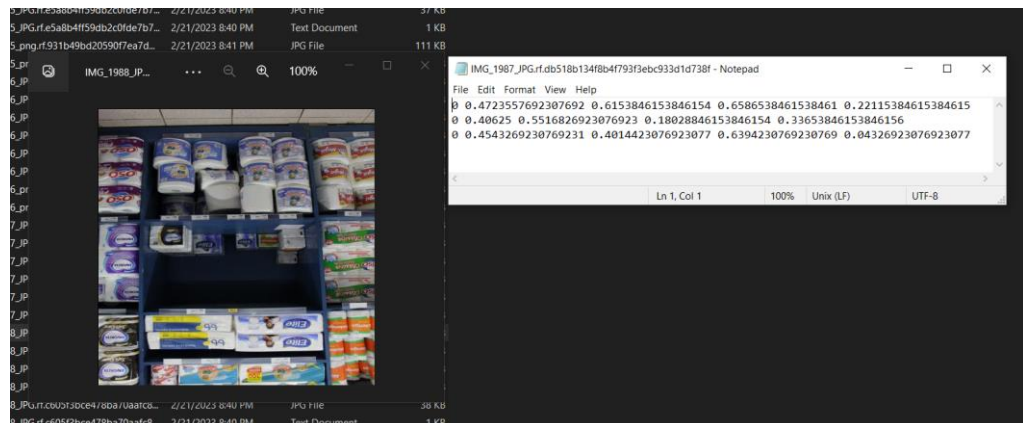
CHƯƠNG 4

THỰC HIỆN MÔ HÌNH VÀ ĐÁNH GIÁ

4.1 HUẤN LUYỆN MÔ HÌNH HỌC SÂU DỰA TRÊN KIẾN TRÚC MẠNG YOLO

4.1.1 Định dạng dữ liệu huấn luyện

Nền tảng darknet yêu cầu bộ dữ liệu đầu vào phải có định dạng là ảnh .jpg kèm với tệp .txt chứa các thông số tọa độ của các hộp giới hạn và hai tệp này phải có cùng tên với nhau. Tất cả các tệp dữ liệu này đều phải được bỏ vào trong cùng một đường dẫn **darknet/data/obj**.



Hình 4. 1: Chuẩn bị dữ liệu huấn luyện

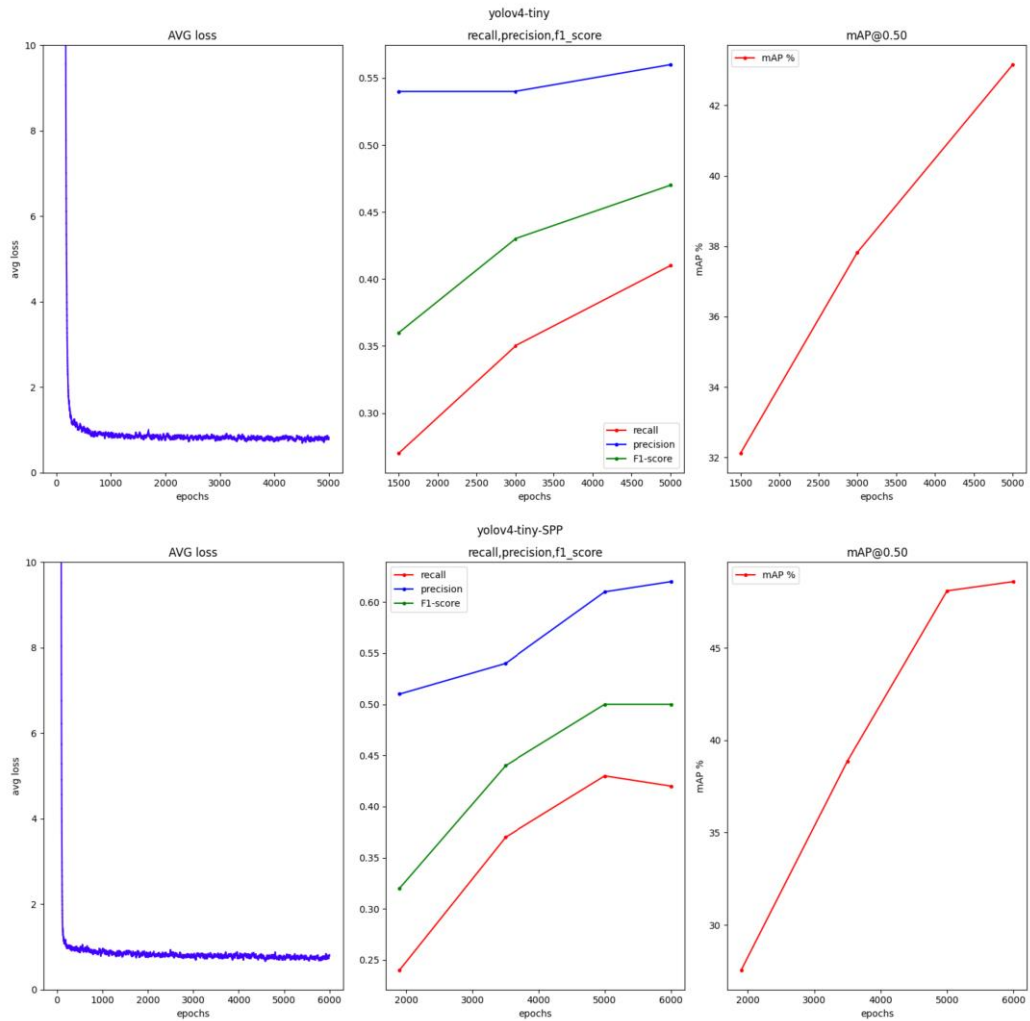
Bảng 4. 1: Tệp obj.data.txt và obj.names.txt

Obj.data.txt	Obj.names.txt
Classes = ‘số lượng lớp đối tượng’	‘Tên lớp đối tượng’
Train = ‘đường dẫn tới file train.txt’	
Valid = ‘đường dẫn tới file val.txt’	

Names = <i>‘đường dẫn tới file names.txt’</i>	
Backup= <i>‘đường dẫn tới file thư mục sao lưu’</i>	

4.1.2 Đánh giá kết quả huấn luyện

Kết quả sau khi được huấn luyện với hai lớp bao gồm lớp phát hiện OOS với số lượng ảnh là 30.000 ảnh và lớp phát hiện IS dựa trên tập ảnh với số lượng ảnh là 9.000 ảnh. Trong chương này sẽ đánh giá lại kết quả huấn luyện dựa trên các mô hình YOLOv4-tiny, YOLOv3-tiny do nhóm **AlexeyAB** phát triển và so sánh với kết quả đạt được trên hai mô hình được đề xuất trong đề tài này là YOLOv4-tiny-SPP và YOLOv4-tiny-G3l dành cho hai tập ảnh phát hiện OOS và tập ảnh phát hiện IS. Dưới đây là biểu đồ so sánh YOLOv4-tiny và YOLOv4-tiny-SPP được huấn luyện trên tập ảnh phát hiện OSS với 30.000 ảnh.



Hình 4. 2: YOLOv4-tiny và YOLOv4-tiny-SPP trên tập ảnh OSS

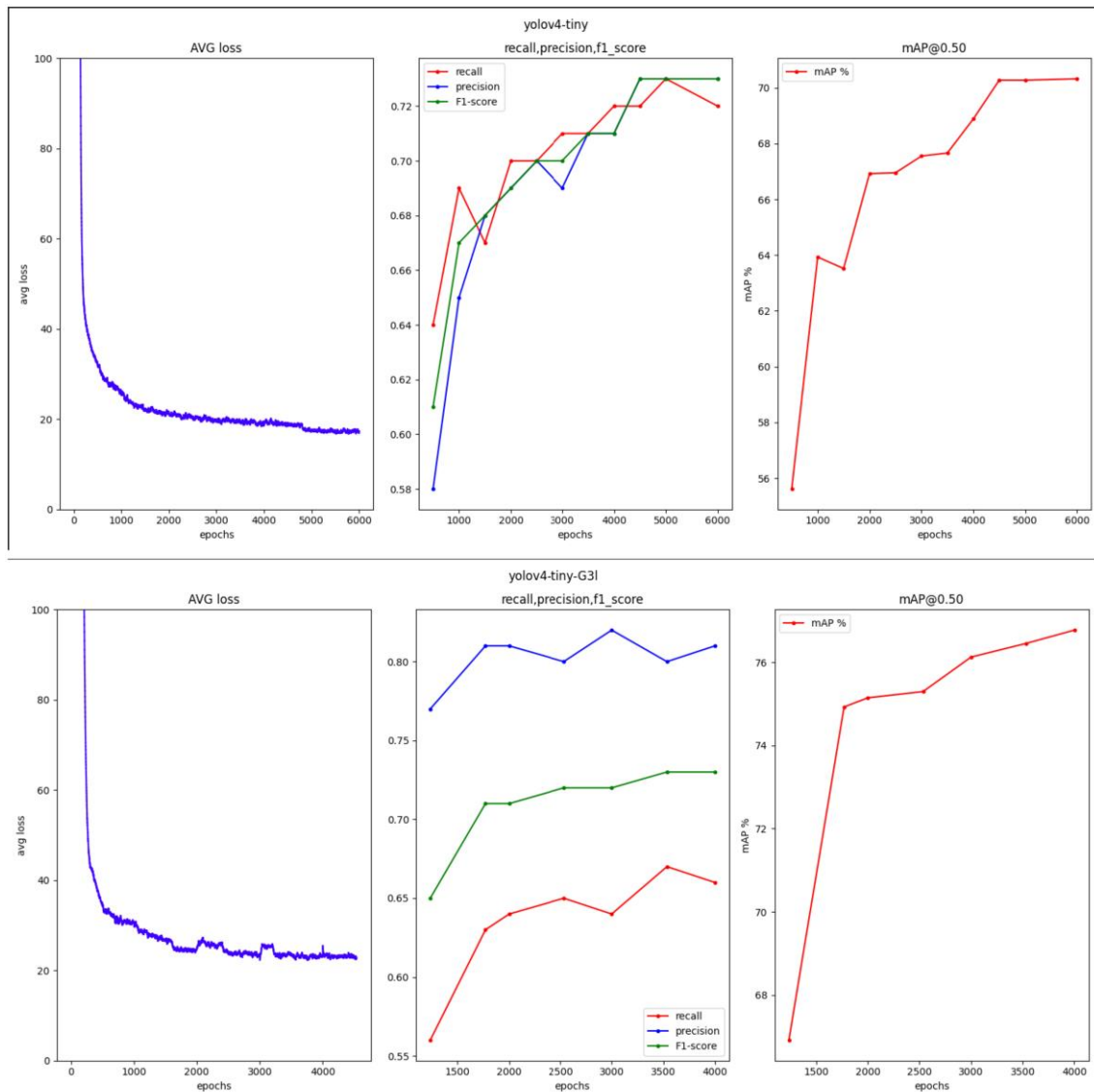
Dựa vào hình ảnh so sánh YOLOv4-tiny và YOLOv4-tiny-SPP trên tập ảnh phát hiện OSS có thể đánh giá được tỉ lệ accuracy của mô hình YOLOv4-tiny-SPP đã được cải thiện từ 41.63% lên 48.59% mặc dù độ phức tạp BFLOPs chỉ tăng 69 phép tính. Điều này cho thấy được sự hiệu quả của lớp SPP (Spatial pyramid pooling) khi được thêm vào trước lớp YOLO head đã cải thiện được accuracy so với YOLOv4-tiny mà vẫn giữ được độ phức tạp BFLOPs tăng không quá cao.

Bảng 4. 2: So sánh mAP giữa các mô hình trên tập ảnh phát hiện OOS

Kiến trúc	mAP-50	BFLOPs	Precision	Recall	F1-score	Kích thước
YOLOv4-tiny	41.63	6.787	0.56	0.41	0.47	22.4 MB
YOLOv4-tiny-SPP	48.59	6.856	0.62	0.42	0.5	22.7 MB
YOLOv3-tiny	34.95	5.560	0.63	0.26	0.37	33.0 MB

Như vậy có thể kết luận kiến trúc YOLOv4-tiny-SPP đã đem lại sự hiệu quả vượt trội so với mục tiêu của đề tài mAP đã tăng được 6.96% trong khi kích thước của mô hình chỉ tăng 0.7MB và BFLOPs chỉ tăng 69 so với kiến trúc YOLOv4-tiny.

Trên tập dữ liệu phát hiện vị trí IS bao gồm 9.000 ảnh có đặc điểm chung là các đối tượng phát hiện là các đối tượng có kích thước nhỏ vì vậy kiến trúc mô hình YOLOv4-tiny-G3l sẽ đem lại hiệu quả về accuracy trên tập dữ liệu này.



Hình 4. 3: YOLOv4-tiny và YOLOv4-tiny-G3l trên tập ảnh IS

Dựa vào hình ảnh so sánh kết quả giữa YOLOv4-tiny và YOLOv4-tiny-G3l kết quả accuracy đã tăng từ 70.32% lên 76.01%, trong khi đó để đạt được mức độ accuracy cao hơn thì cần phải đánh đổi giữa BFLOPs tăng từ 6.787 lên 7.001. Tuy nhiên, đối với ứng dụng của đề tài thì mức tăng độ phức tạp của mô hình như vậy là không đáng kể và không ảnh hưởng đến hiệu năng của hệ thống. Từ đó có thể đưa ra kết luận kiến trúc YOLOv4-tiny-G3l đã đáp ứng được yêu cầu đặt ra trong đề tài.

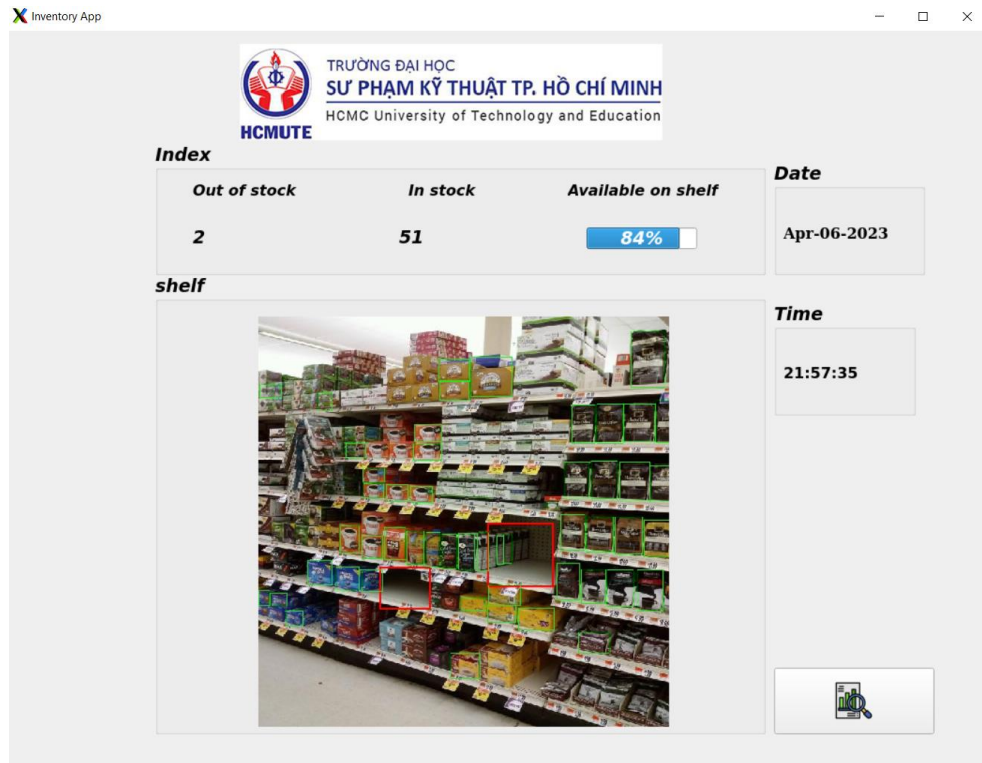
Bảng 4. 3: So sánh mAP giữa các mô hình trên tập ảnh phát hiện IS

Kiến trúc	mAP-50	BFLOPs	Precision	Recall	F1-score	Kích thước
YOLOv4-tiny	70.32	6.787	0.73	0.72	0.73	22.4 MB
YOLOv4-tiny-G3l	76.01	7.001	0.81	0.66	0.73	5.11 MB
YOLOv3-tiny	54.84	5.560	0.66	0.58	0.62	33.0 MB

Với việc áp dụng nhóm các lớp tích chập để giảm số lượng phép tính BFLOPs đi đáng kể từ đó có thể tăng kích thước của ảnh đầu vào từ 416×416 lên 608×608 để tăng cường khả năng phát hiện vật thể nhỏ của mô hình YOLOv4-tiny-G3l đã đem lại hiệu quả accuracy đáng kể cùng với đó việc áp dụng nhóm các lớp tích chập đã nén mô hình đi hơn một nửa so với YOLOv4-tiny. Dựa vào yêu cầu đặt ra trong đề tài có thể kết luận được kiến trúc đề xuất YOLOv4-tiny-G3l đã đáp ứng được nhu cầu của đề tài.

4.2 LẬP TRÌNH HỆ THỐNG

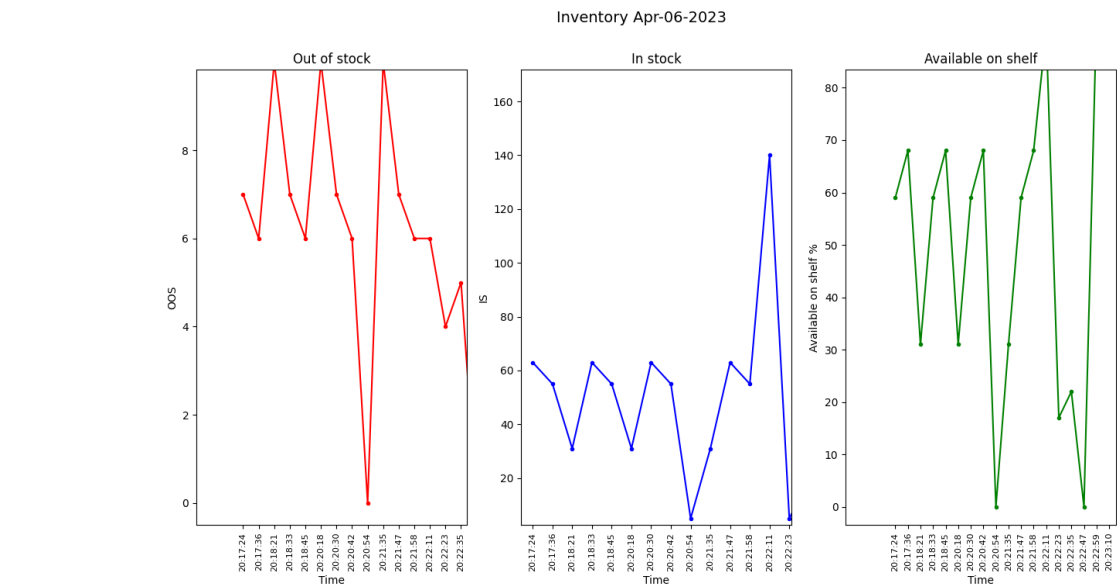
Thành phần giao diện sẽ được thiết kế dựa trên nền tảng pyqt5 để thiết kế giao diện đồng thời xử lý các tác vụ như hiển thị dữ liệu được lấy từ thành phần lưu trữ đám mây thời gian thực và hiển thị hình ảnh thực trạng tại kệ hàng và dữ liệu kiểm kê được thu thập từ phần cứng Jetson Nano. Sau đó dữ liệu được cập nhật từ firebase vào tệp csv và kiểm kê theo ngày, người dùng có thể lựa chọn ngày cần kiểm kê hàng hóa giao diện sẽ hỗ trợ vẽ đồ thị hiển thị số lượng OOS, IS và tỉ lệ hàng hóa có sẵn trên kệ dưới dạng biểu đồ. Giao diện có thể được sử dụng để giám sát từ xa tại một máy tính khác.



Hình 4. 4: Hình ảnh giao diện hiển thị

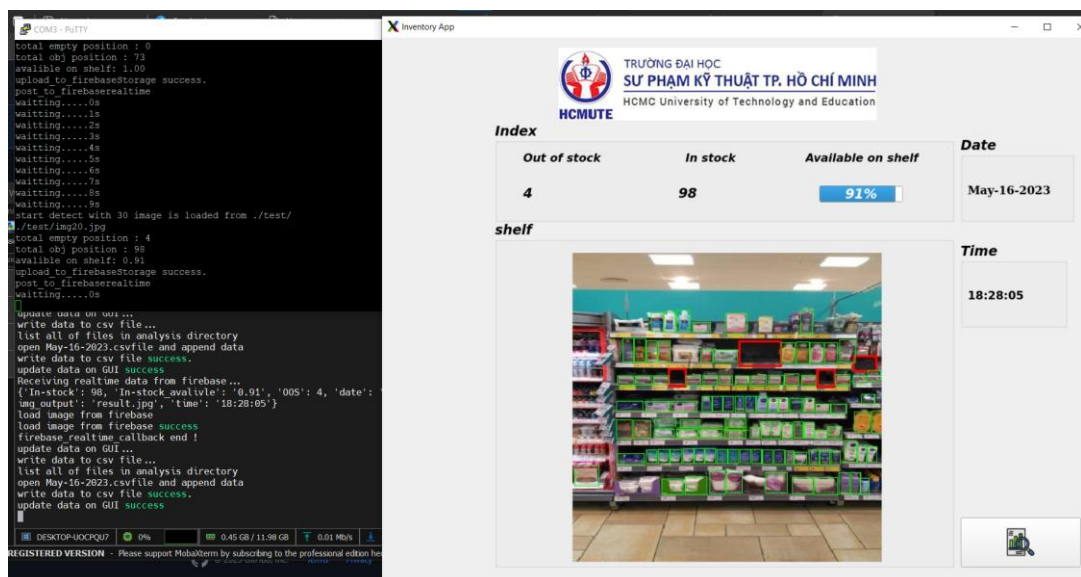
$$Available_on_shelf = \frac{S_{IS}}{S_{IS} + S_{OOS}}$$

Trong đó S_{IS} là tổng diện tích các vật thể IS có trong hình ảnh và S_{OOS} là tổng diện tích các vị trí OOS có trong khung hình. Dựa vào cách tính này có thể ước lượng được tương đối chính xác tỉ lệ hàng hóa đang nằm trên kệ hàng.

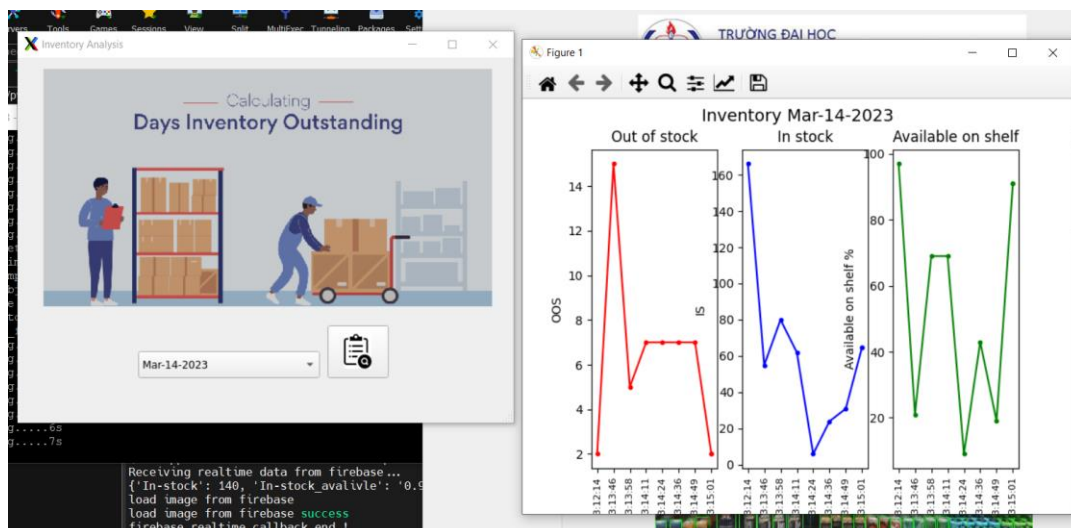


Hình 4. 5: Giao diện hiển thị biểu đồ kiểm kê hàng hóa

4.3 ĐÁNH GIÁ KẾT QUẢ



Hình 4. 6: Kết quả chạy thử mô hình



Hình 4. 7: Giao diện kiểm kê hàng hóa trong ngày

Để tiến hành đánh giá hoạt động của hệ thống trong đề tài này sẽ sử dụng máy tính nhúng Jetson Nano 4GB để chạy tác vụ xử lý hình ảnh và một máy tính khác chạy giao diện hiển thị và cả hai thành phần được kết nối thông qua firebase. Máy tính nhúng Jetson Nano sẽ được cung cấp hình ảnh đầu vào thông qua camera sẽ chụp lại tại các cửa hàng và sẽ được kết nối internet để gửi dữ liệu thông qua firebase. Cùng lúc đó ứng dụng **Inventory_app** sẽ được chạy ở một máy tính khác có kết nối internet, khi dữ liệu được tải lên firebase thời gian thực và hình ảnh tại kệ hàng cũng được tải lên firebase storage thì firebase sẽ gửi tín hiệu báo xuống cho ứng dụng chạy **Inventory_app** khi đó ứng dụng sẽ lấy dữ liệu từ firebase về và hiển thị các thông số như OOS, IS, Available on shelf, ngày và giờ tại thời điểm mà máy tính nhúng Jetson Nano thu thập dữ liệu. Từ đó có thể kết luận được hệ thống đã chạy và đáp ứng đúng với sơ đồ khối hệ thống và sơ đồ trình tự hệ thống được thiết kế ở trong chương 3. Độ trễ của hệ thống sẽ phụ thuộc khá lớn vào đường truyền mạng wi-fi vì vậy cần phải cung cấp một đường truyền mạng với tốc độ ổn định để đảm bảo hệ thống không bị gián đoạn.

CHƯƠNG 5

KẾT LUẬN

5.1 TỔNG KẾT LẠI QUÁ TRÌNH THỰC HIỆN ĐỀ TÀI

5.1.1 Nghiệm thu kết quả

Kết quả đạt được trong đề tài này thực hiện đã đạt yêu cầu với mục tiêu ban đầu đặt ra của đề tài bao gồm hai phần xử lý hình ảnh được chụp tại vị trí các kệ hàng và xây dựng giao diện hiển thị để nhân viên tại các siêu thị lớn hoặc cửa hàng tiện lợi có thể giám sát từ xa và kịp thời xử lý tình trạng OOS tại cửa hàng. Ngoài ra với chức năng hiển thị biểu đồ và lưu lại dữ liệu về tình trạng hàng hóa theo ngày đề tài đã hoàn thành đầy đủ các mục tiêu đề ra. Phát triển ứng dụng kiểm kê hàng hóa tại các cửa hàng tiện lợi giúp cho các nhà quản lý thuận tiện hơn trong việc giám sát tình hình kinh doanh tại cửa hàng và từ đó đưa ra các chiến lược phù hợp để gia tăng doanh thu.

5.1.2 Tổng kết lại quá trình thực hiện đề tài

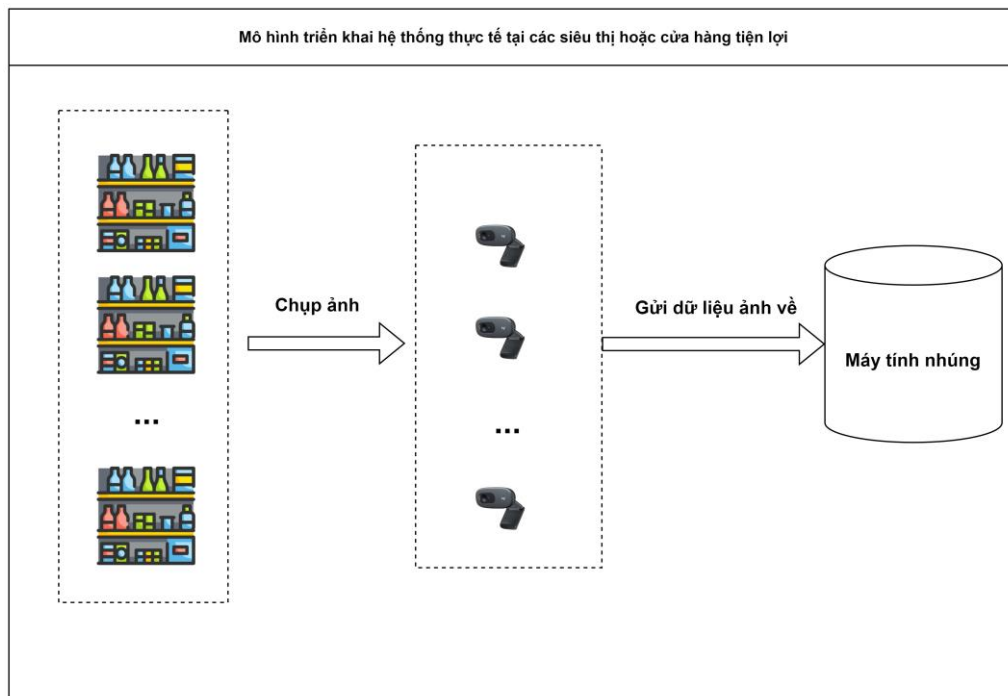
Tổng kết lại toàn bộ quá trình nghiên cứu và thực hiện đề tài ứng dụng học sâu cho việc giám sát và quản lý kiểm kê hàng hóa trong ngành bán lẻ đã đạt được các mục tiêu đề ra như sau:

- Nghiên cứu về Học máy và trí tuệ nhân tạo.
- Áp dụng học sâu để đào tạo mô hình nhận diện OOS và IS.
- Nghiên cứu và tìm hiểu các kiến trúc trong học sâu và áp dụng để tăng cường độ chính xác của mô hình.

- Tìm hiểu các thiết bị máy tính nhúng và triển khai mô hình hệ thống đã thiết kế.
- Xây dựng được hệ thống gồm phần cứng và phần mềm để thu thập hình ảnh tại kệ hàng và quản lý kiểm kê các thông số trên kệ hàng.

5.2 HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

Định hướng phát triển của đề tài ứng dụng học sâu cho việc giám sát và quản lý kiểm kê hàng hóa trong ngành bán lẻ có thể áp dụng tại các siêu thị lớn, cùng với việc sử dụng giao diện OPENCV DNN nên có thể tương tác với rất nhiều phần cứng khác nhau và cơ chế chụp ảnh để thu thập dữ liệu sau một khoảng thời gian nhất định thì hệ thống sẽ không cần khả năng xử lý quá mạnh mẽ của máy tính nhúng. Từ đó có thể cân nhắc sử dụng nhiều camera kết nối với một máy tính nhúng Jetson Nano và thu thập dữ liệu tại nhiều kệ hàng cùng lúc giúp giảm tối đa chi phí lắp đặt thiết bị.



Hình 5. 1: Mô hình áp dụng tại các siêu thị lớn

TÀI LIỆU THAM KHẢO

- [1] K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015.
- [2] Kalinina, Maria, and Pavel Nikolaev. "Research of yolo architecture models in book detection." In 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2020), pp. 218-221. Atlantis Press, 2020.
- [3] P. Gibson, J. Cano, J. Turner, E. J. Crowley, M. O'Boyle and A. Storkey, "Optimizing Grouped Convolutions on Edge Devices," 2020 IEEE 31st International Conference on Application-specific Systems, Architectures and Processors (ASAP), Manchester, UK, 2020, pp. 189-196.
- [4] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In International conference on machine learning, pp. 6105-6114. PMLR, 2019.
- [5] Mahasin, Marsa, and Irma Amelia Dewi. "Comparison of CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0 Backbones on YOLO V4 as Object Detector." *International Journal of Engineering, Science and Information Technology* 2, no. 3 (2022): 64-72.
- [6] Gothane, Suwarna. "A Practice for Object Detection Using YOLO Algorithm." *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 7, no. 2 (2021): 268-272.
- [7] Juan K. Leonard, Ph.D. "Image Classification and Object Detection Algorithm Based on Convolutional Neural Network" On Nov 30, 2019.

- [8] Janiesch, Christian, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning." *Electronic Markets* 31, no. 3 (2021): 685-695.
- [9] Li, Ping, Taiyu Han, Yifei Ren, Peng Xu, and Hongliu Yu. "Improved YOLOv4-tiny based on attention mechanism for skin detection." *PeerJ Computer Science* 9 (2023): e1288.

PHỤ LỤC

[1]. Link mã nguồn trên phần cứng Jetson Nano.

https://github.com/Winxkin/cv_detect_inventory

[2]. Link mã nguồn phần giao diện.

https://github.com/Winxkin/pyqt5_inventory

[3]. Link kết quả chạy thử mô hình hệ thống.

<https://www.youtube.com/watch?v=zMn6YRYxsWE>