LVTN_Huan

by Thien Huynh-The

Submission date: 27-May-2023 04:27PM (UTC+0900)
Submission ID: 2102856801
File name: LVTN_Huan.pdf (3.99M)
Word count: 15366
Character count: 60745

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

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

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. HUỲNH 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 MSSV: 19119182

Tel: 0866078421 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

Tp.HCM, ngày tháng năm 2023

Xác nhận của Bộ Môn

Giáo viên hướng dẫn

(Ký ghi rõ họ tên và học hàm học vị)

TS. 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 tin cậy dựa trên cơ sở lý thuyết; có phân tích và đánh giá phù hợp; có tính mới và tính sáng tạo.

Có cải tiến 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. Có 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 hiệu năng (cải thiện khác)

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.

Phù hợp 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.

Tác phong chuyên nghiệp và có tính sáng tạo

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.

Đầy đủ 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 cải tiến 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 đánh giá trên nhiều tập dataset thực tế hơn
- Thời gian thi hành

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

* Giảng viên hướng dẫn có thể nhận xét thêm vào mặt sau

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 X

DANH MỤC BẢNG XII

CÁC TỪ VIẾT TẮT XIII

CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI 1

1.1 ĐẶT VẤN ĐỀ 1

1.2 MỤC TIÊU ĐỀ TÀI 2

1.3 PHƯƠNG PHÁP NGHIÊN CỨU 3

1.4 BỐ CỤC TRÌNH BÀY 4

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 5

2.1 HỌC SÂU TRONG THỊ GIÁC MÁY TÍNH 5

2.1.1 Giới thiệu chung 5

2.1.2 Mạng Nơ-ron (Neuron) nhân tạo 6

2.1.3 Sự khác nhau giữa học sâu, học máy và trí tuệ nhân tạo 7

2.1.4 Bài toán nhận diện vật thể 9

2.1.5 Bài toán phân loại 10

2.2 MÔ HÌNH MẠNG CNN 11

2.2.1 Giới thiệu mô hình mạng CNN (Convolutional Neural Networks) 11

2.2.2 Kiến trúc mạng CNN 12

2.2.3 Phép tích chập trong xử lý ảnh (Convolution) 13

2.3 MẠNG HỌC SÂU YOLO 14

2.3.1 Giới thiệu về mạng học sâu YOLO 14

2.3.2 Kiến trúc mạng YOLO phiên bản đầu tiên 16

2.3.3 Một số thuật ngữ cơ bản 19

2.3.4 Các phiên bản của mạng YOLO 22

2.3.5 Giới thiệu kiến trúc mạng YOLOv4-tiny 24

2.3.6 Lớp Phân bổ chồng không gian (Spatial pyramind pooling) 25

2.3.7 Nhóm các lớp tích chập (Group convolution) *26*

2.4 GIỚI THIỆU VỀ OPENCV DNN 27

2.4.1 Tổng quan về thư viện OPENCV *27*

2.4.2 OPENCV DNN *28*

2.5 PHẦN CỨNG JETSON NANO 29

2.5.1 Giới thiệu *29*

2.5.2 Cấu hình phần cứng *31*

CHƯƠNG 3 MÔ HÌNH ỨNG DỤNG 32

3.1 THIẾT KẾ MÔ HÌNH ỨNG DỤNG 32

3.1.1 Tổng quan mô hình *32*

3.1.2 Nguyên lý hoạt động *33*

3.2 THIẾT KẾ SƠ ĐỒ THUẬT TOÁN XỬ LÝ TRÊN JESON NANO 35

3.2.1 Mô hình kết hợp xử lý phát hiện phân loại OOS và IS *35*

3.2.2 Đề xuất kiến trúc YOLOv4-tiny-SPP *36*

3.2.3 Đề xuất kiến trúc YOLOv4-tiny-G3l *38*

3.2.4 Sơ đồ thuật toán *41*

3.3 THIẾT KẾ SƠ ĐỒ THUẬT TOÁN XỬ LÝ TRÊN GIAO DIỆN 42

3.3.1 Sơ đồ thuật toán *42*

3.3.2 Sơ đồ khối giao diện hiển thị *44*

CHƯƠNG 4 THỰC HIỆN MÔ HÌNH VÀ ĐÁNH GIÁ 46

4.1 HUẤN LUYỆN MÔ HÌNH HỌC SÂU DỰA TRÊN KIẾN TRÚC MẠNG YOLO 46

4.1.1 Giới thiệu google colab *46*

4.1.2 Nền tảng darknet *47*

4.1.3 Cấu hình các thông số của mạng YOLO *48*

4.1.4 Định dạng dữ liệu huấn luyện *51*

4.1.5 Đánh giá kết quả huấn luyện *52*

4.2 THIẾT LẬP PHẦN CỨNG JETSON NANO 56

4.3 LẬP TRÌNH HỆ THỐNG 57

4.4 ĐÁNH GIÁ KẾT QUẢ 59

CHƯƠNG 5 KẾT LUẬN 61

5.1 TỔNG KẾT LẠI QUÁ TRÌNH THỰC HIỆN ĐỀ TÀI 61

5.1.1 Nghiệm thu kết quả 61

5.1.2 Tổng kết lại quá trình thực hiện đề tài 61

5.2 HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI 62

TÀI LIỆU THAM KHẢO 63

PHỤ LỤC 65

DANH MỤC HÌNH

Hình 1. 1: OOS tại các cửa hàng (nguồn Internet) 2
Hình 2. 1: Hình ảnh minh họa về Deep learning (nguồn Internet) 5
Hình 2. 2:Mô hình mạng Nơ-ron (nguồn Internet) 6
Hình 2. 3: Mô hình phân cấp AL, ML, DL (nguồn Internet).....8
Hình 2. 4: Deep learning và Machine learning (nguồn Internet).....9
Hình 2. 5:Nhận diện vật thể (nguồn Internet).....10
Hình 2. 6: Phân loại vật thể (nguồn Internet) 11
Hình 2. 7: Kiến trúc mạng CNN (nguồn Internet).....12
Hình 2. 8: Tính toán tích chập trên ma trận ảnh (nguồn Internet) 14
Hình 2. 9: Hình ảnh minh họa với YOLOv4-tiny (nguồn Internet).....15
Hình 2. 10: So sánh FPS mạng YOLO (nguồn Internet) 15
Hình 2. 11: Kiến trúc mạng YOLOv1 với 24 lớp (nguồn Internet).....16
Hình 2. 12: Chia ảnh đầu vào thành dạng lưới 4×4 (nguồn Internet) 17
Hình 2. 13: Chọn ra các hộp giới hạn chứa đối tượng (nguồn Internet).....18
Hình 2. 14: Hàm dự đoán IOU (nguồn Internet) 19
Hình 2. 15: Kiến trúc chi tiết mạng YOLOv4-tiny.....24
Hình 2. 16: spatial pyramind pooling 25
Hình 2. 17: Group convolution (nguồn Internet) 26
Hình 2. 18: Giao diện OPENCV.....28
Hình 2. 19: OPENCV hỗ trợ nhiều định dạng mô hình (nguồn Internet).....29
Hình 2. 20: NVIDIA Jetson nano Developer kit (nguồn Internet) 30
Hình 3. 1: Mô hình hệ thống giám sát OOS và IS 33
Hình 3. 2: Sơ đồ trình tự hoạt động hệ thống 34
Hình 3. 3: Sơ đồ khối mô hình xử lý phát hiện vị trí OOS và IS.....35
Hình 3. 4: Kiến trúc YOLOv4-tiny-SPP.....37
Hình 3. 5: kiến trúc YOLOv4-tiny-G3l 39
Hình 3. 6: Nhóm lớp tích chập (nguồn Internet) 39
Hình 3. 7: Sơ đồ thuật toán xử lý trên Jetson nano 41
Hình 3. 8: Sơ đồ thuật toán trên giao diện hiển thị 43

Hình 3. 9: Sơ đồ khối giao diện hiển thị 44
Hình 4. 1: Google colab huấn luyện mô hình AI (nguồn Internet).....46
Hình 4. 2: Chuẩn bị dữ liệu huấn luyện 51
Hình 4. 3: Giao diện bộ công cụ SDK manager 56
Hình 4. 4: Hình ảnh giao diện hiển thị.....58
Hình 4. 5: Giao diện hiển thị biểu đồ kiểm kê hàng hóa 59
Hình 4. 6: Kết quả chạy thử mô hình.....59
Hình 4. 7: Giao diện kiểm kê hàng hóa trong ngày 60
Hình 5. 1:Mô hình áp dụng tại các siêu thị lớn.....62

DANH MỤC BẢNG

Bảng 3. 1: So sánh BFLOPS giữa YOLOv4-tiny và YOLOv4-tiny-SPP 38
Bảng 3. 2: So sánh BFLOPS giữa YOLOv4-tiny và YOLOv4-tiny-SPP 40
Bảng 4. 1: Cài đặt thông số môi trường 48
Bảng 4. 2: Một vài cấu hình thông dụng trong darknet 48
Bảng 4. 3: Tệp obj.data.txt và obj.names.txt 52
Bảng 4. 4 : So sánh mAP giữa các mô hình trên tệp ảnh phát hiện OOS..... 54
Bảng 4. 5: So sánh mAP giữa các mô hình trên tệp ảnh phát hiện IS 56

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 pyramind 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 tồn kho cao do sản phẩm sẽ 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). Điều này cũng là một trong những yếu tố quan trọng để hỗ trợ cho các nhà quản lý trong việc thu thập thông tin về mức độ tiêu thụ hàng hóa tại các địa điểm khác nhau, từ đó nâng cao năng suất trong quản lý hàng hóa và mục đích tối đa hóa lợi nhuận cho doanh nghiệp. Bên cạnh đó, đối với các địa điểm bán lẻ, thông tin về lượng hàng có sẵn trên kệ IS (In-stock) là một yêu cầu tối thiểu cần phải có cho bất kì chuỗi cung ứng nào. Những số liệu này sẽ giúp cho các nhà quản lý kiểm kê hàng hóa một cách dễ dàng, tiết kiệm thời gian và không cần đến cửa hàng.

Hình 1. 1: OOS tại các cửa hàng (nguồn Internet)

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 là một trong những yêu cầu tối thiểu cần phải có cho bất kì chuỗi cung ứng nào. Dễ để có thể nhận ra rằng hiện tượng OOS sẽ khiến cho doanh thu tại cửa hàng bị ảnh hưởng nghiêm trọng về lâu về dài và dẫn đến những hậu quả xấu khác như sự hài lòng của khách hàng, thất bại trong việc thu hút người mua hàng và nghiêm trọng hơn là mất uy tín trong ngành bán lẻ. Việc các kệ hàng không được tận dụng một cách tối đa sẽ dẫn đến phát sinh các chi phí khác như không tối đa hóa được lợi nhuận và chi phí thuê mặt bằng.

1.2 MỤC TIÊU ĐỀ TÀI

Để giải quyết vấn đề về OOS trong đề tài này sẽ tập trung vào việc phát triển một 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 OOS đó nhằm tối đa hóa công

năng của kệ hàng, tránh tình trạng OOS diễn ra quá lâu mà không được khắc phục dẫn đến những hệ lụy liên quan đến như sự hài lòng của khách hàng, mất cơ hội tiêu thụ hàng hóa và nhiều vấn đề liên quan khác. 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 đạt được độ chính xác cao nhất.
- 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 cơ sở lý thuyết về các khái niệm mạng học sâu (Deep Learning), máy học (Machine Learning), 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 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.

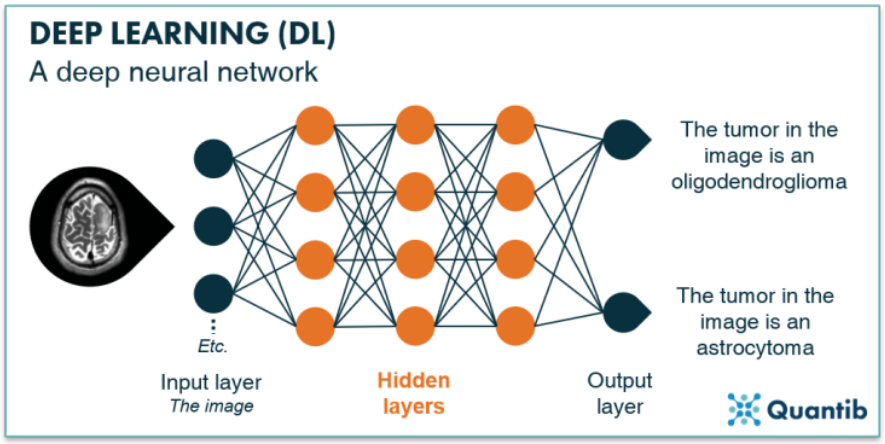
CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

2.1 HỌC SÂU TRONG THỊ GIÁC MÁY TÍNH

2.1.1 Giới thiệu chung

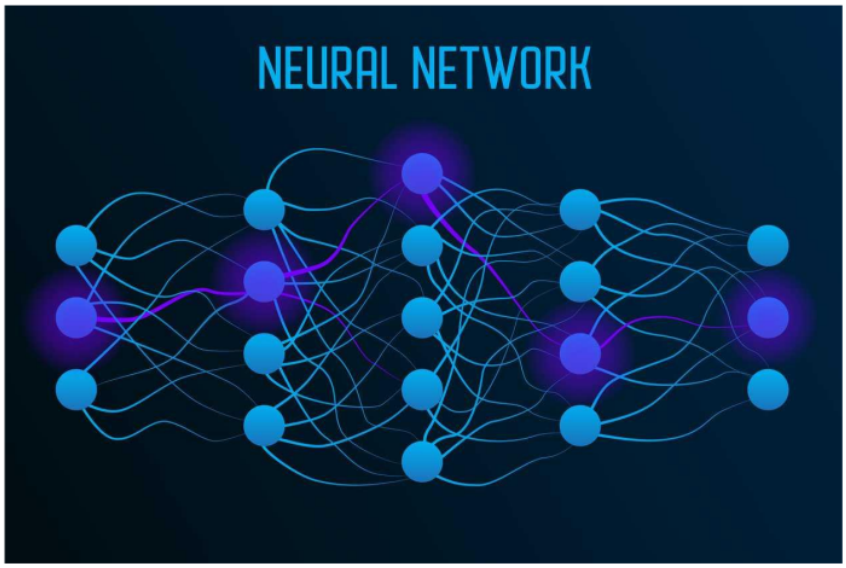
Trong thời đại hiện nay, khả năng trí tuệ nhân tạo được cung cấp bởi các hệ thống thông minh dựa trên máy học (ML - Machine Learning) và học sâu (DL - Deep Learning). Máy học miêu tả khả năng của các hệ thống học từ dữ liệu đào tạo để tự động hóa các quá trình phân tích mô hình và giải quyết các công 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 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.

2.1.2 Mạng Nơ-ron (Neuron) nhân tạo



Hình 2. 2:Mô hình mạng Nơ-ron (nguồn Internet)

Khái niệm Nơ-ron (Neuron) là một đơn vị thần kinh được liên kết với hàng triệu đơn vị Nơ-ron khác tạo nên một hệ thống thần kinh phức tạp và hoạt động bằng cách nhận dữ liệu từ đầu vào của nó và truyền đi cho các nơ-ron khác trong hệ thống.

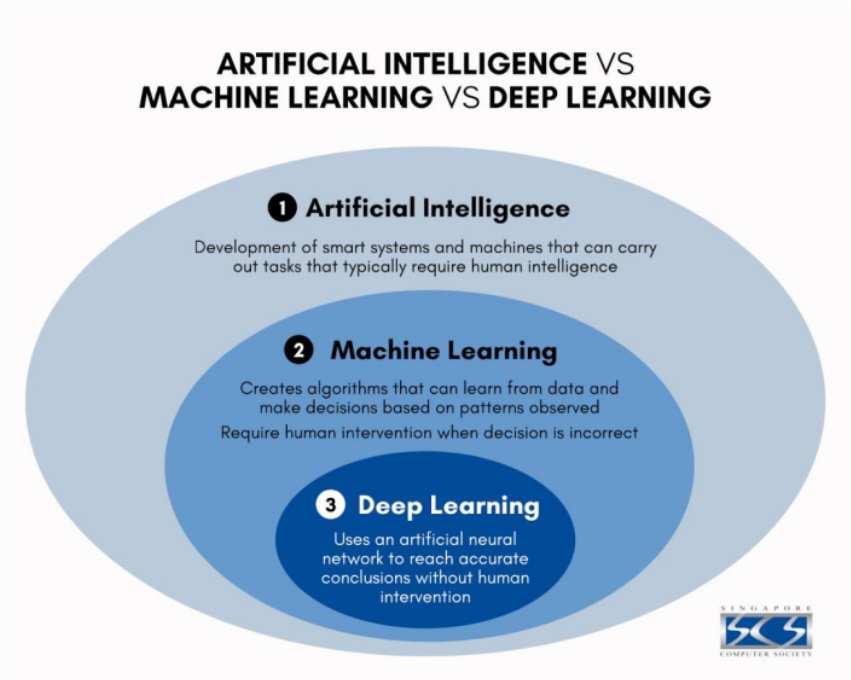
Ngày nay, Học sâu (DL) dựa trên cách hoạt động của mạng nơ-ron thần kinh để xây dựng lên một hệ thống mạng thần kinh gồm các lớp nơ-ron nhân tạo và bắt chước theo các tính năng của nó để áp dụng vào trong lĩnh vực trí tuệ nhân tạo (AI – Artificial intelligence). Từ đó mô hình mạng

nơ-ron ra đời và giúp cho lĩnh vực AI phát triển một cách vượt bậc nhờ khả năng suy luận gần giống như hành vi của con người. Nói cách khác mô hình mạng nơ-ron nhân tạo cố gắng mô phỏng lại gần chính xác với quy trình xử lý thông tin như là bộ não của con người.

2.1.3 Sự khác nhau giữa học sâu, học máy và trí tuệ nhân tạo

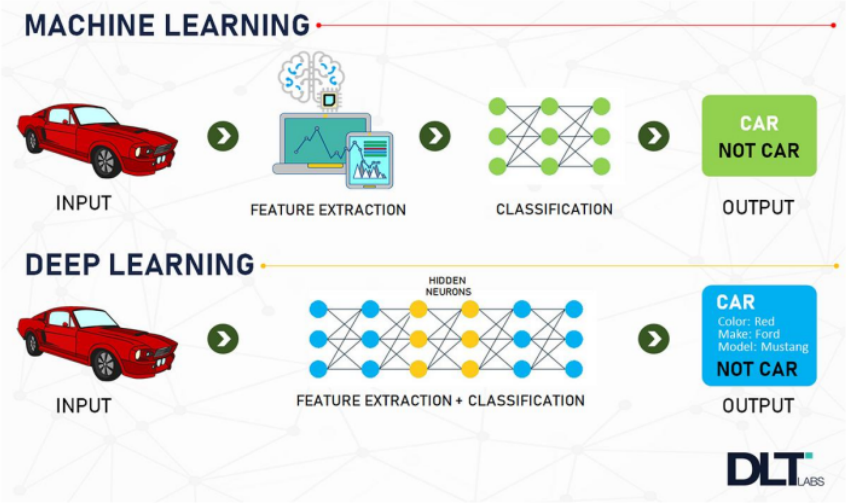
Thay vì hệ thống hóa kiến thức vào máy tính, máy học (ML) sẽ tìm cách tự động học các mối quan hệ và mẫu có ý nghĩa từ các ví dụ và quan sát. Những tiến bộ trong ML đã cho phép sự gia tăng gần đây của các hệ thống thông minh với khả năng nhận thức giống như con người, thâm nhập vào công việc kinh doanh và cuộc sống cá nhân của chúng ta, đồng thời định hình các tương tác được kết nối mạng trên thị trường điện tử theo mọi cách có thể.

Các hệ thống máy học (ML) được gọi là trí tuệ nhân tạo, nhờ khả năng xử lý và giải quyết các vấn đề phức tạp, phụ thuộc vào các mô hình phân tích và dự đoán các quy tắc để đưa ra câu trả lời hoặc đề xuất kết quả tương tự như con người. ML đã đóng góp vai trò quan trọng dựa trên việc giảm gánh nặng tính toán và dự đoán cho con người trong nhiều ứng dụng cụ thể. Tóm lại lĩnh vực trí tuệ nhân tạo phát triển phụ thuộc vào các mô hình DL và ML trong đó khái niệm về DL là một phương pháp được nâng cấp từ phương pháp học của ML trước đó bằng việc áp dụng mô hình mạng nơ-ron nhân tạo vào để xử lý các bài toán với cách thức tương tự như con người.



Hình 2. 3: Mô hình phân cấp AL, ML, DL (nguồn Internet)

Từ đó lĩnh vực ML đã tạo ra nhiều tiến bộ trong ngành máy tính nói riêng và các lĩnh vực liên quan khác nói chung. Một trong những tiến bộ này là sự phát triển và ra đời của các mạng thần kinh nơ tron nhân tạo từ đó khái niệm học sâu (DL – Deep Learning) được ra đời [8]. Đối với các ứng dụng cụ thể như nhận diện vật thể hoặc phân loại đối tượng, việc ứng dụng DL đem lại nhiều hiệu quả vượt trội nhờ cách huấn luyện đặt biệt thông qua mạng thần kinh nơ tron nhân tạo. Bằng cách trích xuất ra những đặc trưng của vật thể tại từng lớp mạng nơ tron nhân tạo và điều chỉnh các trọng số (weights) sau mỗi chu kì huấn luyện cho đến khi tỉ lệ dự đoán đạt được là nhỏ nhất.



Hình 2. 4: Deep learning và Machine learning (nguồn Internet)

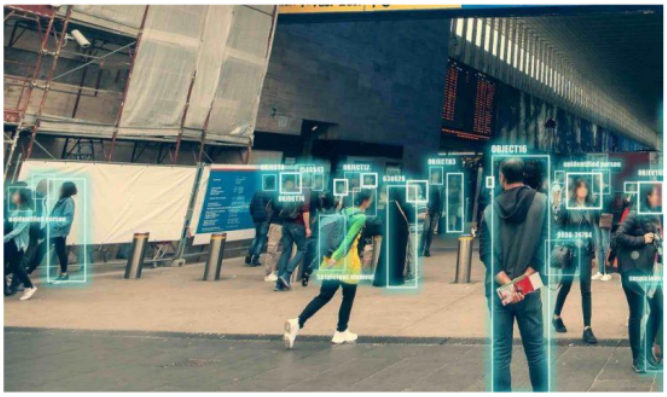
Trong DL nguồn dữ liệu đầu vào là cơ sở giúp cho mô hình có thể bắt đầu quá trình học. Dữ liệu có thể là các đoạn văn bản, hình ảnh, biểu tượng, v.v, và những dữ liệu đó được gọi là các dữ liệu chưa được xử lý. Khi dữ liệu đã được xử lý hay còn gọi là gán nhãn thì nó được gọi là các thông tin và DL sẽ dựa vào các thông tin đó để ghi nhớ hình dạng cũng như các đặc điểm của vật thể từ đó điều chỉnh các trọng số phù hợp để đưa ra các dự đoán chính xác. Từ sự phát triển của DL đã mở ra khả năng xử lý được nhiều vấn đề thực tế và đưa các hành vi của máy tính trở lại gần giống với cách con người suy luận. Từ đó lĩnh vực AL đã được mở rộng và bao quát hơn nhờ DL.

2.1.4 Bài toán nhận diện vật thể

Bài toán nhận diện vật thể (Object Detection) là một trong những bài toán cơ bản cũng như là quan trọng nhất trong lĩnh vực thị giác náy tính (Computer Vision). Có thể nói bài toán nhận diện vật thể là bước khởi đầu cho những bài toán khác như bài toán phân loại vật thể, truy vết vật thể, v.v. Nhận diện vật thể là nhiệm vụ phát hiện các thể hiện của một đối

tượng thuộc một lớp nhất định trong một hình ảnh và video [7]. Các phương pháp nhận diện vật thể tiên tiến nhất hiện nay có thể chia thành hai loại như sau:

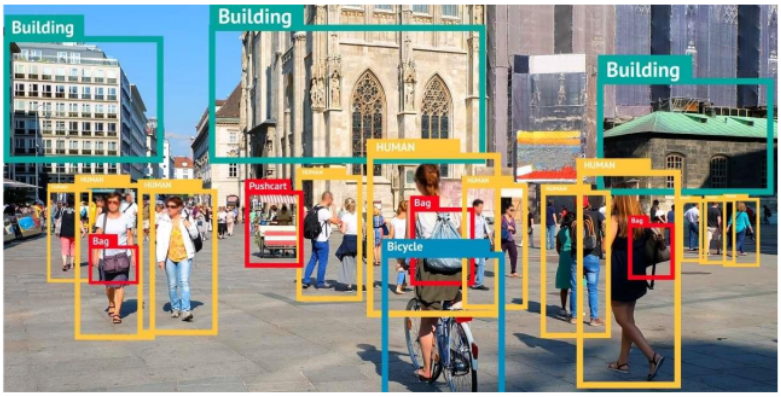
- **Phương pháp một giai đoạn:** Phương pháp này sẽ ưu tiên vào tốc độ suy luận của mô hình và thường được ứng dụng vào các ứng dụng thời gian thực cần tốc độ xử lý nhanh.
- **Phương pháp hai giai đoạn:** Phương pháp hai giai đoạn sẽ ưu tiên về độ chính xác của mô hình hơn là tốc độ suy luận của mô hình.



Hình 2. 5: Nhận diện vật thể (nguồn Internet)

2.1.5 Bài toán phân loại

Bài toán phân loại (Classification) là một trong những phương pháp được ứng dụng nhiều nhất trong lĩnh vực máy học. Bài toán phân loại sẽ thực hiện nhiệm vụ quyết định đối tượng sẽ thuộc phân lớp nào trong tổng số phân loại đối tượng được nhận diện. Bài toán phân loại sẽ dựa trên đặc điểm nhận dạng của các lớp khác nhau nhằm đưa ra dự đoán chính xác về vật thể [7].



Hình 2. 6: Phân loại vật thể (nguồn Internet)

Điều kiện tiên quyết để thực hiện được bài toán phân loại vật thể là phải nhận diện được các đối tượng trước từ đó mới dựa trên các đặc điểm của đối tượng để tiến hành phân loại đối tượng. Bài toán phân loại vật thể sẽ chia làm 2 giai đoạn bao gồm cả bài toán nhận diện đối tượng.

- **Giai đoạn 1:** Nhận diện và phát hiện vật thể trong khung hình dựa trên bài toán nhận diện vật thể từ đó trả về các giá trị tọa độ của vật thể trong khung hình.
- **Giai đoạn 2:** Sau khi có được vị trí tọa độ của vật thể ở giai đoan 1, giai đoạn 2 sẽ tiến hành thực hiện trích xuất các đặt trưng của đôi tượng và đưa ra dự đoán để phân loại đối tượng thuộc về lớp nào.

2.2 MÔ HÌNH MẠNG CNN

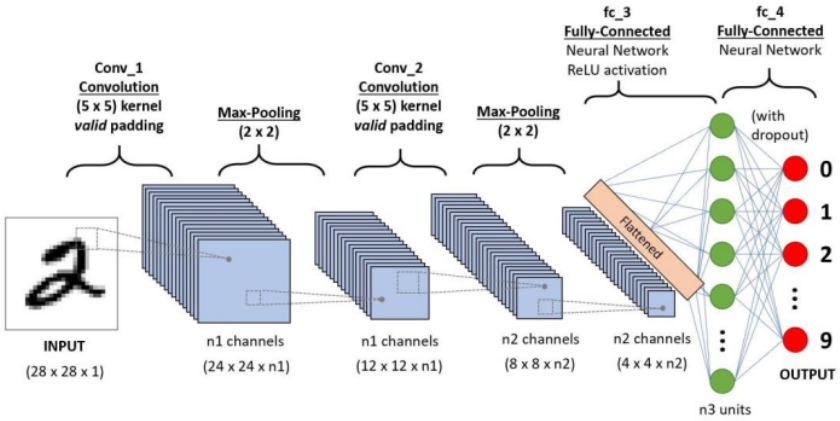
2.2.1 Giới thiệu mô hình mạng CNN (Convolutional Neural Networks)

Mô hình mạng CNN (Convolutional Neural Networks) là mạng tích chập được ứng dụng phổ biến trong các mô hình ứng dụng dùng để nhận diện và phân loại hình ảnh. Mô hình mạng CNN được ứng dụng rộng rãi trong các ứng dụng nhận diện và phân loại hình ảnh bằng cách sử dụng nhiều lớp tích chập, mỗi lớp tích chập sẽ trích xuất ra các đặc trưng của ảnh.

Mô hình mạng CNN cơ bản sử dụng các lớp tích chập (Convolution layer) và các lớp tổng hợp (Pooling layer) chồng lên nhau để trích xuất ra đặt trưng của ảnh đầu vào. Lớp tích chập đóng vai trò trích xuất ra các đặc trưng của ảnh đầu vào bằng cách sử dụng các mặt nạ (kernels) để lướt trên từng điểm ảnh và sử dụng phép nhân chập để trích suất ra các thông số đặc trưng của ảnh đầu vào. Lớp tổng hợp đóng vai trò tổng hợp lại các tích chất của ảnh từ lớp tích chập và loại bỏ đi các tính chất không cần thiết và đưa ra kết quả.

2.2.2 Kiến trúc mạng CNN

Kiến trúc của mạng CNN được xây dựng dựa trên các lớp cơ bản như lớp tích chập (convolution layer), lớp kích hoạt (Relu layer), lớp tổng hợp (Pooling layer) và lớp kết nối đầy đủ (Fully connected layer).



Hình 2. 7: Kiến trúc mạng CNN (nguồn Internet)

- **Lớp tích chập (Convolution layer):** Lớp tích chập được coi là có vai trò quan trọng nhất trong kiến trúc mạng CNN, ở lớp này sẽ thực hiện các quá trình tính toán như nhận chập ảnh với mặt nạ (kernals), lọc (fillter), trích xuất đặc trưng (feature map). Trích xuất đặc trưng là yếu tố quan trọng nhất trong lớp tích chập.

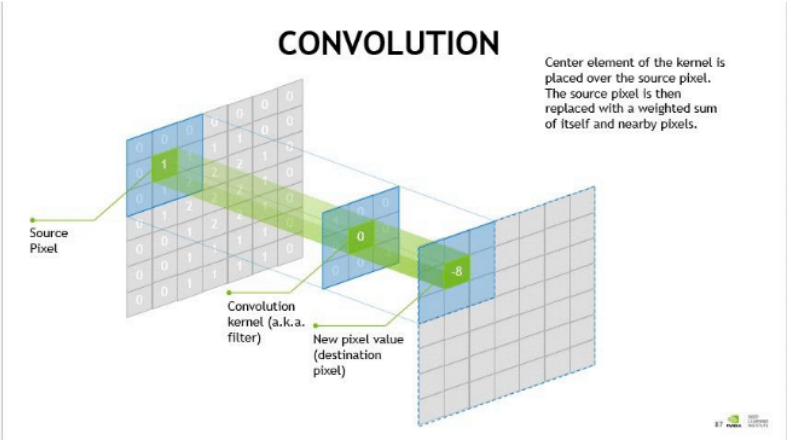
- **Lớp Kích hoạt (Relu layer):** Lớp kích hoạt dùng để tính toán và mô phỏng lại tỉ lệ truyền xung qua axon của hệ nơ-tron thần kinh dùng các hàm hàm kích hoạt (activation function) như leaky, mish, swish, v.v.
- **Lớp tổng hợp (Pooling layer):** Lớp tổng hợp sẽ lượt bỏ đi các thông tin không cần thiết từ lớp tích chập từ đó cho ra kết quả mong muốn.
- **Lớp kết nối đầy đủ (Fully connected layer):** Ở lớp kết nối đầy đủ sẽ lấy kết quả đầu vào từ lớp trước đó và tiến hành nén thông tin ảnh về dưới dạng một véc tơ và dùng kết quả đó để làm đầu vào cho lớp tiếp theo.

2.2.3 Phép tích chập trong xử lý ảnh (Convolution)

Phép tích chập là một trong những phép toán quan trọng trong lĩnh vực xử lý ảnh, bằng cách sử dụng các mặt nạ để lướt trên từng điểm ảnh và tính tích chập tại từng điểm ảnh. Công thức tích chập tại một điểm ảnh $f(x,y)$ với mặt nạ $k(x,y)$ được biễu diễn như sau.

$$f(x,y)*k(x,y)=\sum_{u=-m/2}^{m/2}\sum_{v=-n/2}^{n/2}k(u,v)f(x-u,y-v)$$

Phép tích chập có thể được hình dung như bằng cách dùng mặt nạ K dịch chuyển trên từng điểm ảnh từ góc trái ảnh theo hàng ngang và theo hàng dọc khi đi tới điểm ngoài cùng bên phải của ảnh. Tương ứng với mỗi lần dịch chuyển sẽ thực hiện tính toán giá trị tại điểm ảnh đang xét bằng công thức tính tích chập.



Hình 2. 8: Tính toán tích chập trên ma trận ảnh (nguồn Internet)

2.3 MẠNG HỌC SÂU YOLO

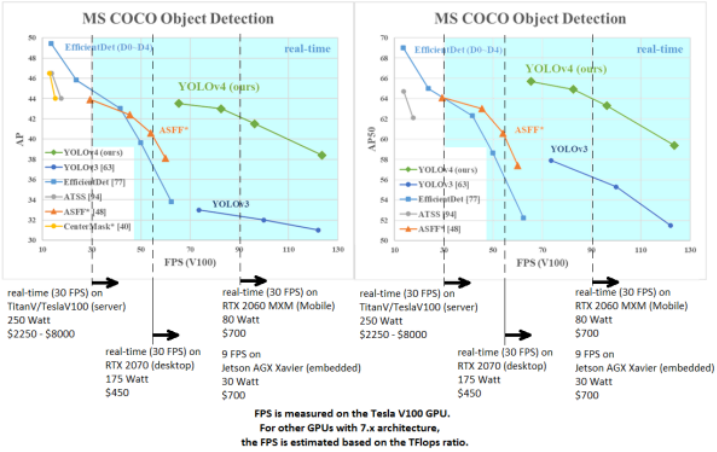
2.3.1 Giới thiệu về mạng học sâu YOLO

Trong những thập kỷ gần đây, việc ứng dụng các công nghệ để tạo ra các hệ thống thông minh dựa trên hình ảnh và phát hiện đối tượng đã trở thành một nhu cầu thiết yếu. Điều này được thúc đẩy bởi sự phát triển của tốc độ tính toán của các máy tính và sự phát triển của các mô hình mạng học sâu như mạng tích chập (CNN). Tuy nhiên, để đáp ứng được nhu cầu này, cần sử dụng các nguồn thông tin dữ liệu khổng lồ có sẵn.

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. 9: Hình ảnh minh họa với YOLOv4-tiny (nguồn Internet)

YOLO (You Only Look Once) là một kiến trúc mạng học sâu được thiết kế để xử lý một pha duy nhất, giúp tối ưu hóa tốc độ xử lý trong khi vẫn đảm bảo độ chính xác của mô hình. Với ưu điểm này, mạng YOLO được sử dụng mạnh mẽ trong các ứng dụng cần tính thời gian thực. Khả năng phát hiện đối tượng nhanh của YOLO đem lại tốc độ xử lý trên một khung hình đạt mức cao, mà không ảnh hưởng nhiều đến độ chính xác so với các mạng học sâu khác [6].



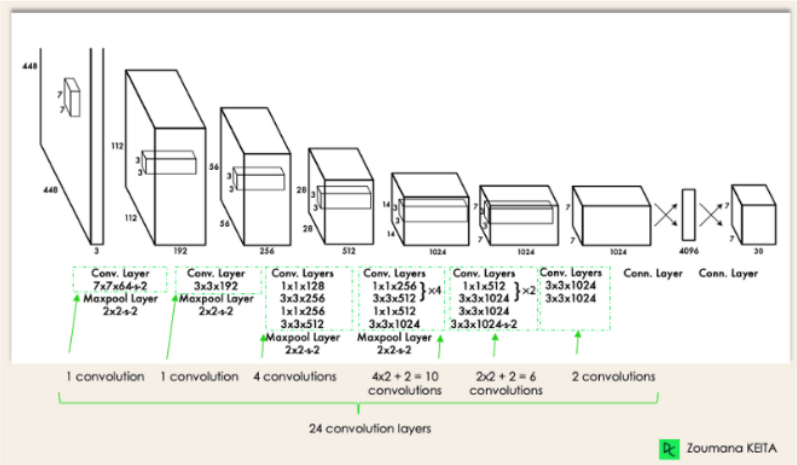
Hình 2. 10: So sánh FPS mạng YOLO (nguồn Internet)

Ngoài ra với tốc độ dự đoán của mạng YOLO, mạng YOLO đặc biệt hữa ích để giải quyết các vấn đề khác nhau như phân tích lưu lượng xe

trong lĩnh vực giao thông, hệ thống hỗ trợ người lái trong lĩnh vực ô tô và nhiều lĩnh vực khác cần thời gian đáp ứng ở tốc độ cao.

2.3.2 Kiến trúc mạng YOLO phiên bản đầu tiên

Kiến trúc mạng YOLO ban đầu được xây dựng dựa trên kiến trúc mạng CNN. Không giống như cách thức thực hiện của mạng Fast R-CNN sẽ sử dụng thuật toán *Selective Seach* để cắt các vùng cần phát hiện ra thành các ảnh nhỏ có kích thước bé hơn rồi đưa qua mạng CNN để xử lý và cho ra kết quả thì YOLO lại theo một cách thức khác thay vì lựa chọn và cắt các đối tượng trong ảnh ra YOLO sẽ chia ảnh thành một lưới ô vuông với kích thước MxN rồi duyệt qua từng ô trong ảnh sau đó tiến hành dự đoán và phân loại. Cuối cùng đối với các ô có xác xuất cao trên ngưỡng cho phép thì sẽ được giữ lại và cho ra kết quả.

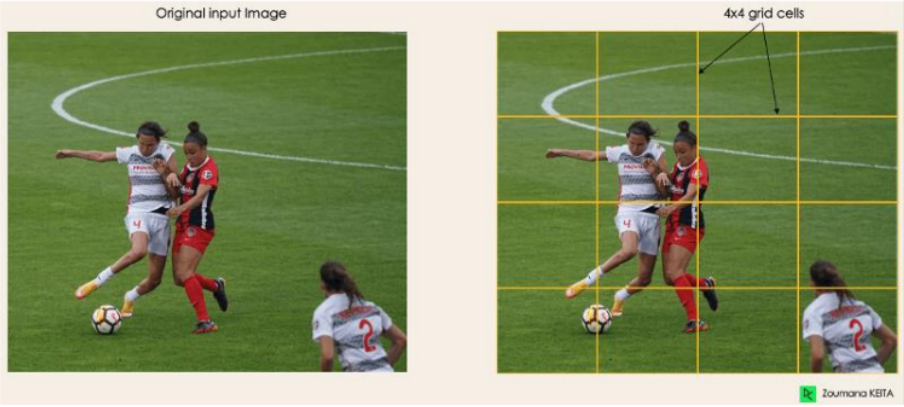


Hình 2. 11: Kiến trúc mạng YOLOv1 với 24 lớp (nguồn Internet)

Kiến trúc mạng YOLO sẽ dùng hình ảnh đầu vào và thay đổi kích thước của chúng thành 448×448 bằng cách giữ nguyên tỉ lệ khung hình và thực hiện kỹ thuật đệm, sau đó hình ảnh sẽ được đưa qua lớp CNN. Với mô hình mạng YOLO ban đầu sẽ bao gồm 24 lớp tích chập (Convolution layers), 4 lớp tổng hợp (Pooling layers) và theo sau là 2 lớp kết nối đầy đủ (Fully connected layers). Đầu tiên, Lớp tích chập 1×1 sẽ đươc áp dụng để

giảm số lượng kênh, sau đó áp dụng các lớp tích chập 3×3 để tạo ra các khối đầu ra. Lớp cuối cùng của mạng sẽ là lớp dự đoán đầu ra của ảnh.

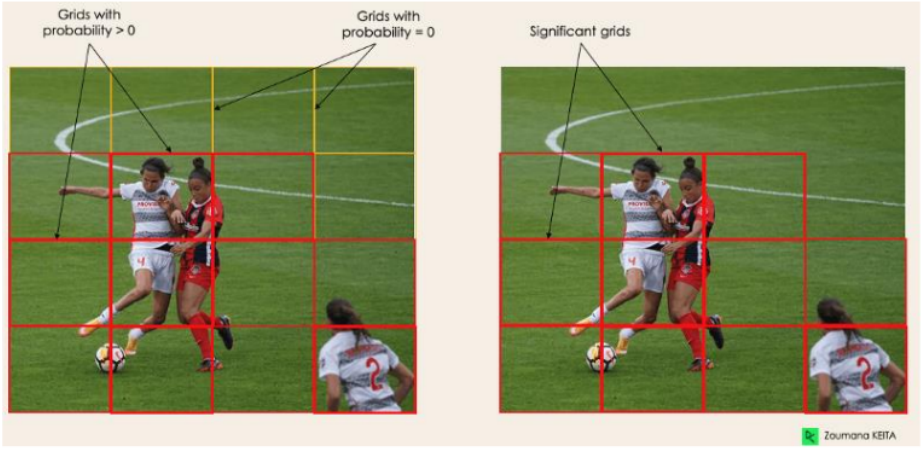
Để có thể dự đoán được vật thể có trong ảnh đầu tiên YOLO sẽ phân chia hình ảnh đầu vào thành các lưới hình hộp có kích thước M×N, mỗi ô trong hình sẽ phải chịu trách nhiệm giới hạn đối tượng mà nó bao quanh và dự đoán đối tượng đó.



Hình 2. 12: Chia ảnh đầu vào thành dạng lưới 4×4 (nguồn Internet)

Ở bước tiếp theo, YOLO sẽ lựa chọn ra các hộp giới hạn (Bouding boxes) có chứa đối tượng trong ảnh. Thuộc tính của mỗi hộp giới hạn sẽ được biểu diễn dưới dạng một véc tơ.

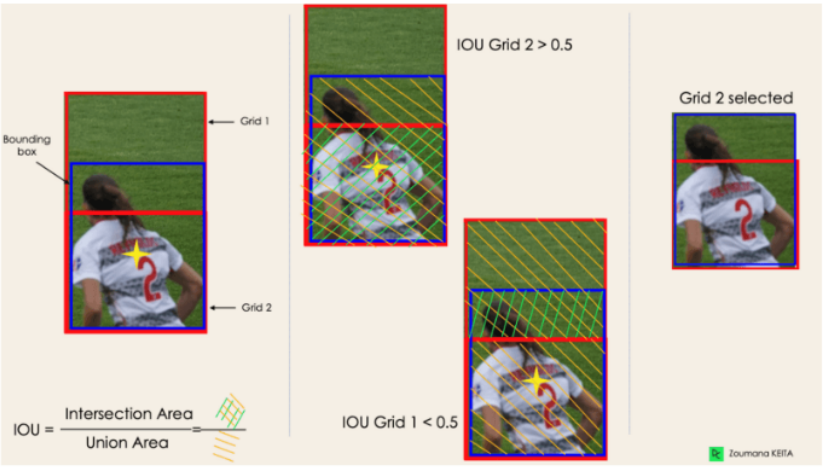
Y = [pc, bx, by, bh, bw, c1, c2, …, cn] Trong đó:



Hình 2. 13: Chọn ra các hộp giới hạn chứa đối tượng (nguồn Internet)

- Tham số bx, by đại diện cho vị trí tọa độ x, y của điểm trung tâm của hộp giới hạn so với toàn bộ các lưới ảnh.
- Tham số bw, bh sẽ tương ứng với chiều rộng và chiều cao của hộp giới hạn trên lưới ảnh.
- Các tham số c1, c2, …, cn đại diện cho từng lớp đối tượng có trong ảnh.

Ở bước tiếp theo YOLO sẽ tiến hành dự đoán các đối tượng có trong ảnh, thông thường một đối tượng được dự đoán sẽ bao gồm nhiều hộp giới hạn bao quanh đối tượng đó, thậm chí các hộp giới hạn này còn không liên quan tới nhau. Vì vậy YOLO dùng thông số IOU (Intersection over union) để đánh giá độ chính xác của đối tượng dự đoán, IOU sẽ giúp loại bỏ các hộp giới hạn mà không liên quan đến đối tượng dự đoán. Giá trị IOU nằm trong ngưỡng từ 0 đến 1.



Hình 2. 14: Hàm dự đoán IOU (nguồn Internet)

Tùy vào cấu hình mà người sử dụng có thể đặt ra một ngưỡng tin cây (IOU selection threshold). YOLO sẽ tiến hành tính toán giá trị IOU của mỗi hộp giới hạn và so sánh với ngưỡng tin cậy nếu giá trị IOU mà lớn hơn ngưỡng tin cậy thì hộp giới hạn đó được chọn, ngược lại nếu giá trị IOU bé hơn ngưỡng tin cậy thì hộp giới hạn đó sẽ bị loại bỏ.

2.3.3 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 trên 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:

$$L_{coord} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (x_i - x_i^{'})^2 + (y_i - y_i^{'})^2$$

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

$$L_{coord} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (\sqrt{w_i} - \sqrt{w_i^{'}})^2 + (\sqrt{h_i} - \sqrt{h_i^{'}})^2$$

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

$$L_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - C_i^{'})^2 + \lambda_{noobject} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - C_i^{'})^2$$

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:

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

Trong đó $\amalg_i^{obj}$ sẽ bằng 1 nếu hộp giới hạn đang xét có chưa 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 (Toltal loss): Hàm lỗi tổng hợp được tính bằng tổng của 3 hàm lỗi 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. Tỉ 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

Tỉ lệ này cũng tương tự như tỉ lệ dự đoán (precision), recall cho thấy được tỉ lệ bỏ sót các điểm TP (true positive). Công thức của tỉ 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-Socre

Tỉ lệ F1-Socre được lựa chọn khi mô hình cần cả hai Precision và Recall cao vì vậy tỉ 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 tỉ 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 tỉ lệ dự đoán chính xác đối với một lớp đối tượng, N là tổng số lớp đối tượng được dự đoán.

2.3.4 Các phiên bản của mạng YOLO

Cho tới hiện nay YOLO đã trải qua nhiều lần cải cách cùng với các phiên bản khác nhau nhằm nâng cao độ chính xác và tối ưu các thuật toán để giải quyết các vấn đề. Dưới đây sẽ trình bày một số đặc điểm của các phiên YOLOv2 tới YOLOv4.

1. **YOLO2**

   -Kiến trúc mạng: YOLOv2 sử dụng kiến trúc mạng Darknet-19 với 19 lớp tích chập.

   -Batch Norm: Các lớp Batch Norm được thêm vào giúp cải thiện

thời gian huấn luyện nhanh hơn, ổn định hơn.

-Fine-grained features: YOLOv2 sử dụng kích thước ma trận ảnh đặc trưng là 13×13 (gấp đôi so với phiên bản YOLOv1 trước đó). Vì vậy sẽ tăng cường được việc phát hiện những đối tượng nhỏ của YOLOv2 tốt hơn.

-Multi scale tranning: Với việc thay đổi kích thước ảnh đầu vào liên tục trong lúc huấn luyện, YOLOv2 có khả năng phát hiện đối tượng tốt hơn với các kích thước ảnh khác nhau.

-Anchor box: Việc ứng dụng anchor box trong YOLOv2 tăng cường khả năng nhận diện nhiều vật thể cùng lúc nhưng bù lại khả năng nhận diện chính xác của YOLOv2 bị giảm.

2. **YOLOv3**

-Kiến trúc mạng: YOLOv3 sử dụng mạng Darknet-53 bao gồm 53 lớp tích chập.

-Neck: YOLOv3 áp dụng mô hình mạng thực hiện phát hiện đối tượng với 3 mức tỉ lệ khác nhau của ma trận đặc trưng.

-Classification prediction: Thay vì sử dụng hàm *softmax* như các phiên bản YOLO trước thì ở phiên bản YOLOv3 sử dụng hàm *sigmoid* ở đầu ra của mạng.

3. **YOLOv4**

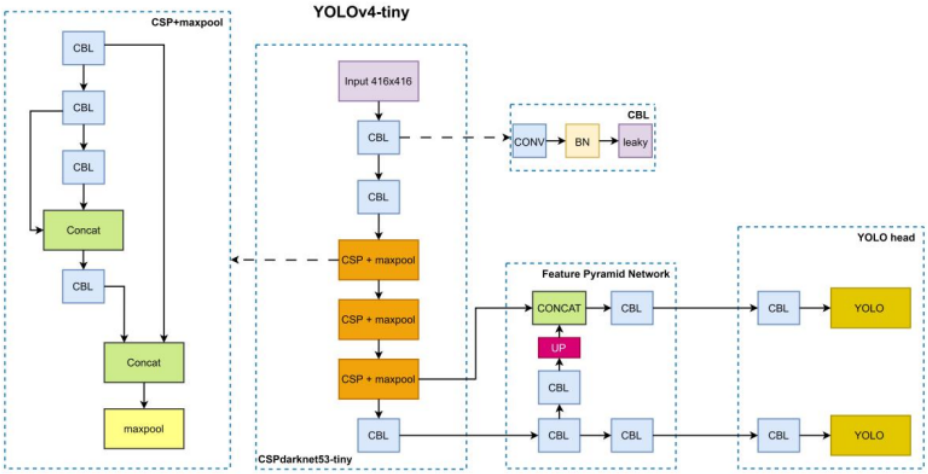
-Kiến trúc mạng: YOLOv4 sử dụng mạng Darknet-137 với 137 lớp tích chập. Với một số cải tiến dựa trên thuật toán của phiên bảng trước đó là YOLOv3, YOLOv4 đã cải thiện được độ chính xác trung bình (mAP) lên đến 10%.

-Backbone: Với việc chia lớp đầu vào hiện tại ra làm hai thành phần một phần đi qua lớp tích chập và một phần sẽ không đi qua lớp tích chập, sau đó kết quả được tổng hợp lại ở đầu ra.

2.3.5 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 đặt 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 ra lớp YOLO để dự đoán. Độ phức tạp tính toán của mạng YOLOv4-tiny rơi vào khoảng 6700 BFLOPS.



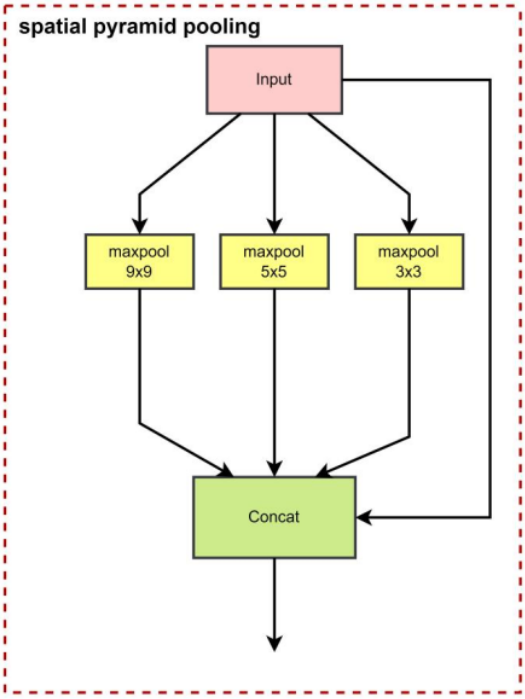
Hình 2. 15: Kiến trúc chi tiết mạng YOLOv4-tiny

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.

Ngoài ra với kiến trúc nhỏ gọn của mạng YOLOv4-tiny tốc độ tính toán được giảm đi đáng kể và vẫn dự được mức tỉ lệ dự đoán tốt, từ đó YOLOv4-tiny trở thành lựa chọn ưu tiên hàng đầu cho các nền tảng phần cứng nhúng với các nhược điểm về sức mạnh phần cứng để giải quyết các vấn đề trong đời sống bằng các ứng dụng trong thị giác máy tính.

2.3.6 Lớp Phân bổ chồng không gian (Spatial pyramind pooling)



Hình 2. 16: spatial pyramind pooling

Các lớp tích chập (convolution) sẽ có thể chấp nhận kích thước ảnh đầu vào với bất kì 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 pyramind pooling) có thể được áp đụng để tăng cường Bow bằng cách sử dụng ba lớp maxpool trích xuất đặt 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×W×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×W×4C. Dưới đây là biểu thức biểu thị cho đầu vào và đầu ra của lớp SPP.

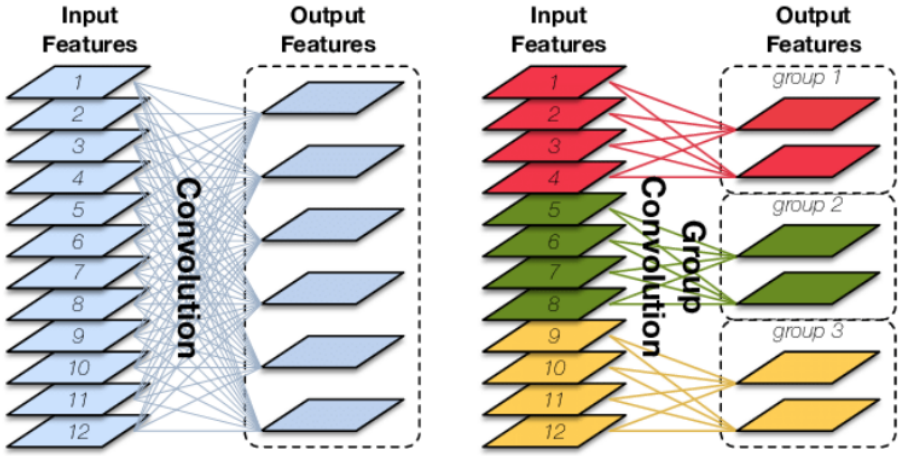
$$F_{mp9x9} = F_{\max pool9x9}(F_{in})$$

$$F_{mp5x5} = F_{\max pool5x5}(F_{in})$$

$$F_{mp3x3} = F_{\max pool3x3}(F_{in})$$

$$F_{out} = F_{concat}(F_{mp9x9}, F_{mp5x5}, F_{mp3x3}, F_{in})$$

2.3.7 Nhóm các lớp tích chập (Group convolution)



Hình 2. 17: Group convolution (nguồn Internet)

Dựa theo nhiều nghiên cứu trước đó đã chứng minh được sự hiệu quả của nhóm các lớp tích chập (group convolution) so với lớp tích chập truyền thống. 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 CNN. 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.

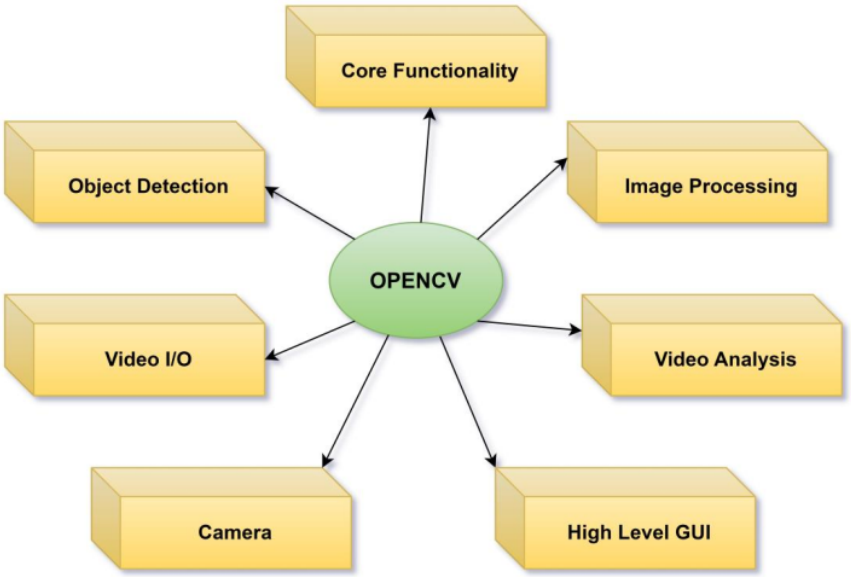
$$PARAM_{conv} = H \times W \times Din \times Dout$$

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

2.4 GIỚI THIỆU VỀ OPENCV DNN

2.4.1 Tổng quan về thư viện OPENCV

OPENCV là một giao diện cung cấp các chức năng để sử dụng các mô hình học sâu dưới dạng suy luận. Với OPENCV, ta có thể sử dụng các mạng thần kinh được đào tạo trước từ các khung phổ biến như YOLO, TensorFlow, PyTorch, Caffe, v.v. và tích hợp chúng trực tiếp vào trong OPENCV để thực hiện các chức năng học sâu và xử lý ảnh. Nhờ đó, OPENCV cho phép phát triển các ứng dụng kết hợp giữa các mô hình mạng thần kinh và xử lý ảnh một cách hiệu quả.

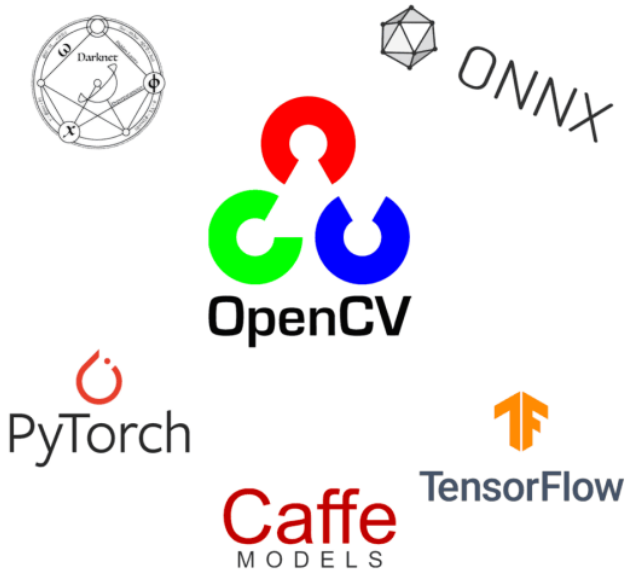


Hình 2. 18: Giao diện OPENCV

OPENCV là một thư viện mã nguồn mở rất phổ biến và hỗ trợ cả Python và C++. Điều này cho phép lập trình viên lựa chọn ngôn ngữ phù hợp với nhu cầu của mình. Ngoài ra, OPENCV cũng hỗ trợ linh hoạt trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS, giúp cho việc phát triển ứng dụng trở nên dễ dàng và tiện lợi hơn.

2.4.2 OPENCV DNN

Giao diện OPENCV DNN là một giao diện OPENCV được tích hợp thành phần giao diện DNN dùng để tương thích với các tài nguyên GPU (Graphic Processing Unit) và cho phép sử dụng các tài nguyên GPU có trên phần cứng nhằm mục đích tối đa hóa hiệu năng. Điểm ưu thế của giao diện OPENCV DNN là có thể tương thích với bất kì phần cứng nào, do đó đem lại sự linh hoạt và tiết kiệm thời gian xây dựng ứng dụng cho các nền tảng khác nhau.



Hình 2. 19: OPENCV hỗ trợ nhiều định dạng mô hình (nguồn Internet)

Ngoài ra với tính phổ biến OPENCV DNN hỗ trợ rất nhiều định dạng mô hình học sâu phổ thông hiện nay như darknet, caffe, pytorch, tensorflow, onnx, v.v.

2.5 PHẦN CỨNG JETSON NANO

2.5.1 Giới thiệu

Bộ phát triển NVIDIA jetson nano (NVIDIA jetson nano Developer Kit) được phát triển bởi nhà sản suất NVIDIA với tốc độ tính toán cao cùng với việc được tích hợp GPU trên bộ xử lý giúp cải thiện khả năng tính toán trong các ứng dụng liên quan tới AI (Artificial Intelligence) đã đem đến sự lựa chọn thích hợp để triển khai các ứng dụng dựa trên các mạng học sâu trên nền tảng jetson nano.

Hình 2. 20: NVIDIA Jetson nano Developer kit (nguồn Internet)

Bộ phát triển NVIDIA Jetson nano là một máy tính AI dành cho các các nhà phát triển và đem lại hiệu năng xử lý cho các ứng dụng trí tuệ nhân tạo hiện nay. Bắt đầu nhanh chóng với hỗ trợ sẵn có cho nhiều ứng dụng phổ biến thiết bị ngoại vi, tiện ích bổ sung và dự án sẵn sàng sử dụng.

Bộ phát triển NVIDIA Jetson nano được hỗ trợ toàn diện với bộ công cụ phát triển phần mềm NVIDIA Jetpack (NVIDIA Jetpack SDK), nhờ vào đó bộ phát triển NVIDIA Jetson nano có đủ hiệu suất và khả năng để chạy các tác vụ liên quan đến AI. Bộ phát triển phần mềm Jetpack sẽ bao gồm:

- Cung cấp tất cả các trình điều khiển dựa trên nhân hệ điều hành linux
- Cung cấp các thư viện chuyên dụng cho việc xử lý các tác vụ liên quan đến AI
- Cung cấp các công cụ phát triển
- Cung cấp tài liệu và các ví dụ

2.5.2 Cấu hình phần cứng

Bộ phát triển NVIDIA jetson nano (NVIDIA jetson nano Developer Kit) hiện nay có rất nhiều phiên bản được bán trên thị trường nhằm đáp ứng các nhu cầu phát triển và triển khai các ứng dụng lớn nhỏ ở cấp độ nghiên cứu và học tập. Điển hình nhất là hai phiên bản Bộ phát triển NVIDIA jetson nano A02 và B01 được sử dụng phổ biến cho các ứng dụng và nghiên cứu liên quan tới lĩnh vực thị giác máy tính. Dưới đây là thông số cơ bản của Bộ phát triển NVIDIA jetson nano:

- Bộ xử lý trung tâm (CPU): 128 lõi Maxwell
- Bộ xử lý đồ họa (GPU): Quad-core lõi ARM A57 @ 1.43GHz
- Bộ nhớ tạm thời (RAM): 4GB hoặc 2GB LPDDR4
- Camera: Hỗ trợ cổng MIPI CSI-2
- Các ngoại vi khác: USB, Ethernet, HDMI, GPIO, I2C, UART, v.v
- Nguồn cung cấp: cổng microUSB (5V 2.5A) hoặc cổng DC (5V 4A)

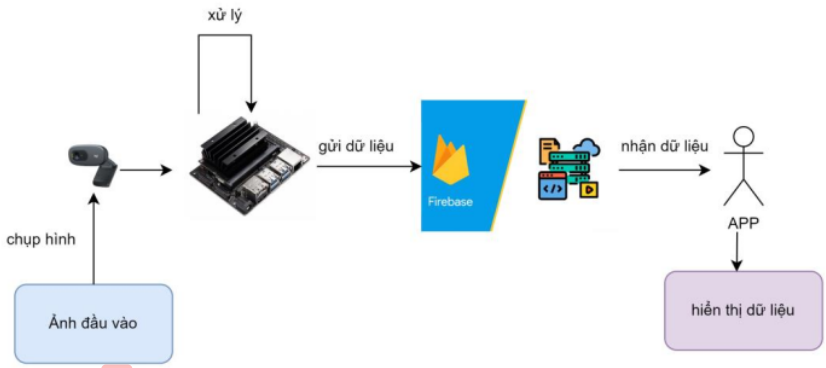
CHƯƠNG 3

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. Hệ thống sẽ hữu ích trong việc giám sát các kệ hàng từ xa mà nhân viên của các cửa hàng không cần thiết phải đi đến vị trí đặt kệ hàng, điều này giúp giảm thiểu được số lượng nhân viên giám sát, đồng thời các vị trí OOS sẽ được phát hiện kịp thời bởi các nhân viên tại cửa hàng sớm nhất để nhân viên có thể đưa ra biện pháp giải quyết giúp giảm thiểu đi tình trạng OOS quá lâu trên kệ hàng gây ảnh hưởng thiệt hại về mặt kinh tế cho cửa hàng.



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

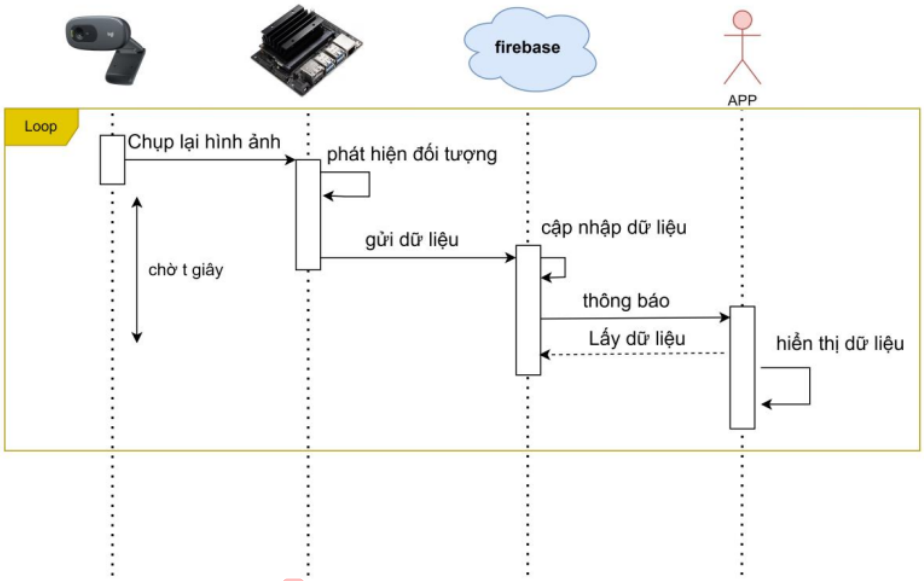
Mô hình hệ thống được thể hiện như hình 3.1, hệ thống sẽ bao gồm máy ảnh được đặt tại vị trí đối diện các kệ hàng và được kết nối đến máy tính nhúng Jetson nano. Hình ảnh sẽ được hệ thống chụp từ camera về và đưa hình ảnh của kệ hàng hiện tại về máy tính nhúng Jetson nano xử lý phát hiện các vị trí OOS, IS cũng như tính toán được tỉ lệ hàng có sẵn trên kệ hàng. Kết quả thu được sẽ được gửi về hệ thống đám mây lưu trữ thời gian thực đám mây từ xa thông qua giao thức wi-fi. Khi có sự cập nhập dữ liệu từ hệ thống đám mây lưu trữ thời gian thực sẽ gửi thông báo đến ứng dụng người dùng, từ đó ứng dụng người dùng sẽ cập nhập dữ liệu từ hệ thống đám mây lưu trữ thời gian thực và hiển thị dữ liệu OOS, IS và tỉ lệ hàng có sẵn trên kệ hàng để nhân viên tại các cửa hàng tiện lợi hoặc các siêu thị có thể khắc phục tình trạng OOS kịp thời.

3.1.2 Nguyên lý hoạt động

Nguyên lý hoạt đông 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

Sơ đồ trình tự hoạt động của hệ thống phát hiện OOS và IS được trình bày như hình 3.2. Trình tự hoạt động của hệ thống phát hiện OOS và IS sẽ được biễu diễn như sau:

- Bước 1: Thành phần thu thập dữ liệu (máy ảnh) sẽ chụp lại hình ảnh của kệ hàng tại thời điểm cụ thể (thời điểm cách nhau mỗi lần chụp là t giây).
- Bước 2: Dữ liệu hình ảnh thu thập được từ thành phần thu thập dữ liệu sẽ được gửi về cho thành phần xử lý dữ liệu (máy tính nhúng Jetson nano) và tiến hành xử lý phát hiện các vị trí OOS và IS từ hình ảnh thu được dựa trên mô hình được huấn luyện bởi mạng YOLO. Sau đó kết quả sẽ được tổng hợp lại và gửi dữ liệu

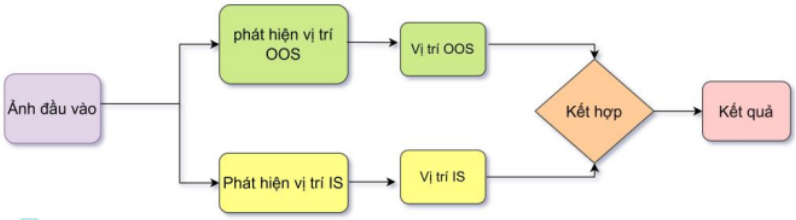
sau khi được xử lý lên dịch vụ lưu trữ đám mây thời gian thực (firebase).

- Bước 3: Thành phần lưu trữ (Dịch vụ lưu trữ đám mây thời gian thực firebase) sẽ cập nhập dữ liệu được gửi từ thành phần xử lý và tạo một tín hiệu thông báo đến thành phần hiển thị.
- Bước 4: Thành phần hiển thị (Giao diện ứng dụng) sẽ tiến hành cập nhập lại dữ liệu từ thành phần lưu trữ sau khi nhận được thông báo cập nhập dữ liệu từ thành phần lưu trữ và thực hiện tác vụ hiển thiện dữ liệu OOS và IS mới lên màn hình
- Bước 5: Hệ thống chờ đợi sau khoảng thời gian đặt trước t giây rồi quay lại thực hiện từ bước 1.

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

Mô hình xử lý phát hiện đối tượng trong đề tài này sẽ sử dụng kết hợp hai mô hình phát hiện đối với vị trí OOS và mô hình phát hiện các vị trí IS và thực hiện phân loại dựa trên các kết quả trả về từ hai mô hình nhận diện OOS và IS dựa trên mô hình huấn luyện mạng yolo.



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

Sơ đồ khối của mô hình xử lý phát hiện kết hợp hai vị trí OOS và IS được thể hiện như hình 3.3. Hình ảnh đầu vào được chụp từ máy ảnh sẽ được đưa vào độc lập lần lược hai mô hình phát hiện OOS và IS được huấn luyện dựa trên kiến trúc mạng yolo và đưa ra hai kế quả lần lược 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.

Bài toán phát hiện đối tượng được áp dụng dựa trên việc huấn luyện mô hai mô hình phát hiện hai đối vị trí riêng biệt OOS và IS, sau đó thực hiện bài toán phân loại từ hai kết quả biệt lập thu được bằng cách kết hợp hai kết quả vị trí lại và vẽ lại kết quả hộp giới hạn thu lên ảnh đầu vào. Từ đó cho ra được kết quả của mô hình nhận diện hai vị trí OOS và IS trên kệ hàng.

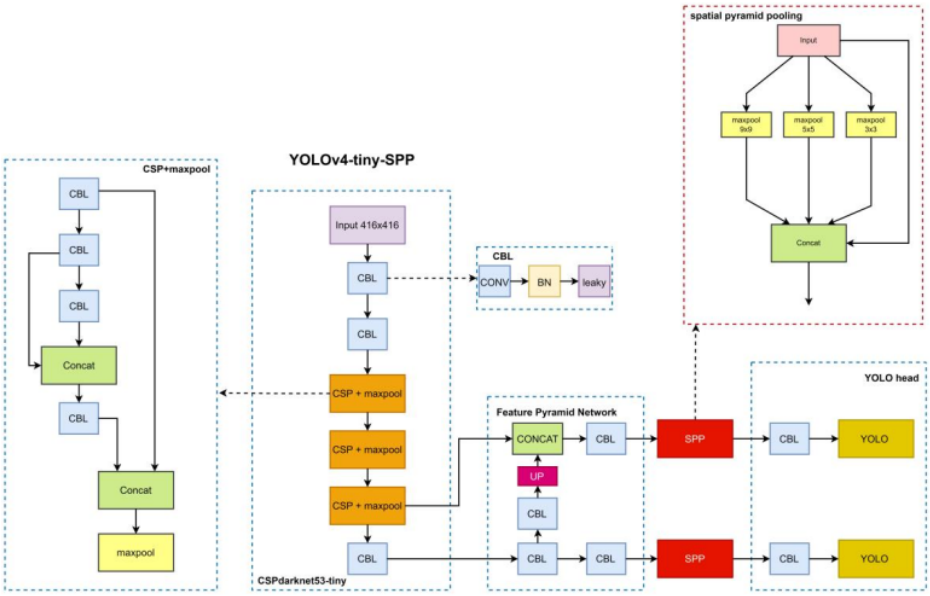
3.2.2 Đề xuất kiến trúc YOLOv4-tiny-SPP

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 data 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 maxpooling 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 maxpooling 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

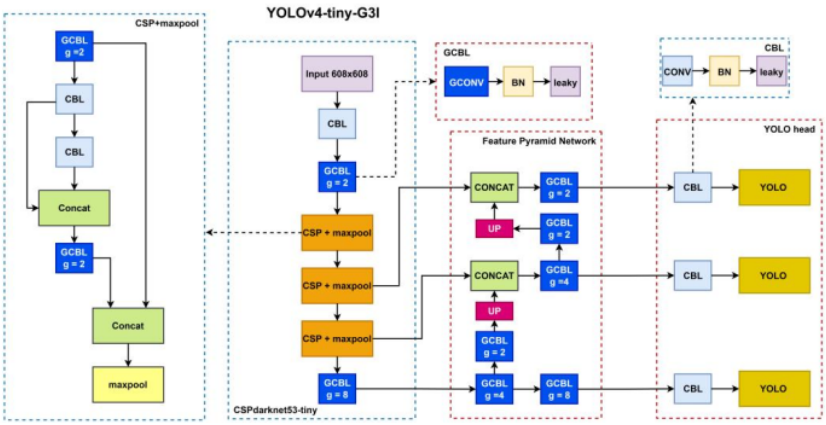
nhận diện *YOLO head*. Tuy nhiên kiến trúc YOLOv4-tiny-SPP sẽ có số lượng phép tính lớn hơn so với YOLOv4-tiny nhưng điều này cũng không đáng kể và ảnh hưởng đến độ ổn định của hệ thống. Bảng dưới đây sẽ đưa ra so sánh giữa YOLOv4-tiny và YOLOv4-tiny-SPP.

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

	BFLOPS
YOLOv4-tiny	6.787
YOLOv4-tiny-SPP	6.856

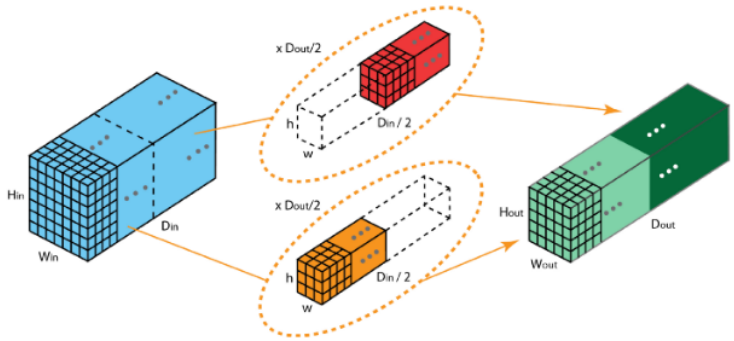
3.2.3 Đề xuất kiến trúc YOLOv4-tiny-G3l

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 (group CNN) để giảm tối đa số lượng phép tính BFLOPs mà vẫn giữ được độ chính xác của mô hình ở mức kì vọng.



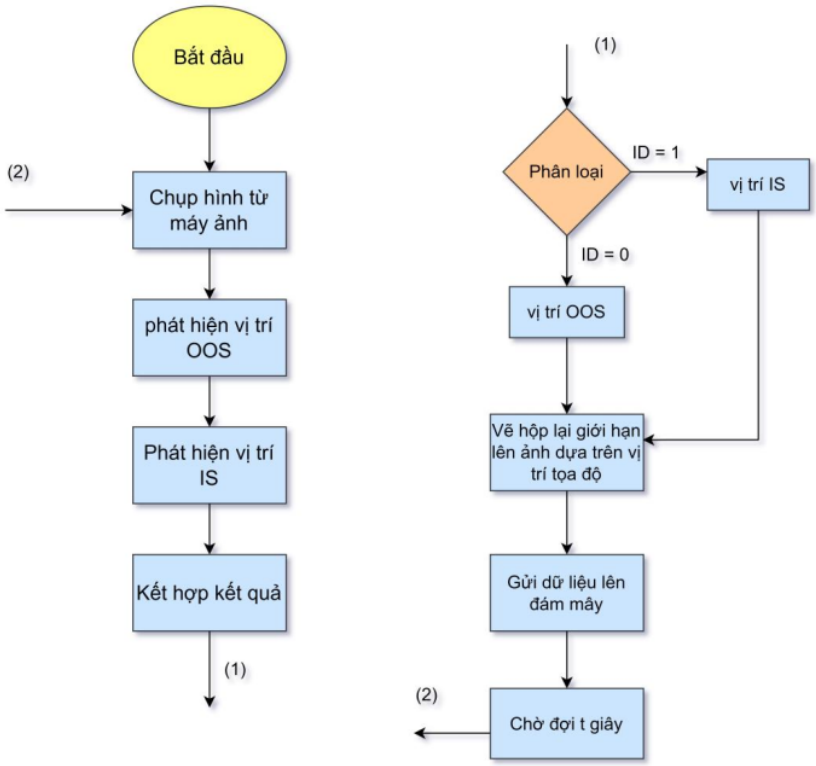
Hình 3. 6: Nhóm lớp tích chập (nguồn Internet)

Nhóm các lớp tích chập (group convolution) 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 CNN vào kiến trúc yolov4-tiny-G3l sẽ giảm tham số đi một nữa so với dùng CNN truyền thống.

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

	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 ảnh đầu vào từ thành phần thu thập dữ liệu (máy ảnh) và tạm thời lưu lại hình ảnh để xử lý. Dưới đây sẽ trình bày các bước xử lý cụ thể cho lưu đồ giải thuật:

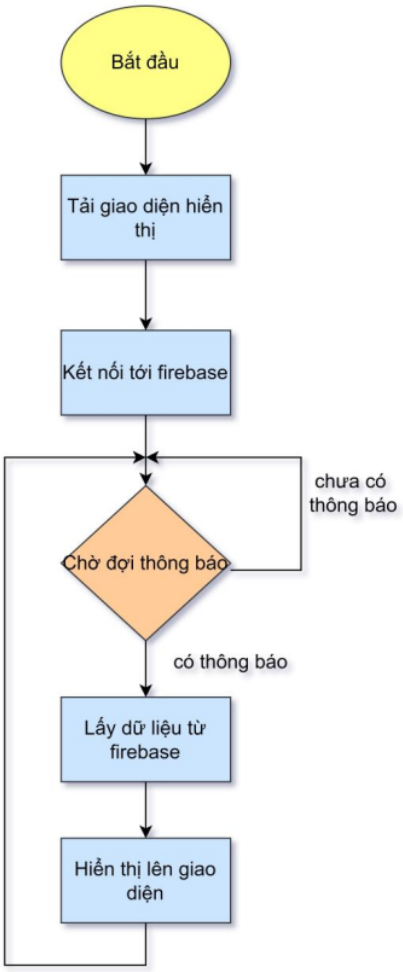
- Bước 1: Lấy dữ liệu hình ảnh được thu thập từ máy ảnh thông qua giao diện CSI hoặc USB.
- Bước 2: Đưa hình ảnh thu được qua mô hình xử lý phát hiện vị trí OOS và thu thập lại kết quả trả về.
- Bước 3: Đưa hình ảnh thu được qua mô hình xử lý phát hiện vị trí IS và thu thập lại kết quả trả về.

- Bước 4: Thực hiện quá trình kết hợp hai dữ liệu kết quả trả về từ bước 2 và bước 3.
- Bước 5: Tiến hành định vị dữ liệu vị trí OOS và IS bằng thông số ID và vẽ lại các hộp giới hạn từ dữ liệu trả về ở bước 4.
- Bước 6: Vẽ lại các hộp giới hạn bao quanh các vị trí phát hiện được dựa trên dữ liệu về tọa độ và vị trí thu được.
- Bước 7: Tiến hành gửi dữ liệu đã được xử lý lên trên dịch vụ lưu trữ đám mây thời gian thực.
- Bước 8: Chờ đợi trong khoảng thời gian t giây và quay về lại bước 1.

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.



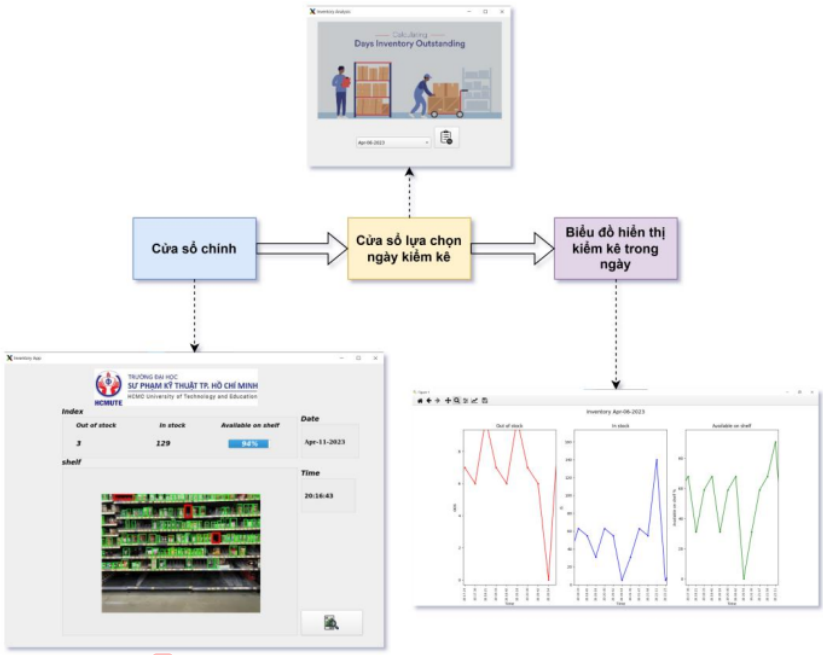
Hình 3. 8: Sơ đồ thuật toán trên giao diện hiển thị

Sơ đồ thuật toán xử lý trên giao diện được xây dựng dựa trên việc kết nối đến thành phần lưu trữ đám mây để lấy dữ liệu về đồng thời nhận thông báo mỗi khi có sự cập nhập dữ liệu từ thành phần lưu trữ. Dưới đây sẽ trình bày chi tiết về lưu đồ thuật toán:

- Bước 1: Khi khởi động hệ thống sẽ tải lên giao diện hiển thị được thiết kế trước đó và hiển thị ra màn hình.
- Bước 2: Hệ thống sẽ tiến hành kết nối tới thành phần lưu trữ đám mây thời gian thực.

- Bước 3: Chờ đợi thông báo có dữ liệu được cập nhập từ thành phần lưu trữ.
- Bước 4: Nếu có thông báo được gửi từ thành phần lưu trữ thì sẽ tiến hành lấy dữ liệu mới từ thành phần lưu trữ về và hiển thị dữ liệu mới lên trên giao diện sau đó lưu vào tệp csv. Nếu không có thông báo từ thành phần lưu trữ thì sẽ tiếp tục chờ ở bước 4.

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.

Giao diện cửa sổ chính sẽ hiển thị các chỉ số OOS, IS và tỉ lệ phần trăm hàng còn trên kệ cùng với hình ảnh của kệ hàng hiện tại được chụp từ phần cứng gửi qua đám mây và đưa về giao diện hiển thị. Tiếp đến khi lựa chọn vào phần kiểm kê người dùng sẽ lựa chọn theo ngày, các giá trị sẽ được lưu dưới dạng file csv và đồ thị sẽ được vẽ lại tại giao diện biểu đồ hiển thị kiểm kê trong ngày dựa vào các dữ liệu được ghi vào file csv.

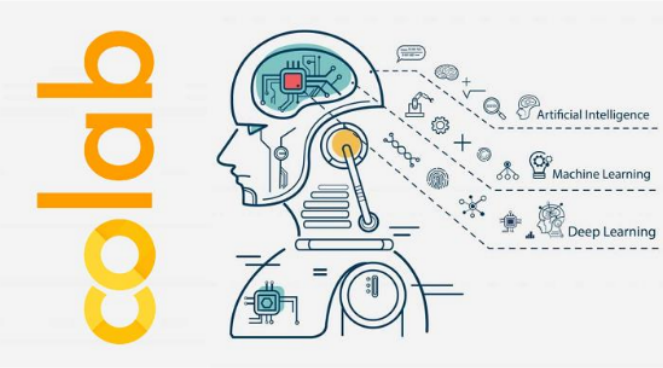
CHƯƠNG 4

THỰC HIỆN MÔ HÌNH VÀ ĐÁNH GIÁ

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

4.1.1 Giới thiệu google colab

Google colab là một hệ thống máy chủ được phát triển bởi google, google colab hỗ trợ người dùng có thể truy cập vào các máy chủ từ xa của google thông qua mạng không dây. Các máy chủ của google colab được phát triển đặc biệt dành cho các tác vụ huấn luyện và xây dựng các mô hình trí tuệ nhân tạo, các mô hình này thường đòi hỏi rất nhiều về sức mạnh tính toán của GPU. Từ đó tạo điều kiện cho các nhà phát triển đặt biệt trong lĩnh vực trí tuệ nhân tạo có thể triển khai các mô hình huấn luyện một cách hiệu quả và tiết kiệm thời gian.



Hình 4. 1: Google colab huấn luyện mô hình AI (nguồn Internet)

Với việc cung cấp GPU miễn phí và có sức mạnh tính toán nhanh, hiện nay google colab là lựa chọn hàng đầu cho các cá nhân nghiên cứu về

lĩnh vực trí tuệ nhân tạo mà không cần phải đầu tư các thiết GPU bị với giá thành rất cao trên thị trường.

4.1.2 Nền tảng darknet

Trong đề tài này sử dụng nền tảng darknet được xây dựng bởi tác giả Alexey cùng với đội nhóm AlexeyAB phát triển, vì vậy khi chuẩn bị các dữ liệu để huấn luyện mô hình nhận diện cần phải tuân theo các nguyên tắc mà nền tảng darknet đặt ra. Dưới đây sẽ trình bày các cụ thể trình tự cài đặt một nền tảng darknet dùng cho huấn luyện mô hình mạng YOLO.

Đầu tiên cần phải sử dụng công cụ git để tải bộ thư viện của darknet về máy tính hoặc google colab. Tại thư mục darknet sau khi tải về tiến hành cấu hình lại Makefile và sửa đổi các cấu hình thông số như sau:

- **OPENCV = 1:** tham số này sẽ chỉ định Makefile xây dựng nền tảng darknet sử dụng với thư viện OPENCV.
- **GPU = 1:** tham số này sẽ chỉ định cho Makefile xây dựng nền tảng darknet huấn luyện các mô hình trí tuệ nhân tạo sử dụng GPU nhằm tăng tốc quá trình huấn luyện.
- **CUDNN = 1, CUDNN_HALF = 1:** tham số này sẽ chỉ định cho Makefile xây dựng nền tnagr darknet huấn luyện sử dụng các nhân cuda trên GPU để tăng tốc khả năng tính toán giúp tiết kiệm thời gian huấn luyện.
- **LIBSO = 1:** tham số này chỉ định cho Makefile xây dựng nền tảng darknet bằng các thư viện liên kết.

Bảng 4. 1: Cài đặt thông số môi trường

Tham số môi trường	Giá trị tham số
OPENCV	1
GPU	1
CUDNN	1
CUDNN_HALF	1
LIBSO	1

Kết quả sau khi xây dựng nền tảng darknet sử dụng Makefile các thư mục đối tượng sẽ được lưu lại trong đường dẫn **/darknet/obj/** và công cụ darknet được sinh ra tại đường dẫn **/darknet/**, công cụ này sẽ dùng để chạy chương trình darknet cho việc huấn luyện mô hình trí tuệ nhân tạo. Các thông số cũng như cấu trúc của mô hình mạng huấn luyện sẽ được nằm trong đường dẫn **/darknet/cfg/**.

4.1.3 Cấu hình các thông số của mạng YOLO

Trong nền tảng darknet kiến trúc mạng học sâu sẽ được định nghĩa thông qua tệp cấu hình (.cfg) nằm tại đường dẫn **darknet/cfg/**. Dưới đây sẽ trình bày cụ thể về ý nghĩa của một số cú pháp để biểu diễn kiến trúc của mạng học sâu YOLO thông qua thư mục cấu hình.

Bảng 4. 2: Một vài cấu hình thông dụng trong darknet

Lớp	**Thông số cấu hình**	**Định nghĩa**
[net]		Lớp tổng quan của mạng dùng để cài đặt các thông số khi huấn luyện
	Width	Chiều ngang của ảnh đầu vào
	Height	Chiều dọc của ảnh đầu vào
	Channels	Số kênh màu sẽ được chuyển hóa

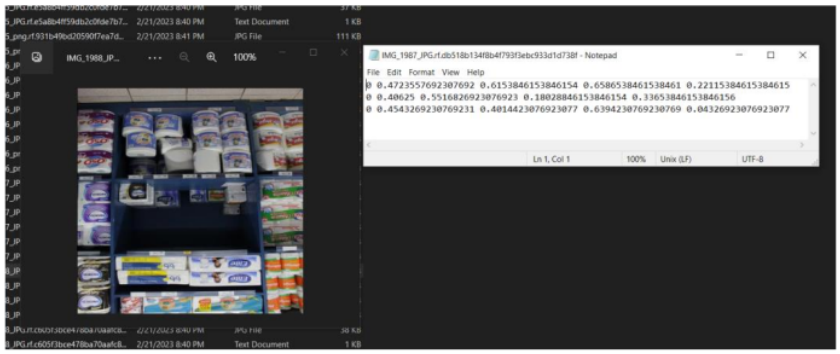
	Momentum	Thông số momentum thường được mặc định là 0.9
	Decay	Thông số dùng để cập nhập trọng số
	Learning_rate	Thông số này sẽ gây ảnh hưởng đến quá trính cập nhập trọng số
	Batch	Số lượng ảnh sẽ được load vào RAM của GPU
[convolution]		Đại diện cho lớp tích chập
	Filters	Số lượng tính chất của lớp tích chập
	Size	Kích thước lớp tích chập
	Group	Nếu thông số này được định nghĩa sẽ chỉ ra nhóm lớp tích chập (mặt định = 1)
	Stride	Thông số này sẽ giảm kích thước ảnh đi theo tỉ lệ mà nó được cài đặt
	Dilation	Nếu thông số này được cài đặc lớp tích chập sẽ sự dụng phương pháp sự giãn nở để thực hiện phép tính
	Activasion	Chỉ định hàm kích hoạt (leaky, mish, swish, v.v)
[maxpool]		Lớp gộp sẽ lấy giá trị lớn nhất trên vùng điểm ảnh đang xét
	Size	Kích thước lớp gộp maxpool
	Stride	Thông số này sẽ giảm kích thước ảnh đi theo tỉ lệ mà nó được cài đặt
[avgpool]		Lớp gộp sẽ lấy giá trị trung bình

		trên vùng điểm ảnh đang xét
[upsample]		Lớp này hỗ trợ tăng kích thước ảnh lên
	Stride	Thông số tỉ lệ tăng kích thước ảnh
[route]		Lớp này giúp gộp hai hoặc nhiều lớp có cùng kích thước lại với nhau
	layers	Chỉ định ra lớp nào sẽ được gộp lại với nhau
[yolo]		Lớp [yolo] cuối cùng dùng để dự đoán và phân loại vật thể
	Anchors	Dùng để khởi tạo các kích thước của hộp giới hạn
	Num	Tổng hộp giới hạn với các kích thước khác nhau
	classes	Chỉ ra số đối tượng nhận diện
	Ignore_thresh	Định nghĩa tỉ lệ trùng lặp đươc cho phép giữ lại
	Truth_thresh	Định nghĩa tỉ lệ độ tin cây khi phát hiện vật thể

Tùy vào mỗi kiến trúc của mạng học sâu mà các thông số trong thư mục cấu hình được cài đặt để tạo ra kiến trúc mạng trong lúc huấn luyện. Ngoài ra nền tảng darknet còn cung cấp rất nhiều các thông số cấu hình cho những mục đích khác nhau, tuy nhiên trong đề tài này chỉ đề cập tới một số thông số cũng như các lớp thông thường được sử dụng để xây dựng mạng học sâu.

4.1.4 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 điều phải được bỏ vào trong cùng một đường dẫn **darknet/data/obj**.



Hình 4. 2: Chuẩn bị dữ liệu huấn luyện

Tệp định dạng các hộp giới hạn có định dạng như hình 4.2. Với cột đầu tiên sẽ chứa số định danh cho đối tượng và các cột còn lại hiển thị lần lược điểm đầu, điểm cuối, chiều rộng và chiều dài của hộp giới hạn tương ứng với hình ảnh huấn luyện đã được đánh dấu sẵn.

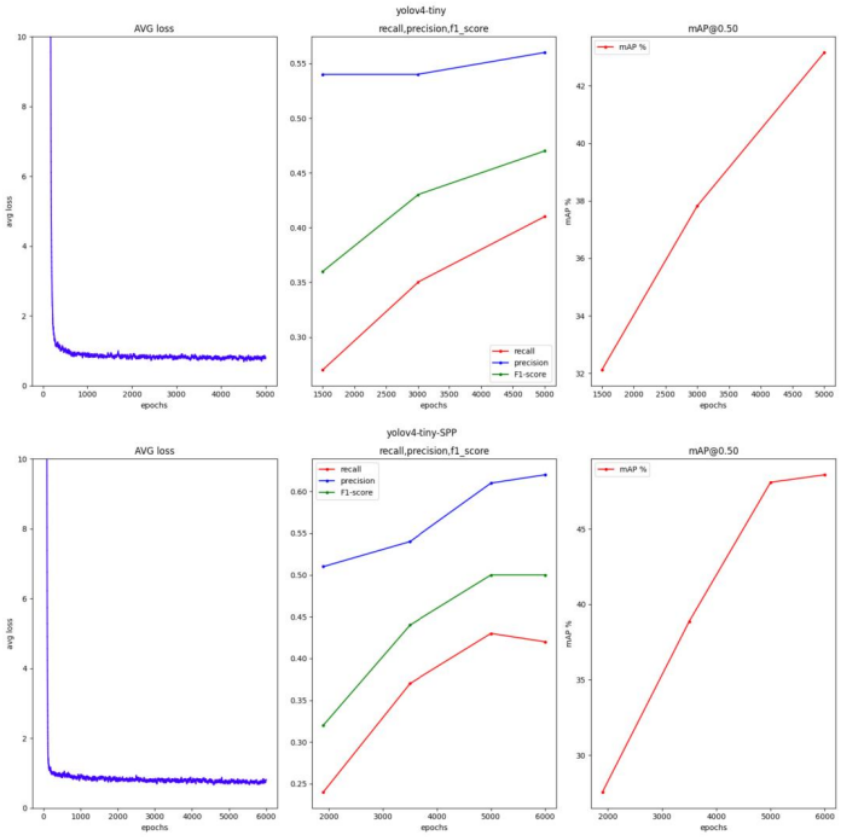
Tiếp theo tạo hai tệp **train.txt** và **test.txt** trong đó tệp **train.txt** sẽ chưa đường dẫn đến tất cả các ảnh dùng cho huấn luyện, còn tệp **test.txt** sẽ chứa tát cả đường dẫn tới ảnh dùng để đánh giá mô hình. Ngoài mô hình còn bắc buộc phải tạo thêm hai tệp **obj.data** và **obj.names** trong đó tệp obj.data sẽ chưa các thông tin như số lượng lớp phân loại, đường dẫn sao lưu và đường dẫn đến các tệp **train.txt**, **test.txt** và **obj.names**. Còn tệp **obj.names** sẽ chứa lần lược các tên của các lớp phân loại nhằm mục đích giúp mô hình nhận biết được đó là đối tượng nào. Tất cả các tệp này đều được đưa vào thư mục **darknet/data**.

Bảng 4. 3: Tệp obj.data.txt và obj.names.txt

Obj.data.txt	Obj.names.txt
Classes = *'số lượng lớp đối tượng'* Train = *'đường dẫn tới file train.txt'* Valid =*'đường dẫn tới file val.txt'* Names = *'đường dẫn tới file names.txt'* Backup=*'đường dẫn tới file thư mục sao lưu'*	*'Tên lớp đối tượng'*

4.1.5 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. 3: YOLOv4-tiny và YOLOv4-tiny-SPP trên tập ảnh OSS

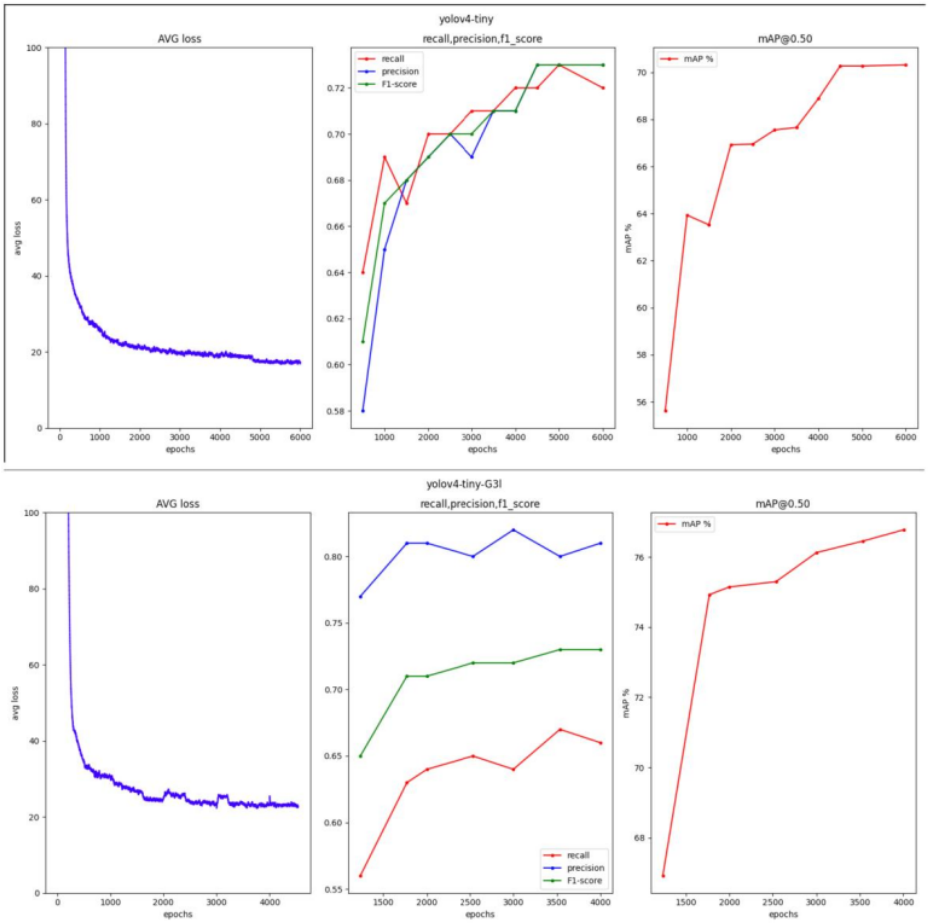
Dựa vào hình ảnh so sánh YOLOv4-tiny và YOLOv4-tiny-SPP trên tập ảnh phát hiện OSS có thể đánh giá được tỉ lệ accuracy của mô hình YOLOv4-tiny-SPP đã được cải thiện từ 41.63% lên 48.59% mặc dù độ phức tạp BFLOPs chỉ tăng 69 phép tính. Điều này cho thấy được sự hiệu quả của lớp SPP (spatial pyramind pooling) 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. 4 : So sánh mAP giữa các mô hình trên tệp 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 tệp 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. 4: YOLOv4-tiny và YOLOv4-tiny-G3l trên tập ảnh IS

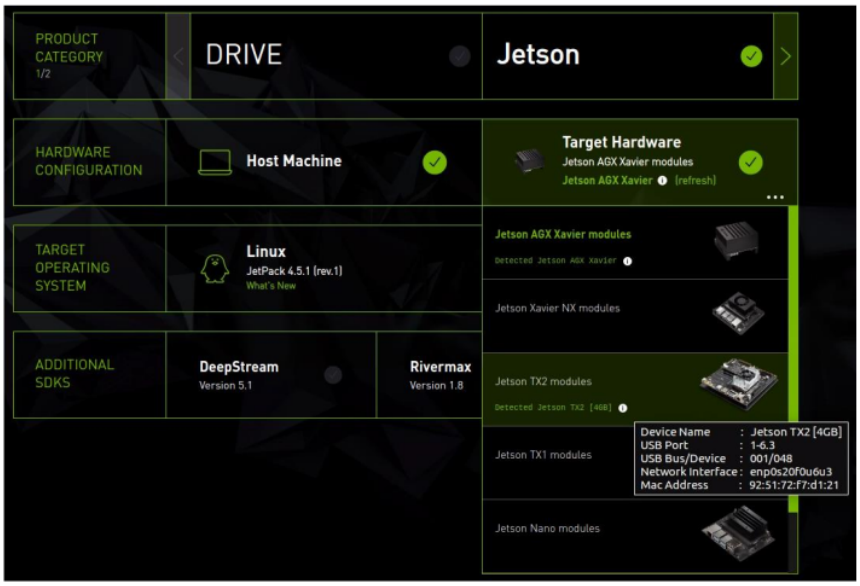
Dựa vào hình ảnh so sánh kết quả giữa YOLOv4-tiny và YOLOv4-tiny-G3l kết quả accuracy đã tăng từ 70.32% lên 76.01%, trong khi đó để đạt được mức accuracy cao hơn thì cần phải đánh đổi giữa BFLOPs tăng từ 6.787 lên 7.001. Tuy nhiên đối với ứng dụng của đề tài thì mức tăng độ phức tạp của mô hình như vậy 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. 5: So sánh mAP giữa các mô hình trên tệp 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 nữa 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 THIẾT LẬP PHẦN CỨNG JETSON NANO



Hình 4. 3: Giao diện bộ công cụ SDK manager

Để tiến hành cài đặt hệ điều hành cho máy tính nhúng Jetson nano trong đề tài này sẽ sử dụng bộ công cụ có sẵn được nhà phát triển Nvidia phát triển là SDK manager. Trong Bộ công cụ này đã được tích hợp sẵn nhân linux và các gói phụ trợ đi kèm hỗ trợ cho lĩnh vực ML như thư viện Cuda, OPENCV, v. v.

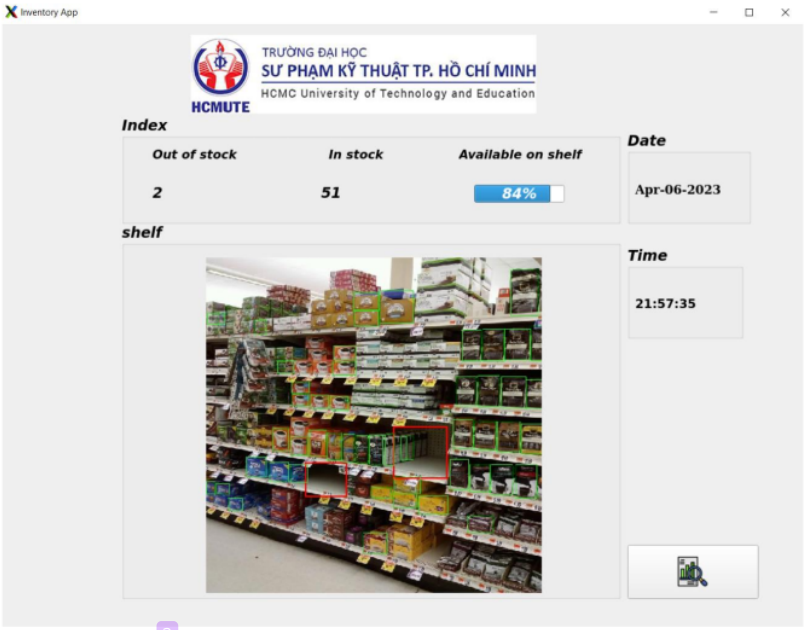
Để cài đặt môi trường để chạy ứng dụng trong đề tài này cần sử dụng thư viện OPENCV có hỗ trợ DNN. Tuy nhiên, giao diện DNN không được hỗ trợ cài bởi SDK manager, vì thế để có thể cài đặt được giao diện OPENCV DNN sử dụng GPU trên máy tính nhúng cần phải cài đặt OPENCV DNN thủ công từ mã nguồn của OPENCV DNN.

4.3 LẬP TRÌNH HỆ THỐNG

Tiến hành lập trình hệ cho hệ thống được chia thành hai thành phần, một là lập trình phần xử lý ảnh dùng ngôn ngữ python trên phần cứng jetson nano sử dụng thư viện OPENCV DNN kết hợp với hai mô hình được huấn luyện từ hai kiến trúc yolov4-tiny-SPP và yolov4-tiny-G3l để phát hiện các vị trí OOS và IS, đồng thời thu thập dữ liệu và gửi lên thành phần lưu trữ đám mây thời gian thực firebase theo trình tự luu đồ giải thuật đã được trình bày ở chương 3.

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 nhập 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.

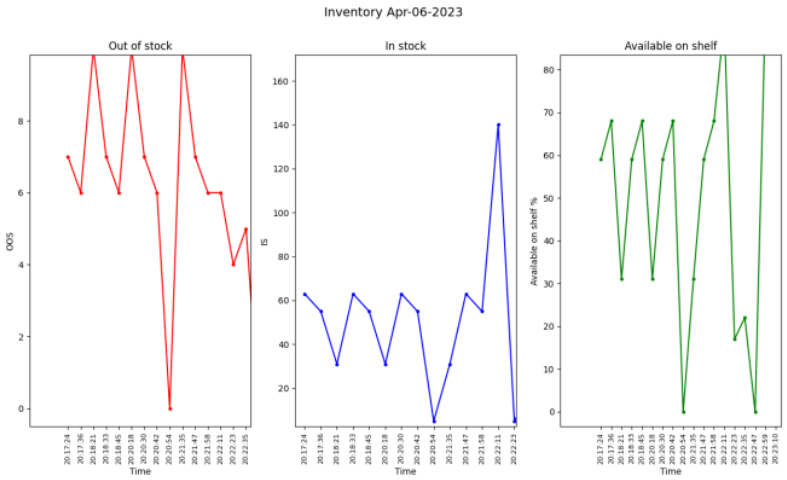


Hình 4. 4: Hình ảnh giao diện hiển thị

Giao diện hiển thị trạng thái kệ hàng bao gồm chỉ số OOS, IS và tỉ lệ mặt hàng có sẵn trên kệ đồng thời hiển thị thời gian như ngày và giờ tại thời điểm mà dữ liệu được cập nhập từ firebase. Đồng thời ứng dụng cũng lưu lại dữ liệu theo từng ngày phục vụ cho mục đích kiểm kê hàng hóa. Tỉ lệ hàng hóa có sẵn trên kệ được tính toán bằng công thức như sau:

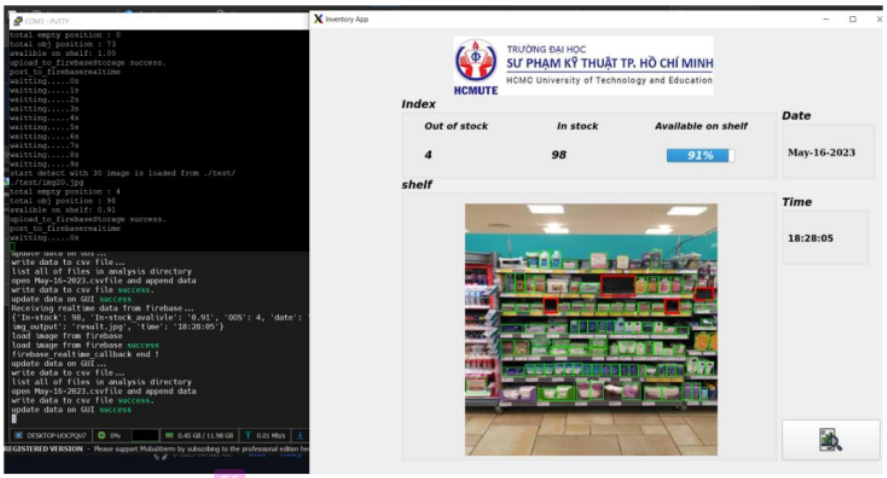
$$Available_on_shelf = \frac{S_{IS}}{S_{IS} + S_{OOS}}$$

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.

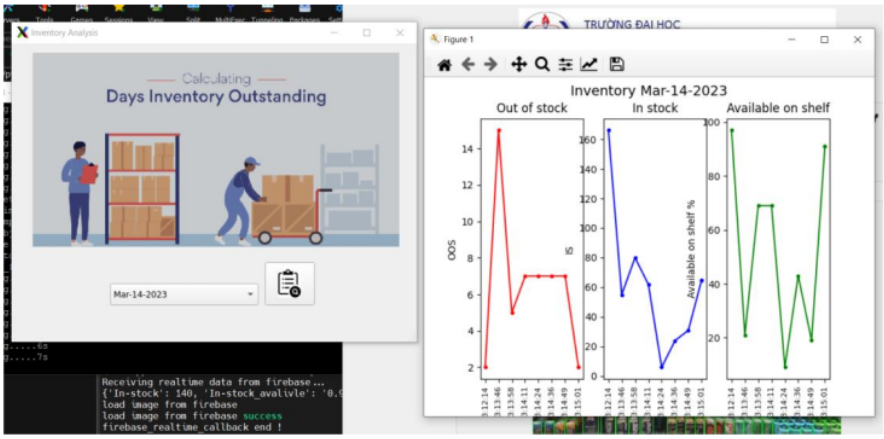


Hình 4. 5: Giao diện hiển thị biểu đồ kiểm kê hàng hóa

4.4 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 phần 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 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 tính 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

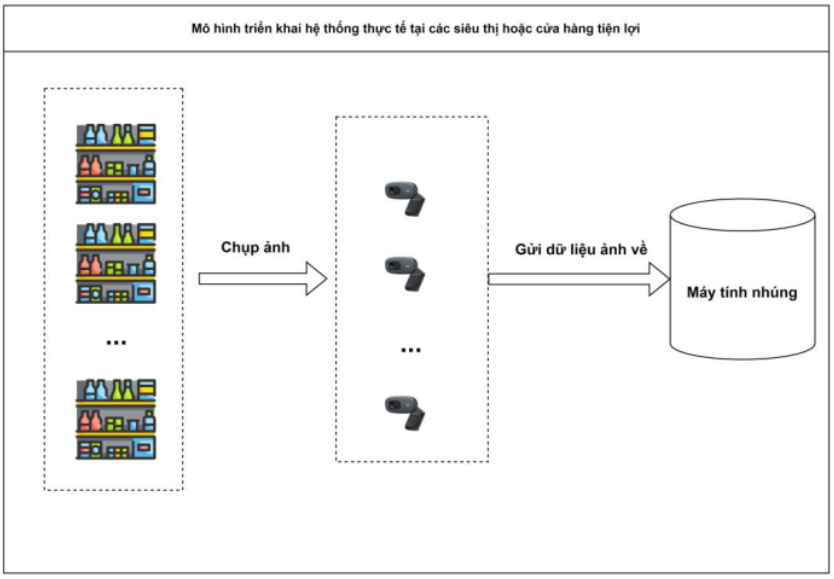
Tổng kết lại toàn bộ quá trình nghiên cứu và thực hiện đề tài ứng dụng học sâu cho việc giám sát và quản lý kiểm kê hàng hóa trong ngành bán lẻ đã đạt được các mục tiêu đề ra như sau:

- 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 kệ 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

LVTN_Huan

ORIGINALITY REPORT

29%	23%	16%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Ho Chi Minh University of Technology and Education Student Paper	3%
2	www.ctu.edu.vn Internet Source	2%
3	tailieu.vn Internet Source	2%
4	text.123docz.net Internet Source	1%
5	VNUA Publication	1%
6	thuvienso.bvu.edu.vn Internet Source	1%
7	Submitted to Vietnam Maritime University Student Paper	1%
8	Submitted to National Economics University Student Paper	1%
9	www.slideshare.net Internet Source	1%

10	Banking Academy Publication	1 %
11	vilas.edu.vn Internet Source	<1 %
12	text.xemtailieu.net Internet Source	<1 %
13	doc.edu.vn Internet Source	<1 %
14	My H. Nguyen, Doanh C. Bui, Nguyen D. Vo, Khang Nguyen. "SURVEY ON THE EFFICIENCY OF IOU AND L1 LOSS ON VEHICLE DETECTION PROBLEM IN AERIAL IMAGES USING DETR", PROCEEDINGS OF THE 14TH NATIONAL CONFERENCE ON FUNDAMENTAL AND APPLIED INFORMATION TECHNOLOGY RESEARCH, 2021 Publication	<1 %
15	Hanoi University Publication	<1 %
16	daotao.vku.udn.vn Internet Source	<1 %
17	mientayvn.com Internet Source	<1 %
18	Submitted to Ho Chi Minh City Open University Student Paper	<1 %

19	mic.gov.vn Internet Source	<1 %
20	vieclam.vietnamnet.vn Internet Source	<1 %
21	www.irjmets.com Internet Source	<1 %
22	Submitted to Industrial University of Ho Chi Minh City Student Paper	<1 %
23	xemtailieu.net Internet Source	<1 %
24	www.svcl.ucsd.edu Internet Source	<1 %
25	glints.com Internet Source	<1 %
26	portal.ptit.edu.vn Internet Source	<1 %
27	www.thuvientailieu.vn Internet Source	<1 %
28	Hanoi Pedagogycal University 2 Publication	<1 %
29	Submitted to Nha Trang University Student Paper	<1 %
30	123docz.net Internet Source	

		<1 %
31	Phan Anh Cang, Phan Thuong Cang, Tran Ho Dat. "LARGE-SCALE LUNG LESION DETECTION WITH CONVOLUTIONAL NEURAL NETWORKS IN SPARK", PROCEEDINGS OF THE 14TH NATIONAL CONFERENCE ON FUNDAMENTAL AND APPLIED INFORMATION TECHNOLOGY RESEARCH, 2021 Publication	<1 %
32	Baranidharan Balakrishnan, Rashmi Chelliah, Madhumitha Venkatesan, Chetan Sah. "Comparative Study On Various Architectures Of Yolo Models Used In Object Recognition", 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2022 Publication	<1 %
33	fwt.fialda.com Internet Source	<1 %
34	tckh.dlu.edu.vn Internet Source	<1 %
35	luanvan.co Internet Source	<1 %
36	Rapti Chaudhuri, Suman Deb. "Machine Learning approaches for Microscopic Image analysis and Microbial Object Detection(MOD)	<1 %

as a decision support system", 2022 First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR), 2022
Publication

37	www.researchgate.net Internet Source	<1 %
38	Submitted to Gitam University Student Paper	<1 %
39	www.acornstartup.com Internet Source	<1 %
40	techinsight.com.vn Internet Source	<1 %
41	Submitted to University of Finance – Marketing Student Paper	<1 %
42	viblo.asia Internet Source	<1 %
43	www.turcomat.org Internet Source	<1 %
44	github.com Internet Source	<1 %
45	Manh Cuong Dong. "Forecasting economic growth at provincial level in Vietnam: A systematic dynamics modeling approach",	<1 %

VNU JOURNAL OF ECONOMICS AND BUSINESS, 2022
Publication

46	Nguyen Hai Yen, Nguyen Thi Dinh, Nguyen Van Thinh, Van The Thanh, Le Manh Thanh. "A SEMANTIC-BASED IMAGE RETRIEVAL MODEL BASE ON k-NN ALGORITHM AND BAG OF WORD FEATURES", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XIII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN - Proceedings of the 13th National Conference on Fundamental & Applied Information Technology Research, 2020 Publication	<1 %
47	docs.microsoft.com Internet Source	<1 %
48	luanvan.net.vn Internet Source	<1 %
49	123doc.org Internet Source	<1 %
50	Lê Trung Kiên, Phạm Thị Ngọc Mỹ, Nguyễn Hoài Quốc Trung, Phạm Hoàng Anh. "ỨNG DỤNG BLOCKCHAIN ĐỂ TĂNG CƯỜNG TÍNH TOÀN VẸN VÀ BẢO MẬT TRONG QUẢN LÝ LƯU TRỮ VÀ CHIA SẺ DỮ LIỆU IOT", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA	<1 %

LẦN THỨ XII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN, 2019
Publication

---

51 Phan Anh Cang, Le Thi Nguu Huynh, Phan Thuong Cang. "CLASSIFICATION OF THE LIVER LESIONS BY HOUNSFIELD VALUES AND DEEP LEARNING TECHNIQUES", PROCEEDINGS OF THE 14TH NATIONAL CONFERENCE ON FUNDAMENTAL AND APPLIED INFORMATION TECHNOLOGY RESEARCH, 2021
Publication

<1 %

---

52 Phan Anh Cang, Tran Phan An Truong, Vo Van Quyen, Phan Thuong Cang. "CLASSIFICATION AND QUANTIFICATION OF BRAIN HEMORRHAGIC LESIONS USING HOUNSFIELD AND R-FCN", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XIII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN - Proceedings of the 13th National Conference on Fundamental & Applied Information Technology Research, 2020
Publication

<1 %

---

53 Song Luong, Nguyen Vo, Khang Nguyen. "EVALUATING VIDEO FACE RECOGNITION SYSTEM STREAM FROM DRONE", PROCEEDINGS OF THE 14TH NATIONAL CONFERENCE ON FUNDAMENTAL AND

<1 %

APPLIED INFORMATION TECHNOLOGY RESEARCH, 2021
Publication

54	ban-palletnhua.blogspot.com Internet Source	<1 %
55	ijesty.org Internet Source	<1 %
56	www.berrybase.de Internet Source	<1 %
57	blog.vietnamlab.vn Internet Source	<1 %
58	codegym.vn Internet Source	<1 %
59	nthbinh.weebly.com Internet Source	<1 %
60	ved2.topcv.vn Internet Source	<1 %
61	Linh Nguyễn Tuấn, Cuong Pham, Thủy Nguyễn Văn. "Phát hiện hoạt động bất thường của người bằng mạng học sâu tích chập kết hợp mạng bộ nhớ dài ngắn", Journal of Research and Development on Information and Communication Technology, 2020 Publication	<1 %
62	Submitted to Vietnam Commercial University Student Paper	<1 %

63	ddt.hitu.edu.vn Internet Source	<1 %
64	oa.upm.es Internet Source	<1 %
65	pdfcookie.com Internet Source	<1 %
66	peerj.com Internet Source	<1 %
67	Submitted to Hoa Sen University Student Paper	<1 %
68	Submitted to International University - VNUHCM Student Paper	<1 %
69	Nguyễn Văn Phương, Đào Khánh Hoài, Tống Minh Đức. "Tăng tốc độ phát hiện dị thường trên ảnh đa phổ và siêu phổ ứng dụng trong tìm kiếm cứu nạn", Research and Development on Information and Communication Technology, 2019 Publication	<1 %
70	Submitted to University of St Mark and St John Student Paper	<1 %
71	ceur-ws.org Internet Source	<1 %

72	fastest.com.vn Internet Source	<1 %
73	khcn.vimaru.edu.vn Internet Source	<1 %
74	www.stdio.vn Internet Source	<1 %
75	Submitted to Da Nang University of Economics Student Paper	<1 %
76	Hà Mạnh Hùng, Đặng Việt Hưng. "Phương pháp đa nhiệm xác định hư hỏng trong kết cấu giàn sử dụng dữ liệu định hướng", Tạp chí Khoa học Công nghệ Xây dựng (KHCNXD) - ĐHXD, 2021 Publication	<1 %
77	baotintuc.vn Internet Source	<1 %
78	hanghoavacongluan.vn Internet Source	<1 %
79	ictnews.vietnamnet.vn Internet Source	<1 %
80	khambenhnamkhoa.vn Internet Source	<1 %
81	khoahocvacongnghevietnam.com.vn Internet Source	<1 %

82	thuvien.hce.edu.vn:8080 Internet Source	<1 %
83	tusach.thuvienkhoahoc.com Internet Source	<1 %
84	vnresource.vn Internet Source	<1 %
85	www.petrovietnam.com.vn Internet Source	<1 %
86	www.tinhhuongkhogo.com Internet Source	<1 %
87	www.warse.org Internet Source	<1 %
88	3gmobifones.com Internet Source	<1 %
89	Submitted to Hanoi Pedagogical University 2 Student Paper	<1 %
90	Lê Hữu Vinh, Nguyễn Đình Thuân. "DỰ BÁO GIÁ BITCOIN BẰNG KẾT HỢP MÔ HÌNH ARIMA VÀ MẠNG NƠRON", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN, 2019 Publication	<1 %
91	allimage.vn Internet Source	<1 %

92	coin98.net Internet Source	<1 %
93	eudl.eu Internet Source	<1 %
94	gpsolar.vn Internet Source	<1 %
95	in5day.com Internet Source	<1 %
96	m.antoanthongtin.vn Internet Source	<1 %
97	moha.gov.vn Internet Source	<1 %
98	myshop.com.vn Internet Source	<1 %
99	nhathoang.vn Internet Source	<1 %
100	taichatzalo.net Internet Source	<1 %
101	tailieufree.vn Internet Source	<1 %
102	vietreview.vn Internet Source	<1 %
103	vn.sputniknews.com Internet Source	<1 %

104	vneconomy.vn Internet Source	<1 %
105	www.cione.com.vn Internet Source	<1 %
106	www.duongbo.vn Internet Source	<1 %
107	www.giaiphaptoandien.com Internet Source	<1 %
108	www.thongtincongnghe.com.vn Internet Source	<1 %
109	www.vinaheadhunter.com Internet Source	<1 %
110	www.yellowpages.vnn.vn Internet Source	<1 %
111	xehyundaidothanh.com Internet Source	<1 %
112	Phan Anh Cang, Nguyễn Thanh Hoàng, Trần Hồ Đạt, Nguyễn Văn Hiếu, Phan Thượng Cang. "NHẬN DẠNG HÌNH ẢNH THỰC PHẨM BẰNG PHƯƠNG PHÁP DEEP LEARNING", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XIII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN - Proceedings of the 13th National Conference on Fundamental &	<1 %

Applied Information Technology Research, 2020
Publication

113	Phạm T.T., Nông Nguyễn K.N., Ngô H.C.. "Lợi ích phi các-bon từ rừng: Góc nhìn cần xem xét trong quá trình xây dựng chính sách phát triển lâm nghiệp mới", Center for International Forestry Research (CIFOR) and World Agroforestry Centre (ICRAF), 2020 Publication	<1 %
114	Ping Li, Taiyu Han, Yifei Ren, Peng Xu, Hongliu Yu. "Improved YOLOv4-tiny based on attention mechanism for skin detection", PeerJ Computer Science, 2023 Publication	<1 %
115	Submitted to University of Economics & Law Student Paper	<1 %
116	chuyenvochong.net Internet Source	<1 %
117	daitrangia.com.vn Internet Source	<1 %
118	haophuong.com Internet Source	<1 %
119	ictexhibitions.com.vn Internet Source	<1 %
120	quantrimang.com Internet Source	

		<1 %
121	scholar.archive.org Internet Source	<1 %
122	sites.google.com Internet Source	<1 %
123	thanhnien.vn Internet Source	<1 %
124	timtailieu.vn Internet Source	<1 %
125	vinacode.net Internet Source	<1 %
126	vnexpress.net Internet Source	<1 %
127	www.ba8.vn Internet Source	<1 %
128	www.cdtm.edu.vn Internet Source	<1 %
129	www.charliethellama.com Internet Source	<1 %
130	www.mofficeviettel.com Internet Source	<1 %
131	xehiendai.blogspot.com Internet Source	<1 %

132 (7-5-13) http://115.78.239.30/index.php/vi/component/content/article/general/38-khainiemvetuchugdh <1%
Internet Source

133 Bui Thanh Hung, Pham Hoang Phuong. "IMAGE SEARCH USING DEEP LEARNING METHOD", PROCEEDINGS OF THE 14TH NATIONAL CONFERENCE ON FUNDAMENTAL AND APPLIED INFORMATION TECHNOLOGY RESEARCH, 2021 <1%
Publication

134 Submitted to Foreign Trade University <1%
Student Paper

135 Nguyễn Chí Thanh, Võ Thị Huyền Trang, Lê Minh Hưng, Hoàng Lê Uyên Thục. "KẾT HỢP ĐẶC TRƯNG SÂU TRONG HỖ TRỢ CHẨN ĐOÁN UNG THƯ VÚ TRÊN NHŨ ẢNH X-QUANG", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XIII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN - Proceedings of the 13th National Conference on Fundamental & Applied Information Technology Research, 2020 <1%
Publication

136 Nguyễn Hiếu Cường. "Một lược đồ mới phát hiện ảnh giả mạo dựa trên luật Benford", Research and Development on Information and Communication Technology, 2017 <1%

Publication

137 Phan Anh Cang, Nguyễn Thị Mỹ Nga, Phan Thượng Cang. "CHẨN ĐOÁN XUẤT HUYẾT NÃO DỰA TRÊN CHỈ SỐ HOUNSFIELD VÀ KỸ THUẬT MẠNG NƠ-RON TÍCH CHẬP", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XI NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN, 2018 <1%

Publication

138 Phan Anh Cang, To Huynh Thien Truong, Cao Hùng Phi, Phan Thuong Cang. "FACE DETECTION AND IDENTIFICATION USING DEEP LEARNING TECHNOLOGY WITH CGAN AND SRGAN", PROCEEDINGS OF THE 14TH NATIONAL CONFERENCE ON FUNDAMENTAL AND APPLIED INFORMATION TECHNOLOGY RESEARCH, 2021 <1%

Publication

139 Phan Đình Duy, Nguyễn Văn Thanh, Vũ Đức Lung. "NHẬN DẠNG TRANG WEB CÓ NỘI DUNG KHIÊU DÂM DỰA TRÊN TEXT VÀ IMAGE", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN, 2019 <1%

Publication

140 Phạm Vũ Hồng Sơn, Nguyễn Đình Bảo. "Giám sát tiến độ và khối lượng thực tế bằng mô <1%

hình 3D trên dữ liệu đám mây thông qua nền tảng Autodesk Forge", Tạp chí Khoa học Công nghệ Xây dựng (KHCNXD) - ĐHXD, 2021
Publication

141	congngheweb.edu.vn Internet Source	<1 %
142	daichung10.blogspot.com Internet Source	<1 %
143	hocphanmemketoanfast.blogspot.com Internet Source	<1 %
144	intphcm.com Internet Source	<1 %
145	justpassion.net Internet Source	<1 %
146	khuvuonmini.com Internet Source	<1 %
147	kiendovn.net Internet Source	<1 %
148	lithaco.vn Internet Source	<1 %
149	luatvietnam.vn Internet Source	<1 %
150	mailcongty.blogspot.com Internet Source	<1 %

151	nasco.com.vn Internet Source	<1 %
152	nckh.huph.edu.vn Internet Source	<1 %
153	patentscope.wipo.int Internet Source	<1 %
154	phongkhamhongha.vn Internet Source	<1 %
155	researchgo.wordpress.com Internet Source	<1 %
156	solarpower.vn Internet Source	<1 %
157	taioperaminitiengvietchoandroid.blogspot.com Internet Source	<1 %
158	text.123doc.org Internet Source	<1 %
159	thianhphatgiao2020.com Internet Source	<1 %
160	thongtinphapluatdansu.edu.vn Internet Source	<1 %
161	thucphamchucnangvn.com Internet Source	<1 %
162	thuvienphapluat.vn Internet Source	<1 %

163	tinhte.vn Internet Source	<1 %
164	toihocdesignpattern.com Internet Source	<1 %
165	vaas.org.vn Internet Source	<1 %
166	vi.wikipedia.org Internet Source	<1 %
167	www.dieukhientudong.com Internet Source	<1 %
168	www.donamed.vn Internet Source	<1 %
169	www.ibom.com.vn Internet Source	<1 %
170	www.lephuongtruong.com Internet Source	<1 %
171	www.mypage.vn Internet Source	<1 %
172	www.nhandan.org.vn Internet Source	<1 %
173	www.pacisoft.vn Internet Source	<1 %
174	www.semtek.com.vn Internet Source	<1 %

175	www.tailieudaihoc.com Internet Source	<1 %
176	www.vanchuyenhangdidanang.com Internet Source	<1 %
177	www.zbook.vn Internet Source	<1 %
178	Đoàn Thiện Minh, Trần Văn Lăng, Văn Đình Vỹ Phương, Phan Mạnh Thường. "KỸ THUẬT HỌC SÂU ĐỂ GIẢI QUYẾT BÀI TOÁN CHẨN ĐOÁN BỆNH LAO PHỔI", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN, 2019 Publication	<1 %
179	Đào Vũ Hiệp, Mạc Đình Hiếu, Tống Việt Hùng, Vũ Văn Phán, Hà Quốc Trung. "IVA-EC: GIẢI PHÁP PHÂN TÍCH VIDEO THÔNG MINH DỰA TRÊN ĐIỆN TOÁN BIÊN", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XIII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN - Proceedings of the 13th National Conference on Fundamental & Applied Information Technology Research, 2020 Publication	<1 %
180	Hà Thị Phương Anh, Phạm Thế Phi, Đỗ Thanh Nghị. "ĐÓNG GÓP CỦA ĐẶC TRƯNG VĂN BẢN	<1 %

TRONG BÀI TOÁN PHÂN LỚP ẢNH", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XI NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN, 2018
Publication

---

181 Thị Uyên Nhi Nguyễn, Thế Thành Văn, Mạnh Thạnh Lê. "Tìm kiếm ảnh dựa vào Ontology", Journal of Research and Development on Information and Communication Technology, 2021
Publication

<1 %

---

182 Duong Phi Long, Nguyen Thanh Tuong Vi, Vo Trung Hieu, Vo Duy Nguyen, Nguyen Tan Tran Minh Khang. "TABLE DETECTION IN DOCUMENT IMAGES USING CORNERNET CORNER LOCATING METHODS", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XIII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN - Proceedings of the 13th National Conference on Fundamental & Applied Information Technology Research, 2020
Publication

<1 %

---

183 Hoang Van Hieu, Do Phuc. "BUILDING A KNOWLEDGE BASE FOR CHATBOT FOUNDATION USING BERT TECHNOLOGY", PROCEEDINGS OF THE 14TH NATIONAL CONFERENCE ON FUNDAMENTAL AND

<1 %

APPLIED INFORMATION TECHNOLOGY RESEARCH, 2021
Publication

---

184 Phạm Hải Sơn, Lê Hoàng Sơn, Byeongnam Yoon, Đặng Vũ Tuấn. "ĐỀ XUẤT KHUNG KIẾN TRÚC ỨNG DỤNG CHO CHÍNH PHỦ DI ĐỘNG DỰA TRÊN KIẾN TRÚC TỔNG THỂ TẠI VIỆT NAM", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XIII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN - Proceedings of the 13th National Conference on Fundamental & Applied Information Technology Research, 2020
Publication

<1%

---

185 Phạm T.T., Đào Thị L.C., Hoàng T.L., Nguyễn D.T., Lê M.T., Nông H.H., Đặng T.N.. "Tác động của chi trả dịch vụ môi trường rừng (PFES) tại Sơn La, Việt Nam", Center for International Forestry Research (CIFOR) and World Agroforestry Centre (ICRAF), 2018
Publication

<1%

---

186 Trần Văn Quang, Trần Đình Toàn, Lê Mậu Long, Lê Minh Hưng. "PHÂN LOẠI HẠCH UNG THƯ PHỔI BẰNG MÔ HÌNH 3D-CNN", KỶ YẾU HỘI NGHỊ KHOA HỌC CÔNG NGHỆ QUỐC GIA LẦN THỨ XIII NGHIÊN CỨU CƠ BẢN VÀ ỨNG DỤNG CÔNG NGHỆ THÔNG TIN - Proceedings of the 13th National Conference on

<1%

Fundamental & Applied Information Technology Research, 2020
Publication

187	Trần Đức Học, Lê Tấn Tài. "Tối ưu hóa ước tính mức tiêu thụ năng lượng trong các tòa nhà dựa trên các thuật toán trí tuệ nhân tạo", Tạp chí Khoa học Công nghệ Xây dựng (KHCNXD) - ĐHXD, 2020 Publication	<1 %
188	tiendientu.org Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

LVTN_Huan

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

PAGE 77