## ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG





Báo cáo đồ án chuyên ngành

ĐỀ TÀI: XÂY DỰNG VÀ ĐÁNH GIÁ CÁC MÔ HÌNH ỨNG DỤNG PHÁT HIỆN LỬA CHẠY TRÊN THIẾT BỊ JETSON NANO

Giảng viên hướng dẫn: **TS. Lê Kim Hùng**

` Lớp: **NT114.L11.MMCL**

Sinh viên thực hiện: **Vũ Hà Anh**

*TP HCM, Ngày 12 tháng 01 năm 2020*



**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

## MỤC LỤC

[ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG 1](#_Toc61188447)

[MỤC LỤC 3](#_Toc61188448)

[PHỤC LỤC HÌNH ẢNH 4](#_Toc61188449)

[TÓM TẮT ĐỀ TÀI 5](#_Toc61188450)

[CHƯƠNG 1: GIỚI THIỆU 6](#_Toc61188451)

[CHƯƠNG 2: TỔNG QUAN 7](#_Toc61188452)

[TÀI LIỆU THAM KHẢO 8](#_Toc61188453)

## PHỤC LỤC HÌNH ẢNH

Table of Contents

Type chapter title (level 1)1

Type chapter title (level 2)2

Type chapter title (level 3)3

Type chapter title (level 1)4

Type chapter title (level 2)5

Type chapter title (level 3)6

## TÓM TẮT ĐỀ TÀI

Tại đề tài này em tập trung vào các vấn đề chính như sau: chuẩn bị và gán nhãn các dữ liệu đưa vào, xây dựng mô hình phát hiện đám cháy, phát hiện vật thể trong thời gian thực qua Camera và đánh giá hiệu năng giữa các mô hình nhận diện vật thể. Các mô hình được sử dụng để phát hiện lửa trong đề tài này bao gồm: các phiên bản của Yolo (Yolov3, Yolov4, Yolov5), Darknet.Conv.29 chạy trên các nền tảng khác nhau, mô hình Tensorflow Lite được chuyển đổi từ Darknet.conv29. Tuy nhiên, các mô hình này chỉ thích hợp chạy trên các thiết bị có cấu hình cao với CPU và GPU mạnh, để đưa vào chạy trên các thiết bị cấu hình thấp ta cần phải qua các bước cấu hình và thiết lập khác nhau, các phương pháp sẽ được trình bày sau trong bản báo cáo này.

Thiết bị Jetson Nano phiên bản 2gb được em sử dụng để xây dựng các mô hình trong đồ án. Với bộ nhớ RAM chỉ có 2gb nên thiết bị luôn xảy ra tình trạng thiếu bộ nhớ trong quá trình phát hiện lửa. Để đảm bảo quá trình này được diễn ra thông suốt, em đã sử dụng một số cấu trúc rút gọn của các mô hình tuy nhiên vẫn đảm bảo độ chính xác ở trong mức ổn định khi đào tạo đưa dữ liệu vào. Đồng thời, việc chuẩn bị và gán nhãn dữ liệu cũng phải được đảm bảo cẩn thận, nhằm tránh các trường hợp mô hình nhận diện sai các vật thể. Ví dụ như khi thực thi mô hình phát hiện lửa, ánh đèn của bóng đèn điện thường bị nhận diện nhầm, cho nên việc lựa chọn hình ảnh cần lựa chọn kỹ càng, tránh đi những hình ảnh khiến mô hình bị hiểu nhầm trong khi tìm kiếm vật thể. Ngoài ra, giải pháp Tensorflow Lite cũng được em triển khai để tối ưu hóa mô hình nhằm đảm bảo độ ổn định của mô hình khi chạy trên thiết bị Jetson Nano.

Sau khi thực hiện xây dựng những mô hình được nói trên, kết luận được đưa ra là giải pháp sử dụng Tensorflow Lite là phương án thích hợp nhất trong việc phát triển ứng dụng phát hiện lửa trên thiết bị Jetson Nano nói riêng, cũng như các thiết bị có cấu hình thấp hơn so với Laptop và PC nói riêng. Mặc dù hệ điều hành Jetpack mà Jetson Nano đang sử dụng chưa hỗ trợ chạy trên GPU cho Tensorflow Lite, nhưng đây vẫn là giải pháp tối ưu nhất để chạy ứng dụng phát hiện lửa và dễ dàng tích hợp đưa ứng lên Web Server.

## CHƯƠNG 1: GIỚI THIỆU

Hiện nay để phát hiện lửa có rất nhiều phương pháp, ví dụ như: quan sát bằng mắt thường của con người, hệ thống vệ tinh, cảm biến không khí MQ-135, hệ thống xử lý hình ảnh,… Quan sát bằng mắt thường là một trong những phương pháp truyền thống, tuy nhiên không thiết thực khi có hỏa hoạn xảy ra. Hệ thống vệ tinh cần thời gian quét dài và không thể cung cấp hình ảnh đám cháy theo thời gian thực. Cảm biến MQ-135 mang lại sự nhầm lẫn giữa khói bụi từ môi trường bình thường và khói của đám cháy, hơn nữa việc phân biến số lượng cảm biến lớn ngoài tự nhiên không phù hợp với tính chất tiết kiệm chi phí. Từ các phương pháp này ta có thể thấy, kỹ thuật xử lý hình ảnh đem lại ưu điểm vượt trội, chỉ cần một máy quay chất lượng cao, ta có thể giám sát một cánh rừng vài hecta, đồng thời phương pháp này còn cung cấp chi tiết về ba yếu tố của đám cháy là màu sắc, chuyển động và kết cấu.

Trong đề tài này, em đánh giá hiệu năng phát hiện lửa của ba phiên bản Yolov3, Yolov4, Yolov5, bản cải tiến của Yolov4-tiny có tên là Darknet.Conv.29 và mô hình Tensorflow Lite được chuyển đổi từ Darknet.Conv.29. Từ kết quả so sánh sau quá trình đào tạo mô hình, nhận thấy rằng phiên bản càng về sau, mô hình Yolo có độ chính xác càng cao, tuy Yolov5 mang đến hiệu quả cao nhất nhưng một số thư viện trong mô hình này không tương thích với hệ điều hành Jetpack của thiết bị Jetson Nano. Vì thế, mô hình Tensorflow Lite là một lựa chọn thích hợp, tuy không hỗ trợ GPU cho Tensorflow Lite, nhưng kết quả trong lúc thực hiện phát hiện lửa Tensorflow Lite vẫn đem lại kết quả ổn định nhất, có tốc độ khung hình mỗi giây giao động từ 8 đến 10. Từ kết quả này, em tin rằng, nếu như có thể cấu hình mô hình này theo một định dạng khác như TensorRT đang được Nvdia (công ty sản xuất Jetson Nano) hỗ trợ thì có thể đạt kết quả tốt hơn nhiều.

Yolo là mô hình phát hiện vật thể nổi tiếng với khả năng xử lý hình ảnh trong thời gian thực, nhưng đối với vật thể nhỏ, nó luôn có tỷ lệ phát hiện chính xác thấp hơn. Tuy nhiên ở đề tài này, em đã có khắc nhược điểm này bằng cách đưa hình ảnh ngọn lửa có kích thước nhỏ vào tệp huấn luyện, điều chỉnh các thông số về kích thước, độ lọc ảnh,… trong mô hình

Đối với thiết bị Jetson Nano bản 2gb em sử dụng trong đề tài, với bộ nhớ ít cho nên việc chạy các mô hình trong một thời gian dài là bất khả thi. Việc ra đời các mô hình rút gọn, được biết với cái tên là tiny mang lại hiệu quả cao, rút ngắn thời gian đào tạo và có khả năng chạy trên các thiết bị cấu hình thấp. Các thực nghiệm trên thiết bị Jetson Nano là các mô hình Yolov3-tiny, Yolov4-tiny và Darknet.Conv.29.

## CHƯƠNG 2: TỔNG QUAN

#### Mô hình Yolo

##### Giới thiệu về mạng Yolo

Yolo là tên viết tắt từ cụm từ “You only look once”, tức là chúng ta chỉ cần nhần 1 lần, điều đó chứng minh hiệu quả của Yolo trong bài toán phát hiện vật thể. Khác với bài toán Classification chỉ có thể dự đoán nhãn của vật thể. Yolo giải quyết bài toán Object Detection, không chỉ có thể phát hiện nhiều vật thể với nhiều nhãn khác nhau mà Yolo còn xác định vị trí của từng vật thể đó. Do đó, Yolo có thể phát hiện nhiều vật thể có nhãn khác nhau trong một khung ảnh, điều này rất phù hợp cho vấn đề xác định vật thể trong thời gian thực được đặt ra trong đề tài này.

##### Lịch sử phát triển Yolo

Phiên bản đầu tiên của Yolo là Yolov1 được ra mắt vào tháng 5 năm 2016 của tác giả Joseph Redmon với paper “You only look once: Unified, Real-Time Object Detection”. Nó là bước ngoặt cho việc phát triển của các bài toàn phát hiện vật thể trong thời gian thực. Vào tháng 12 năm 2017 thì tác giả lại đưa ra một phiên bản khác mang tên Yolo9000.

Chưa đầy một năm sau, tháng 4 năm 2018, một phiên bản mới được phát triển với tên gọi Yolov3, đây là mô hình được sử dụng nhiều nhất cho đến thời điểm hiện tại, được hỗ trợ trên các nền tảng khác nhau: Pytorch, Darknet, Keras, Tensorflow Lite. Khác hẳn với hai phiên bản trước, ở phiên bản này, tốc độ huấn luyên mô hình được cải thiện đáng kể, dễ dàng cài đặt và cấu hình. Đặc biệt với nên tảng Darknet, người dùng có thể dễ dàng tạo ra một model riêng cho bản thân sau vài tiếng đồng hồ.

Vào đầu năm 2020, Alexey Bochkovskiy đã giới thiệu Yolov4, nó vượt trội hơn hẳn Yolov3 về độ chính xác trung bình, số khung hình trên một giây, tiết kiệm tài nguyên của thiết bị, thời gian đào tạo mô hình một cách đáng kể. Tính tới thời điểm hiện tại, tuy ra mắt chưa đầy một năm, nhưng số lượng các chủ đề về yolov4 luôn là một đề tài nóng trên các trang của ngành công nghệ thông tin.

Sau đó không lâu thì **Glenn Jocher** đã phát hành YOLOv5, có rất nhiều những tranh cãi xảy ra với cái tên Yolov5 này. Cho đến hiện nay, vẫn chưa có một Paper nào nói về vấn đề này. Phiên bản này hiện nay chỉ chạy được trên nền tảng Pytorch. Yolov5 là một mô hình khá gọn nhẹ, nhưng hiệu năng vẫn cao hơn các phiên bản trước.

#### Các đề tài liên quan

Hiện nay có hai mô hình về ứng dụng phát hiện lửa được tham khảo nhiều nhất, đó là:

* Đề tài **“Fire Detection CNN”** của tác giả **Toby Breakon** sử dụng mạng CNN hay còn được biết đến với tên gọi “mạng nơ-ron tính chập”, đây là một mô hình sử dụng bài toán Classification. Đề tài này dựa vào bản Tensorflow 1.5, với cách cài đặt và triển khai dễ dàng, người dùng không tốn thời gian để đào tạo mô hình. Mô hình này chỉ có thể nhân diện trong khung ảnh là có lửa hay không có lửa và không thể xác định vị trí ngọn lửa trong khung ảnh.
* Đề tài “**Fire Net**” của tác giả **Olafenwa Moses** sử dụng mô hình Yolov3 trên nền tảng Keras, các tập tin sau khi được huấn luyện sẽ lưu dưới định dạng .h5. Mô hình được chạy bằng các câu lệnh trong thư viên Python là ImageAI cũng do tác giả tạo ra, điều này giúp người dùng chỉ cần chạy mô hình với 6 dòng code. Tuy tác giả nói rằng đây là một mô hình của Yolov3 nhưng tác giả đã thay đổi các thông số về anchor, cho nên dẫn tới việc chỉ có thể sử dụng thư viện ImageAi để chạy mô hình này và không thể chuyển đổi mô hình sang chạy trên các nên tảng khác như Pytorch, Darknet. Một nhược điểm nữa chính là tất cả câu lệnh đều dựa trên thư viện ImageAi dẫn đến việc không tích hợp được OpenCV, không thể sử dụng mô hình này phát hiện lửa trên thời gian thực.

#### Các vấn đề đồ án thực hiện và giải quyết

* Trong quá trình thực hiện đồ án, em tập trung giải quyết các vấn đề: xử lý được bài toán phát hiện lửa (Object Detection) trong thời gian thực, tối ưu hóa hiệu suất mô hình Yolo ở các phiên bản để có thể thiết lập trên thiết bị Jetson Nano phiên bản 2gb nói riêng và các thiết bị có cấu hình thấp nói chung, đồng thời có thể phát hiện lửa với các input khác nhau như: hình ảnh, video, các giao thức stream (http, rtmp, rtsp) , camera,…
* Đưa được ứng dụng lên Web Server bằng Flask của Python, có khả năng Stream trên chương trình quản lý các thiết bị IOT là thingsboard, từ đó có thể mở rộng và phát triển đề tài.

## CHƯƠNG 3: CƠ SỞ LÝ THUYẾT