|  |
| --- |
| **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  **KHOA CÔNG NGHỆ THÔNG TIN**  \*\*\*\*\*\* Description: bs00975_ \*\*\*\*\*\*  Description: bieu tuong  **ĐỒ ÁN I**  **NHẬN DIỆN GIỐNG LOÀI DOG AND CAT**  Giáo viên hướng dẫn : Phan Thị Hải Hồng  SV thực hiện : Nguyễn Bá Hoàn      Hà Nội, 2020 |

|  |  |  |  |
| --- | --- | --- | --- |
| **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  **KHOA CÔNG NGHỆ THÔNG TIN**  **BỘ MÔN HỆ THỐNG THÔNG TIN** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **ĐỘC LẬP - TỰ DO - HẠNH PHÚC** |  |  |

# NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ và tên: NGUYỄN BÁ HOÀN Lớp: Hệ Thống Thông Tin Khoá: 15

Ngành: Hệ Thống Thông Tin Chuyên ngành: Hệ Thống Thông Tin Quản Lý

**1. Tên đề tài:** Nghiên cứu các kỹ thuật Phân loại giống chó mèo sử dụng mạng học sâu và ứng dụng.

**3. Nội dung thuyết trình:**

LỜI NÓI ĐẦU.

Chương I: TỔNG QUAN.

Chương II: DECTEC VÀ MẠNG HỌC SÂU CHO BÀI TOÁN PHÂN LOẠI GIỐNG DOG AND CAT

Chương III: THỰC NGHIỆM VÀ KẾT QUẢ.

KẾT LUẬN.

TÀI LIỆU THAM KHẢO.

**5. Cán bộ hướng dẫn:** Thiếu tá, GV,TS Phan Thị Hải Hồng, giảng viên bộ môn Khoa học máy tính, hướng dẫn toàn bộ.

Hà Nội, ngày 28 tháng 11 năm 2020.

**Chủ nhiệm bộ môn Cán bộ hướng dẫn**

(Ký, ghi rõ hột tên, học hàm, học vị)

**Học viên thực hiện**

Đã hoàn thành và nộp đồ án ngày 28 tháng 11 năm 2020

(Ký và ghi rõ họ tên)

MỤC LỤC

[NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP i](#_Toc57647966)

[LỜI CẢM ƠN iv](#_Toc57647967)

[DANH MỤC CÁC TỪ VIẾT TẮT v](#_Toc57647968)

[DANH SÁCH HÌNH ẢNH vi](#_Toc57647969)

[LỜI NÓI ĐẦU vii](#_Toc57647970)

[CHƯƠNG I: TỔNG QUAN 1](#_Toc57647971)

[1.1. Tóm tắt về bài toán. 1](#_Toc57647972)

[1.2. Trí tuệ nhân tạo và ứng dụng. 1](#_Toc57647973)

[1.3. Các loại thuật toán học máy 5](#_Toc57647974)

[1.4. Lý thuyết về thị giác máy tính (Computer vision): Tìm hiểu các phương pháp trích chọn và biểu diễn đặc trưng ảnh. 6](#_Toc57647975)

[CHƯƠNG II. DECTEC VÀ MẠNG HỌC SÂU CHO BÀI TOÁN NHẬN DẠNG GIỐNG LOÀI. 10](#_Toc57647976)

[2.1. Thuật toán phát hiện đối tượng ( Object detection ). 10](#_Toc57647977)

[2.1.1. YOLO 10](#_Toc57647978)

[2.1.2. YOLOv2 (2016) và YOLOv3 (2018) 11](#_Toc57647979)

[2.1.3. Nhóm các mô hình họ YOLO. 13](#_Toc57647980)

[2.1.4. Cách hoạt động của YOLO. 13](#_Toc57647981)

[2.2. Các mô hình mạng học sâu cho bài toán. 16](#_Toc57647982)

[2.2.1. Mô hình MobileNetV1 16](#_Toc57647983)

[2.2.2. Mô hình MobileNetV2 20](#_Toc57647984)

[CHƯƠNG III. THỰC NGHIỆM VÀ ĐÁNH GIÁ 34](#_Toc57647985)

[3.1. Quá trình thực nghiệm. 34](#_Toc57647986)

[3.1.1. Xác định bài toán cần thực hiện. 34](#_Toc57647987)

[3.2. Các bước xây dựng và ứng dụng model train dữ liệu. 35](#_Toc57647988)

[KẾT LUẬN 40](#_Toc57647989)

[TÀI LIỆU THAM KHẢO 41](#_Toc57647990)

# LỜI CẢM ƠN

Để hoàn thành đồ án này, em xin chân thành gửi lời cảm ơn đến TS Phan Thị Hải Hồng, giảng viên trực tiếp phụ trách hướng dẫn đề tài đồ án. Sự hướng dẫn tận tình của cô đã giúp em có thêm nhiều kiến thức, khắc phục được những lỗi em thường mắc phải để đồ án được hoàn thiện tốt hơn. Qua những hướng dẫn tận tâm của cô đã giúp chúng em hiểu sâu hơn về những kiến thức đã học, và trau dồi thêm những kiến thức mới bổ ích.

Em xin chân thành gửi lời cảm ơn quý thầy cô trong bộ môn cũng như tất cả các thầy cô trong khoa Công nghệ thông tin của Học Viện Kỹ Thuật Quân Sự đã tận tình giúp đỡ và trang bị cho em những kiến thức cơ bản, cần thiết để em có thể hoàn thành tốt đồ án này.

Ngoài ra, tác giả xin gửi lời cảm ơn tới Ban Giám hiệu Học Viện Kỹ Thuật Quân Sự đã tạo điều kiện để em thực hiện tốt đồ án này.

Mặc dù có nhiều cố gắng để thực hiện đồ án tốt nghiệp một cách hoàn chỉnh nhất. Song do năng lực còn nhiều hạn chế nên không thể tránh khỏi những thiếu sót nhất định.

Và cuối cùng, em xin được gửi lời cảm ơn tới gia đình, người thân và bạn bè - những người đã luôn ở bên em những lúc khó khăn nhất, luôn động viên và khuyến khích em trong cuộc sống cũng như trong học tập và công việc.

Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các quý thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Xin trân trọng cảm ơn tất cả!

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| 1 | AI | Artificial intelligence |
| 2 | BPTT | Backpropagation Through Time |
| 3 | CV | Computer Vision |
| 4 | CNN | Convolutional Neural Network |
| 5 | DL | Deep Learning |
| 6 | Mnet V1 | MobileNetV1 |
| 7 | Mnet V2 | MobileNetV2 |
| 8 | IoU | Intersection over Union |
| 9 | MP | Multi-Scale |
| 10 | ML | Machine Learning |
| 11 | RNN | Recurrent Neural Network |
| 12 | R-CNN | Region with CNN feature |
| 13 | ROI | Region of Interest |
| 14 | YOLO | You Only Look Once |
| 15 | GPU | **Graphic Proccessing Unit** |
| 16 | GPGPU | General Purpose Computing on Graphics Processing Units |
| 17 | ASPP | Atrous Spatial Pyramid Pooling |

# DANH SÁCH HÌNH ẢNH

[**Hình 1. 1** Hình ảnh mình họa về trí tuệ nhân tạo 13](#_Toc57646333)

[**Hình 1. 2** Xe tự lái 14](#_Toc57646334)

[**Hình 1. 3** Xử lý ảnh được thực hiện trên ảnh tĩnh 17](#_Toc57646335)

[**Hình 1. 4** Lịch sử phát triển Dêp Learning 19](#_Toc57646336)

[**Hình 2. 1** Các bước xử lý trong mô hình YOLO 21](#_Toc57646353)

[**Hình 2. 2** Sơ đồ tạo prior bounding box 22](#_Toc57646354)

[**Hình 2. 3** Mô hình YOLO 23](#_Toc57646355)

[**Hình 2. 4** Classification loss 25](#_Toc57646356)

[**Hình 2. 5** Biểu diễn tọa độ của đối tượng 26](#_Toc57646357)

[**Hình 2. 6** Giá trị hàm lỗi 26](#_Toc57646358)

[**Hình 2. 7** Mô hình MobileNet V2 28](#_Toc57646359)

[**Hình 2. 8** Mô hình tổng quát MobileNetV2 30](#_Toc57646360)

[**Hình 2. 9** Độ chính xác mô hình với hệ số a 31](#_Toc57646361)

[**Hình 2. 10** So sánh hai mô hính MobileNetV1 và V2 32](#_Toc57646362)

[**Hình 2. 11** ReLU6 33](#_Toc57646363)

[**Hình 2. 12** Cấu trúc dư ngược 34](#_Toc57646364)

[**Hình 2. 13** Cấu trúc dư ngược 34](#_Toc57646365)

[**Hình 2. 14** cấu trúc MobileNetV2 35](#_Toc57646366)

[**Hình 2. 15** Kiến trúc tổng thể 36](#_Toc57646367)

[**Hình 2. 16** Kiến trúc tổng thể MobileNetV2 36](#_Toc57646368)

[**Hình 2. 17** So sánh các khối khác nhau cho kiến trúc 38](#_Toc57646369)

[**Hình 2. 18** So sánh kênh 39](#_Toc57646370)

[**Hình 2. 19** Tác động của nút cổ chai tuyến tính 39](#_Toc57646371)

[**Hình 2. 20** Tác động của phím tắt 40](#_Toc57646372)

[**Hình 2. 21** MobileNetV2 để Phân loại, Phát hiện và Phân đoạn 41](#_Toc57646373)

[**Hình 2. 22** Độ chính xác hàng đầu của ImageNet 42](#_Toc57646374)

[**Hình 2. 23** So sánh kích thước mô hình và tính toán 43](#_Toc57646375)

[**Hình 2. 24** SSDLite 43](#_Toc57646376)

[**Hình 2. 25** Phát hiện đối tượng MS COCO 44](#_Toc57646377)

[**Hình 2. 26** Bộ xác thực PASCAL VOC 2012 45](#_Toc57646378)

[**Hình 3. 1** Ảnh cần xử lý 47](#_Toc57646379)

[**Hình 3. 2** Ảnh trước và sau khi qua YoloV3 48](#_Toc57646380)

[**Hình 3. 3** Mô hình sau khi đã được tranfer 49](#_Toc57646381)

[**Hình 3. 4** Quá trình training với 20 epochs. 50](#_Toc57646382)

[**Hình 3. 5** Kết quả với 2 con dog 51](#_Toc57646383)

[**Hình 3. 6** Kết quả với nhiều dogs 51](#_Toc57646384)

# LỜI NÓI ĐẦU

1. Lý do chọn đề tài.

Trong những năm gần đây, AI hay còn được biết đến là Trí tuệ nhân tạo đang dần trở nên phổ biến và được nhiều người biết đến. Do đó việc xây dựng các ứng dụng trí tuệ nhân tạo đang là một ngành công nghiệp mới đầy tiềm năng và hứa hẹn nhiều sự phát triển vượt bậc của ngành khoa học kỹ thuật.

Máy tính vốn là một vật vô tri vô giác, nhưng ta có thể biến nó trở nên thông minh hơn bằng cách cài đặt những chương trình cho máy tính tự học, tự nhận diện và cũng từ đó sinh ra những dự đoán, những bước đi tiếp theo. Đó là ý tưởng cơ bản của Trí tuệ nhân tạo. Từ ý tưởng đó, đã có rất nhiều sản phẩm thành công như các phần mềm chơi cờ thông minh hay nhận diện khuôn mặt để mở khóa điện thoại.

Cùng với sự phát triển của khoa học công nghệ của cuộc cách mạng công nghiệp 4.0, các ứng dụng về trí tuệ nhân tạo (TTNT) và học máy ngày càng phát triển và là lĩnh vực được quan tâm nhiều nhất hiện nay, trong đó lĩnh vực học sâu (Deep Learning) là một loại phổ biến của máy học đã có những thành tựu phát triển vượt bậc. Nó đã mở ra một bước ngoặc mới trong việc giải quyết các bài toán về TTNT trước đây đã gặp phải khó khăn như nhận thức sự vật (object perception), nhận diện hình ảnh, hệ thống gợi ý (recommend system) trên các nền tảng dữ liệu lớn….Trong đó phải kể đến sự phát triển của lĩnh vực thị giác máy tính (computer vision), đặt nền tảng cho nhiều ứng dụng trong thực tiễn như các hệ thống xe tự hành, rôbôt thông minh, nhận dạng khuôn mặt người(Face Detect) Lĩnh vực học sâu (Deep Learning) là một kỹ thuật trong học máy sử dụng mạng nơron nhiều lớp để giải quyết các bài toán phức tạp dựa trên phương pháp tích chập và trích chọn các đặc trưng từ tập dữ liệu lớn và đã đem lại kết quả chính xác cao trong giải quyết các bài toán TTNT. Cụ thể là nó đã tập trung giải quyết các vấn đề liên quan đến mạng neural về thị giác máy tính, xử lý ngôn ngữ tự nhiên, xử lý giọng nói…Hiện nay, cũng có nhiều nghiên cứu về phân loại giống chó/mèo.Tuy nhiên đây là lĩnh vực này tương đối rộng và hiện có nhiều kỹ thuật khác nhau để giải quyết bài toán này. Vì vậy, trong phạm vi luận văn này tập trung nghiên cứu lý thuyết và sử dụng các kỹ thuật mạng nơ-ron, tích chập vào việc giải quyết bài toán nhận diện giống loài chó/mèo, tập trung hướng đến các giống loài phổ biến của chó và mèo. Thông qua đó nghiên cứu đề xuất cải tiến các tham số đầu vào và các thuật toán để tăng hiệu quả về thời gian xử lý và độ chính xác nhận dạng, so sánh kết quả thực hiện với một số cơ sở dữ liệu chuẩn đã có nhằm đánh giá tính hiệu quả của các kỹ thuật mới được áp dụng.

1. Mục đích nghiên cứu.

Mục tiêu chính của đề tài là nghiên cứu nhận dạng loài chó mèo dựa trên các kỹ thuật học sâu trong lĩnh vực thị giác máy tính. Thực nghiệm áp dụng kỹ thuật trích xuất đặc trưng hình ảnh để dự đoán giống loài của chó/mèo. Từ đó phân tích, đánh giá hiệu quả và độ chính xác trên các mô hình học sâu. Xây dựng tập dữ liệu và thực nghiệm phân loại từng loài của chó và mèo.

1. Tổng quan về các nghiên cứu liên quan.

Trong những năm gần đây, lĩnh vực trí tuệ nhân tạo đã được các nhà khoa học, giới chuyên gia và các công ty công nghệ quan tâm và đầu tư nghiên cứu ứng dụng vào thực tiển nhiều như Google, Baidu, Apple,… ). Các cuộc hội thảo về các lĩnh vực trí tuệ nhân tạo cũng được tổ chức thường xuyên để giải quyết các bài toán về thị giác máy tính ( Computer Vision), nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên…Song song với đó, kỹ thuật học sâu sử dụng trong lĩnh vực thị giác máy tính ngày càng phổ biến (nhận diện hình ảnh, các lĩnh vực khác) và phát triển nhờ vào tính ưu việt và hỗ trợ của thiết bị phần cứng để tăng tốc độ xử lý (GPU/ GPGPU). Đối với bài toán nhận dạng giống loài cũng là một lĩnh vực không mới, có rất nhiều bài báo tại các hội thảo trình bày này đã được trình bày. Có nhiều giải pháp, kiến trúc mạng học sâu được đề xuất và đưa ra với các kết quả khác nhau như phương pháp “[ResNet50](https://github.com/keras-team/keras-applications/blob/master/keras_applications/resnet50.py)’’ cho ra kết quả độ chính xác 90% ( trên tập dogImages), phương pháp [VGG19](https://github.com/keras-team/keras-applications/blob/master/keras_applications/vgg19.py) cho ra kết quả độ chính xác 89.9% (trên tập dogImages), [MobileNetV2(alpha=0.50)](https://github.com/keras-team/keras-applications/blob/master/keras_applications/mobilenet_v2.py) cho ra kết quả độ chính xác 75% ( trên tập Cats\_and\_Dogs [1]…Ở Việt Nam, việc nghiên cứu về thị giác máy tính (Computer Vision) và các kỹ thuật xử lý ảnh, trí tuệ nhân tạo (AI) để áp dụng giải quyết các bài toán nhận diện giống loài, nhận diện khuôn mặt …chưa được nghiên cứu nhiều. Các sản phẩm thực tế ứng dụng từ các kỹ thuật và công nghệ này chưa phổ biến. Vì thế trong luận văn này, đồ án nghiên cứu các kỹ thuật học sâu ứng dụng nhận dạng giống loài nhằm ứng dụng vào app để người chưa có kiến thức về dog và cat có thể biết được giống loài của mình từ đó cung cấp cho người nuôi rõ nguồi gốc xuất xứ để đưa ra những phương án chăm sóc và sinh sản phù hợp với thú cưng.

1. Đối tượng và phạm vi nghiên cứu.

Nghiên cứu về các phương pháp Học máy (Machine Learning), các lĩnh vực học máy và thị giác máy tính trong nhận dạng giống loài cat/dog.

- Lý thuyết về trí tuệ nhân tạo, mạng neural và mạng học sâu.

- Nghiên cứu các model classifire tốt nhất và thực hiện train trên toàn bộ dataset chuẩn bị.

1. Phương pháp nghiên cứu.

Nghiên cứu lý thuyết: Tổng hợp thu thập và nghiên cứu các tài liệu có liên quan đến đề tài như đã nêu trong mục 3 bao gồm: lý thuyết về xử lý ảnh, lý thuyết về học máy, mạng neural tích chập, kỹ thuật học sâu trong nhận dạng phân loại giống loài.

Phương pháp thực nghiệm: Đánh giá một số kỹ thuật thị giác máy tính trong nhận dạng giống loài.

Đề xuất mô hình và thể hiện cụ thể những kết quả đã nghiên cứu, kết quả thực nghiệm so với các cơ sở dữ liệu chuẩn để so sánh đánh giá.

1. Ý nghĩa khoa học và ý nghĩa thực tiễn.

Với việc nghiên cứu và thực nghiệm kỹ thuật học sâu trong kiến trúc mạng nơ-ron để nhận dạng loài dog/cat cho ra kết quả mong muốn, sử dụng tập dữ liệu huấn luyện lớn đóng vai trò quan trọng trong việc xây dựng mô hình để dự đoán kết quả đầu ra chính xác hơn. Với các tập/bộ dữ liệu 100 loài phổ biến nhất để trainning

# CHƯƠNG I: TỔNG QUAN

## Tóm tắt về bài toán.

1. Giới thiệu chó mèo?

Ngày nay, các con vật nuôi trong nhà trở thành những **'vật cưng'** và đóng một vai trò rất quan trọng trong cuộc sống hằng ngày của chủ nhân. Do vậy, sự có mặt của chúng cũng cần được xem xét thật kỹ trước khi đến với ngôi nhà của chúng ta.

Chó là một trong những động vật đầu tiên được con người thuần hóa, loài vật này đã trở thành bạn đồng hành của chúng ta trong suốt hơn 18000 năm vừa qua. Thậm chí, nhiều nhà khoa học đã tìm ra những lý do để mỗi người nên sở hữu 1 chú cún ở nhà.

Bên cạnh đó mèo cũng là loài vật nuôi được khá nhiều người lựa chọn để nuôi bởi sự đáng yêu cũng như tính tự lập của nó.

Thế nhưng vẫn có nhiều người nuôi pet nhưng không biết rõ pet mình đang nuôi thuộc giống/ loài nào chúng có nhưng ưu nhược điểm gì hay cách chăm sóc ra sao sẽ phù hợp để tránh được các bệnh thường gặp.

**Bài nhận dạng giống loài giải quyết vấn đề gì ?**

Bài toán nhận dạng giống loài đã giải quyết cho chúng ta về việc phân loại giống của chúng. Trên thực tế có rất nhiều loài dog/cat giống nhau nếu như không phải chuyên gia khó có thể phân biệt được giữa 2 loài dog hay cat ra, mỗi người nuôi pet thì đều muốn biết tên, thông tin giống loài mà mình đang nuôi và chăm sóc. Đó chính là lý do bài toán được ra đời.

## Trí tuệ nhân tạo và ứng dụng.

1. Một số khái niệm chung.

Trí tuệ nhân tạo hay trí thông minh nhân tạo (Artificial intelligence – viết tắt là AI) là một ngành thuộc lĩnh vực khoa học máy tính (Computer science). Là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người.

Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc ứng dụng các hệ thống học máy (machine learning) để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính.

Cụ thể, trí tuệ nhân tạo giúp máy tính có được những trí tuệ của con người như: biết suy nghĩ và lập luận để giải quyết vấn đề, biết giao tiếp do hiểu ngôn ngữ, tiếng nói, biết học và tự thích nghi,…

Tuy rằng trí thông minh nhân tạo có nghĩa rộng như là trí thông minh trong các tác phẩm khoa học viễn tưởng, nó là một trong những ngành trọng yếu của tin học. Trí thông minh nhân tạo liên quan đến cách cư xử, sự học hỏi và khả năng thích ứng thông minh của máy móc.

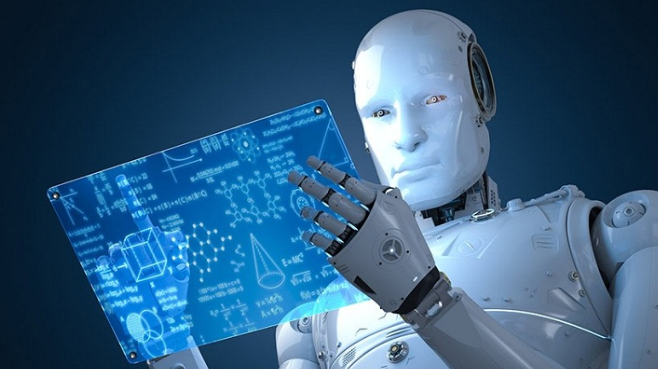
1. Có bao nhiêu loại AI ?

Công nghệ AI được chia làm 4 loại chính:

**Loại 1: Công nghệ AI phản ứng.**

Công nghệ AI phản ứng có khả năng phân tích những động thái khả thi nhất của chính mình và của đối thủ, từ đó, đưa ra được giải pháp tối ưu nhất.

Một ví dụ điển hình của công nghệ AI phản ứng là Deep Blue. Đây là một chương trình chơi cờ vua tự động, được tạo ra bởi IBM, với khả năng xác định các nước cờ đồng thời dự đoán những bước đi tiếp theo của đối thủ. Thông qua đó, Deep Blue đưa ra những nước đi thích hợp nhất [3].



**Hình 1. 1** Hình ảnh mình họa về trí tuệ nhân tạo

**Loại 2: Công nghệ AI với bộ nhớ hạn chế.**

Đặc điểm của công nghệ AI với bộ nhớ hạn chế là khả năng sử dụng những kinh nghiệm trong quá khứ để đưa ra những quyết định trong tương lai. Công nghệ AI này thường kết hợp với cảm biến môi trường xung quanh nhằm mục đích dự đoán những trường hợp có thể xảy ra và đưa ra quyết định tốt nhất cho thiết bị.

Ví dụ như đối với xe không người lái, nhiều cảm biến được trang bị xung quanh xe và ở đầu xe để tính toán khoảng cách với các xe phía trước, công nghệ AI sẽ dự đoán khả năng xảy ra va chạm, từ đó điều chỉnh tốc độ xe phù hợp để giữ an toàn cho xe [3].



**Hình 1. 2** Xe tự lái

**Loại 3: Lý thuyết trí tuệ nhân tạo.**

Công nghệ AI này có thể học hỏi cũng như tự suy nghĩ, sau đó áp dụng những gì học được để thực hiện một việc cụ thể. Hiện nay, công nghệ AI này vẫn chưa trở thành một phương án khả thi [3].

**Loại 4: Tự nhận thức.**

Công nghệ AI này có khả năng tự nhận thức về bản thân, có ý thức và hành xử như con người. Thậm chí, chúng còn có thể bộc lộ cảm xúc cũng như hiểu được những cảm xúc của con người. Đây được xem là bước phát triển cao nhất của công nghệ AI và đến thời điểm hiện tại, công nghệ này vẫn chưa khả thi [3].

1. Học máy.

Học máy (ML) là một công nghệ phát triển từ lĩnh vực trí tuệ nhân tạo. Các thuật toán ML là các chương trình máy tính có khả năng học hỏi về cách hoàn thành các nhiệm vụ và cách cải thiện hiệu suất theo thời gian.

ML vẫn đòi hỏi sự đánh giá của con người trong việc tìm hiểu dữ liệu cơ sở và lựa chọn các kĩ thuật phù hợp để phân tích dữ liệu. Đồng thời, trước khi sử dụng, dữ liệu phải sạch, không có sai lệch và không có dữ liệu giả.

Các mô hình ML yêu cầu lượng dữ liệu đủ lớn để "huấn luyện" và đánh giá mô hình. Trước đây, các thuật toán ML thiếu quyền truy cập vào một lượng lớn dữ liệu cần thiết để mô hình hóa các mối quan hệ giữa các dữ liệu. Sự tăng trưởng trong dữ liệu lớn (big data) đã cung cấp các thuật toán ML với đủ dữ liệu để cải thiện độ chính xác của mô hình và dự đoán.

## Các loại thuật toán học máy

Có 3 loại ML chính bao gồm học có giám sát (supervised learning), học không giám sát (unsupervised learning) và Học tăng cường (reinforcement learning).

**Học có giám sát.**

Trong học có giám sát, máy tính học cách mô hình hóa các mối quan hệ dựa trên dữ liệu được gán nhãn (labeled data). Sau khi tìm hiểu cách tốt nhất để mô hình hóa các mối quan hệ cho dữ liệu được gắn nhãn, các thuật toán được huấn luyện được sử dụng cho các bộ dữ liệu mới.

Ứng dụng của kĩ thuật học có giám sát: Xác định tín hiệu hay biến số tốt nhất để dự báo lợi nhuận trong tương lai của cổ phiếu hoặc dự đoán xu hướng thị trường chứng khoán.

**Học không giám sát.**

Trong học không giám sát, máy tính không được cung cấp dữ liệu được dán nhãn mà thay vào đó chỉ được cung cấp dữ liệu mà thuật toán tìm cách mô tả dữ liệu và cấu trúc của chúng.

Ứng dụng của học không giám sát: Phân loại các công ty thành các nhóm công ty tương đồng dựa trên đặc điểm của chúng thay vì sử dụng tiêu chuẩn của các nhóm ngành hoặc các quốc gia.

**Học tăng cường.**

Có thể xem học tăng cường là môt mảng con của học máy (tùy quan điểm); đối với deep learning thì học tăng cường chủ yếu là học chuỗi các phản ứng nhằm tối đa hóa lợi ích.

Reinforcement learning là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximizing the performance). Hiện tại, Reinforcement learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi (Game Theory), các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất.

**Ứng dụng**

Các thuật toán ML đang được sử dụng để phân tích dữ liệu lớn (big data) để giúp dự đoán xu hướng hoặc sự kiện thị trường, ví dụ như dự đoán giá nhà đất.

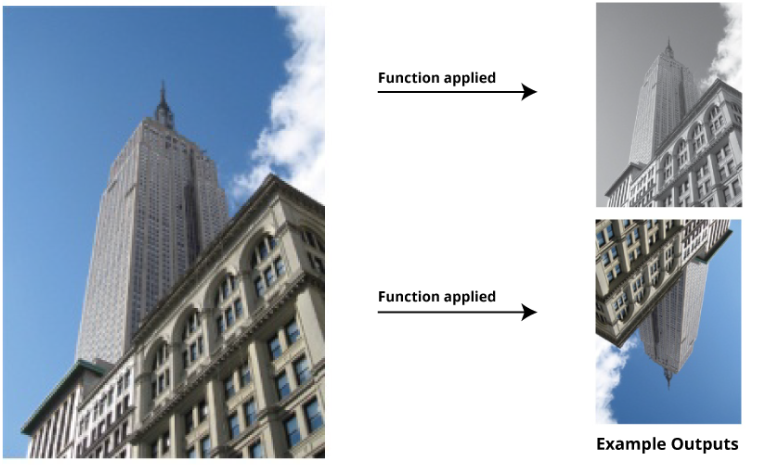
Các thuật toán nhận dạng hình ảnh hiện có thể phân tích dữ liệu từ các hệ thống chụp ảnh vệ tinh để cung cấp thông tin về số lượng khách hàng tại các bãi đậu xe của cửa hàng bán lẻ, phát hiện giạn lận, phát hiện xâm nhập, hoạt động vận chuyển và cơ sở sản xuất, và sản lượng nông nghiệp...Những thông tin này sẽ cung cấp dữ liệu đầu vào cho các mô hình định giá hoặc các mô hình kinh tế.

## Lý thuyết về thị giác máy tính (Computer vision): Tìm hiểu các phương pháp trích chọn và biểu diễn đặc trưng ảnh.

Thị giác máy tính, Computer Vision, được coi là một định nghĩa duy nhất, là khả năng và quy trình để máy tính hiểu được môi trường xung quanh thông qua việc sử dụng một hoặc nhiều mắt kỹ thuật số. Rõ ràng, điều này không được thực hiện bằng cách sử dụng một nhiệm vụ độc lập. Thay vào đó, đó là một loạt các bước bắt đầu bằng việc có được hình ảnh đầu tiên, và sau đó đạt được sự hiểu biết thông qua xử lý và phân tích hình ảnh.

Thị giác của con người là một quá trình phức tạp và việc mô phỏng điều này luôn là một nhiệm vụ đầy thách thức đối với máy tính.Thông qua việc sử dụng các kỹ thuật machine learning truyền thống và gần đây với những tiến bộ trong DeepLearning,có những tiến bộ đáng chú ý đang được thực hiện trong các máy tính có khả năng diễn giải và phản ứng với những gì chúng “thấy”.

Một điều kiện tiên quyết và quan trọng cho Thị giác máy tính khác với Xử lý hình ảnh là việc thao tác trên nhiều hình ảnh. Trong khi Xử lý hình ảnh hoàn toàn hoạt động với duy nhất một tấm hình ở dạng số hóa, Computer Vision hoạt động phù hợp hơn trên một luồng hình ảnh (stream of images) có mối quan hệ tạm thời đã biết trước [4].



**Hình 1. 3** Xử lý ảnh được thực hiện trên ảnh tĩnh

1. Kỹ thuật học sâu (Deep learning): Khái niệm, mạng neuron học sâu, mạng neuron tích chập.

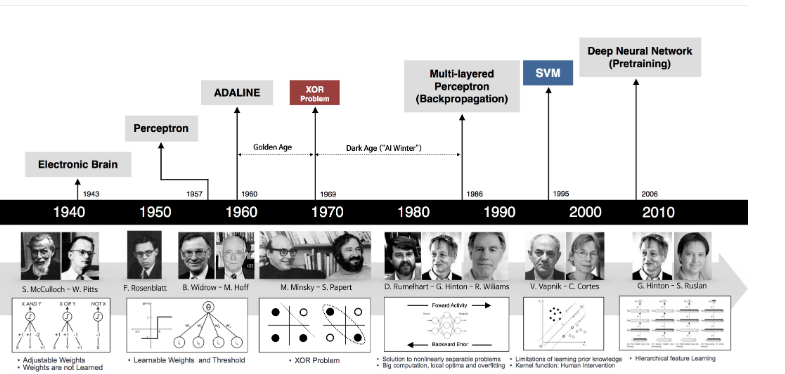
Học sâu (Deep learning) là một loại hình học máy đã phát triển nhanh chóng trong vài năm qua, do những cải tiến về sức mạnh xử lý và kỹ thuật học sâu. Thông thường, học sâu để cập đến các mạng lưới nơ-ron sâu, hoặc nhiều lớp, nổi trội trong việc thực hiện các nhiệm vụ rất phức tạp, thường là trung tâm của con người, như nhận dạng hình ảnh và dịch ngôn ngữ.

Các mô hình học tập sâu học cách xem dữ liệu huấn luyện của họ như một hệ thống phân cấp các khái niệm lồng nhau, cho phép chúng thể hiện các mẫu cực kỳ phức tạp. Nói cách khác, các mô hình này không chỉ xem xét các đặc trưng ban đầu mà bạn cung cấp cho chúng mà còn tự động kết hợp các đặc trưng này để tạo thành các đặc trưng meta mới, được tối ứu hóa, sau đó chúng kết hợp để tạo thành nhiều đặc trưng hơn.

“Deep” cũng đề cập đến kiến trúc được sử dụng để thực hiện điều này, thường bao gồm nhiều lớp đơn vị xử lý, mỗi lớp sử dụng các đầu ra lớp trước rất mạnh mẽ làm đầu vào của nó. Mỗi đơn vị xử lý này được gọi là nơ ron và toàn bộ kiến trúc mô hình được gọi là mạng nơ ron hoặc hoặc mạng nơ ron sâu khi có nhiều lớp.

Để xem kiếm trúc này có thể hữu ích như thế nào, hãy để ý nghĩ về một chương trình cố gắng phân loại hình ảnh là xe đạp hoặc xe đạp một bánh. Đối với một con người, đây là nhiệm vụ dễ dàng, nhưng lập trình một máy tính để xem xét một lưới các pixel và cho biết đối tượng mà nó đại diện là khó khăn. Một số pixel nhất định chỉ ra rằng một bánh xe đạp tồn tại trong một hình ảnh sẽ có nghĩa là một cái gì đó hoàn toàn khác trong chiếc tiếp theo nếu chiếc xe đạp đó hơi di chuyển, được đặt ở một góc khác hoặc có màu khác [5].

Các mô hình học tập sâu đã vượt qua điều này bằng cách chi nhỏ vấn đề thành nhiều phần dễ quản lý hơn. Ví dụ, một mạng nơ-ron sâu, lớp tế bào nơ-ron đầu tiên có thể phân tách hình ảnh thành các phần và chỉ xác định các đặc điểm hình ảnh ở mức độ thấp, như các cạnh và đường viền của hình dạng trong hình ảnh. Các đặc trưng được tạo này được đưa vào lớp tiếp theo của mạng để tìm các mẫu trong số các đặc trưng. Các mẫu này sau đó được đưa vào các lớp tiếp theo, cho đến khi mạng xác định các hình dạng chung và cuối cùng là các đối tượng hoàn chỉnh. Trong ví dụ về bánh xe đạp của chúng ta, lớp đầu tiên có thể tìm thấy các đường, lớp thứ hai có thể thấy các đường tạo thành các vòng tròn và lớp thứ ba có thể xác định rằng các vòng tròn nhất định thực sự là bánh xe. Theo cách này, thay vì nhìn vào một khối lượng pixel, mô hình có thể thấy rằng mỗi hình ảnh có một số đặc trưng meta nhất định của bánh xe. Ví dụ nó có thể học được rằng hai bánh xe có khả năng chỉ ra một chiếc xe đạp, trong khi một bánh xe có nghĩa là một bánh xe.



**Hình 1. 4** Lịch sử phát triển Dêp Learning

# CHƯƠNG II. DECTEC VÀ MẠNG HỌC SÂU CHO BÀI TOÁN NHẬN DẠNG GIỐNG LOÀI.

## Thuật toán phát hiện đối tượng ( Object detection ).

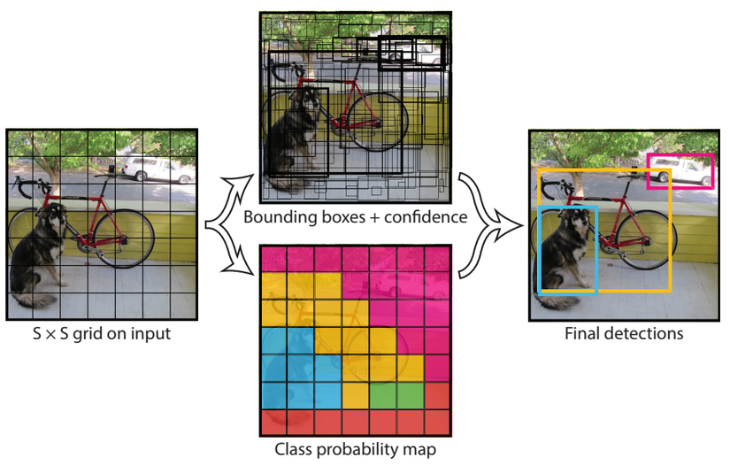
### YOLO

Mô hình YOLO được mô tả lần đầu tiên bởi Joseph Redmon, và các cộng sự. trong bài viết năm 2015 có tiêu đề Bạn chỉ nhìn một lần: Phát hiện đối tượng theo thời gian thực - You Only Look Once: Unified, Real-Time Object Detection [6]. Trong công trình này thì một lần nữa Ross Girshick, người phát triển mạng R-CNN, cũng là một tác giả và người đóng góp khi ông chuyển qua Facebook AI Research.

Phương pháp chính dựa trên một mạng neural network duy nhất được huấn luyện dạng end-to-end model. Mô hình lấy input là một bức ảnh và dự đoán các bounding box và nhãn lớp cho mỗi bounding box. Do không sử dụng region proposal nên kỹ thuật này có độ chính xác thấp hơn (ví dụ: nhiều lỗi định vị vật thể - localization error hơn), mặc dù hoạt động ở tốc độ 45 fps (khung hình / giây) và tối đa 155 fps cho phiên bản tối ưu hóa tốc độ. Tốc độ này còn nhanh hơn cả tốc độ khung hình của máy quay phim thông thường chỉ vào khoảng 24 fps [6].

Mô hình hoạt động bằng cách trước tiên phân chia hình ảnh đầu vào thành một lưới các ô (grid of cells), trong đó mỗi ô chịu trách nhiệm dự đoán các bounding boxes nếu tâm của nó nằm trong ô. Mỗi grid cell (tức 1 ô bất kì nằm trong lưới ô) dự đoán các bounding boxes được xác định dựa trên tọa độ x, y (thông thường là tọa độ tâm, một số phiên bản là tọa độ góc trên cùng bên trái) và chiều rộng (width) và chiều cao (height) và độ tin cậy (confidence) về khả năng chứa vật thể bên trong. Ngoài ra các dự đoán nhãn cũng được thực hiện trên mỗi một bonding box.

Ví dụ: một hình ảnh có thể được chia thành lưới 7 × 7 và mỗi ô trong lưới có thể dự đoán 2 bounding box, kết quả trả về 98 bounding box được đề xuất. Sau đó, một sơ đồ xác suất nhãn (gọi là class probability map) với các confidence được kết hợp thành một tợp hợp bounding box cuối cùng và các nhãn. Hình ảnh được lấy từ bài báo dưới đây tóm tắt hai kết quả đầu ra của mô hình.



**Hình 2. 1** Các bước xử lý trong mô hình YOLO

### YOLOv2 (2016) và YOLOv3 (2018)

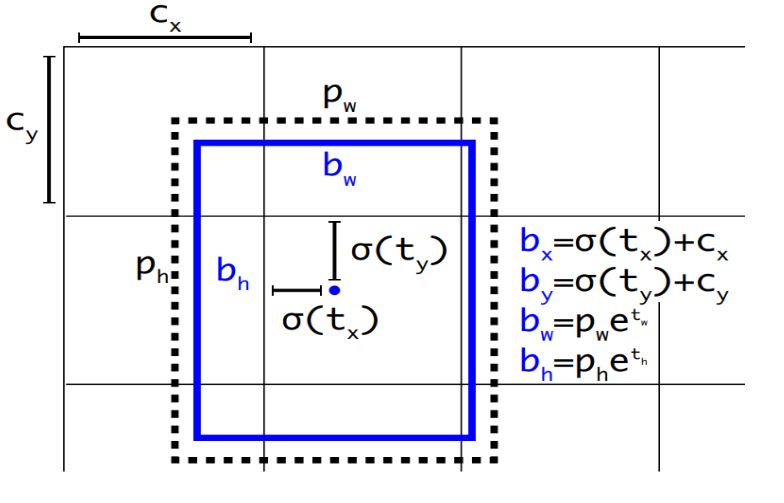
Mô hình YOLOv2 được Joseph Redmon và Ali Farhadi cập nhật nhằm cải thiện hơn nữa hiệu suất trong bài báo năm 2016 có tựa đề là YOLO9000: Better, Faster, Stronger [7].

Mặc dù biến thể của YOLO được gọi là YOLOv2, một instance của mô hình theo như mô tả đã được đào tạo trên hai bộ dữ liệu nhận dạng đối tượng, và có khả năng dự đoán lên tới 9000 loại đối tượng khác nhau, do đó được đặt tên là YOLO9000. Với con số này thì mô hình này đã tiến xa hơn rất nhiều so với mọi mô hình trước đó về số lượng các loại đối tượng có khả năng phát hiện.

Một số thay đổi về huấn luyện và kiến trúc đã được thực hiện, chẳng hạn như việc sử dụng batch normalization cho hàng loạt và hình ảnh đầu vào phân giải cao.

Giống như Faster R-CNN, mô hình YOLOv2 sử dụng anchor boxes, bounding box được xác định trước với hình dạng và kích thước hợp lý được tùy chỉnh trong quá trình huấn luyện. Sự lựa chọn các bounding boxes cho hình ảnh được xử lý trước bằng cách sử dụng thuật toán phân cụm k-mean trên tập dữ liệu huấn luyện.

Điều quan trọng, các predicted bounding box được tinh chỉnh để cho phép các thay đổi nhỏ có tác động ít hơn đến các dự đoán, dẫn đến mô hình ổn định hơn. Thay vì dự đoán trực tiếp vị trí và kích thước, các offsets (tức tọa độ tâm, chiều dài và chiều rộng) được dự đoán để di chuyển và định hình lại các pre-defined anchor boxes tại mỗi một grid cell thông qua hàm logistic.



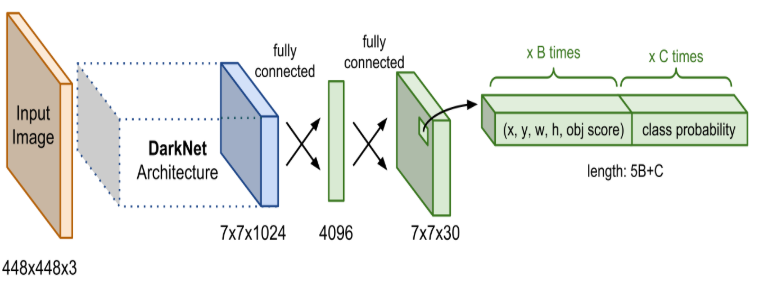
**Hình 2. 2** Sơ đồ tạo prior bounding box

Sơ đồ giúp tạo prior bounding box có chiều rộng pwpw và chiều cao phph đã xác định từ grid cell có tọa độ (cx,cy)(cx,cy). Khi đó tọa độ tâm (bx,by)(bx,by) được tính theo mức độ tịnh tiến hàm sigmoid. Đồng thời, chiều rộng và chiều cao (bw,bh)(bw,bh) được tính như công thức scale số mũ của cơ số tự nhiên e.

Những cải tiến xa hơn của mô hình đã được đề xuất bởi Joseph Redmon và Ali Farhadi trong bài báo năm 2018 với tiêu đề [YOLOv3: An Incremental Improvement](https://arxiv.org/abs/1804.02767). Những cải tiến này khá là nhỏ, chủ yếu là thay đổi mô hình deep CNN trong trích xuất feature.

### Nhóm các mô hình họ YOLO.

Yolo là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. Yolo được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers.Trong đóp các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng [8].



**Hình 2. 3** Mô hình YOLO

### Cách hoạt động của YOLO.

Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh. Ảnh đầu vào được chia thành thành S x S ô thường thì sẽ là 3 x 3, 7 x 7, 9 x 9 ... việc chia ô này có ảnh hưởng tới việc mô hình phát hiện đối tượng, mình xin trình bày ở phần sau.

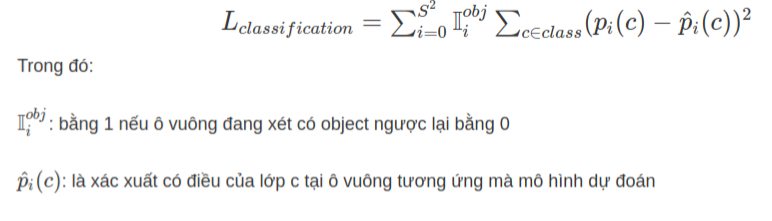
Với Input là 1 ảnh, đầu ra mô hình là một ma trận 3 chiều có kích thước S x S x (5 x N + M) [6]với số lượng tham số mỗi ô là (5 x N + M) [6] với N và M lần lượt là số lượng Box và Class mà mỗi ô cần dự đoán. Ví dụ với hình ảnh trên chia thành 7 x 7 ô, , mỗi ô cần dự đóan 2 bounding box và 3 object :  con chó, ô tô, xe đạp thì output là 7 x 7 x 13, mỗi ô sẽ có 13 tham số, kết quả trả về (7 x 7 x 2 =98) bounding box. Chúng ta sẽ cùng giải thích con số (5 x N + M) [6] được tính như thế nào.  
Dự đoán mỗi bounding box gồm 5 thành phần : (x, y, w, h, prediction) với (x, y ) là tọa độ tâm của bounding box, (w, h) lần lượt là chiều rộng và chiều cao của bounding box, prediction được định nghĩa Pr(*Object*)∗ *IOU*(*pred*,*truth*)  [6] xin trình bày sau. Với hình ảnh trên như ta tính mỗi ô sẽ có 13 tham số, ta có thể hiểu đơn giản như sau tham số thứ 1 sẽ chỉ ra ô đó có chứa đối tượng nào hay không P(Object), tham số 2, 3, 4, 5 sẽ trả về x, y ,w, h của Box1. Tham số 6, 7, 8, 9, 10 tương tự sẽ Box2, tham số 11, 12, 13 lần lượt là xác suất ô đó có chứa object1( P(chó|object), object2(P(ô tô|object)), object3(P( xe đạp|object)) [6]. Lưu ý rằng tâm của bounding box nằm ở ô nào thì ô đó sẽ chứa đối tượng, cho dù đối tượng có thể ở các ô khác thì cũng sẽ trả về là 0. Vì vậy việc mà 1 ô chứa 2 hay nhiều tâm của bouding box hay đối tượng thì sẽ không thể detect được, đó là một hạn chế của mô hình YOLO1, vậy ta cần phải tăng số lượng ô chia trong 1 ảnh lên đó là lí do vì sao mình nói việc chia ô có thể làm ảnh hưởng tới việc mô hình phát hiện đối tượng.

* **Loss Function.**

Hàm lỗi trong YOLO được tính trên việc dự đoán và nhãn mô hình để tính [6]. Cụ thể hơn nó là tổng độ lôĩ của 3 thành phần con sau :

* Độ lỗi của việc dự đoán loại nhãn của object - Classifycation loss.
* Độ lỗi của dự đoán tọa độ tâm, chiều dài, rộng của boundary box (x, y ,w, h) - Localization loss.
* Độ lỗi của việc dự đoán bounding box đó chứa object so với nhãn thực tế tại ô vuông đó - Confidence loss.
* **Classification loss**

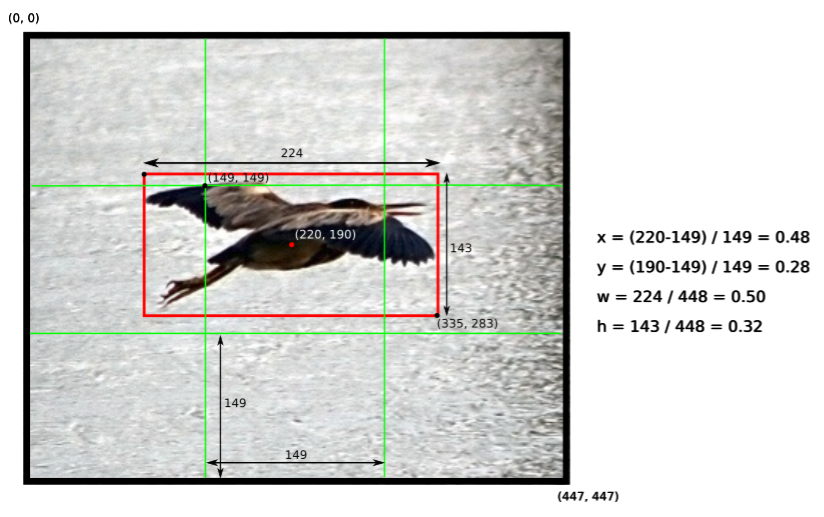
Classification loss - độ lỗi của việc dự đoán loại nhãn cuả object, hàm lỗi này chỉ tính trên những ô vuông có xuất hiện object, còn những ô vuông khác ta không quan tâm. Classification loss [6] được tính bằng công thức sau:



**Hình 2. 4** Classification loss

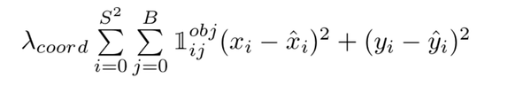
* **Localization loss**

Localization loss là hàm lỗi dùng để tính giá trị lỗi cho boundary box [6] được dự đoán bao gồm tọa độ tâm, chiều rộng, chiều cao của so với vị trí thực tế từ dữ liệu huấn luyện của mô hình. Lưu ý rằng chúng ta không nên tính giá trị hàm lỗi này trực tiếp từ kích thức ảnh thực tế mà cần phải chuẩn hóa về [0, 1] so với tâm của bounding box. Việc chuẩn hóa này kích thước này giúp cho mô hình dự đoán nhanh hơn và chính xác hơn so với để giá trị mặc định của ảnh. Hãy cùng xem một ví dụ:



**Hình 2. 5** Biểu diễn tọa độ của đối tượng

Giá trị hàm Localization loss được tính trên tổng giá trị lỗi dự đoán toạ độ tâm (x, y) và (w, h) của predicted bounding box với grouth-truth bounding box. Tại mỗi ô có chưa object, ta chọn 1 boundary box có IOU (Intersect over union) [6] tốt nhất, rồi sau đó tính độ lỗi theo các boundary box này.  
Giá trị hàm lỗi dự đoán tọa độ tâm (x, y) của predicted bounding box và (x̂, ŷ) là tọa độ tâm của truth bounding box được tính như sau :



**Hình 2. 6** Giá trị hàm lỗi

Giá trị hàm lỗi dự đoán (w, h ) của predicted bounding box so với truth bounding box được tính như sau :



* **Total loss**



## Các mô hình mạng học sâu cho bài toán.

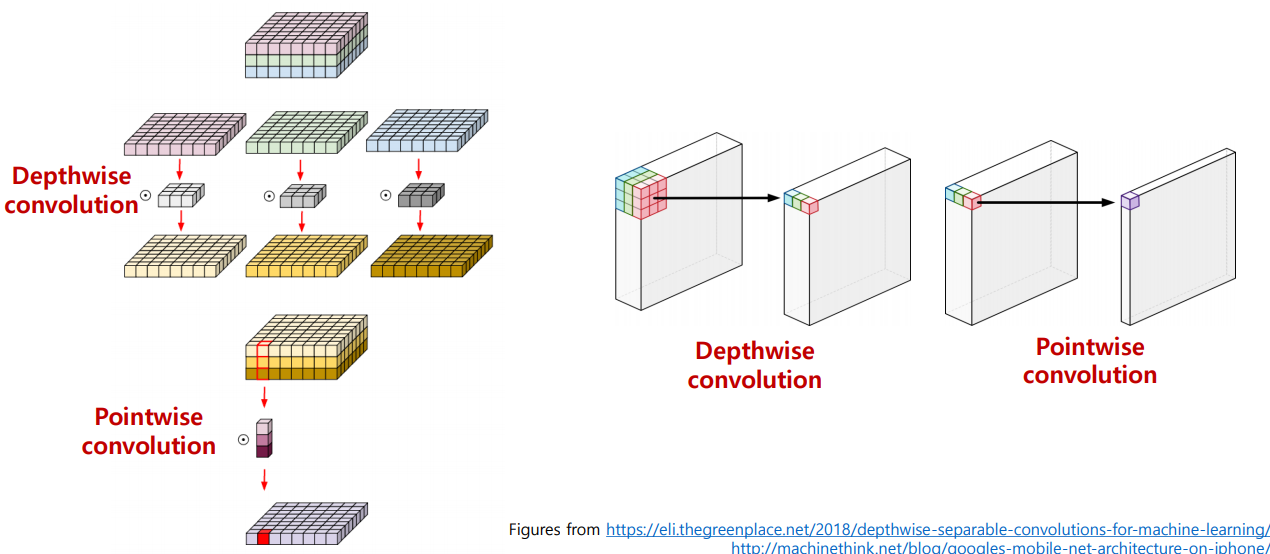
### Mô hình MobileNetV1

Các mạng hiện đại đòi hỏi tính toán cao tài nguyên ngoài khả năng của nhiều thiết bị di động và nhúng các ứng dụng

Gần đây đã mở ra một hướng mới để tối ưu hóa các phương pháp bao gồm thuật toán di truyền và học tăng cường để tìm kiếm kiến trúc. Tuy nhiên, một nhược điểm là kết quả mạng kết thúc rất phức tạp.

[MobileNetV1](https://research.googleblog.com/2017/06/mobilenets-open-source-models-for.html) , một nhóm mạng thần kinh thị giác máy tính đa năng được thiết kế với các thiết bị di động để hỗ trợ phân loại, phát hiện và hơn thế nữa. Khả năng chạy các mạng sâu trên thiết bị di động cá nhân cải thiện trải nghiệm người dùng, cung cấp quyền truy cập mọi lúc, mọi nơi, với các lợi ích bổ sung về bảo mật, quyền riêng tư và tiêu thụ năng lượng. Khi các ứng dụng mới xuất hiện cho phép người dùng tương tác với thế giới thực trong thời gian thực, thì nhu cầu về các mạng thần kinh hiệu quả hơn bao giờ hết cũng tăng theo.

MobileNetV1 Depthwise Separable Convolution (depthwise conv + pointwise conv)



**Hình 2. 7** Mô hình MobileNet V2

Standard convolution takes  𝑖 input tensor  và applies convolutional kernel  để tạo ra  output tensor 

Standard convolutions have the computational cost of



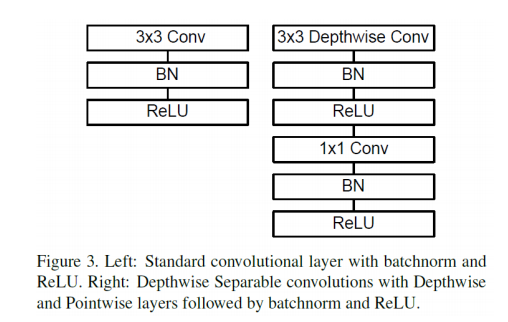
Depthwise separable convolutions cost

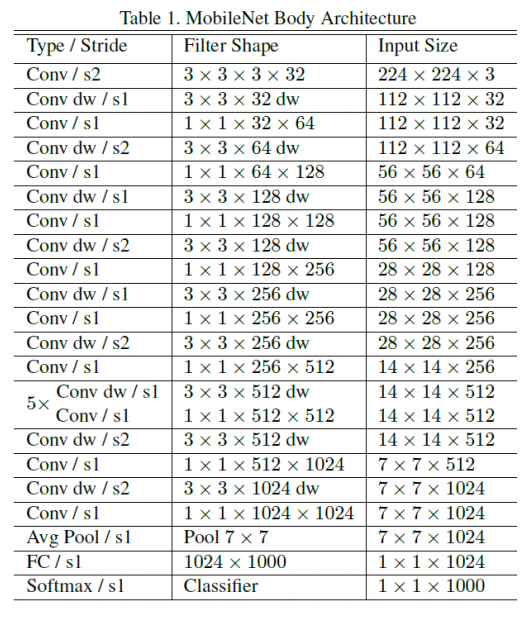


Reduction in computations



Nếu sử dụng 3x3 depthwise separable convolutions, chúng ta sẽ tính toán ít hơn 8-9 lần.





**Hình 2. 8** Mô hình tổng quát MobileNetV2

Hệ số Multiplier và Resolution Multiplier:

Đối với 1 lớp Multiplier nhất định hệ số α, số kênh đầu vào M trở thành αM và số đầu ra N trở thành αN trong đó với các cài đặt điển hình là 1, 0,75, 0,5 và 0,25

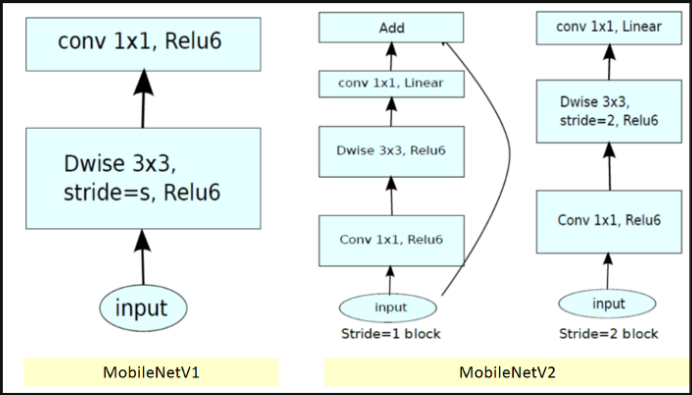


**Hình 2. 9** Độ chính xác mô hình với hệ số a

Siêu tham số thứ hai để giảm chi phí tính toán của một nơron mạng là hệ số resolution multiplier ρ.

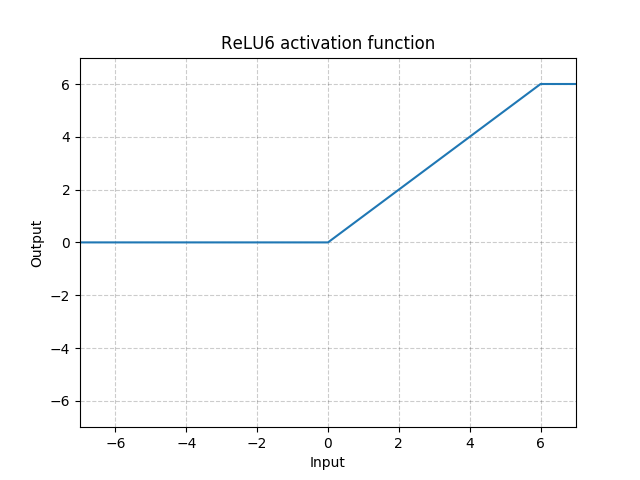
### Mô hình MobileNetV2

Trong các phiên bản trước [MobileNetV1](https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69) , **Depthwise tách Convolution** được giới thiệu mà làm giảm đáng kể chi phí phức tạp và kích thước mô hình của mạng, đó là phù hợp với các thiết bị di động, hoặc bất kỳ thiết bị với sức mạnh tính toán thấp. Trong MobileNetV2, một mô-đun tốt hơn được giới thiệu với **cấu trúc phần dư ngược** . Lần này, các điểm **không tuyến tính trong các lớp hẹp sẽ bị loại bỏ** . Với MobileNetV2 là xương sống để trích xuất tính năng, các hiệu suất hiện đại cũng đạt được để phát hiện đối tượng và phân đoạn ngữ nghĩa



**Hình 2. 10** So sánh hai mô hính MobileNetV1 và V2

* Trong [MobileNetV1](https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69) , có 2 lớp.
* Lớp **đầu tiên** được gọi là **tích chập theo chiều sâu** , nó thực hiện lọc nhẹ bằng cách áp dụng một bộ lọc chập duy nhất cho mỗi kênh đầu vào.
* Các**lớp thứ hai** là một **1 × 1 tích chập**, được gọi là **tích chập điểm**, chịu trách nhiệm xây dựng các tính năng mới thông qua tính toán kết hợp tuyến tính của các kênh đầu vào.
* **ReLU6** được sử dụng ở đây để so sánh. (Trên thực tế, trong báo cáo công nghệ [MobileNetV1](https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69) , tôi không thể tìm thấy bất kỳ gợi ý nào cho thấy họ sử dụng ReLU6… Có lẽ chúng ta cần kiểm tra mã trong Github…), tức là **min (max ( x , 0), 6)** như sau:



**Hình 2. 11** ReLU6

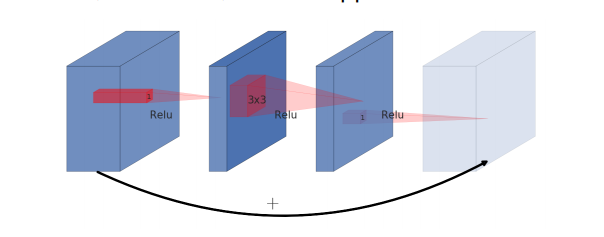
ReLU6 được sử dụng do tính mạnh mẽ của nó khi được sử dụng với tính toán có độ chính xác thấp, dựa trên [27] MobileNetV1.

* + - 1. Cấu trúc phần dư ngược

Các khối blocks kết nối phần đầu và phần cuối của một khối blocks bằng kết nối convolutional. Bằng cách thêm hai trạng thái này, mạng có cơ hội truy cập các kích hoạt trước đó

không được sửa đổi trong khối convolutional .

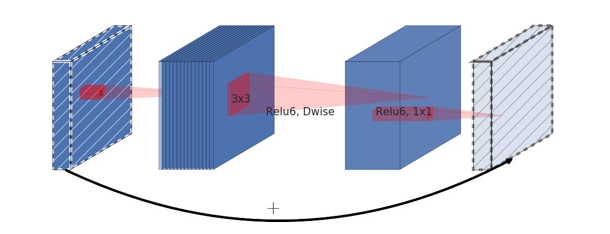
Rộng 🡪 Hẹp(Nút cổ chai) 🡪 Tiếp cận rộng



**Hình 2. 12** Cấu trúc dư ngược

Các nút thắt cổ chai chứa tất cả thông tin cần thiết , trong khi lớp mở rộng chỉ hoạt động như một chi tiết triển khai đi kèm với một biến đổi phi tuyến tính của tensor.

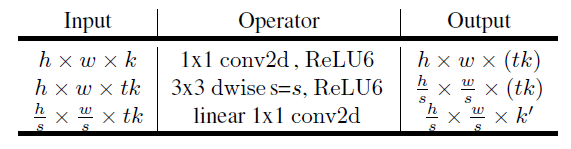
Tiếp cận Hẹp 🡪 Rộng 🡪 Hẹp

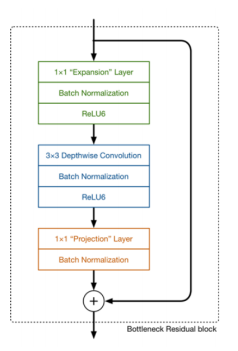


**Hình 2. 13** Cấu trúc dư ngược

Trong MobileNetV2, có hai loại khối. Một là khối còn lại với sải chân là 1. Một khối khác là khối có sải chân là 2 để giảm kích thước.

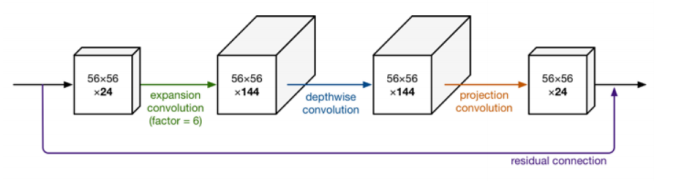
* Có 3 lớp cho cả hai loại khối.
* Lần này, **lớp đầu tiên** là **1 × 1 tích chập với ReLU6.**
* Lớp **thứ hai** là **tích chập theo chiều sâu** .
* Lớp **thứ ba** là một **tích chập 1 × 1 khác nhưng không có bất kỳ sự phi tuyến tính nào.**Người ta khẳng định rằng nếu ReLU được sử dụng lại, các mạng sâu chỉ có sức mạnh của bộ phân loại tuyến tính trên phần khối lượng khác 0 của miền đầu ra.



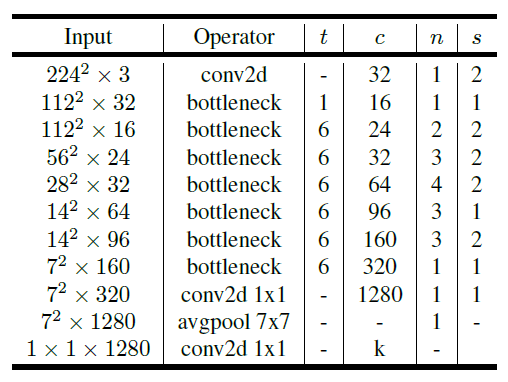


**Hình 2. 14** cấu trúc MobileNetV2

Và có một hệ số mở rộng t . Và t = 6 cho tất cả các thí nghiệm chính.

Nếu đầu vào có 64 kênh, thì đầu ra bên trong sẽ nhận được 64 × t = 64 × 6 = 384 kênh.

**Hình 2. 15** Kiến trúc tổng thể



**Hình 2. 16** Kiến trúc tổng thể MobileNetV2

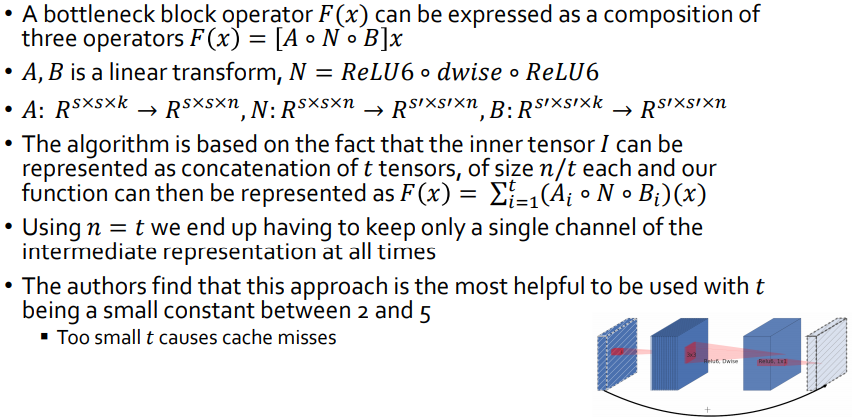
trong đó t : hệ số mở rộng, c : số kênh đầu ra, n : số lặp lại, s: bước sóng. Hạt nhân 3 × 3 được sử dụng để tích chập không gian.

Điển hình, **mạng chính** (hệ số chiều rộng 1, **224 × 224** ), có chi phí tính toán là **300 triệu phép nhân-cộng** và sử dụng **3,4 triệu tham số** . (Hệ số chiều rộng được giới thiệu trong [MobileNetV1](https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69) .)

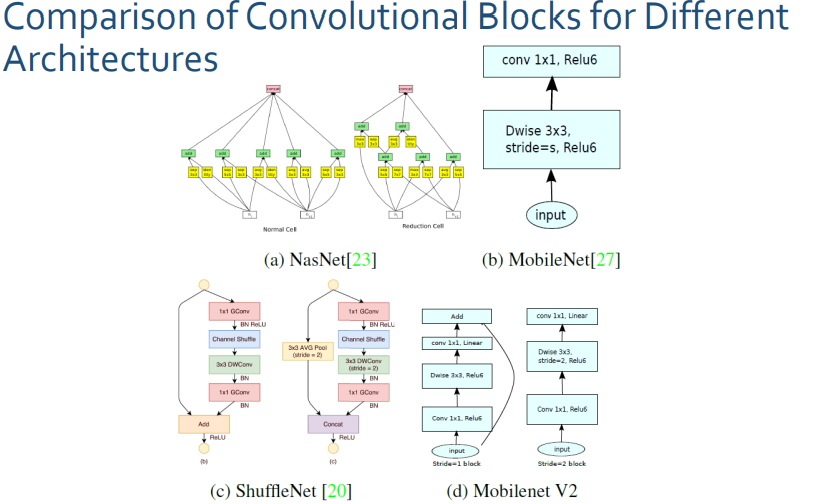
Sự cân bằng hiệu suất được khám phá thêm, đối với **độ phân giải đầu vào từ 96 đến 224** , và **hệ số nhân rộng từ 0,35 đến 1,4** .

Chi phí tính toán mạng lên đến 585 triệu MAdds, trong khi kích thước mô hình thay đổi giữa các thông số 1,7 triệu và 6,9 triệu.

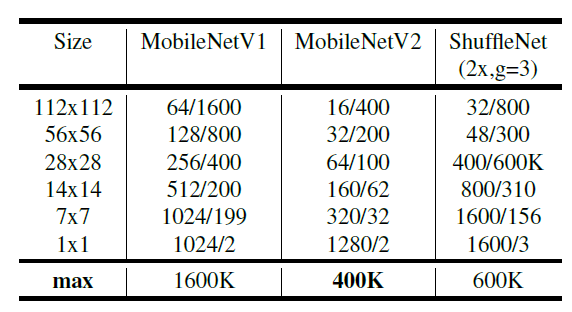
Để đào tạo mạng, 16 GPU được sử dụng với kích thước lô là 96.



* Giảm yêu cầu bộ nhớ



**Hình 2. 17** So sánh các khối khác nhau cho kiến trúc

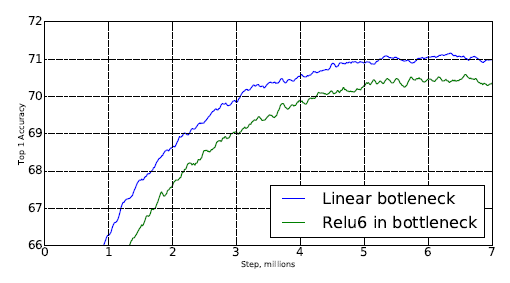


**Hình 2. 18** So sánh kênh

**Số kênh / bộ nhớ tối đa tính bằng Kb) ở mỗi độ phân giải không gian cho kiến ​​trúc khác nhau với phao 16 bit để kích hoạt**

**2.2.2.2 Nghiên cứu cắt bỏ**

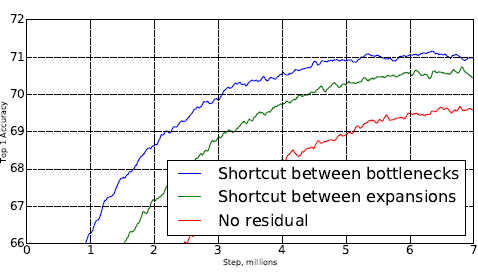
Tác động của nút cổ chai tuyến tính



**Hình 2. 19** Tác động của nút cổ chai tuyến tính

* Với việc loại bỏ ReLU6 ở đầu ra của mỗi mô-đun nút cổ chai, độ chính xác được cải thiện.

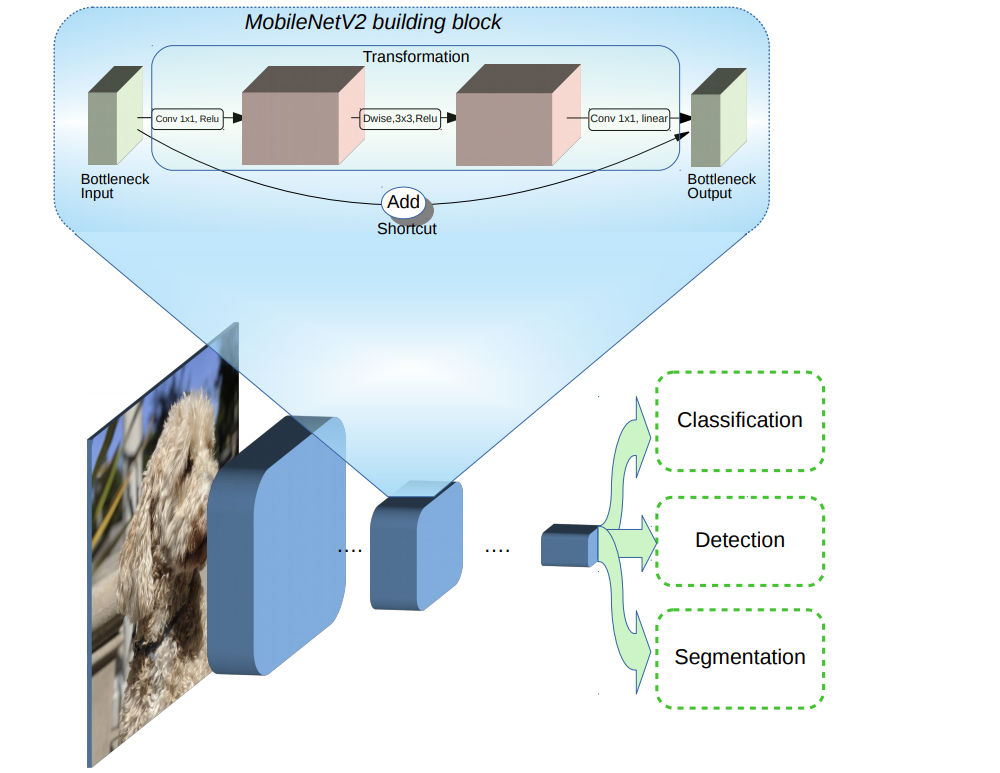
Tác động của phím tắt



**Hình 2. 20** Tác động của phím tắt

* Với phím tắt giữa các nút thắt cổ chai, nó hoạt động tốt hơn phím tắt giữa các bản mở rộng và phím tắt không có bất kỳ kết nối nào.

**2.2.2.1. Kết quả thực nghiệm**



**Hình 2. 21** MobileNetV2 để Phân loại, Phát hiện và Phân đoạn

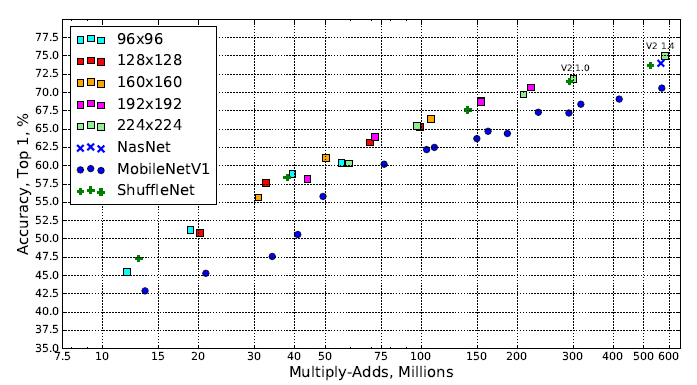
**(Từ**[**https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html**](https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html)**)**

**Phân loại mạng hình ảnh**



**Hình 2. 22** Độ chính xác hàng đầu của ImageNet

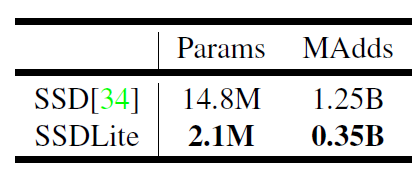
* MobileNetV2 nhanh hơn so với [MobileNetV1](https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69) và [ShuffleNet](https://towardsdatascience.com/review-shufflenet-v1-light-weight-model-image-classification-5b253dfe982f) (1.5) với kích thước mô hình so sánh và tính toán chi phí.
* Với hệ số nhân rộng là 1,4, MobileNetV2 (1,4) vượt trội hơn [ShuffleNet](https://towardsdatascience.com/review-shufflenet-v1-light-weight-model-image-classification-5b253dfe982f) (× 2) và [NASNet](https://medium.com/@sh.tsang/review-nasnet-neural-architecture-search-network-image-classification-23139ea0425d) với thời gian suy luận nhanh hơn.



**Hình 2. 23** So sánh kích thước mô hình và tính toán

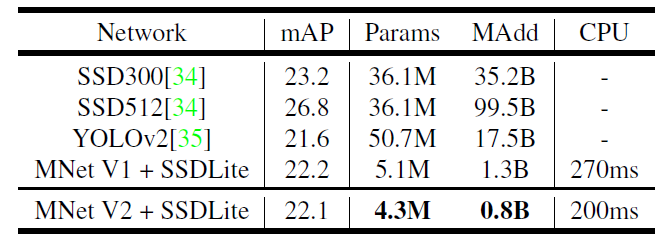
* Như được hiển thị ở trên, các độ phân giải đầu vào khác nhau và số nhân chiều rộng được sử dụng. Nó luôn hoạt động tốt hơn [MobileNetV1](https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69) .

**Phát hiện đối tượng MS COCO**



**Hình 2. 24** SSDLite

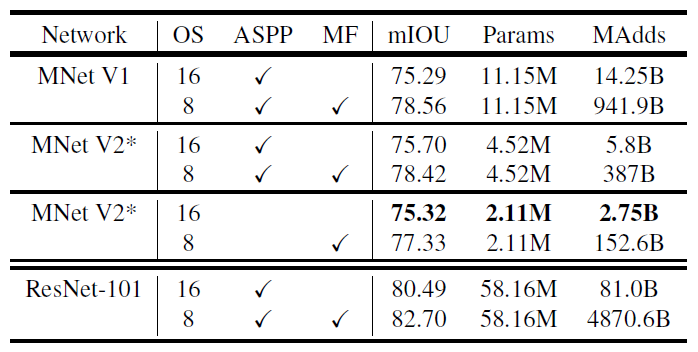
* Đầu tiên, SSDLite được giới thiệu bằng cách sửa đổi các chập thông thường trong [SSD](https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11) với các chập có thể phân tách theo chiều sâu ( [MobileNetV1](https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69) một).
* SSDLite giảm đáng kể cả số lượng tham số và chi phí tính toán.



**Hình 2. 25** Phát hiện đối tượng MS COCO

* MobileNetV2 + SSDLite đạt được độ chính xác cạnh tranh với ít tham số hơn đáng kể và độ phức tạp tính toán nhỏ hơn.
* Và thời gian suy luận nhanh hơn [MobileNetV1](https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69) một.
* Đáng chú ý, MobileNetV2 + SSDLite hiệu quả hơn 20 lần và nhỏ hơn 10 lần trong khi vẫn hoạt động tốt hơn YOLOv2 trên tập dữ liệu COCO.

**PASCAL VOC 2012** Phân đoạn ngữ nghĩa



**Hình 2. 26** Bộ xác thực PASCAL VOC 2012

Ở đây, MobileNetV2 được sử dụng làm trình trích xuất tính năng cho [DeepLabv3](https://towardsdatascience.com/review-deeplabv3-atrous-convolution-semantic-segmentation-6d818bfd1d74) .

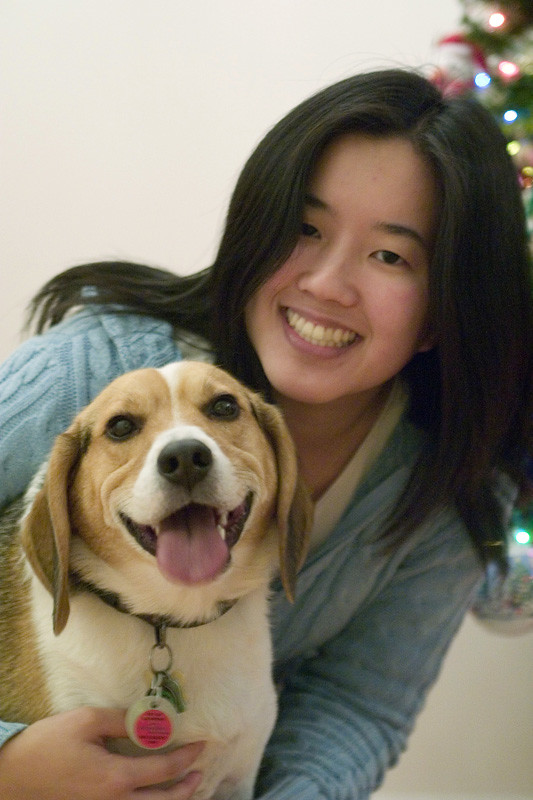
* Với việc vô hiệu hóa Atrous Spatial Pyramid Pooling (ASPP) cũng như Multi-Scale và Flipping (MP), đồng thời thay đổi bước đầu ra từ 8 thành 16, thu được mIOU 75,32%, với kích thước mô hình và chi phí tính toán thấp.

# CHƯƠNG III. THỰC NGHIỆM VÀ ĐÁNH GIÁ

## Quá trình thực nghiệm.

### Xác định bài toán cần thực hiện.

Bài toán nhận dạng giống loài chó mèo là sự kết hợp giữa hai bài toán là dectec ra con chó/ mèo và nhận dạng loài của chúng. Với cuộc sống ngày càng phát triển việc nuôi pet đã trở thành xu thế vậy nhưng để có thể biết hết về thông tin thú cưng thì không phải ai cũng biết thế nên bài toán đã được ra đời.



**Hình 3. 1** Ảnh cần xử lý

## Các bước xây dựng và ứng dụng model train dữ liệu.

Sử dụng yoloV3 để detec ra chó và mèo trong ảnh đê cho vào mô hình.

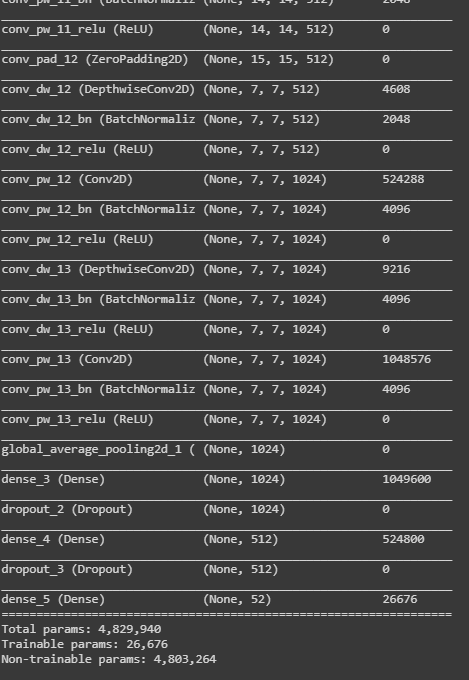


**Hình 3. 2** Ảnh trước và sau khi qua YoloV3

Sau khi sử dụng YoLoV3 ta sẽ tiếp tục sử dụng mô hình MobileNet V2 để nhận diện ra loài của chúng. Ở đây ta xây dựng một mạng MobileNetV2, với input đầu vào có chiều là input\_shape =(24x24x3) tương ứng với số chiều của dữ liệu.

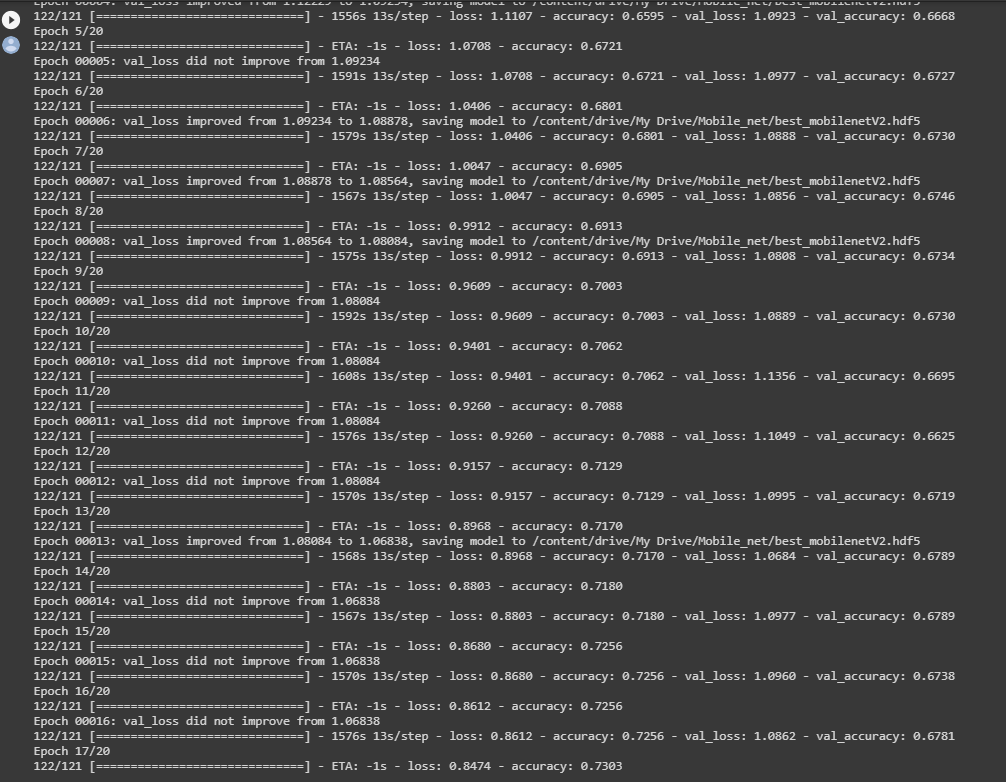
Với hàm activation và các tham số được chọn trên những thử nghiệm trong quá trình training.

Sau khi cắt giảm mô hình cho phù hợp với dữ liệu đã chuẩn bị thì mô hình cần học 26,676 prams.

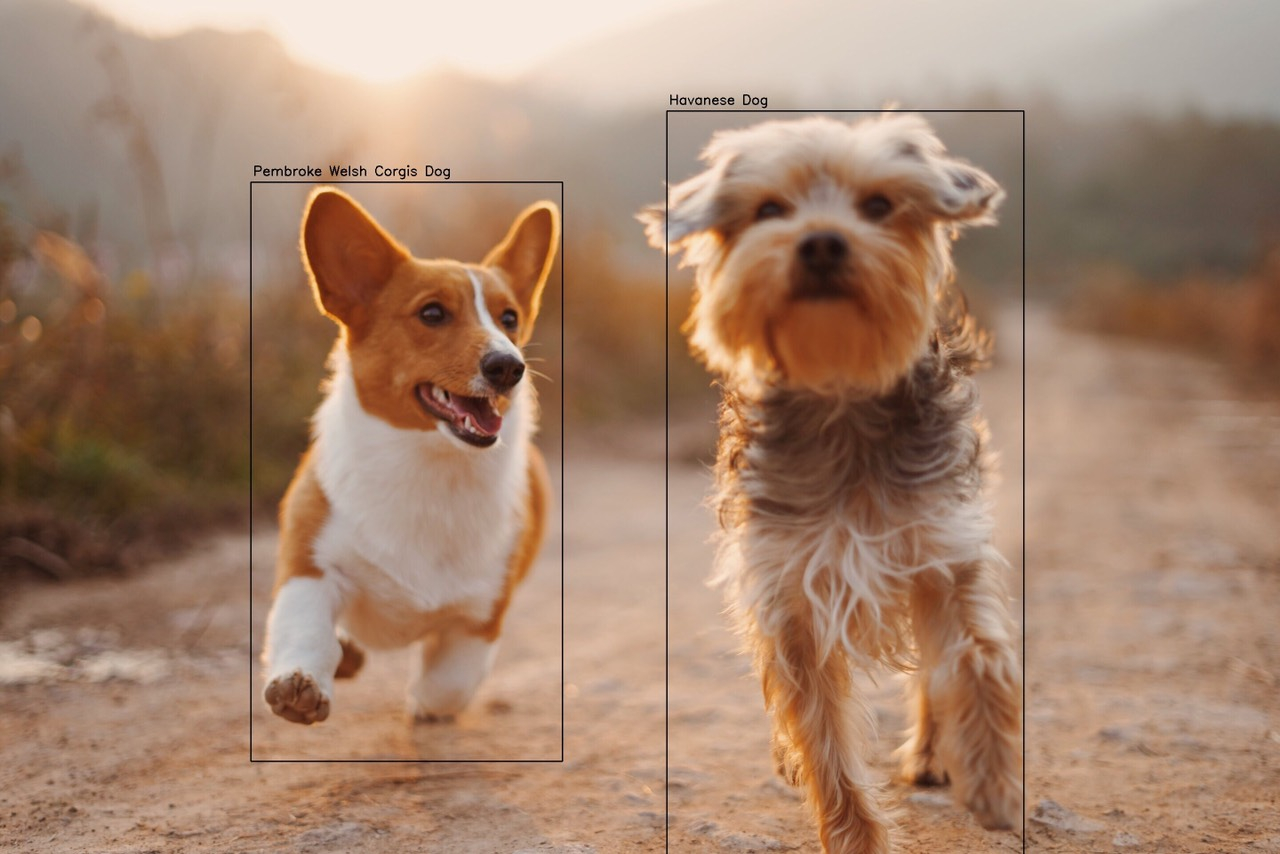


**Hình 3. 3** Mô hình sau khi đã được tranfer

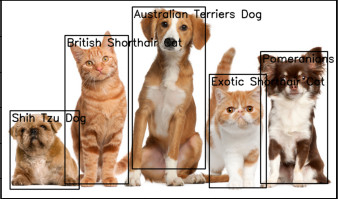
Sau quá trình training với tập dữ liệu mới trên model ta thấy được độ chính xác đạt độ chính xác 73% và chí phí giảm dần.



**Hình 3. 4** Quá trình training với 20 epochs.

* Demo chương trình

**Hình 3. 5** Kết quả với 2 con dog



**Hình 3. 6** Kết quả với nhiều dogs

# KẾT LUẬN

Với bài toán nhận dạng giống loài, ta có thể tiếp cận giải quyết bài toán với các phương pháp công nghệ hiện đại. Với những kết quả trong quá trình tìm hiểu và nghiên cứu ở trên ta có thể phát triển và áp dụng vào cuộc sống.

So sánh với các công nghệ trước đây về nhận dạng giống loài, khi đó người dùng phải lên mạng tra tìm kiếm và kiểm tra xem có đúng hay không, điều đó gây khá nhiều phiền toái với người dùng. Tuy nhiên với giải pháp ứng dụng các công nghệ về xử lý ảnh kết hợp với học sâu, giúp người dùng không cần phải tìm đến chuyên gia hay tìm kiếm một cách khổ sở mà vẫn đáp ứng được kỳ vọng.

Mục tiêu tiếp theo là sẽ tiến tới bài toán image seach để đẩy mọi thông tin tìm kiếm được cho người dùng

# TÀI LIỆU THAM KHẢO

[1] https://arxiv.org/pdf/1905.11946.pdf

[2] https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet

[3] https://www.dienmayxanh.com/kinh-nghiem-hay/tri-tue-nhan-tao-ai-la-gi-ung-dung-nhu-the-nao-tro-1117277

[4]https://smartfactoryvn.com/technology/ai-ml/thi-giac-may-tinh-computer-vision-trong-san-xuat/

[5] https://www.thegioimaychu.vn/blog/ai-deep-learning/lam-the-nao-de-giai-thich-deep-learning-hoc-sau-mot-cach-de-hieu-nhat-p2073/

[6] https://phamdinhkhanh.github.io/2019/09/29/OverviewObjectDetection.html

[7] https://viblo.asia/p/deep-learning-thuat-toan-faster-rcnn-voi-bai-toan-phat-hien-duong-luoi-bo-faster-rcnn-object-detection-algorithm-for-nine-dash-line-detection-bJzKmREOZ9N

[8] https://nttuan8.com/bai-11-object-detection-voi-faster-r-cnn/

[9] https://dominhhai.github.io/vi/2017/10/what-is-rnn/

[10]https://www.google.com/search?q=cnn+l%C3%A0+g%C3%AC&oq=cnn+&aqs=chrome.1.69i57j0l2j69i59j0j69i60l3.1727j0j4&sourceid=chrome&ie=UTF-8

[11]https://blog.chappiebot.com/h%C6%B0%E1%BB%9Bng-d%E1%BA%ABn-chi-ti%E1%BA%BFt-v%E1%BB%81-c%C6%A1-ch%E1%BA%BF-c%E1%BB%A7a-lstm-v%C3%A0-gru-trong-nlp-a1bd9346b209

[12] ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf

[13] https://arxiv.org/pdf/1406.1078v3.pdf

[14] https://arxiv.org/pdf/1508.03790v2.pdf

[15] https://arxiv.org/pdf/1508.03790v2.pdf

[16] https://arxiv.org/pdf/1503.04069.pdf

[17] http://jmlr.org/proceedings/papers/v37/jozefowicz15.pdf

[18] https://arxiv.org/pdf/1611.08050.pdf

[19] https://arxiv.org/pdf/1812.08008.pdf

[20]https://medium.com/beyondminds/an-overview-of-human-pose-estimation-with-deep-learning-d49eb656739b

[21] https://arxiv.org/pdf/1511.06645.pdf

[22] https://en.wikipedia.org/wiki/Integer\_programming

[23] https://arxiv.org/abs/1511.06645

[24] https://arxiv.org/abs/1612.00137

[25] https://arxiv.org/pdf/1612.00137.pdf

[26] https://arxiv.org/abs/1703.06870

[27] https://github.com/CMU-Perceptual-Computing-Lab/openpose

[28]https://unsplash.com/photos/XuN44TajBGo?utm\_source=unsplash&utm\_medium=referral&utm\_content=creditCopyText

[29]https://pan.baidu.com/s/1XT8pHtNP1FQs3BPHgD5f-A#list/path=%2Fsharelink1864347102-902260820936546%2Fopenpose%2Fopenpose%20graph%20model%20coco&parentPath=%2Fsharelink1864347102-902260820936546

[30] COCO dataset, (2015). COCO 2018 Keypoint Detection Task. [online] Available from: