

ĐHQG TP.HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

🙢-------🕮-------🙠

**ĐỒ ÁN NHẬN DIỆN BIỂN SỐ XE**



LỚP: Nhận dạng - CS338

GIÁO VIÊN HƯỚNG DẪN: Đỗ Văn Tiến

MỤC LỤC

[**CHƯƠNG I: TỔNG QUAN** 4](#_30j0zll)

[**1. Đặt vấn đề** 4](#_1fob9te)

[**2. Thách thức, mục tiêu, phạm vi:** 6](#_3znysh7)

[**2.1. Thách thức** 6](#_2et92p0)

[**2.2. Mục tiêu** 6](#_tyjcwt)

[**2.3. Phạm vi** 7](#_3dy6vkm)

[**3. Đóng góp của đồ án** 7](#_1t3h5sf)

[**4. Cấu trúc của đồ án** 8](#_4d34og8)

[**CHƯƠNG II: CƠ SỞ LÝ THUYẾT VÀ MỘT SỐ BÀI TOÁN CÓ LIÊN QUAN** 8](#_2s8eyo1)

[**1. Mô hình phát hiện vật thể YOLO** 8](#_17dp8vu)

[**2. Mô hình WPOD** 16](#_3rdcrjn)

[**3. Mô hình MobileNET** 20](#_26in1rg)

[**4. Một số bài toán liên quan đến nhận dạng** 27](#_lnxbz9)

[**CHƯƠNG III: PHƯƠNG PHÁP** 29](#_35nkun2)

[**CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 43](#_44sinio)

[**1.**](#_2jxsxqh) **Kết luận** 43

[**2. Hướng phát triển** 44](#_z337ya)

[**TÀI LIỆU THAM KHẢO** 44](#_3j2qqm3)

# 

# **CHƯƠNG I: TỔNG QUAN**

## **1. Đặt vấn đề**

Hiện nay, với sự phát triển kinh tế xã hội bùng nổ số lượng phương tiện giao thông. Gây ra những khó khăn trong công tác quản lý, xử lý vi phạm giao thông mà nguồn nhân lực con người khó có thể đảm đương được. Vì vậy vấn đề cần thiết là có một hệ thống quản lý, xử lý vi phạm giao thông tự động.

Để xây dựng hệ thống quản lý giao thông thì “Bài toán nhận dạng biển số xe” là tiền đề để xây dựng những mô hình quản lý đó. Tuy nhiên, hiện nay ở Việt Nam những hệ thống quản lý đó chưa nhiều và lĩnh vực nhận dạng còn đang phát triển.

Từ thực tế đó, trong thời gian làm đồ án môn học, nhóm chúng em đã quyết định lựa chọn bài toán nhận dạng trong đó chú trọng vào việc nhận dạng biển số xe. Để phục vụ cho bước đầu trong việc quản lý giao thông tại Việt Nam. Trong điều kiện năng lực và thời gian có hạn, nên báo cáo của chúng em không tránh những sai sót. Kính mong thầy và các bạn có thể đóng góp ý kiến để chúng em có thể chỉnh sửa và bổ sung những phần thiếu sót để em hoàn thiện đề tại của mình.

Chúng em xin cảm ơn những lời góp ý chân thành của thầy để chúng em có thể hoàn thành đồ án này một cách tốt nhất có thể.

Bài toán nhận dạng biển số xe được mô tả như sau:

Input : 1 ảnh có thể chứa một hay nhiều biển số xe



Output: Vẽ bounding box cho biển số, hiện số của biển số đã được nhận diện.



## **2. Thách thức, mục tiêu, phạm vi:**

### **2.1. Thách thức**

* Về phương pháp: Đối với bài toán nhận diện biển số xe cần thực hiện hai phần chính là phát hiện và nhận diện biển số. Kết quả của phần trước sẽ ảnh hưởng rất lớn đến phần sau. Cho đến thời điểm này có rất nhiều phương pháp Deep Learning nhưng việc lựa chọn các phương pháp phù hợp để đảm bảo độ chính xác, thời gian xử lý và tài nguyên sử dụng là một vấn đề cần quan tâm.
* Về mặt dữ liệu: dữ liệu hiện nay còn hạn chế, đa phần các ảnh sẽ không rõ nét. Đây là một thách thức khi áp dụng các phương pháp học sâu vốn cần rất nhiều dữ liệu để huấn luyện. Áp dụng với dữ liệu thực tế, số lượng phương tiện giao thông rất đa dạng về mẫu mã, kích thước,...
* Về mặt công nghệ: Khi xây dựng một hệ thống thực tế, việc lựa công nghệ và kiến trúc hệ thống để đảm bảo hệ thống hoạt động một hiệu quả là vấn đề đáng được cân nhắc và xem xét thận trọng.

### **2.2. Mục tiêu**

Đồ án tập trung vào tìm hiểu bài toán nhận diện biển số xe, để thực hiện và đánh giá đồ án nhóm em đã đề ra những mục tiêu như sau:

- Tìm hiểu tổng quan về bài toán nhận diện biển số xe đối với ảnh tĩnh và một số phương pháp hiện nay.

- Tìm hiểu một số bài toán có liên quan và phân tích phương pháp thực hiện.

- Xây dựng chương trình minh họa áp dụng các mô hình đã thực hiện để nhận diện biển số xe trong ảnh tĩnh.

### **2.3. Phạm vi**

Trong phạm vi của đồ án chúng em tập trung hoàn thành các nội dung sau:

- Tìm hiểu tổng quan về kiến thức trong lĩnh vực xử lý ảnh liên quan đến bài toán nhận diện biển số xe. Tập trung vào các hướng tiếp cận, phương pháp sử dụng mô hình Deep Learning, cụ thể là các mô hình YOLO, WPOD và mô hình MobileNet.

- Tập trung đánh giá và phân tích các phương pháp đã thực hiện trên bộ dữ liệu bộ ảnh biển số xe máy của công ty GreenParking với các độ đo mAP cho bài toán Detection và Accuracy cho bài toán Classification. Qua đó, tập trung áp dụng mô hình có kết quả tốt nhất và nhanh nhất vào bài toán nhận diện biển số xe trong thực tế.

- Xây dựng chương trình minh họa cho bài toán nhận diện biển số xe trên ảnh tĩnh.

## **3. Đóng góp của đồ án**

**-** Tìm hiểu tổng quan về bài toán nhận diện biển số xe và một số kiến thức liên quan.

- Đánh giá và phân tích một số phương pháp Deep Learning tiến tiến hiện nay cho bài toán trên tập dữ liệu.

- Xây dựng demo cho bài toán nhận diện biển số xe. Cho phép người dùng nhập vào một hình ảnh chứa phương tiện giao thông và xuất ra biển số của phương tiện đó.

## **4. Cấu trúc của đồ án**

Các phần còn lại của đồ án như sau:

Chương II: Trình bày các cơ sở lý thuyết liên quan đến bài toán nhận diện biển số xe.

Chương III: Trình bày chi tiết các phương pháp thực hiện

Chương IV: Trình bày kết quả thực nghiệm và đánh giá độ chính xác của mô hình thông qua các thông số đánh giá.

Chương V: Trình bày kết luận và hướng phát triển của đồ án trong tương lai

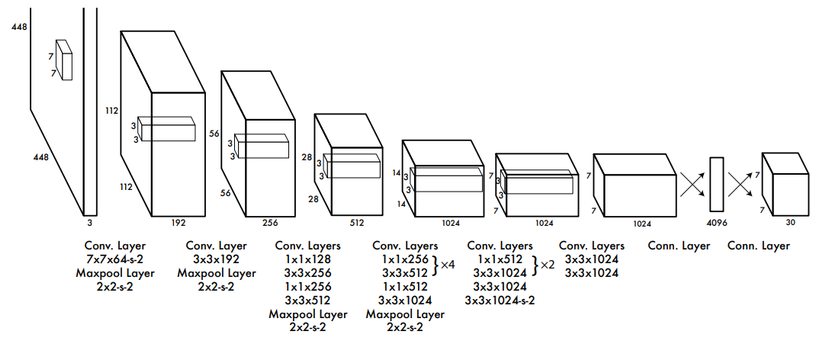
# **CHƯƠNG II: CƠ SỞ LÝ THUYẾT VÀ MỘT SỐ BÀI TOÁN CÓ LIÊN QUAN**

## **1. Mô hình phát hiện vật thể YOLO**

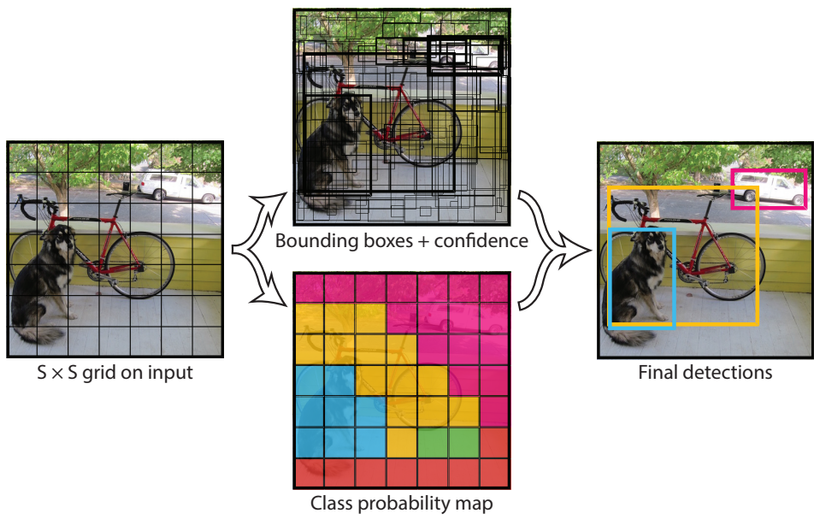
Object Detection là một bài toán quan trọng trong lĩnh vực Computer Vision, thuật toán Object Detection được chia thành 2 nhóm chính:

* Họ các mô hình RCNN ( Region-Based Convolutional Neural Networks) để giải quyết các bài toán về định vị và nhận diện vật thể.
* Họ các mô hình về YOLO (You Only Look Once) dùng để nhận dạng đối tượng được thiết kế để nhận diện các vật thể real-time.

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.



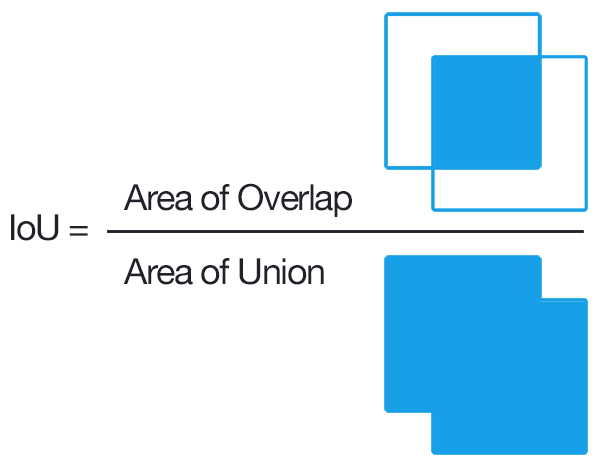
Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh. Ảnh đầu vào được chia thành thành *S*×*S* ô thường thì sẽ là 3×3, 7×7, 9×9... việc chia ô này có ảnh hưởng tới việc mô hình phát hiện đối tượng.



Với Input là 1 ảnh, đầu ra mô hình là một ma trận 3 chiều có kích thước *S*×*S*×(5×*N*+*M*) với số lượng tham số mỗi ô là (5×*N*+*M*) 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 7×7 ô, mỗi ô cần dự đoán 2 bounding box và 3 object : con chó, ô tô, xe đạp thì output là 7×7×13, mỗi ô sẽ có 13 tham số, kết quả trả về (7×7×2=98) bounding box. Chúng ta sẽ cùng giải thích con số (5×*N*+*M*) 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*) 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)). 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 bounding box hay đối tượng thì sẽ không thể detect được, đó là một hạn chế của mô hình YOLOv1, 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.

Trên ta có đề cập prediction được định nghĩa Pr(*Object*)∗ *IOU*(*pred*,*truth*), ta sẽ làm rõ hơn IOU(pred, truth) là gì. IOU (INTERSECTION OVER UNION) là hàm đánh giá độ chính xác của object detector trên tập dữ liệu cụ thể. IOU được tính bằng:

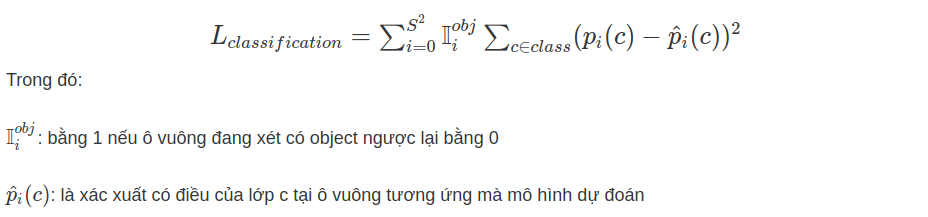


Trong đó Area of Overlap là diện tích phần giao nhau giữa predicted bounding box với ground truth bounding box , còn Area of Union là diện tích phần hợp giữa predicted bounding box với ground truth bounding box. Những bounding box được đánh nhãn bằng tay trong tập training set và test set. Nếu IOU > 0.5 thì prediction được đánh giá là tốt.

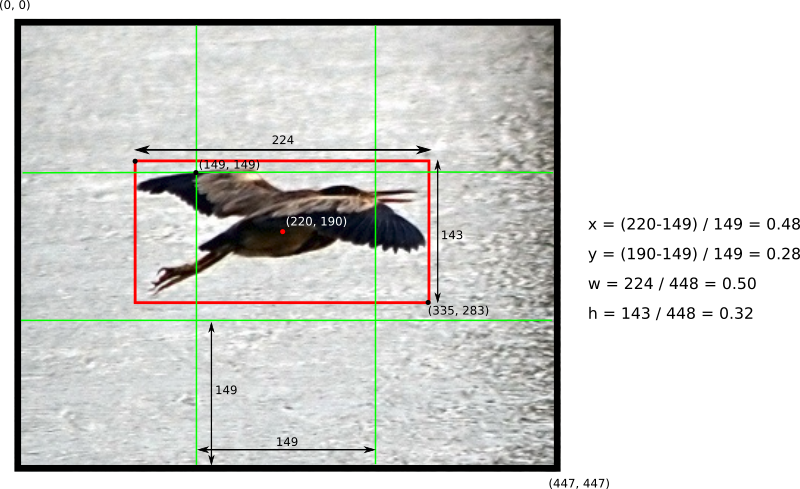
Hàm lỗi trong YOLO được tính trên việc dự đoán và nhãn mô hình để tính. Cụ thể hơn nó là tổng độ lỗi của 3 thành phần con sau :

* Độ lỗi của việc dự đoán loại nhãn của object - Classification 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 - độ lỗi của việc dự đoán loại nhãn của 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 được tính bằng công thức sau:

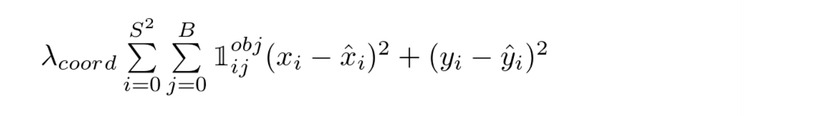


Localization loss là hàm lỗi dùng để tính giá trị lỗi cho boundary box được dự đoán bao gồm tọa độ tâm, chiều rộng, chiều cao của so với vị trí thực tế từ dữ liệu huấn luyện của mô hình. Lưu ý rằng chúng ta không nên tính giá trị hàm lỗi này trực tiếp từ kích thước ảnh thực tế mà cần phải chuẩn hóa về [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:

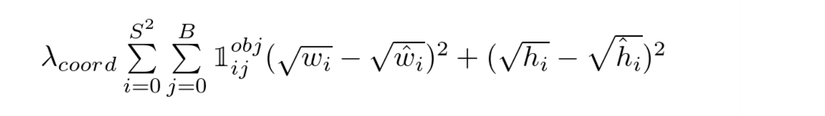


Giá trị hàm Localization loss được tính trên tổng giá trị lỗi dự đoán toạ độ tâm (x, y) và (w, h) của predicted bounding box với ground truth bounding box. Tại mỗi ô có chưa object, ta chọn 1 boundary box có IOU (Intersection over union) tốt nhất, rồi sau đó tính độ lỗi theo các boundary box này.

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 :

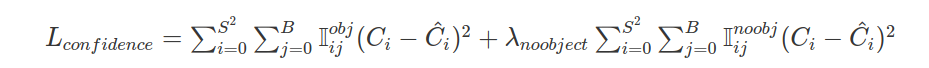


Giá trị hàm lỗi dự đoán (w, h ) của predicted bounding box so với truth bounding box được tính như sau:



Với ví dụ trên thì S =7, B =2, còn λcoord là trọng số thành phần trong paper gốc tác giả lấy giá trị là 5.

Confidence loss là độ lỗi giữa dự đoán boundary box đó chứa object so với nhãn thực tế tại ô vuông đó. Độ lỗi này tính trên cả những ô vuông chứa object và không chứa object.

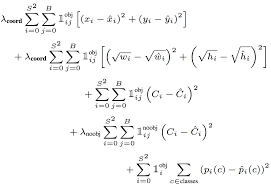


Với ví dụ trên thì S =7, B =2, còn λnoobject là trọng số thành phần trong paper gốc tác giả lấy giá trị là 0.5. Đối với các hộp j của ô thứ i nếu xuất hiệu object thì Ci =1 và ngược lại.

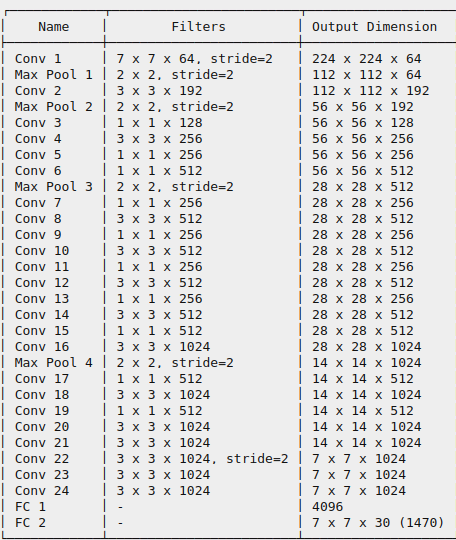
Tổng lại chúng ta có hàm lỗi là tổng của 3 hàm lỗi trên:



hay:



Như ta đã nói ở trên mô hình mạng YOLO là một mô hình mạng CNN thông thường gồm các convolutional layers kết hợp max pooling layers và cuối cùng là 2 lớp fully connected layers, với hàm kích hoạt cho layer cuối cùng là một linear activation function và tất cả các layers khác sẽ sử dụng leaky RELU :

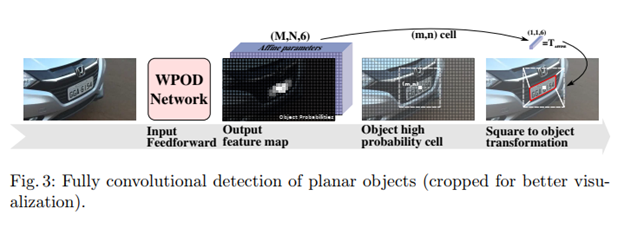


## **2. Mô hình WPOD**

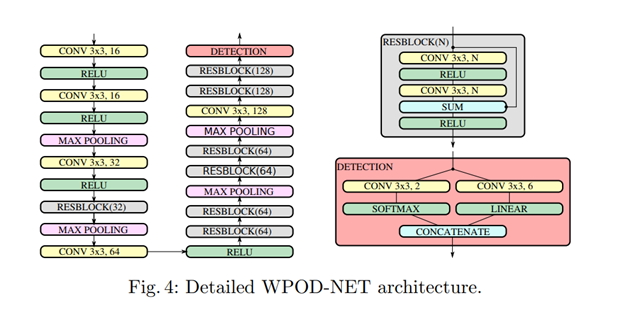
Wpod-Net ( Warped Planar Object Detection Network ). Mô hình này được huấn luyện detect cho các vật thể có tính mặt phẳng trong không gian 2 chiều thích hợp cho các bài toán về License Plate ( LP ) – biển số xe.

Wpod-Net có thể phát hiện các biển số xe trong nhiều tình trạng khác nhau ( biển bị nghiêng, biển dọc, biển mờ, biển ở nhiều tỉ lệ khác nhau, … ) và tạo ra các thông số biến đổi biển số bị biến dạng về dạng chữ nhật như ta thường quan sát.

        Wpod-Net được phát triển dựa trên các ý tưởng bên trong YOLO, SSD và Spartial Transformer Networks (STN), bởi *sergiomsilva*. YOLO và SSD có thể phát hiện nhanh nhiều object và nhận diện chúng trong một lần, nhưng những mô hình này chỉ tạo ra một bounding box hình chữ nhật cho mỗi kết quả detection mà không thực hiện bất kì phép biến đổi không gian nào ( Giảm độ chính xác với các vật thể không phải hình chữ nhật ). STN có thể phát hiện được những vật thể non- rectangular ( không phải hình chữ  nhật ). Tuy nhiên mô hình không thể xử lý quá nhiều phép biến đổi cùng lúc mà chỉ một phép biến đổi cho cả input ( không xử nhiều object cùng lần ).



Quá trình detection sử dụng Wpod-Net được mô tả như hình trên. Khởi tạo, dữ liệu được đưa vào mạng là ảnh đã được resized có chứa biển bảo. Sau bước Feed- forwarding, kết quả là một feature map ( ảnh đặc trưng với 8 channel ) chưa mã hóa object/non-object probabilites ( độ xác nhận vật thể có hay không trong ảnh ) và các tham số cho phép biến đổi affine để sử dụng ở các bước sau. Để lấy ra ảnh biển số biến dạng  ( chẳng hạn ví dụ là biển nghiêng ), giả sử có một hình vuông ảo kích thước cố định bao quanh ô (m,n) cần detect. Nếu object probability tạo ô (m,n) này vượt một ngưỡng detection cho trước ( tức là xác nhận có vật thể nằm trong hình vuông này ). Phần tham số affine trích ra từ trước dùng để tạo ra một ma trận biến đổi affine biến hình vuông giả sử thành hình tứ giác khớp với biển số xe có trong ảnh. Do đó mô hình này có thể giúp biến đổi biển số biến dạng thành một biển số phẳng hình chữ nhật.

****

Kiến trúc mạng Wpod-Net, gồm

-   21 convolution layer với 14 layer trong các Res block ( block như trong Res-Net ). Các lớp convolution đều có kích thước 3x3.

-   Hàm activation được sử dụng là hàm ReLU, trừ block Dectection.

-   4 max-pooling layer 2x2, stride 2, làm giảm số chiều của của input ( về bội của 16 )

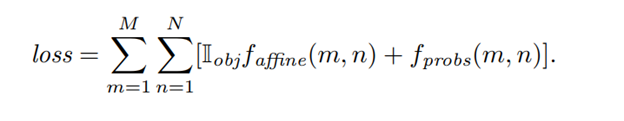
-   Detection block gồm 2 convolution layer song song :

+ 1 layer xử lý probability để đưa vào sofmax dự đoán object

+ 1 layer xử lý các parameter affine, không sử dụng activation ( hoặc sử dụng hàm linear như f(x) = x ).

2 kết quả được nối với nhau bằng phép concatenate để đưa ra kết quả Detection

**Loss Funtion :**



Trong đó :

-     là hàm xác nhận vật thể cần detection có xuất hiện trong hình không ( trả về 1 nếu có, ngược lại trả về 0 )

-   Hàm faffine là hàm trả về độ lỗi giữa phiên bản warped của hình vuông detect với bounding box nhãn có các đỉnh đã được chuẩn hóa



-   Hàm fprobs là hàm xử lý khả năng mà vật thể có hay không có tại ô (m,n)



**Mô hình Pretrained**

Để tạo ra mô pre\_trained, tác giả của mô hình đã sử dụng bộ dataset gồm 196 ảnh, 105 ảnh từ tập Cars Dataset, 40 ảnh từ SSIG dataset và 51 ảnh từ AOLP dataset

Sử dụng phương pháp Augmentation để tăng cường thêm dữ liệu train :

-   Rectification

-   Aspect-ratio

-   Centering

-   Centering

-   Rotation:

-   Mirroring

-   Translation

-   Cropping

-   Colorspace

-   Annotation

Mô hình được train với 100.000 iterations với mini-batche size là 32. Sử dụng Optimizer Adam.

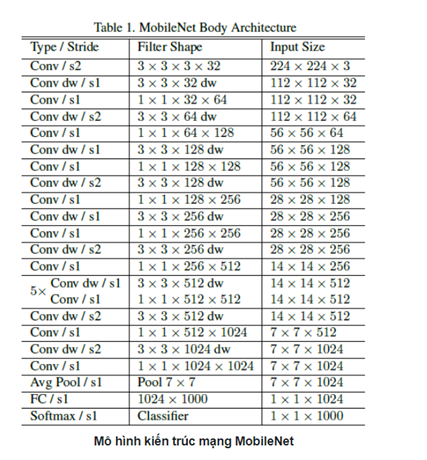
## **3. Mô hình MobileNET**

MobileNet là mô hình được một nhóm nghiên cứu đến từ Google phát triển. Điểm cải tiến của mô hình là phần lớn được xây dựng dựa trên phép tích chập Depthwise Separable Convolution để giảm kích thước mô hình và giảm độ phức tạp tính tóa. Cách tiếp cận đến ý tưởng mô hình dựa trên việc nghiên cứu tạo ra một mô hình có độ chính xác cao nhưng có kích thước nhỏ. MobileNet tập trung vào tối ưu việc tính toán nhanh và là một mạng kích thước nhỏ. Nhờ đó mô hình có thể hữu ích khi xây dựng các ứng dụng trên di động và các thiết bị nhúng :

-       Mô hình có ít tham số hơn nên kích thước sẽ nhỏ hơn, việc tính toán sẽ nhanh và hiệu quả hơn

-       Mô hình có ít phép tính hơn, độ phức tạp mô hình cũng sẽ giảm

**Mô hình kiến trúc mạng MobileNetV1 :**

****

Mô hình gồm có 30 layer bao gồm các layer :

        - Layer 1 : Convolution layer ( stride 2 )

        - Layer 2 : Depthwise layer

        - Layer 3 : Pointwise layer

        - Layer 4 : Depthwise layer ( stride 2 )

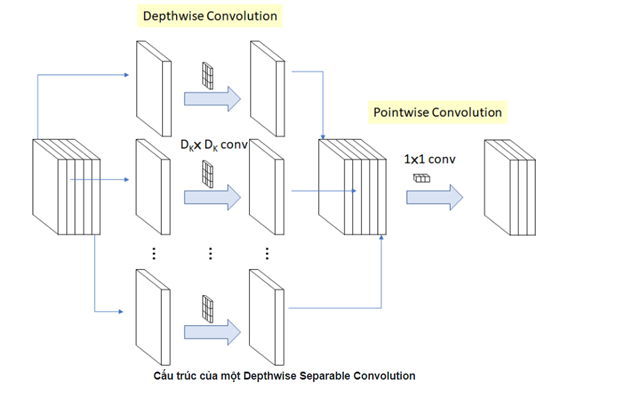
        - Layer 5 : Pointwise layer

        - …

        - Layer 30 : Sofmax layer ( phân lớp )

**Depthwise Separable Convolution**

Depthwise separable convolution là một *depthwise convolution* theo sau bởi một *pointwise convolution* như hình bên dưới :

****

- Depthwise convolution: là một *channel-wise DK×DK spatial convolution*. Ví dụ ở hình trên, ta có 5 channels (các bạn để ý cục đầu tiên có 5 khối hộp, cục thứ 2 là phân tách 5 khối hộp ra thành ma trận mxn, cục thứ 3 là spatial convolution có kích thước kxk, cục thứ 4 là kết quả sau khi convolution, cục thứ 5 là ráp 5 cái kết quả của convolution lại ), do đó chúng ta sẽ có 5 DK×DK spatial convolution tương ứng với 5 channel trên.

- Pointwise convolution: đơn giản là một convolution có kích thước 1x1 (như hình ở trên).

Với M là số lượng input channel, N là số lượng output channel, Dk là kernel size, Df là feature map size (với dataset ImageNet thì input có kích thước là 224, do đó feature map ban đầu có Df = 224), chúng ta có thể tính được:

Chi phí tính toán của Depthwise convolution là :

Dk⋅Dk⋅M⋅Df⋅DfDk⋅Dk⋅M⋅Df⋅Df

Chi phí tính toán của Pointwise convolution là :

M⋅N⋅Df⋅DfM⋅N⋅Df⋅Df

Tổng chi phí tính toán của Depthwise Separable Convolution là:

Dk⋅Dk⋅M⋅Df⋅Df+M⋅N⋅Df⋅DfDk⋅Dk⋅M⋅Df⋅Df+M⋅N⋅Df⋅Df

Nếu chúng ta không sử dụng Depthwise Separable Convolution mà sử dụng phép convolution như bình thường, chi phí tính toán là

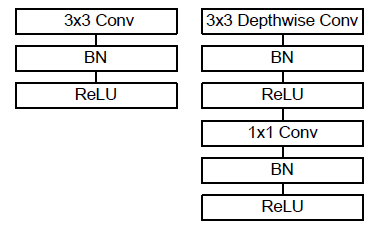
Dk⋅Dk⋅M⋅N⋅Df⋅DfDk⋅Dk⋅M⋅N⋅Df⋅Df

Do đó, chi phí tính toán sẽ giảm:

Dk⋅Dk⋅M⋅Df⋅Df+M⋅N˙Df⋅DfDk⋅Dk⋅M⋅N⋅Df⋅Df=1N+1D2kDk⋅Dk⋅M⋅Df⋅Df+M⋅ND˙f⋅DfDk⋅Dk⋅M⋅N⋅Df⋅Df=1N+1Dk2

Giả sử, chúng ta chọn kernel size Dk = 3, chúng ta sẽ giảm từ 8 đến 9 lần phép tính nhân => giảm chi phí tính toán đi rất nhiều.

Một chú ý nhỏ về kiến trúc ở đây, là sau mỗi convolution MobileNet sẽ sử dụng Batch Normalization (BN) và ReLU như hình bên dưới:



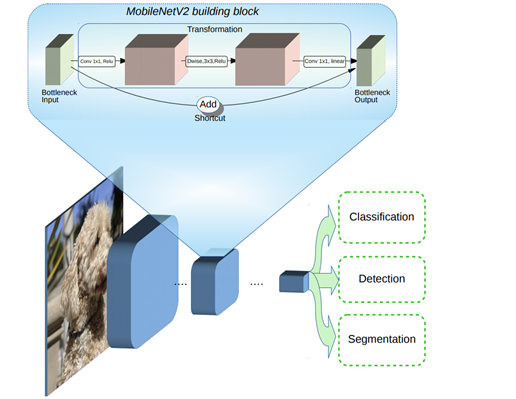
**MobileNetV2**

MobileNetV2 là thế hệ tiếp theo MobileNetV1 với sự cải thiện đáng kể từ mạng tiền nhiệm và thúc đẩy State of the art trong visual recognization bao gồm classification, object detection và semantic segmentation.

MobileNetV2 xây dựng dựa trên ý tưởng MobileNetV1, sử dụng các Depthwise Separable Convolution để tạo nên mô hình. Tuy nhiên model V2 này bổ sung thêm 2 đặc điểm mới vào kiến trúc mô hình :

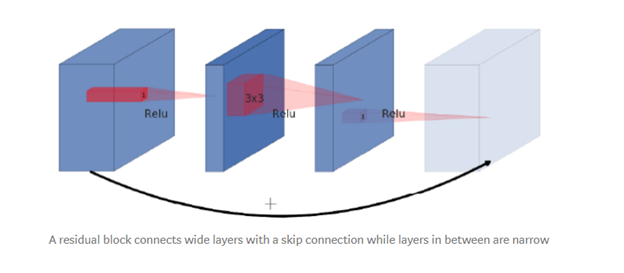
-       Linear Bottleneck giữa các layer

-       Shortcut connection giữa các bottleneck

****

**Inverted Residuals**

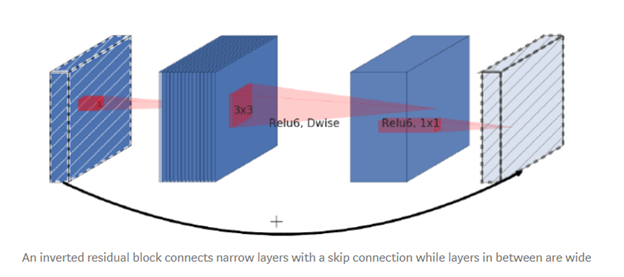
Residual block kết nối giữa layer đầu và cuối của một convolution block bằng một skip connection



Nhờ đó hai trạng thái liên tiếp trong mạng có thể truy cập được đến kết quả hàm activation trước đó mà chưa qua xử lý của convolution block. Hướng tiếp cận này giúp xây dựng mạng sâu hơn.

Một residual block cơ bản sẽ theo một thứ tự wide – narrow – wide ( nếu xét theo số channel của input, output ) như hình trên. Input sẽ có số channel cao ( wide ), qua conv ( 1x1 ) sẽ giảm số channel (narrow ), do đó khi qua tiếp conv 3x3, số parameter sẽ giảm đi. Để đảm có thể kết nối giữa input và output của res block, số channel phải được tăng lên lại nhờ conv ( 1x1 ). Tất cả đều qua hàm activation ReLu.

Tuy nhiên MobileNetV2 lại theo hướng narrow – wide – narrow (Invert Residual ). Đầu tiên là widen ( làm dày ) input bằng convo  1x1, sử dụng depthwise conv (3x3) – giảm rất nhiều các tham số. Cuối cùng một conv (1x1) làm giảm số channels của output để khớp với số channel của input.



**Linear Bottlenecks**

Chúng ta sử dụng các hàm non-linear activation để có thể xây dựng mạng có độ sâu với nhiều layer hơn. Xét hàm non-linearactivation thường được sử dụng trong các mạng neural, ReLU. Hàm này sẽ loại bỏ các giá trị nhỏ hơn 0. Điều này dẫn đên sự mất mát thông tin khi qua các layer. Tuy nhiên có thể khắc phục bằng cách tăng số channel của output khi đưa vào activation.

Với inverted residual block, chúng ta làm giảm số channel của output, điều này ảnh hưởng hiệu quả của mạng neural ( do sự mất thông tin từ ReLU ). Nhóm tác giả của mô hình đã nảy ra ý tưởng sử dụng linear bottleneck đặt ở cuối residual block trước khi linear output của conv cuối được đưa vào hàm activation. Trong coding, ta có thể thực hiện điều này bằng cách xóa đi hàm activation cuối cùng

**Mô hình kiến trúc MobileNetV2 :**



Mô hình thể hiện cách bố trí các bottleneck block.

        - t là expansion rate của channel

        - c là số channel của input

        - n là số lần block lặp lại

## **4. Một số bài toán liên quan đến nhận dạng**

Cùng với sự phát triển không ngừng của kinh tế và xã hội và các ngành kỹ thuật hiện nay. Đòi hỏi sự quản lý và xử lý thông tin chính xác mà nó vượt quá sức của con người. Vì vậy, chúng ta cần có máy móc để làm giảm tải hoặc thay thế công việc nặng nhọc, đòi hỏi sự chính xác cao và nhàm chán cho con người. Việc giúp máy móc nhận dạng (thu thập, phân loại thông tin) như con người sẽ giúp máy móc hoạt động hiệu quả giống như con người với độ chính xác cao hơn rất nhiều.

Một số ứng dụng của bài toán nhận dạng:

Nhận dạng dấu vân tay: ở Việt Nam đã được sử dụng trong việc chấm công, điểm danh và làm khóa an toàn cho các loại cửa sắt, máy tính xách tay,... Nó tạo sự tiện dụng và rất an toàn



Nhận dạng vân tay trên máy ATM

Thay vì phải nhập mã PIN trên máy ATM thì bạn chỉ việc đặt ngón tay của mình lên và giao dịch sẽ bắt đầu.

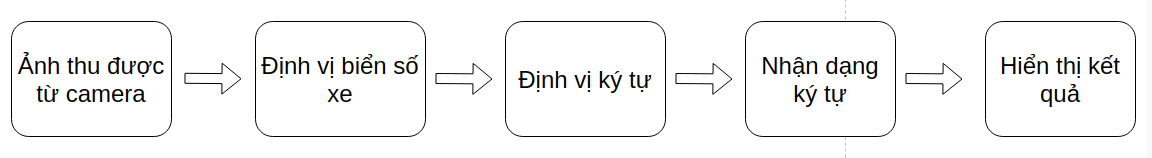
Nhận dạng giọng nói: ứng dụng này được tích hợp trên nhiều sản phẩm, bạn sẽ điều khiển các thiết bị qua giọng nói của mình thay vì phải thao tác trực tiếp bằng tay, như mô hình nhà thông minh, trên điện thoại,... Ví dụ: ứng dụng jibbigo trên điện thoại di động. Ứng dụng này có thể phiên dịch được tiếng nói của người sử dụng với 8 ngôn ngữ khác nhau như tiếng Anh, Pháp, Đức, Nhật, Hàn Quốc, Philipin, Tây Ban Nha, Trung Quốc.



# **CHƯƠNG III: PHƯƠNG PHÁP**

Bài toán nhận diện biển số xe là một bài toán không còn mới. Ứng dụng nhận diện biển số xe được sử dụng rộng rãi trong các bãi giữ xe cấp phát vé tự động, các trạm thu phí trên đường cao tốc hoặc là các hệ thống theo dõi phương tiện tham gia giao thông và phát hiện các phương tiện vi phạm. Hiện nay có rất nhiều các phương pháp khác nhau được đưa ra để giải quyết bài toán này,

Thông thường luồng xử lý của các phương pháp được mô tả như hình dưới đây:



Nhìn vào hình trên ta có thể thấy để giải quyết được bài toán nhận diện biển số xe thì chúng ta phải đi giải quyết ba bài toán con.

* **Bài toán thứ nhất:** Detection / Phát hiện: Định vị vùng chứa biển số xe và tách ra được biển số.
* **Bài toán thứ hai:** Segmentation / Phân tách: Định vị ký tự các ký tự trên biến số.
* **Bài toán thứ ba:** Nhận diện các ký tự chữ số đã tách

1. **Detection / Phát hiện**: Bước đầu tiên phải xác định được biển số nằm ở đâu trong bức ảnh.

Để làm điều đó, chúng em sử dụng pre-trained model wpod-net. Mô hoạt động khá tốt trong việc phát hiện và trích biển số từ ảnh.

* Gọi pre\_trained model
* Load ảnh input và tiền xử lý ảnh (nhị phân hóa ảnh, resize ảnh về kích thước 224x224)
* Lấy ra biển số có trong ảnh ( tọa độ biển số )

Dưới đây là một số kết quả:







1. **Segmentation / Phân tách:** Định vị và tách chữ số trên biển ra.

2.1: Tiền xử lý hình ảnh:

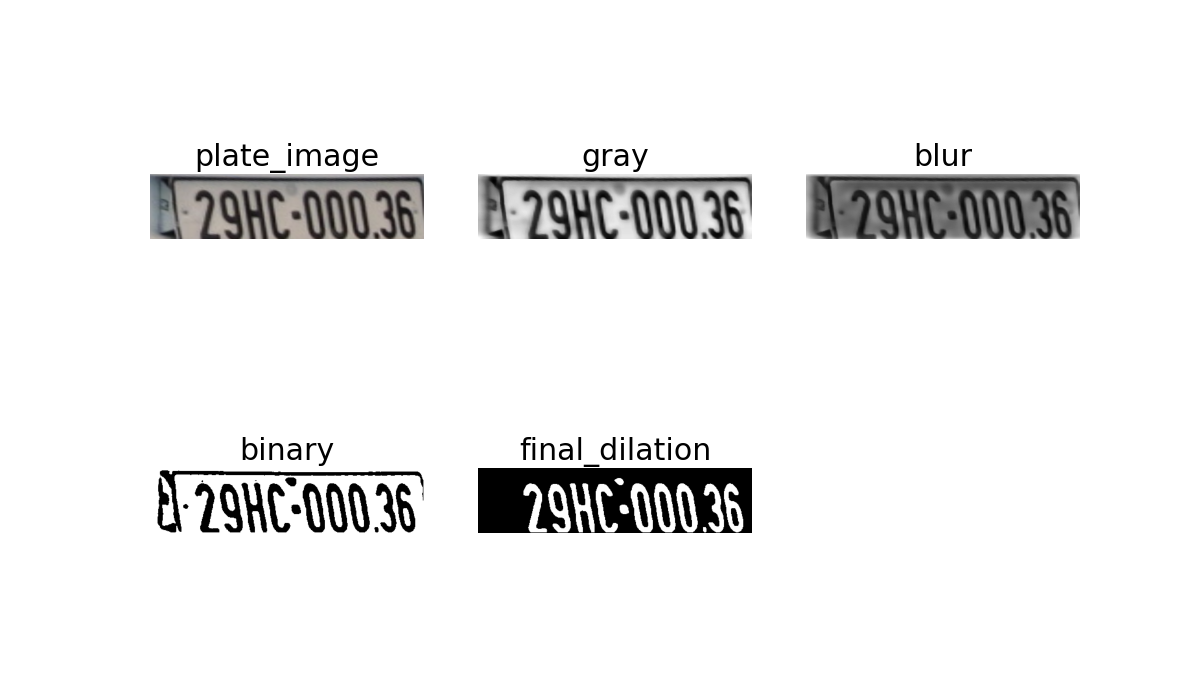
Để giải quyết bài toán **Character segmentation** thì nhóm em sử dụng phương pháp tìm các contour.

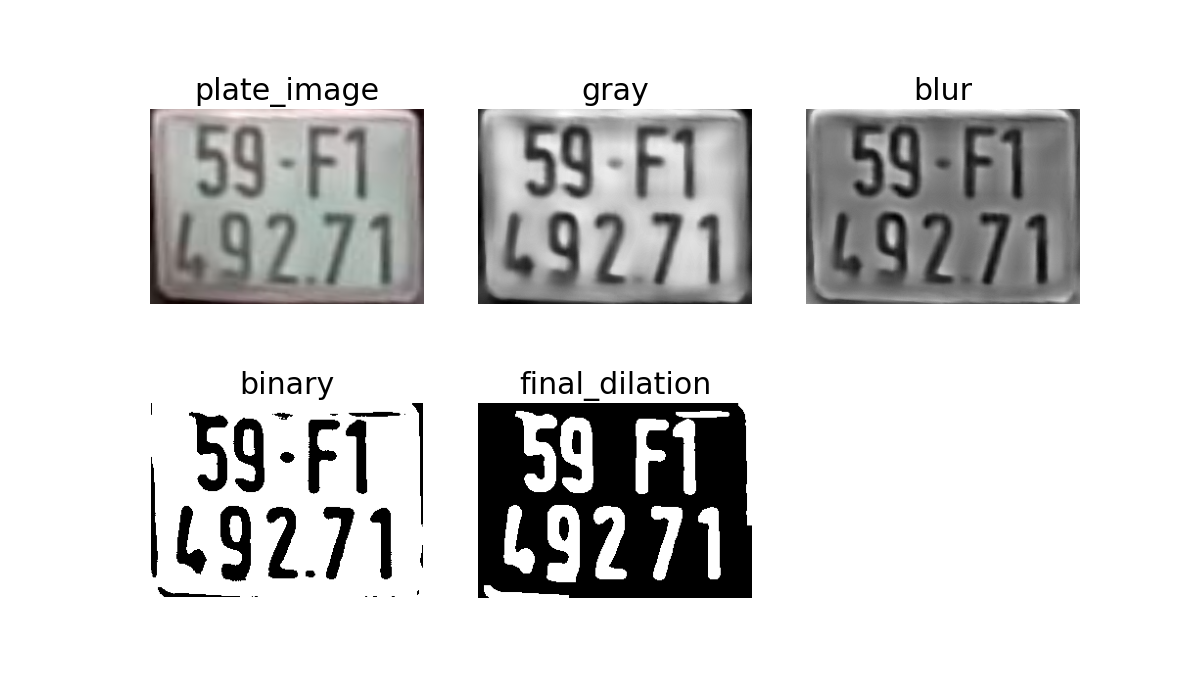
Có thể hiểu một cách đơn giản thì contour là “tập các điểm-liên-tục tạo thành một đường viền (curve) (boundary), và không có khoảng hở trong đường viền đó, đặc điểm chung trong một contour là các các điểm có cùng /gần xấu xỉ một giá trị màu. Contour là một công cụ hữu ích được dùng để phân tích hình dạng đối tượng, phát hiện đối tượng và nhận dạng đối tượng”.

Để tìm contour chính xác, cần phải nhị phân hóa bức ảnh.

Sử dụng các hàm trong thư viện OpenCV để xử lý ảnh chứa biển số xe máy :chuyển về ảnh xám, làm mịn ảnh (Bilateral Filtering),nhị phân hóa ảnh (với ngưỡng động), dilation.

Dưới đây là một số kết quả khi thực hiện tiền xử lý ảnh biển số xe:





2.2: Tách từng ký tự trên biển số:

* Sử dụng hàm cv2.findContours() để tìm các contour:

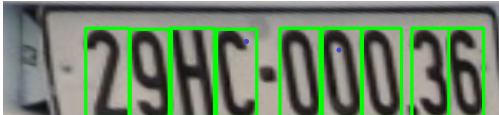
cv2.findContours(binary,cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_SIMPLE)

**mode** :CV\_RETR\_EXTERNAL : khi sử dựng mode này, chỉ lấy ra những đường biên bên ngoài, nhưng biên bên trong của vật thể bị loại bỏ.

**method** : CV\_CHAIN\_APPROX\_SIMPLE : sẽ nén đường viền trước khi lưu trữ, nén phân đoạn theo chiều ngang, chiều dọc và chéo . Ví dụ : một hình chữ nhật sẽ được mã hoá bằng toạ độ của 4 đỉnh.

* Sử dụng hàm cv2.boundingReact() để tìm ra các bounding box hình chữ nhật đứng chứa các contour.
* Thiết lập các giá trị ngưỡng để loại bỏ những bounding box chứa các contour nhiễu:
* Aspect ratio(tỉ lệ rộng / dài) > 1.5 và < 4.5
* Height ratio(tỉ lệ chiều dài kí tự / chiều dài biển số xe): > 0.5 (biển dài) hoặc > 0.33 và < 0.5 (biển vuông)

Kết quả thu được là các ký tự đã được bóc tách:



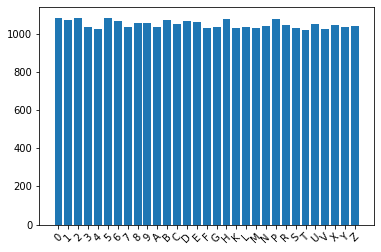


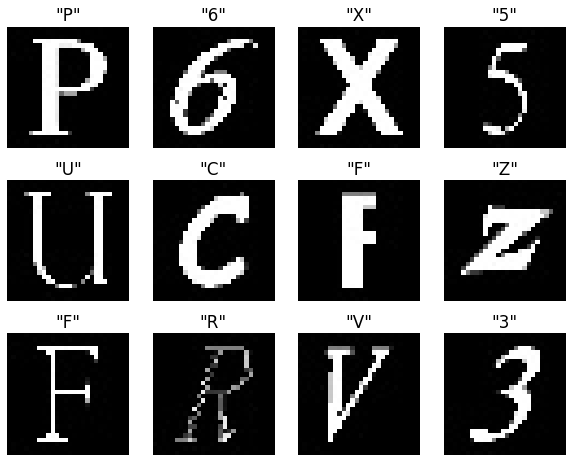


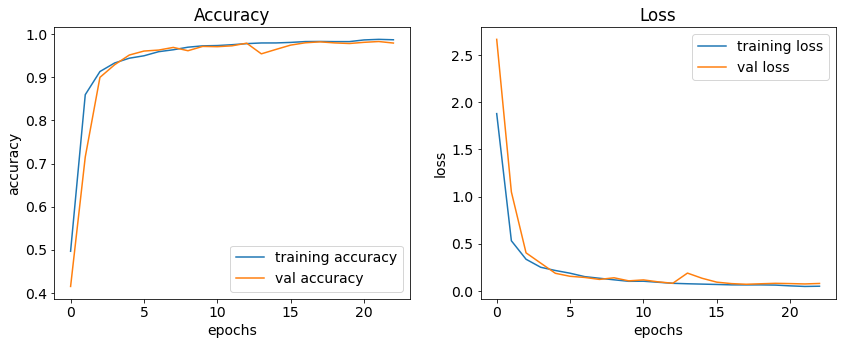
1. **Recognition/ Nhận diện**: Nhận được nội dung biển số và hiển thị trên màn hình

* Ta có thể thấy, các ký tự được bóc tách có hình dạng giống với các ký tự viết tay.
* Vậy nên chúng em quyết đinh sẽ sử dụng phương pháp fine-tuning pre-trained model MobileNetV2 để nhận diện ký tự.
* Ưu điểm:
* Mô hình nhẹ, số lượng parameter ít
* Thời gian train  nhanh
* Độ chính xác cao ( accuracy trên bộ dataset 99% )
* Chúng em sử dụng bộ dataset ký tự viết tay bao gồm chữ số (0..9) và chữ viết trong bảng chữ cái Latin, bỏ đi các ký tự I, J, O, Q, W do chúng không có trong biển số xe
* Bộ dataset này bao gồm 32.543 ảnh thuộc 31 class. Đây là kết quả sau khi visualize data:



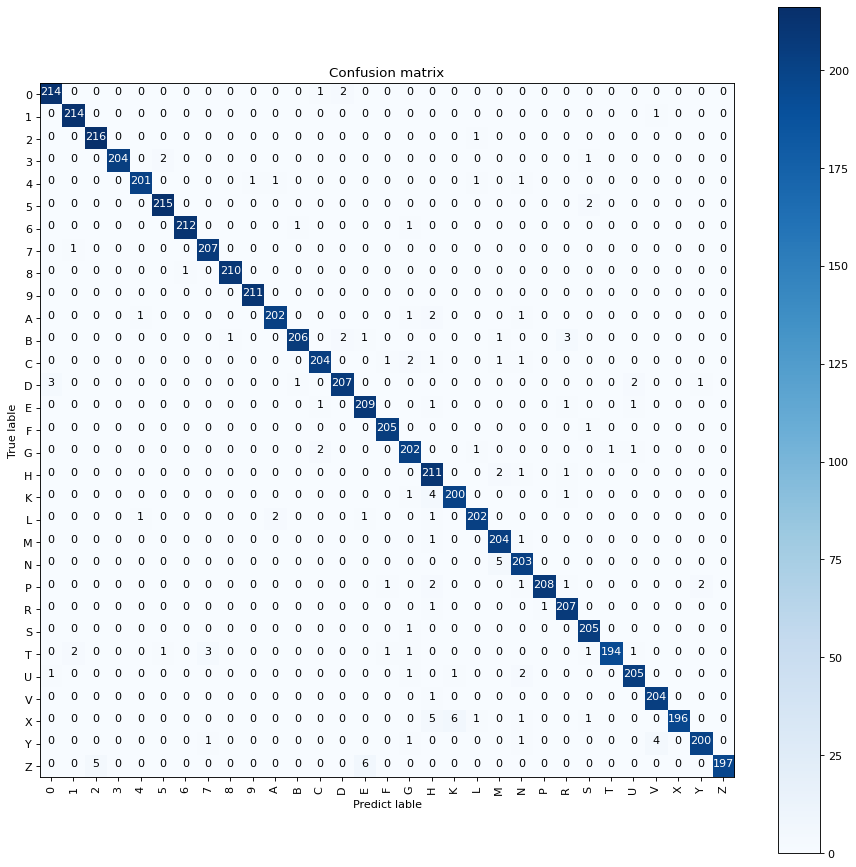


* Bộ dữ liệu được chia thành ba tập train, test và validation theo tỉ lệ 60:20:20. Sau 30 epochs thì thu được độ đo Accuracy lần lượt trên tập train và tập validation là 99% và 98%. Đây là biểu đồ thể hiện Accuracy và Loss của tập train và validation trong suốt quá trình training:

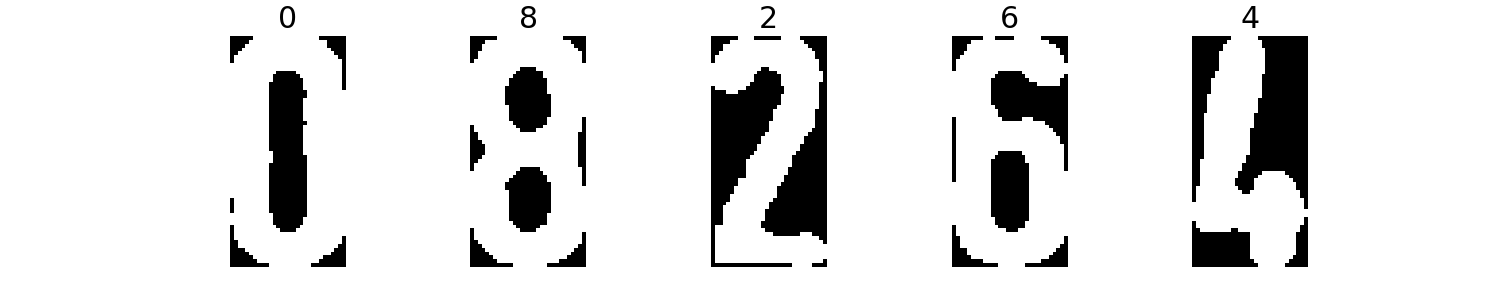
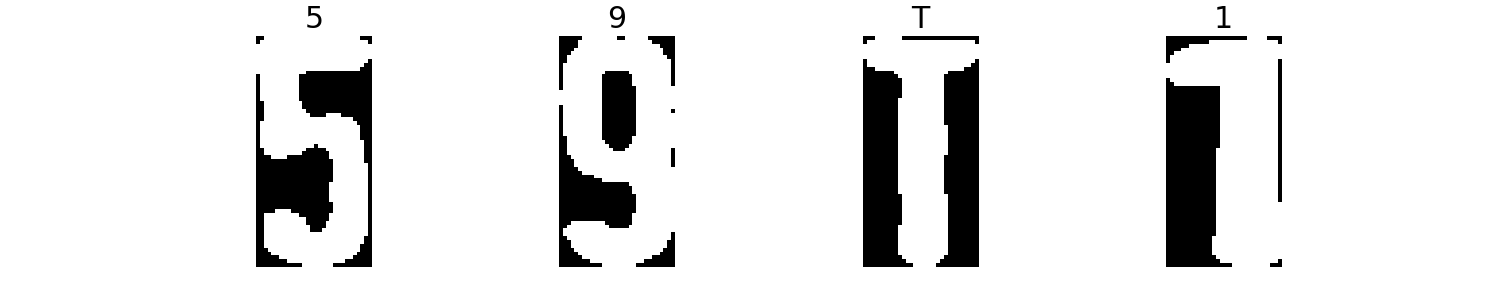


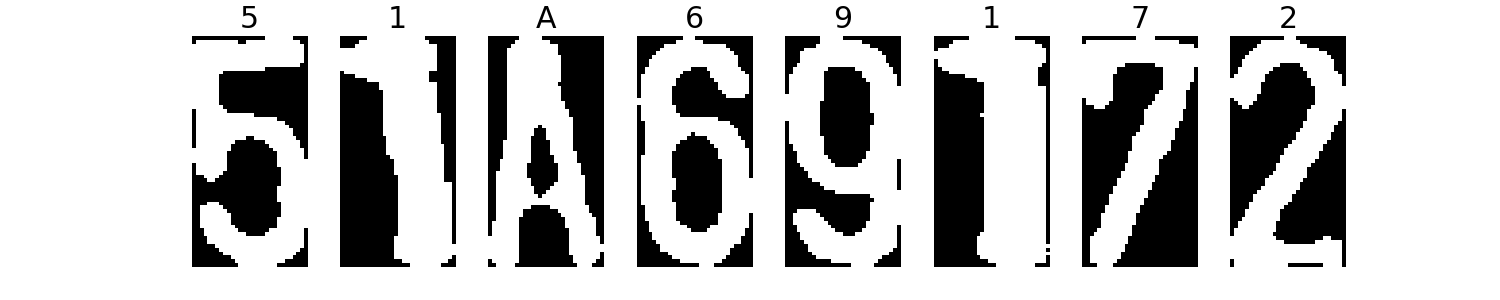
Tiếp theo đó thì nhóm chúng em tiến hành đánh giá độ chính xác mô hình trên tập test thì thu được các kết quả:

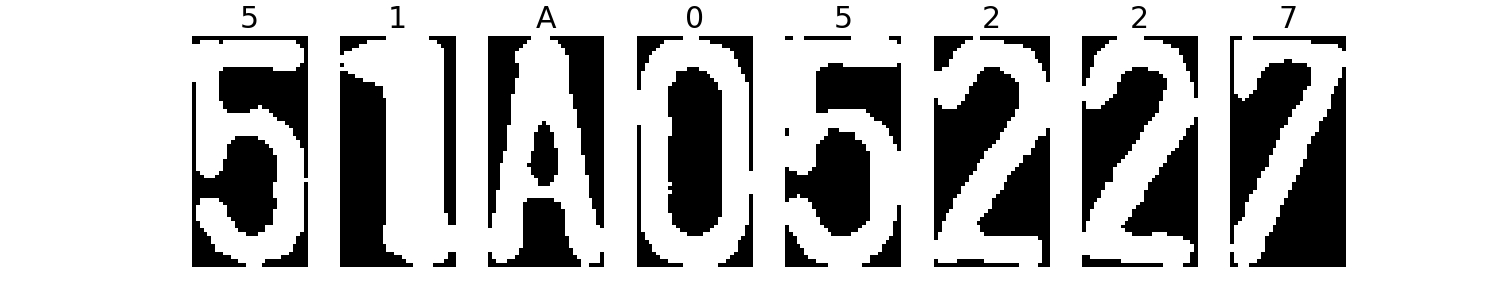




* Kết quả khi sử dụng model để nhận diện các ký tự đã được tách từ biển số:







* Bước cuối cùng là hiển thị các ký tự nhận dạng được lên ảnh. Đây là một số kết quả:









# **CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **Kết luận**

Máy móc dần thay thế con người trên nhiều phương diện giúp nâng cao tối đa chất lượng cuộc sống. Hệ thống tự động hóa quy trình quản lý phương tiện giao thông ở các nhà máy, xí nghiệp, siêu thị, ... dần thay thế công tác quản lý thủ công trước đây, mang lại nhiều lợi ích như tiết kiệm thời gian, nhân lực và nâng cao hiệu quả quản lý. Từ đó có thể nhận thấy việc phát triển bài toán nhận diện biển số xe mang tính cấp thiết nhằm đáp ứng nhu cầu quản lý hệ thống ngày càng cao hiện nay.

Bài toán này tuy không mới nhưng vẫn tồn tại nhiều khó khăn khi thực hiện đó là các thách thức như: có nhiều loại mẫu mã và xe khác nhau, hình ảnh thu được trong các điều kiện ánh sáng khác nhau và đôi khi không rõ nét.

Trải qua quá trình nghiên cứu và thực hiện khóa luận, chúng em đã đạt được một số kết quả sau:

1. Chúng em đã áp dụng được các phương pháp Deep Learning trên bài toán nhận diện biển số xe và chọn ra phương pháp phù hợp nhất.
2. Đánh giá được mô hình thực hiện thông qua các tiêu chí đánh giá khác nhau.
3. Xây dựng demo hiển thị kết quả thực hiện của bài toán.

## **2. Hướng phát triển**

Về tương lai, chúng em mong muốn có thể hoàn thiện tốt hơn đồ án này với một số vấn đề sau:

• Chuẩn bị dữ liệu thực tế các phương tiện giao thông ở Việt Nam để huấn luyện lại các mô hình học sâu cho nhánh Detection và Classification.

• Tìm hiểu thêm về một số mạng học sâu hỗ trợ cho huấn luyện các bài toán ít dữ liệu và với dữ liệu video.

• Xây dựng hệ thống nhận diện biển số xe hoàn thiện hơn, nâng cao hiệu suất của hệ thống nhằm hướng tới mục tiêu có thể áp dụng rộng rãi sau này.

# **TÀI LIỆU THAM KHẢO**

<https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173>

<https://pjreddie.com/darknet/yolo/>

<https://pjreddie.com/media/files/papers/YOLOv3.pdf>

<https://medium.com/@quangnhatnguyenle/detect-and-recognize-vehicles-license-plate-with-machine-learning-and-python-part-1-detection-795fda47e922>

<https://towardsdatascience.com/mobilenetv2-inverted-residuals-and-linear-bottlenecks-8a4362f4ffd5>

<https://arxiv.org/pdf/1801.04381.pdf>

<https://arxiv.org/pdf/1704.04861.pdf>

<https://www.researchgate.net/figure/Detailed-WPOD-NET-architecture_fig3_327861610>

<https://openaccess.thecvf.com/content_ECCV_2018/papers/Sergio_Silva_License_Plate_Detection_ECCV_2018_paper.pdf>

<https://arxiv.org/abs/1804.02767>

<https://pdfs.semanticscholar.org/5402/4d0572a84bc05544e76e94dfc7511e609ecb.pdf>