ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC

PHÂN LOẠI PHƯƠNG TIỆN GIAO THÔNG

*Môn học:* Nhập môn thị giác máy tính - CS231.O22

Giảng viên hướng dẫn: TS. Mai Tiến Dũng

Sinh viên thực hiện:

Đoàn Nhật Tiến - 22521463

Nguyễn Thượng Phúc - 22521134

Lê Dương Minh Thiên - 22521386

Phạm Đình Nhật - 22521023

Trần Hà Sơn - 22521259

*TP Hồ Chí Minh, ngày tháng năm 2024*

Mục lục

[I. Giới thiệu 3](#_3a0ermtxboi1)

[II. Bộ dữ liệu 3](#_ez10j2y4s4ic)

[III. Trích xuất đặc trưng 4](#_hs3b4skzs36b)

[1. Đặc trưng sử dụng cho các thuật toán Machine learning 4](#_26ei065gvpck)

[2. Đặc trưng sử dụng cho các mô hình Deep learning 4](#_s1djy7ncjbd)

[IV. Độ đo 4](#_l76nrg9cwnkm)

[V. Random Forest 5](#_askv5tpl8bds)

[1.Tổng quan: 5](#_k7g0ig5p6cz7)

[2. Nguyên lí hoạt động: 6](#_up9y6da4say1)

[3.Điều chỉnh tham số: 6](#_jkdzqx6ftff)

[4. Kết quả: 6](#_vnlah4ppq0ec)

[VI. KNN 8](#_2qvpwpa6ea91)

[1. Tổng quan 8](#_hgsq18l49bq)

[2. Nguyên lý hoạt động 8](#_bjeyxet10gco)

[3. Quá trình xử lí và huấn luyện 8](#_qmuq54wb4x66)

[4. Kết quả đánh giá 8](#_l6rj91qq7tmq)

[VII. SVM 8](#_s4gujfjsk6b)

[1. Tổng quan và các khái niệm 8](#_6n3j2a325q1e)

[2. Nguyên lý hoạt động và các tham số 9](#_e4pqcv2r4mmf)

[3. Kết quả 11](#_yck5oous74ww)

[VIII. ResNet 18 11](#_umq5rx84c8jt)

[1. Tổng quan 11](#_f8jdyrksnwn4)

[2. Điểm nổi bật 12](#_s4ic0di707m7)

[3. Kiến trúc của mô hình 12](#_92yy5ia3bvsi)

[4. Quá trình huấn luyện 13](#_t832gu6x8rts)

[5. Kết quả đánh giá 14](#_fuhl72b1ql1s)

[IX. GoogleNet 15](#_k6lzkz4ckecg)

[1.Tổng quan mô hình 15](#_2dciffy9myzv)

[2. Điểm nổi bật 15](#_nt4prh2cxk5m)

[3. Quá trình xử lí và huấn luyện 15](#_wr6zjqn0di22)

[4. Kết quả đánh giá trên tập test 16](#_r5vxn130iokz)

[X. Tổng kết 17](#_g816f5w7j1p2)

# 

# 

# 

# Giới thiệu

### **1. Lý do chọn đề tài**

- Trong thế kỷ 21, sự phát triển không ngừng của công nghệ đã thúc đẩy sự tiến bộ đáng kể trong nhiều lĩnh vực, trong đó có lĩnh vực giao thông và vận tải. Tại các thành phố lớn và khu vực đô thị, vấn đề về quản lý và kiểm soát phương tiện giao thông đang trở thành một thách thức ngày càng nghiêm trọng. Điều này phản ánh qua sự gia tăng về mật độ phương tiện và sự tăng cường của nhu cầu di chuyển trong cuộc sống hàng ngày. Những vấn đề này không chỉ ảnh hưởng đến việc đi lại mà còn gây ra các vấn đề liên quan đến môi trường, an toàn và cảm giác tiện lợi của người dân

### **2. Mô tả bài toán**

* - Phân loại và nhận diện phương tiện đã trở thành một phần không thể thiếu của Hệ thống Giao thông Thông minh (ITS). Mục tiêu của ITS là cung cấp an toàn công cộng, cải thiện thông tin di chuyển và vận chuyển, giảm tắc đường và tạo ra quỹ tiết kiệm chi phí cho các nhà vận hành trong các tình huống khẩn cấp.
* Hiệu suất của một hệ thống ITS thường dựa vào hiệu suất của công nghệ nhận diện và phân loại phương tiện.
* Trong quá trình phát triển của ITS, giám sát giao thông là một trong những ứng dụng quan trọng bao gồm phân loại và nhận diện các phương tiện.

Phân loại loại phương tiện có một loạt các ứng dụng bao gồm hệ thống đỗ xe thông minh, thống kê lưu lượng giao thông, hệ thống thu phí, kiểm soát truy cập phương tiện, quản lý tắc đường, hệ thống an ninh và nhiều ứng dụng khác

**Input**: tập ảnh các loại phương tiện giao thông trích xuất từ camera giao thông

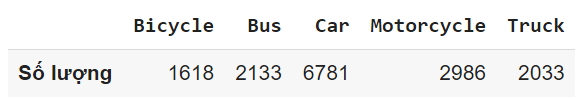
**Output**: tên loại phương tiện giao thông đó

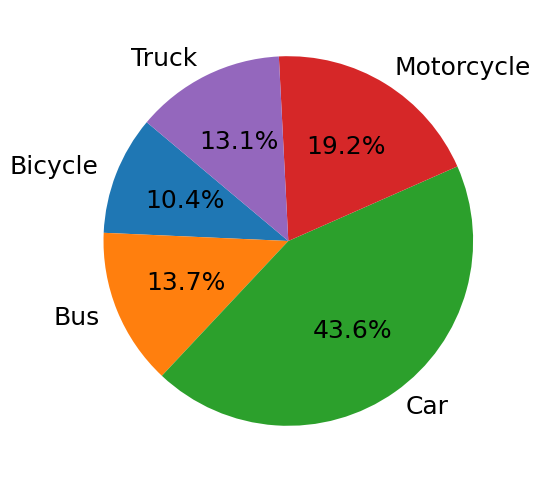
# Bộ dữ liệu

- Nhóm sử dụng bộ dữ liệu được đăng tải trên Kaggle.  
(<https://www.kaggle.com/datasets/iamsandeepprasad/vehicle-data-set>)

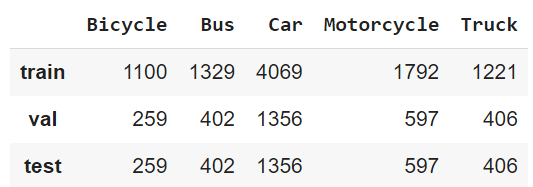
Bao gồm 15551 bức hình được trích xuất từ rất nhiều nguồn như camera an ninh, máy ảnh, điện thoại, …

- Bộ dữ liệu được chia làm 5 lớp: Car, Bus, Truck, Bicycle, Motocycle





* Số lượng ảnh của mỗi lớp được chia như trên hình. Dữ liệu của lớp Car lớn hơn đáng kể so với các lớp còn lại do trong bộ dữ liệu gốc được thiết kế sẵn để chuyên phân biệt Car. Cũng vì số lượng lớn hơn nhiều so với các lớp còn lại của Car này mà sẽ ảnh hưởng khá nhiều đến kết quả dự đoán của các mô hình sau này.
* Sau đó các lớp trên được chia thành các tập để huấn luyện với số lượng như sau:



Bộ dữ liệu được chia làm 3 tập train, val, test với tỉ lệ 6:2:2

# Trích xuất đặc trưng

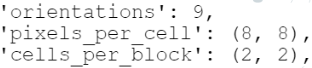
## Đặc trưng sử dụng cho các thuật toán Machine learning

-Chuyển ảnh về kích thước cố định 128x128

-Chuyển sang ảnh xám(grayscale)

-Trích xuất đặc trưng từ ảnh bằng thuật toán HOG

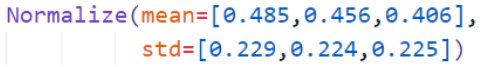
Các tham số của HOG:



## Đặc trưng sử dụng cho các mô hình Deep learning

* Chuyển ảnh về kích thước cố định: 224x224(ResNet-18); 299x299(Inception-v3)
* Chuẩn hóa ảnh theo các chỉ số trung bình và độ lệch chuẩn với công thức :

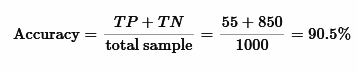
output[channel] = (input[channel] - mean[channel]) / std[channel]



* Các chỉ số trên được tham khảo theo bộ ImageNet

# IV. Độ đo

* Accuracy đơn giản là tỉ lệ các dự đoán chính xác của mô hình trên tất cả các mẫu



* Precision thể hiện tỷ lệ các dự đoán chính xác trong số tất cả các dự đoán mà mô hình đưa ra. Nói cách khác, Precision cho biết có bao nhiêu phần trăm dự đoán của mô hình là đúng.



* Recall đo lường tỉ lệ dự báo chính xác các trường hợp positive trên toàn bộ các mẫu thuộc nhóm positive.



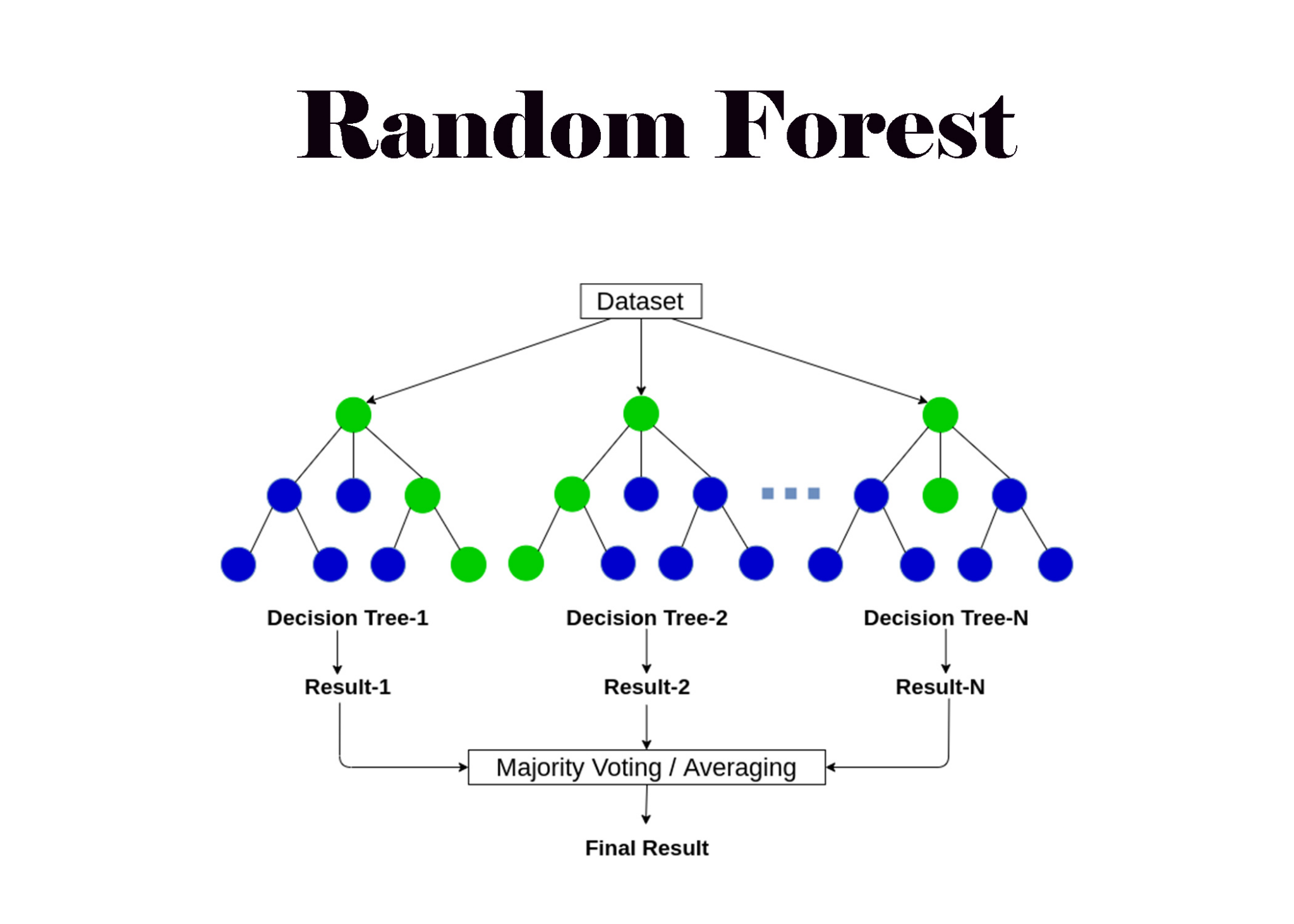
* F1-Score là trung bình điều hòa giữa precision và recall. Do đó nó đại diện hơn trong việc đánh giá độ chính xác trên đồng thời precision và recall



# V. Random Forest

## 1.Tổng quan:

Random forest là một kĩ thuật học máy có giám sát, thường được sử dụng cho các bài toán phân loại và hồi quy. Random forest là một tập hợp của nhiều cây quyết định(decision trees) hoạt động cùng nhau như một mô hình mạnh mẽ và chính xác hơn



## 2. Nguyên lí hoạt động:

Random forest hoạt động bằng cách xây dựng nhiều cây quyết định(Decision Tree) từ các tập con ngẫu nhiên của dữ liệu huấn luyện, sau đó kết hợp dự đoán của các cây này để tạo ra dự đoán cuối cùng. Mục tiêu là tăng sự linh hoạt và cải thiện độ chính xác của mô hình so với việc dùng 1 cây đơn lẻ.

## 3.Điều chỉnh tham số:

1. Tham số n\_estimators: số cây quyết định mà mô hình dựa vào để học và đưa ra kết quả. Thông thường khi tăng số lượng cây làm cải thiện độ chính xác của mô hình vì nó làm giảm phương sai và làm mô hình trở nên mạnh mẽ hơn.
2. Tham số max\_features: số đặc trưng để xem xét để tách tạch mỗi nút của cây quyết định. Khi tăng max\_features, mô hình có thể học chi tiết hơn từ dữ liệu nhưng cũng có nguy cơ bị overfitting. Ngược lại khi giảm max\_features, mô hình học ít chi tiết hơn từ dữ liệu nhưng cũng có nguy cơ bị underfitting.
3. Tham số max\_depth: độ sâu tối đa của cây. Khi tăng max\_features, mô hình có thể học chi tiết hơn từ dữ liệu nhưng cũng có nguy cơ bị overfitting. Ngược lại khi giảm max\_depth, mô hình học ít hơn từ dữ liệu nhưng cũng có nguy cơ bị underfitting

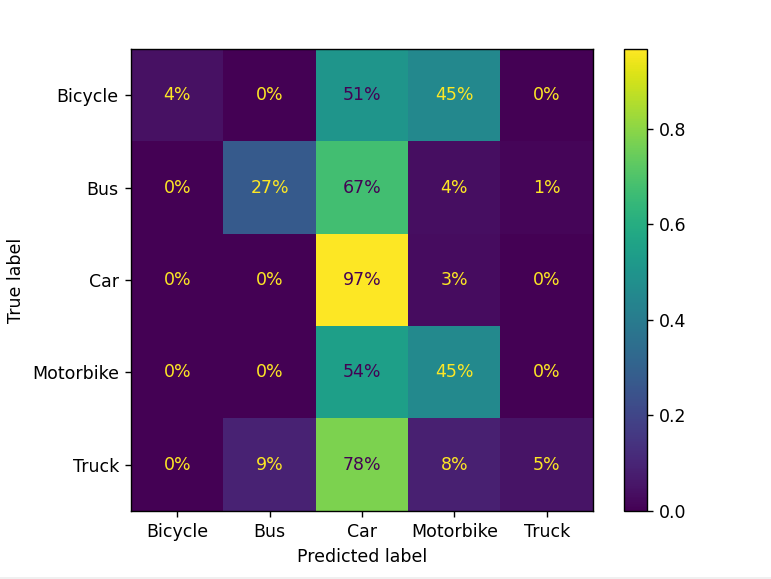
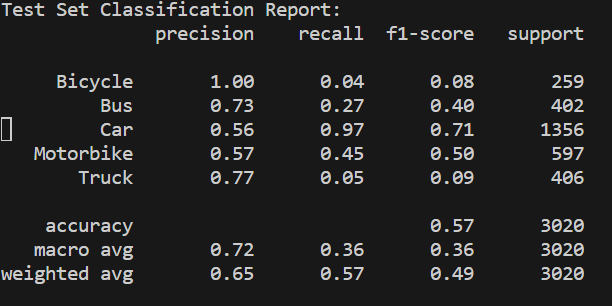
## 4. Kết quả:

Để tìm ra tham số tối ưu của mô hình em đã thử các số thông dụng như [100,200,300] đối với n\_estimators, [“sqrt”, “log2”] với max\_features, [5,8,10,15] với max\_depth. Vì em không dùng các phương pháp tối ưu hóa tham số GridSearchCV mà thế bằng tay nên thời gian làm khá lâu. Các tham số tốt nhất với mô hình lần lượt là:

- n\_estimators: 100 ( 100 cây quyết định)

- max\_features: sqrt( căn bậc hai của tổng các đặc trưng)

- max\_depth: 10( độ sâu tối đa của mỗi cây là 10 tầng)



# VI. KNN

## Tổng quan

KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.  
 Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

## Các bước thực hiện

1. Ta có D là tập các điểm dữ liệu đã được gắn nhãn và A là dữ liệu chưa được phân loại.

2. Đo khoảng cách (Euclidian, Manhattan, Minkowski, Minkowski hoặc Trọng số) từ dữ liệu mới A đến tất cả các dữ liệu khác đã được phân loại trong D.

3. Chọn K (K là tham số mà bạn định nghĩa) khoảng cách nhỏ nhất.

4. Kiểm tra danh sách các lớp có khoảng cách ngắn nhất và đếm số lượng của mỗi lớp xuất hiện.

5. Lấy đúng lớp (lớp xuất hiện nhiều lần nhất).

6. Lớp của dữ liệu mới là lớp mà bạn đã nhận được ở bước 5.

## Quá trình xử lí và huấn luyện

### 3.1. Đặc điểm mô hình

### **- Ưu điểm**

1. Thuật toán đơn giản, dễ dàng triển khai.

2. Độ phức tạp tính toán nhỏ.

3. Xử lý tốt với tập dữ liệu nhiễu

### **- Nhược điểm**

1. Với K nhỏ dễ gặp nhiễu dẫn tới kết quả đưa ra không chính xác

2. Cần nhiều thời gian để thực hiện do phải tính toán khoảng cách với tất cả các đối tượng trong tập dữ liệu.

3. Cần chuyển đổi kiểu dữ liệu thành các yếu tố định tính.

### 3.2. Tham số k

Vấn đề với lựa chọn K neighbor:

- Nếu K là số lẻ : Khi K là số lẻ khi thực hiện voting chúng ta luôn thu về kết quả

voting luôn tồn tại số voting cao hơn nên kết quả sẽ không khiến chúng ta gặp khó khăn

trong việc kết luận.

- Nếu K là số chẳn : Khi K là số chẳn khi thực hiện voting chúng ta sẽ có thể thu về

kết quả voting mà số voting cho mỗi phân loại đều bằng nhau. Lúc này ảnh hưởng đến

kết quả dự đoán không hiệu quả.

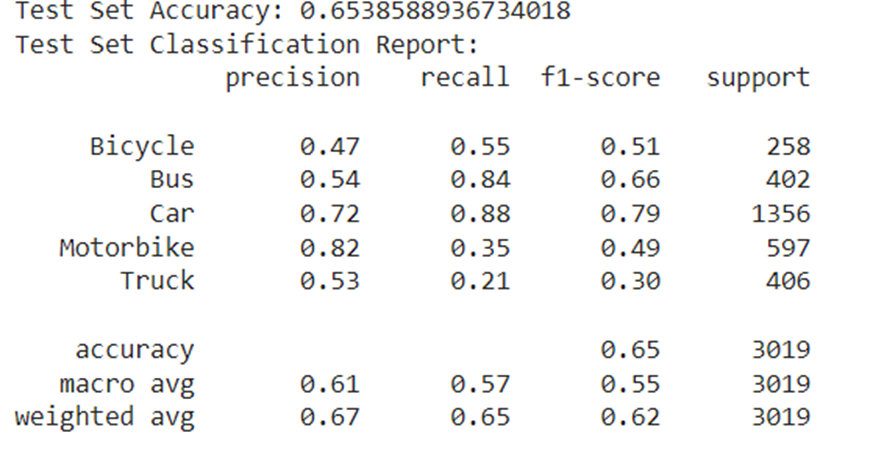
– Nếu k nhỏ, mô hình phức tạp hơn, sai số khớp trên mẫu xây dựng nhỏ hơn, dễ bị overfitting.

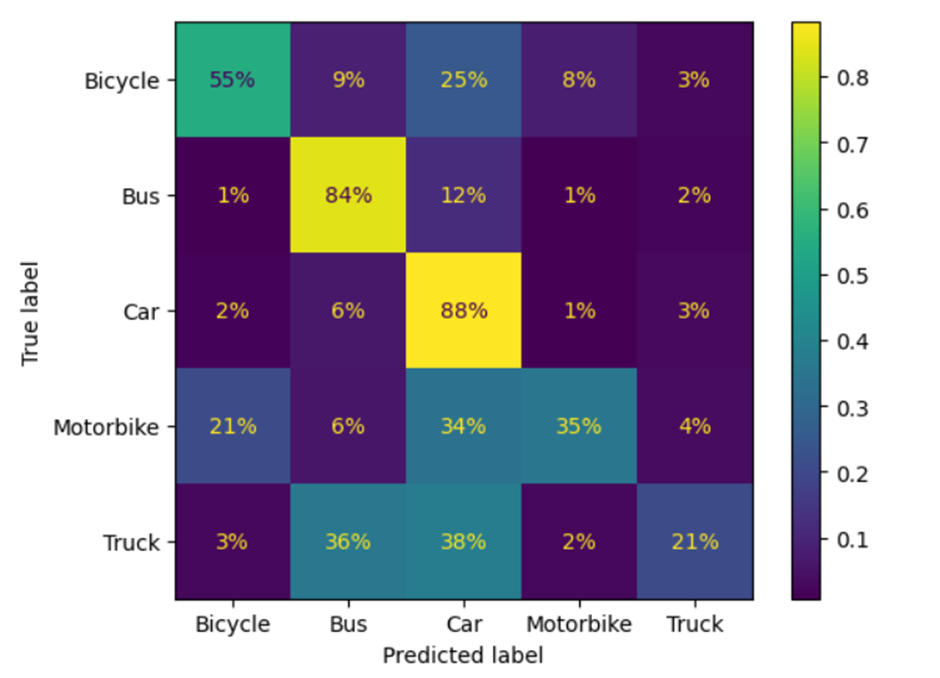
– Nếu k lớn, kết quả dự báo ổn định hơn, do có "sự bình chọn giữa nhiều quan sát.

### 3.3. Tìm tham số tối ưu



## Kết quả đánh giá





* Tỉ lệ accuracy trên tập test là 65%, trong đó mô hình phân loại chính xác cao trên lớp Car, Motorbike và Bus tuy nhiên vẫn phân loại chưa tốt trên lớp Motorbike và Truck.

# VII. SVM

## Tổng quan và các khái niệm

SVM (Support Vector Machine) là thuật toán học máy phổ biến được sử dụng cho các bài toán phân loại và hồi quy. Nó hoạt động bằng cách tìm kiếm một siêu phẳng (hyperplane) trong không gian dữ liệu có khả năng phân chia các điểm dữ liệu thuộc các lớp khác nhau một cách tối ưu nhất.

a. Mục tiêu:

Tìm kiếm siêu phẳng tối ưu giúp phân chia dữ liệu thành các lớp một cách chính xác nhất.

b. Siêu phẳng ( hyperplane)

Là khái quát hóa của đường thẳng trong không gian đa chiều.

Trong trường hợp đơn giản nhất (2 chiều), siêu phẳng sẽ là một đường thẳng.

Với dữ liệu có nhiều chiều hơn, siêu phẳng sẽ là một mặt phẳng đa chiều tương ứng.

c. Vectơ hỗ trợ ( support vector)

Là những điểm dữ liệu nằm gần siêu phẳng nhất, đóng vai trò quan trọng trong việc xác định vị trí của siêu phẳng.

d. Lề ( margin)

Khoảng cách giữa siêu phẳng và các vectơ hỗ trợ. Lề càng lớn, mô hình càng có khả năng phân loại chính xác cao.

e. Ưu điểm:

Hiệu quả cao trong các bài toán phân loại hai lớp và đa lớp.

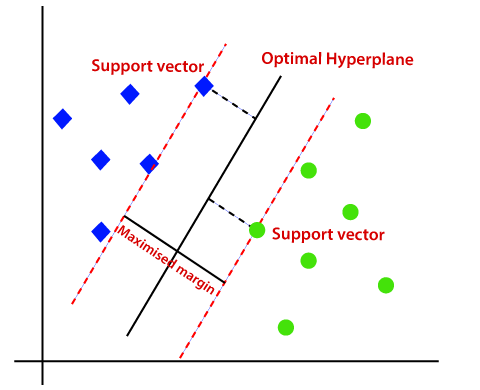
Ít nhạy cảm với nhiễu trong dữ liệu.

Có khả năng giải thích mô hình tốt.

f. Nhược điểm:

Có thể gặp khó khăn khi xử lý dữ liệu có nhiều chiều cao.

Tính toán có thể tốn thời gian cho các tập dữ liệu lớn.

* 

## Nguyên lý hoạt động và các tham số

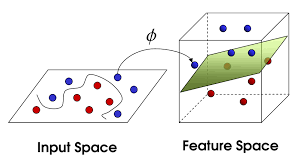
Nguyên lý hoạt động của SVM được mô tả qua các bước chính sau:

**Tìm siêu phẳng tối ưu**:

SVM tìm kiếm một siêu phẳng (hyperplane) tốt nhất để phân chia các điểm dữ liệu của hai lớp khác nhau trong không gian nhiều chiều. Siêu phẳng này được lựa chọn sao cho khoảng cách từ nó đến các điểm dữ liệu gần nhất của cả hai lớp (gọi là các vector hỗ trợ - support vectors) là lớn nhất. Khoảng cách này được gọi là biên (margin). Mục tiêu là tối ưu hóa biên này, để đảm bảo rằng các điểm dữ liệu được phân loại chính xác với khoảng cách lớn nhất có thể từ siêu phẳng.

**Xử lý dữ liệu không tuyến tính**:

Đối với các bài toán mà dữ liệu không thể được phân tách bằng một siêu phẳng trong không gian gốc, SVM sử dụng một thủ thuật gọi là kernel trick. Kernel trick là một phương pháp để ánh xạ dữ liệu từ không gian gốc (input space) sang một không gian cao hơn (feature space), trong đó dữ liệu có thể trở nên tuyến tính và có thể phân tách bởi một siêu phẳng.



**Hàm mục tiêu và điều kiện ràng buộc**:

SVM giải một bài toán tối ưu hóa với hàm mục tiêu là cực tiểu hóa hàm mất mát (loss function) và đồng thời tối đa hóa biên. Điều kiện ràng buộc đảm bảo rằng các điểm dữ liệu của các lớp khác nhau nằm ở hai phía của siêu phẳng tối ưu.

**Các tham số của mô hình SVM:**

* **C (Regularization parameter)**:

Tham số C quyết định mức độ chấp nhận lỗi phân loại trong quá trình tìm kiếm siêu phẳng tối ưu. Giá trị C lớn đặt trọng tâm vào việc phân loại đúng tất cả các điểm dữ liệu, ngay cả khi biên nhỏ. Ngược lại, giá trị C nhỏ cho phép mô hình có thể bỏ qua một số lỗi phân loại, nhưng tối đa hóa biên giữa các lớp.

* **Kernel**:

Các loại Kernel phổ biến:

1. Linear Kernel:
   1. Phù hợp cho dữ liệu có mối quan hệ tuyến tính đơn giản.
   2. Tính toán đơn giản, hiệu quả.
2. Polynomial Kernel:
   1. Dùng cho dữ liệu có mối quan hệ phi tuyến tính dạng đa thức.
   2. Có thể điều chỉnh tham số "d" để phù hợp với bài toán.
3. Radial Basis Function (RBF) Kernel:
   1. Hiệu quả cao với nhiều loại dữ liệu phi tuyến tính.
   2. Tham số "γ" ảnh hưởng đến độ rộng của "hạt nhân".
4. Sigmoid Kernel:
   1. Tương tự RBF Kernel nhưng với hiệu suất thấp hơn.
   2. Ít được sử dụng hơn so với các Kernel khác.

* **Gamma (γ):**

Tham số γ là tham số của RBF Kernel và Polynomial Kernel, quyết định ảnh hưởng của một điểm dữ liệu đơn lẻ. Giá trị γ lớn nghĩa là mỗi điểm dữ liệu sẽ có vùng ảnh hưởng nhỏ hơn, ngược lại, giá trị γ nhỏ sẽ tạo ra vùng ảnh hưởng lớn hơn cho mỗi điểm dữ liệu.

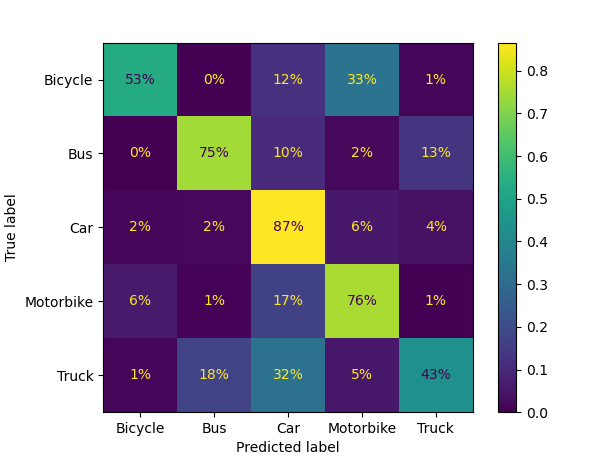
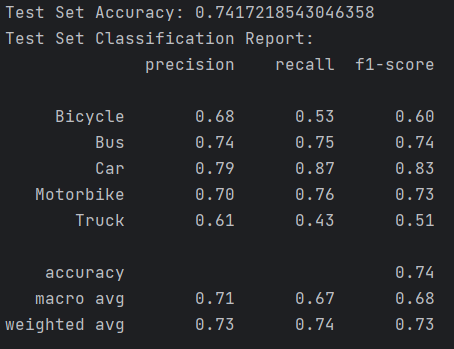
* **Degree (d)**:

Tham số d là bậc của Polynomial Kernel, quyết định mức độ của đa thức được sử dụng để ánh xạ dữ liệu sang không gian cao hơn.

* **Coef0**:

Tham số này xuất hiện trong Polynomial và Sigmoid Kernel, đại diện cho một giá trị không đổi được thêm vào trong tính toán kernel.

## Kết quả



* Tỉ lệ accuracy trên tập test khá ổn ( 0.74%), trong đó mô hình phân loại chính xác cao trên lớp Car, Motorbike và Bus tuy nhiên vẫn phân loại chưa tốt trên lớp Bicycle và Truck.

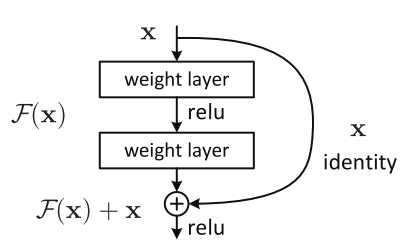
# VIII. ResNet 18

## Tổng quan

* Mạng ResNet-18 là phiên bản có số layer nhỏ nhất trong các biến thể của kiến trúc ResNet (Residual Network) được giới thiệu với công chúng vào năm 2015, bên cạnh đó còn có các mô hình với kích thước lớn hơn như ResNet-34, ResNet-52, ResNet-101 .

## Điểm nổi bật

* Điểm nổi bật của kiến trúc Resnet này chính là giải quyết được vấn đề Vanishing Gradient của các mạng học sâu có kích thước lớn
* Vanishing Gradient là hiện tượng mà khi mô hình huấn luyện và cần phải cập nhật các trọng số theo các giá trị gradient, có thể hiểu là đạo hàm của giá trị hàm mất mát so với từng trọng số tương ứng. Giá trị gradient lúc này sẽ có giá trị nhỏ dần khi đi đến các layer gần với input , khi đó giá trị gradient ở các layer này sẽ vô cùng nhỏ và trọng số cập nhật không đáng kể, dẫn đến mô hình học không được hiệu quả
* Sở dĩ mạng ResNet có thể giải quyết được vấn đề này là do có sử dụng một kỹ thuật “ Skip connection” hay là kết nối “ tắt ”.

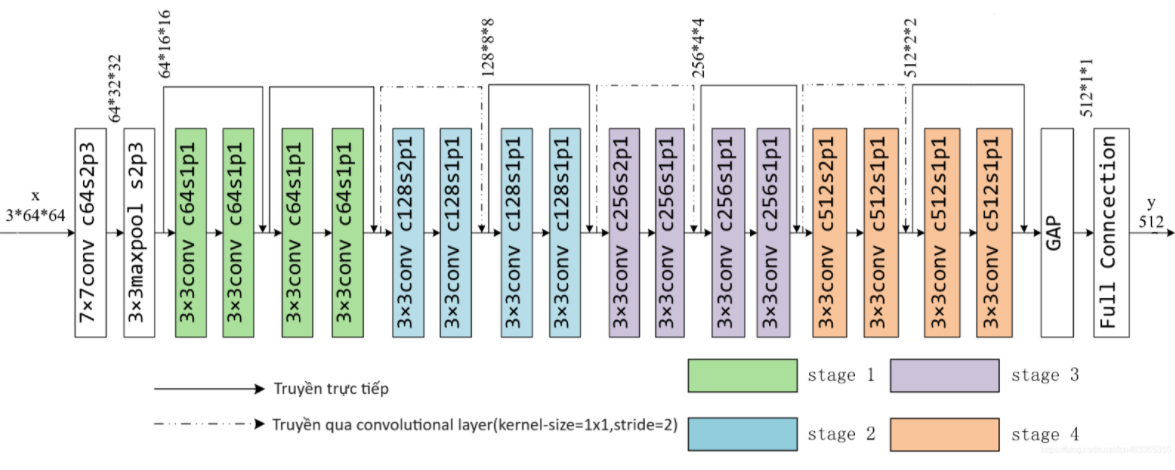


- Input x sau khi đi qua 2 layer, ta sẽ được output F(x), output này sẽ được cộng trở lại input x và đi đến các layer tiếp theo.

- Việc cộng thêm input x vào output sẽ chống lại việc đạo hàm bằng 0. Từ đó giảm thiểu vấn đề Vanishing Gradient.

## Kiến trúc của mô hình

* Các layer chính: 17 layers **convolution**, 1 layer **max pooling**, 1 layer **adaptive** **average pooling** và 1 layer **fully connected** ở cuối. Ngoài ra còn có các layers **batch normalization**.



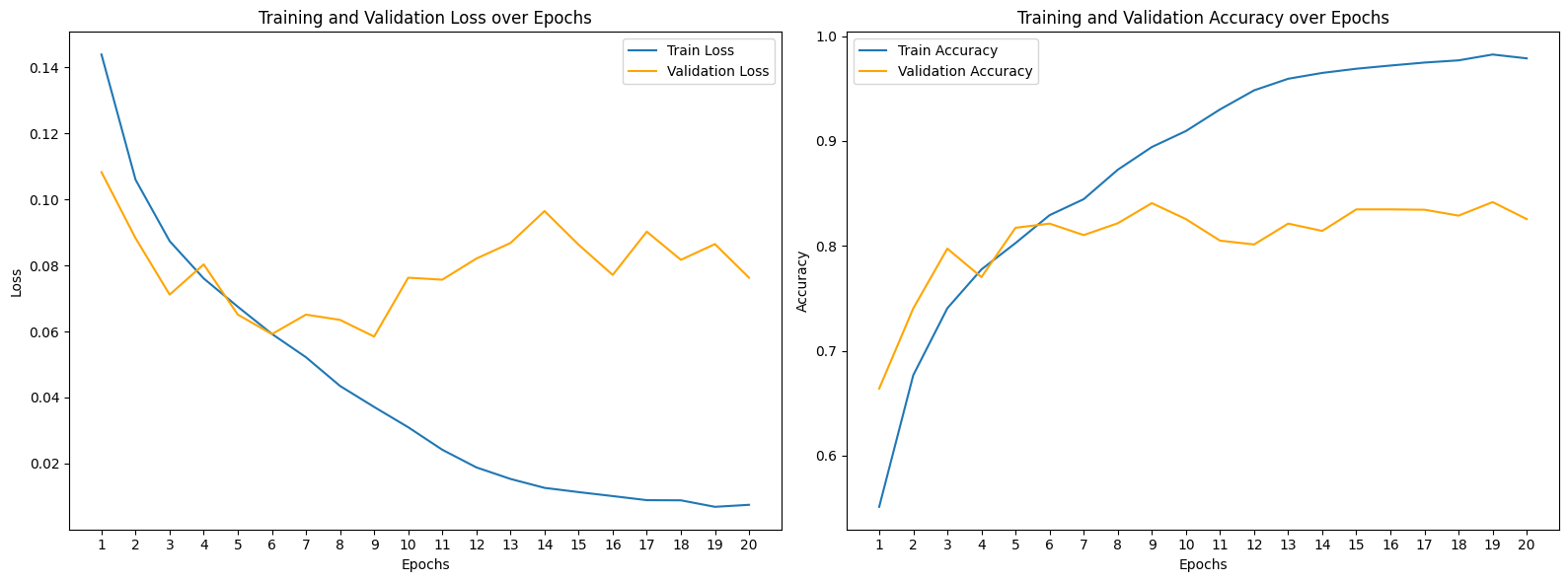
***c*** *: Số lượng kernel* ***MxN*** *: Kích thước kernel*

***s*** *: Giá trị stride* ***p*** *: Giá trị padding*

* Tất cả các phiên bản của mạng ResNet(từ ResNet-18 đến ResNet-152) đều có 2 layer **Convolution** và **Maxpool** ở đầu. Đối với mô hình ResNet-18, sau đó sẽ là 16 layer **Convolution** được chia thành 4 stage, với số lượng kernel gấp đôi dần từ 64 đến 512, các tham số cụ thể của từng layer như trên hình.
* Tiếp theo sẽ là layer **Adaptive** **average pooling,** layer này sẽ tổng hợp lại đặc trưng bằng cách lấy giá trị trung bình của các ma trận con trong ma trận đặc trưng, đầu ra của layer này sẽ là một vector để đưa đến layer **fully connected** ở cuối cùng và phân loại ảnh. Ví dụ như ta thấy trên hình, input của layer GAP có kích thước là 512x2x2, cứ với mỗi ma trận 2x2 ta sẽ lấy giá trị trung bình của ma trận => output sẽ là một vector với kích thước 512.
* Có 2 loại **Skip connection** được sử dụng trong mạng là truyền trực tiếp (ứng với đường nét liền) và truyền qua một convolutional layer với kernel-size có kích thước là 1x1 và stride bằng 2 (ứng với đường nét đứt). Đối với Skip connection truyền trực tiếp thì sẽ giống với ví dụ ở phần trước, output đi ra sẽ được cộng trực tiếp với input trước đó rồi tiếp tục đi qua các layer tiếp theo. Đối với Skip connection còn lại thì input trước đó sẽ được đưa qua một layer tích chấp để kích thước giảm đi còn một nửa rồi mới cộng vào output.

## Quá trình huấn luyện

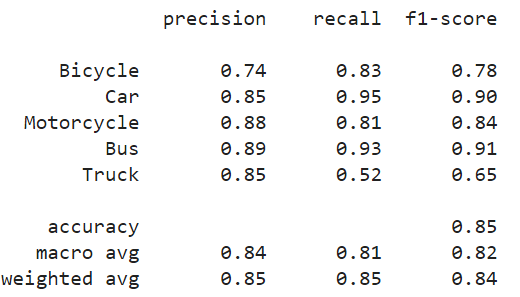
* Trước khi bắt đầu huấn luyện, ta cần phải thay thế layer đầu ra của mô hình để phù hợp với yêu cầu bài toán ( phân loại 5 layer )
* Hàm tối ưu hóa được sử dụng là: Adam với tham số learning\_rate = 0.0003
* Hàm mất mát được sử dụng là: Cross-Entropy
* Cả 2 hàm tối ưu hóa và hàm mất mát đều được sử dụng từ thư viện Pytorch với các tham số còn lại để mặc định.

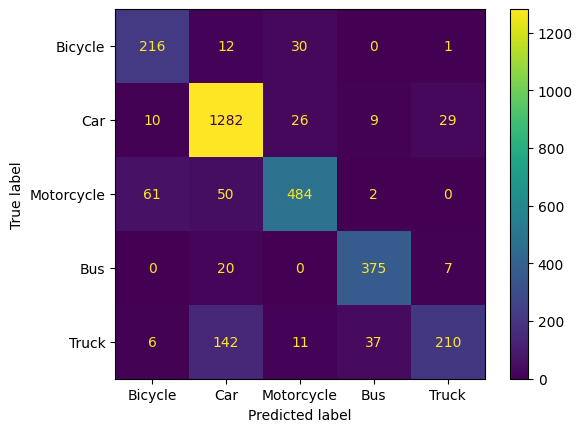


* Dựa vào đồ thị hàm mất mát của mô hình khi huấn luyện qua từng epoch trên hai tập train và validation, ta thấy ở khoảng 6 epochs đầu tiên, kết quả độ lỗi và accuracy được cải thiện khá nhanh và hiệu quả. Tuy nhiên ở những epoch sau, mô hình đã có dấu hiệu bị overfitting khi mà chỉ có kết quả trên tập train là tiếp tục được cải thiện còn tập validation thì lại có kết quả không được khả quan và ổn định.

## Kết quả đánh giá

* Em sẽ sử dụng bộ trọng số ứng với độ lỗi thấp nhất trên tập validation để đánh giá trên tập test
* Sử dụng hàm classification\_report và confusion\_matrix từ thư viện sklearn để đánh giá:





* Với accuracy bằng 85% và giá trị trung bình của precision,recall,f1 nằm trong khoảng từ 81 đến 85%, mô hình dự đoán khá tốt so với các phương pháp còn lại được sử dụng cho đề tài. Tuy nhiên mô hình vẫn còn hay bị nhầm lẫn giữa 2 class Car và Truck, có lẽ một phần bởi vì 2 phương tiện này khá giống nhau.

# IX. GoogleNet

## 1.Tổng quan mô hình

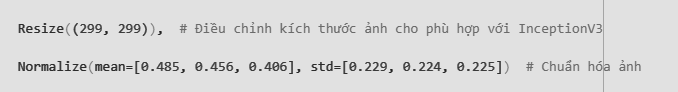
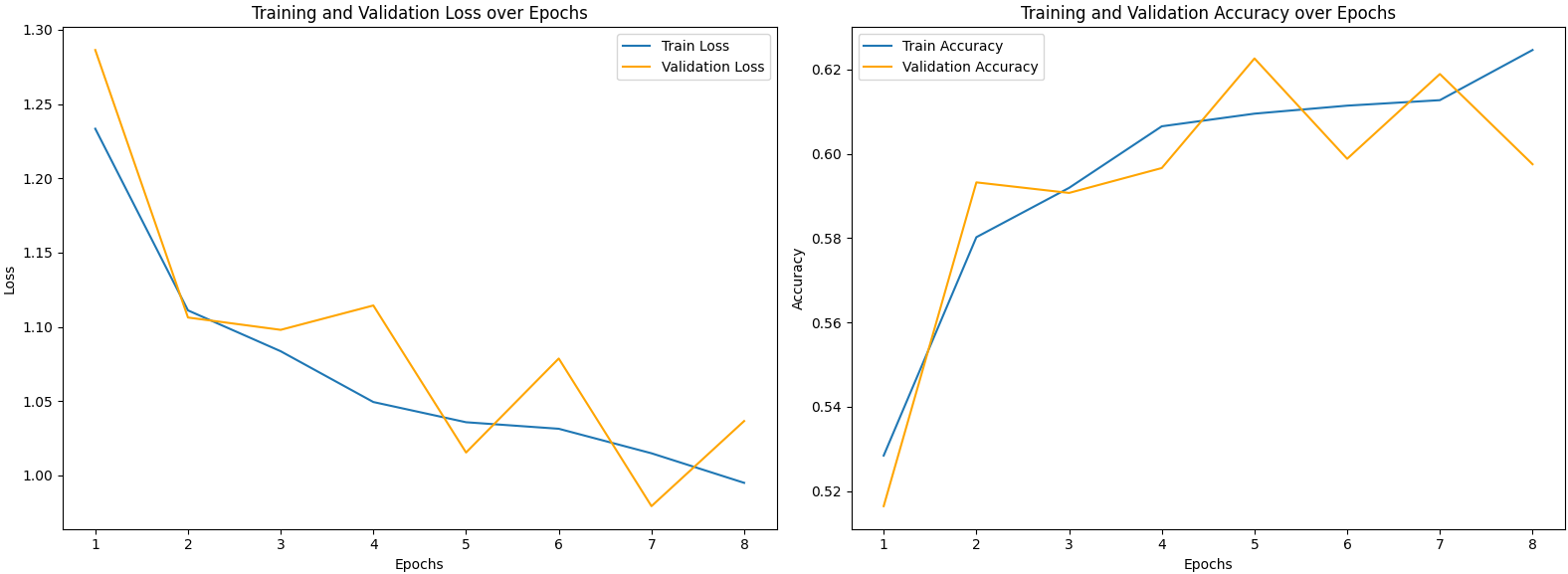
***Inception*** là một *kiến trúc mạng CNN (Convolutional Neural Network)*  được phát triển bởi nhóm nghiên cứu của Google trong dự án Inception. Được ra mắt vào năm 2015. Kiến trúc này đã giải quyết một câu hỏi lớn trong mạng CNN đó là sử dụng *kernel\_size* với kích thước bao nhiêu thì hợp lý.

## 2. Điểm nổi bật

***Inception-v3*** là một kế thừa và nâng cấp của Inception V1,có cấu tạo đặc trưng là sử dụng các *Inception Module*, là một loạt các mô-đun mạng phức tạp, mỗi mô-đun này kết hợp các hoạt động tích chập ở nhiều tỷ lệ và kích thước khác nhau để đạt được *hiệu suất tốt hơn*   
 Khối Inception sẽ bao gồm 4 nhánh song song. Các bộ lọc kích thước lần lượt là 1x1, 3x3, 5x5 được áp dụng trong Inception Module giúp trích lọc được đa dạng đặc trưng trên những vùng nhận thức có kích thước khác nhau

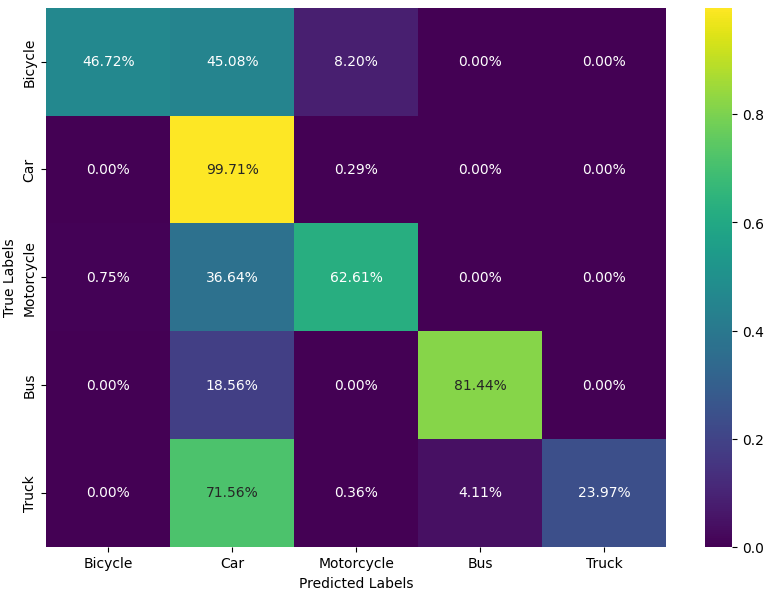
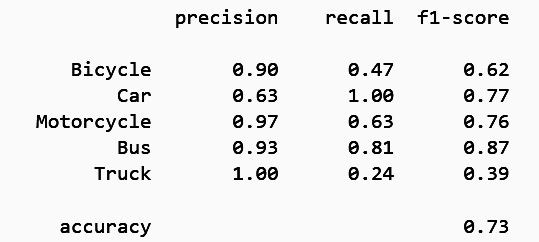
## 3. Quá trình xử lí và huấn luyện

+ *Chuẩn hóa ảnh*

  
299,299 là kích thước mặc định của inceptionv3 trong imagenet nên sử dụng kích thước này để thư viện có thể xử lí tối ưu nhất .Các tham số trên cũng là những tham số mặc định được tham khảo từ thư viện imagenet, điều chỉnh ảnh về mức trung bình  
+ *Quá trình huấn luận mô hình* ******

Về chỉ số của tập train từ biểu đồ ta thấy mô hình có khả năng học nhanh khi mức tăng của accuracy và độ giảm của loss là lớn ở những epoch đầu, tuy sau có tăng chậm những vẫn cho kết quả khá tốt. Về chỉ số của tập validation tuy có chưa ổn định nhiều nhưng vẫn có chiều hướng biến động tốt khi độ đo loss giảm dần và accuracy có tăng dần.  
 Đồ thị vẫn chưa thể hội tụ vì yêu cầu tài nguyên sử dụng cao nên em chỉ mới có thể chạy số lượng epochs hạn chế.

## 4. Kết quả đánh giá trên tập test



Với chỉ số Accuracy cho ra từ mô hình là 73%, mô hình có tỉ lệ dự đoán chính xác khá tốt với khả năng dự đoán tốt nhất ở lớp car với chỉ số recall gần bằng 100%, lý giải cho tỉ lệ xem chừng vô lí này là tập dữ liệu của Car lớn hơn rất nhiều so với các lớp còn lại, tập Car được huấn luyện kĩ và có một phần nhỏ bị dự đoán nhầm nhưng số lượng không đáng kể do bộ dữ liệu lớn.

# 

# 

# 

# 

# 

# 

# X. Tổng kết

| Phương pháp | Ưu điểm | Hạn chế |
| --- | --- | --- |
| Random forest | -Độ ổn định cao | -Kết quả dự đoán không tốt bằng các phương pháp deep learning |
| KNN |  |  |
| SVM | - Kết quả dự đoán tốt. | - Tính toán có thể tốn thời gian cho các tập dữ liệu lớn. |
| ResNet 18 | - Kết quả dự đoán tốt. | - Thời gian huấn luyện lâu hơn các thuật toán Machine learning truyền thống, tốn kém tài nguyên. |
| GoogleNet |  |  |