

# FINAL PROJECT

## Abstract

Dự án này nghiên cứu bài toán phân loại ảnh bằng cách sử dụng hai mô hình học sâu : MLP(Multi-Layer Perceptron) và CNNs (Convolutional Neural Networks).Bộ dữ liệu được sử dụng là bộ dữ liệu CIFAR-10,bao gồm dữ 10 lớp ảnh khác nhau, từ động vật đến phương tiện giao thông.

Chúng tôi đã xây dựng và huấn luyện các mô hình MLP và CNNs để phân loại các ảnh vào các lớp tương ứng. Các mô hình được đánh giá thông qua độ chính xác và độ mất mát (loss).

Kết quả thí nghiệm cho thấy mô hình CNNs đạt hiệu suất vượt trội so với mô hình MLP.Ngoài ra việc tinh chỉnh các siêu tham số như kích thước batch,tốc độ học và số lượng lớp của mô hình cũng ảnh hưởng đến hiệu suất của các mô hình.

Các kết quả và phân tích cung cấp cái nhìn sâu sắc về các đặc điểm của từng mô hình,đồng thời đề xuất các hướng phát triển trong việc cải thiện hiệu suất phân loại ảnh.

## Introduction

Phân loại ảnh là một trong những bài toán cơ bản và quan trọng trong lĩnh vực thị giác máy tính có thể ứng dụng trong nhiều lĩnh vực.Trong đó ,các mô hình học sâu, đặc biệt là MLP(Multi-Layer Perceptron) và CNNs(Convolutional Neural Networks),có hiệu quả cao trong nhiệm vụ phân loại ảnh.

MLP là một mô hình mạng nơ-ron truyền thống với cấu trúc mạng đa lớp, phù hợp với các bài toán dữ liệu có cấu trúc đơn giản. Tuy nhiên, đối với các dữ liệu ảnh có đặc trưng không gian phức tạp, MLP có thể không khai thác hết được các thông tin trong ảnh. Trong khi đó, CNNs được thiết kế đặc biệt để xử lý dữ liệu hình ảnh, với các lớp tích chập giúp phát hiện và học các đặc trưng không gian của ảnh một cách hiệu quả.

Trong bài toán phân loại ảnh, dự án này sử dụng bộ dữ liệu CIFAR-10, một bộ dữ liệu gồm 60,000 ảnh trong 10 lớp khác nhau, bao gồm các đối tượng từ động vật đến phương tiện giao thông. Mục tiêu của dự án này là xây dựng và đánh giá hiệu quả của hai mô hình học sâu: MLP và CNNs. Chúng tôi sẽ phân tích và so sánh hiệu suất của hai mô hình này trong việc phân loại ảnh, thông qua các chỉ số như độ chính xác, độ mất mát (loss), và tác động của việc điều chỉnh các siêu tham số đến hiệu suất của mô hình.

Báo cáo này được tổ chức như sau: phần tiếp theo sẽ giới thiệu về các mô hình và phương pháp huấn luyện được áp dụng. Sau đó, chúng tôi sẽ trình bày kết quả thí nghiệm và phân tích so sánh giữa MLP và CNNs. Cuối cùng, báo cáo sẽ đưa ra các nhận xét và hướng phát triển trong tương lai cho bài toán phân loại ảnh.

## MODELS

## 1. Kiến trúc MLP(Multi-layer Perceptron):

Trong bài toán phân loại ảnh CIFAR-10, mô hình MLP được thiết kế với các lớp fully connected để thực hiện phân loại các ảnh có kích thước  $32 \times 32 \times 3$  (3 kênh màu: Red, Green, Blue).

**Lớp đầu vào:** Lớp đầu vào nhận ảnh có kích thước , được làm phẳng thành một vector dài . Sau đó, ảnh được chuẩn hóa trước khi đưa vào mô hình.

**Lớp ẩn (Hidden Layer):** MLP có một hoặc nhiều lớp ẩn, mỗi lớp có 512 đơn vị (neurons) và sử dụng hàm kích hoạt ReLU. ReLU giúp mô hình học các mối quan hệ phi tuyến giữa các đặc trưng ảnh và các lớp phân loại.

**Lớp Dropout:** Để giảm thiểu overfitting, tôi sử dụng lớp Dropout. Lớp này ngẫu nhiên bỏ qua một số neurons trong quá trình huấn luyện, giúp cải thiện khả năng tổng quát hóa của mô hình.

**Lớp đầu ra:** Lớp này có 10 đơn vị, mỗi đơn vị tương ứng với một nhãn của lớp trong bộ dữ liệu CIFAR-10 (bao gồm 10 lớp: máy bay, ô tô, chim, mèo, hươu cao cổ, chó, ngựa, thuyền, xe tải, con ếch). Softmax được sử dụng để chuẩn hóa đầu ra thành xác suất cho mỗi lớp.

**Tối ưu hóa và loss function:** Tôi sử dụng Adam,SGD() làm thuật toán tối ưu với các learning rate 0.001 và 0.01. Loss function là CrossEntropyLoss, phù hợp với bài toán phân loại đa lớp.

## 2. Kiến trúc CNN(Convolutional Neural Network)

CNN là một trong những mô hình mạnh mẽ và hiệu quả trong bài toán phân loại ảnh, nhờ vào khả năng tự động học các đặc trưng quan trọng từ ảnh mà không cần phải thiết kế thủ công. Kiến trúc CNN của tôi cho bộ dữ liệu CIFAR-10 bao gồm các lớp convolutional, pooling và fully connected.

**Lớp Convolutional (Conv1):**Lớp này sử dụng 32 bộ lọc (filters), mỗi bộ lọc có kích thước  $5 \times 5$ . Lớp này giúp mô hình học các đặc trưng cơ bản của ảnh như các cạnh và đường viền.

**Lớp Pooling (Max Pooling):**Sau lớp convolutional đầu tiên, tôi sử dụng Max Pooling với kích thước  $2 \times 2$  và stride 2. Mục đích là giảm kích thước không gian của ảnh, giữ lại các đặc trưng quan trọng và giảm thiểu tải tính toán cho mô hình.

**Lớp Convolutional (Conv2):**Lớp này sử dụng 64 bộ lọc với kích thước  $5 \times 5$ . Lớp này học các đặc trưng phức tạp hơn từ ảnh, ví dụ như các mẫu chi tiết hơn.

**Lớp Pooling (Max Pooling):** Lớp pooling thứ hai tiếp tục giảm kích thước không gian của ảnh sau lớp convolutional thứ hai, giúp tăng cường sự trừu tượng hóa của các đặc trưng.

**Lớp Fully Connected (FC1):** Sau các lớp convolutional và pooling, tôi sử dụng một lớp fully connected với 120 neurons và hàm kích hoạt ReLU. Lớp này giúp mô hình kết nối các đặc trưng đã học từ các lớp trước và tạo ra các thông tin quan trọng hơn.

**Lớp Fully Connected (FC2):** Lớp fully connected thứ hai có 84 neurons và sử dụng hàm kích hoạt ReLU. Đây là lớp trung gian giúp mô hình tạo ra các kết quả tinh chỉnh hơn.

**Lớp Dropout:** Để giảm thiểu overfitting, tôi sử dụng lớp Dropout với tỷ lệ 0.5. Lớp này ngẫu nhiên bỏ qua một số neurons trong quá trình huấn luyện, giúp cải thiện khả năng tổng quát hóa của mô hình.

**Lớp Fully Connected (FC3):** Cuối cùng, lớp đầu ra có 10 đơn vị, mỗi đơn vị tương ứng với một nhãn trong bộ dữ liệu CIFAR-10. Lớp này không sử dụng hàm kích hoạt ReLU mà trực tiếp đưa ra đầu ra.

**Tối ưu hóa và loss function:** Tôi sử dụng Adam,SGD() làm thuật toán tối ưu với các learning rate 0.001 và 0.01. Loss function là CrossEntropyLoss, phù hợp với bài toán phân loại đa lớp.

## Experiments

### 1. Bộ dữ liệu CIFAR-10:

**CIFAR-10** (Canadian Institute for Advanced Research 10) là một bộ dữ liệu được sử dụng phổ biến trong nghiên cứu và thực nghiệm phân loại ảnh. Bộ dữ liệu này chứa 60,000 ảnh màu với kích thước 32x32 pixel và có 3 kênh màu(RGB), được phân thành 10 lớp khác nhau. Mỗi lớp gồm 6,000 ảnh, và tổng cộng có 10 lớp như sau: Airplane (Máy bay), Automobile (Ô tô), Bird (Chim), Cat (Mèo), Deer (Hươu cao cổ), Dog (Chó), Frog (Ếch), Horse (Ngựa), Ship (Tàu thuyền) và Truck (Xe tải).

### Thông kê dữ liệu

**Số lượng ảnh:** Bộ dữ liệu CIFAR-10 có tổng cộng 60,000 ảnh, chia thành: 50,000 ảnh huấn luyện và 10,000 ảnh kiểm tra.

**Số lượng lớp:** CIFAR-10 gồm 10 lớp khác nhau, mỗi lớp chứa 6,000 ảnh

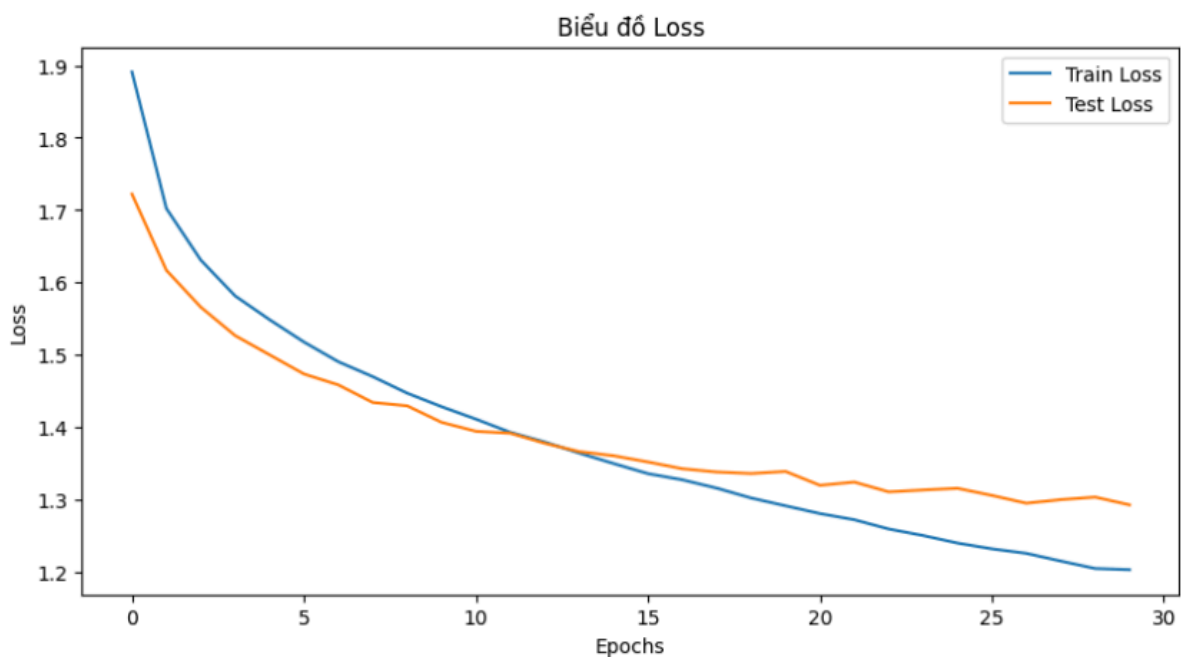
**Phân bổ ảnh theo lớp:** Mỗi lớp trong CIFAR-10 có số lượng ảnh huấn luyện và kiểm tra như sau: **Số lượng ảnh huấn luyện:** 5,000 ảnh cho mỗi lớp và **Số lượng ảnh kiểm tra:** 1,000 ảnh cho mỗi lớp.

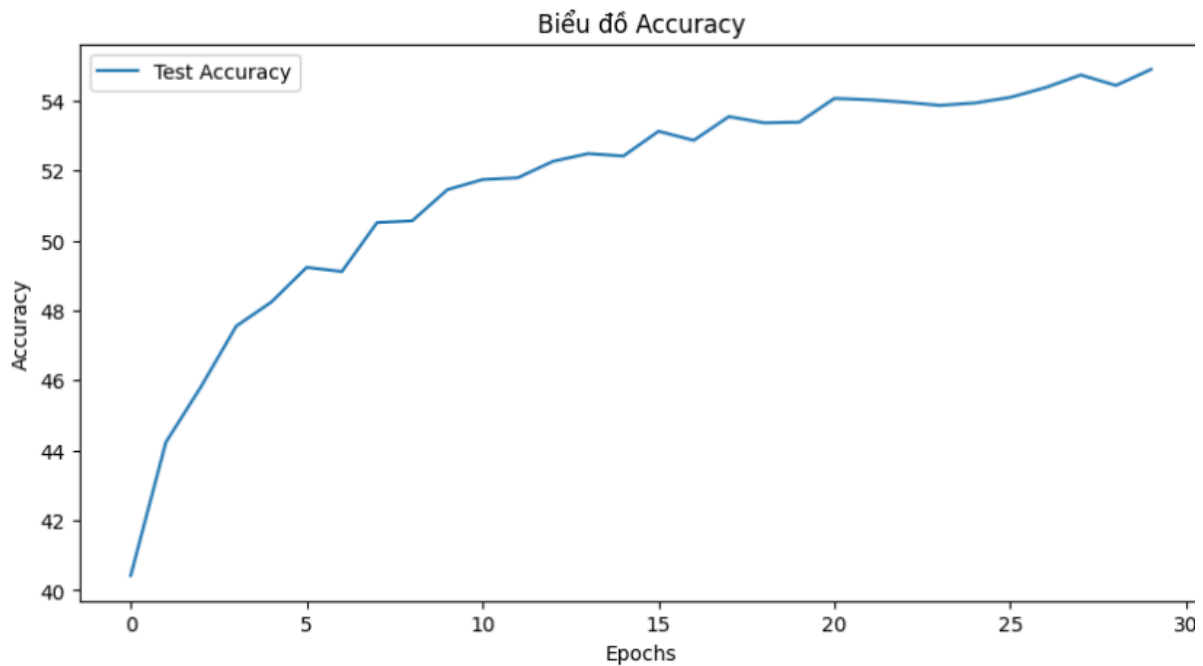
## 2. Kết quả chính:

**Kết quả tốt nhất của mô hình:**

**MLP một lớp ẩn(MLP1):** Mô hình tốt nhất có siêu tham số: Optimizer:SGD, Batch size: 64, Learning rate: 0.01.

Biểu đồ các chỉ số loss, test loss và accuracy sau mỗi epoch





Kết quả cuối cùng của mô hình : Train Loss: 1.2037, Test Loss: 1.2932, Accuracy: 54.89%

Đánh giá:

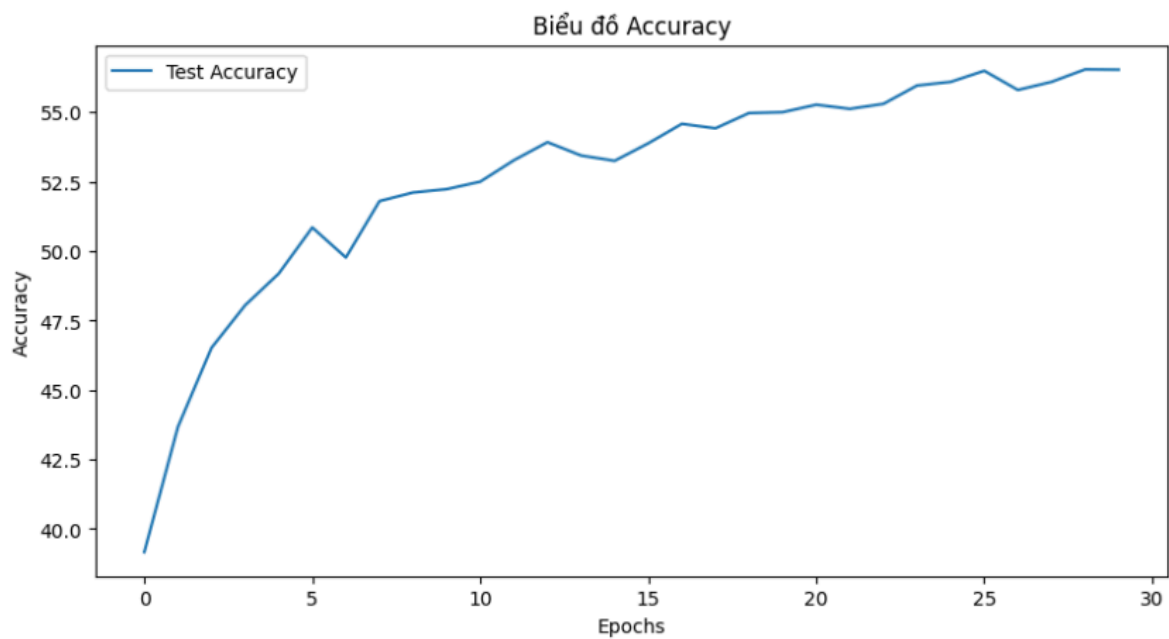
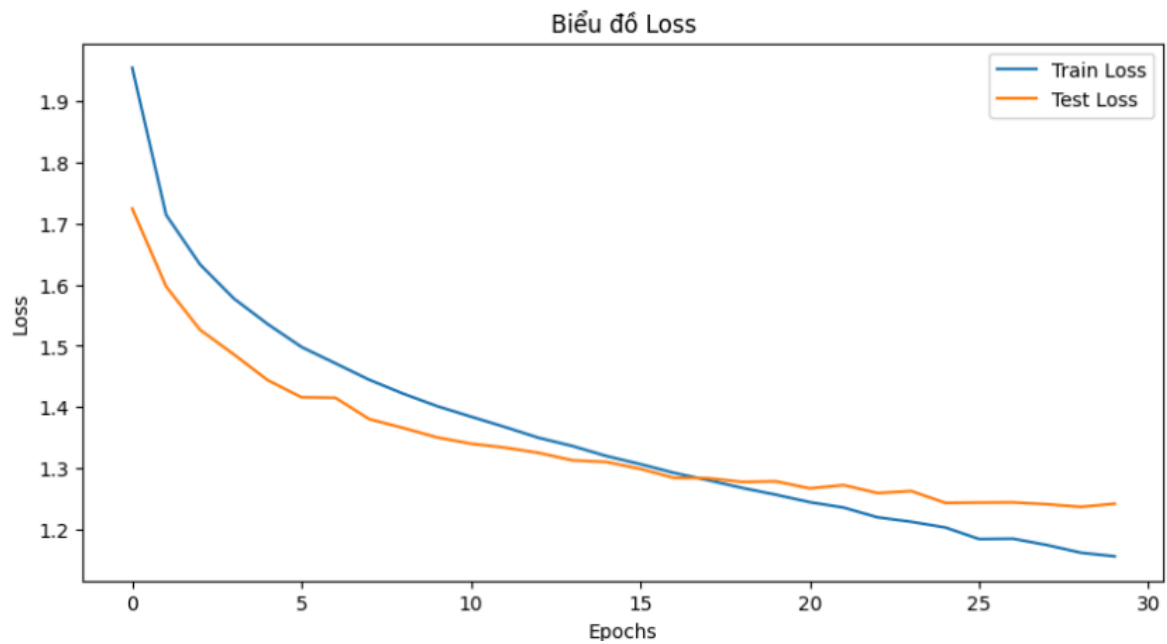
+) Mô hình hoạt động ổn định, không bị overfitting quá rõ ràng.

+) Train Loss (đường xanh) và Test Loss (đường cam) đều giảm dần qua các epoch, cho thấy mô hình đang học tốt. Sau khoảng 20-25 epoch, cả hai đường Loss đều có xu hướng hội tụ, cho thấy việc huấn luyện đã dần ổn định. Khoảng cách giữa Train Loss và Test Loss không quá lớn, nên hiện tượng overfitting không quá nghiêm trọng.

+) Độ chính xác tăng nhanh trong các epoch đầu tiên và có xu hướng ổn định từ epoch 20 trở đi và đạt khoảng 54% sau 30 epoch, điều này cho thấy mô hình đã học được các đặc trưng của dữ liệu.

**MLP hai lớp ẩn(MLP2):** Mô hình tốt nhất có siêu tham số: Optimizer: SGD, learning rate = 0.01, batch size = 32

Biểu đồ các chỉ số loss, test loss và accuracy sau mỗi epoch



Kết quả cuối cùng của mô hình này: Train Loss: 1.1557, Test Loss: 1.2417, Accuracy: 56.53%

Đánh giá:

+ ) Mô hình hoạt động ổn định, không bị overfitting quá rõ ràng.

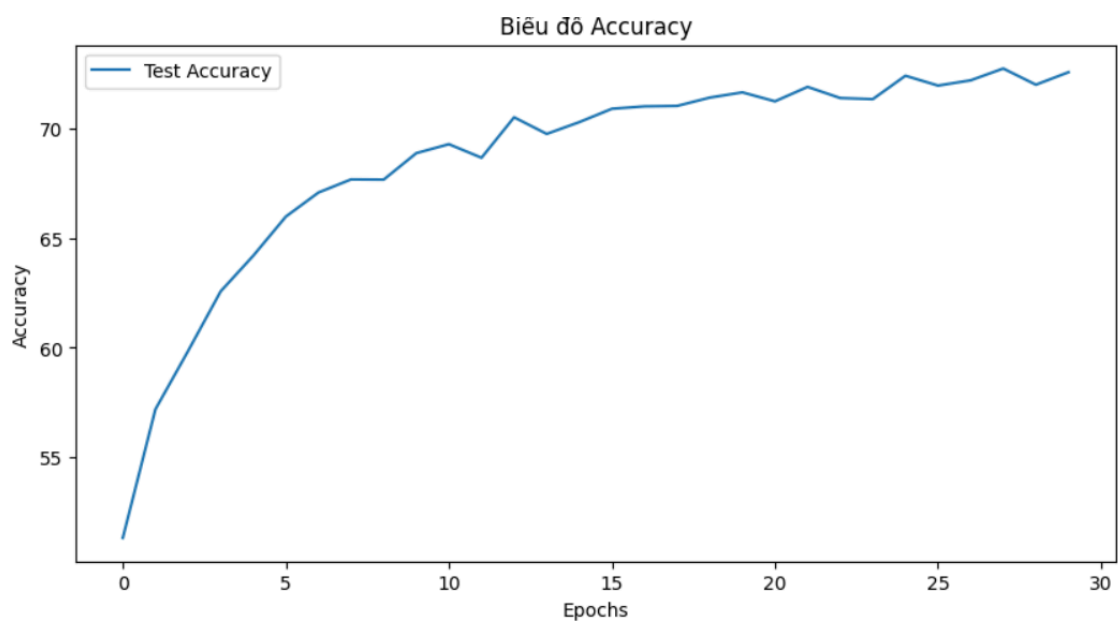
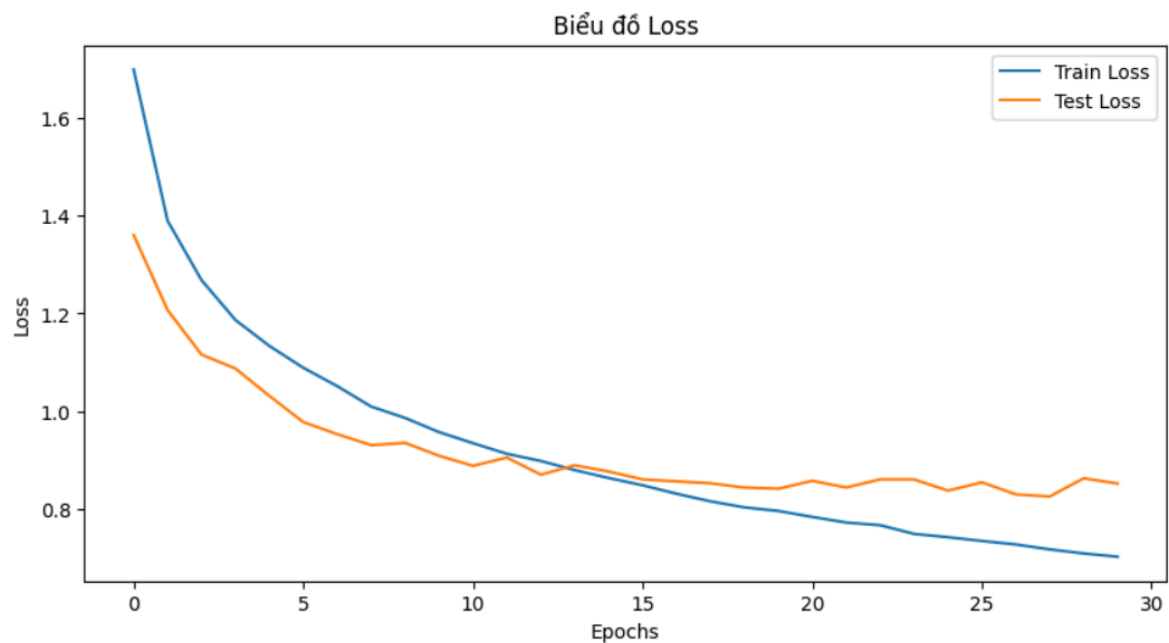
+ ) Train Loss (đường xanh) và Test Loss (đường cam) đều giảm dần qua các epoch, cho thấy mô hình đang học tốt. Sau khoảng 15-20 epoch, cả 2 đường đều có xu hướng hội tụ, cho thấy việc huấn luyện đã dần ổn định. Khoảng cách giữa

Train Loss và Test Loss không quá lớn, nên hiện tượng overfitting không quá nghiêm trọng.

+)Độ chính xác tăng nhanh trong các epoch đầu tiên và dần giảm tốc độ tăng,đạt khoảng 56% sau 30 epoch, điều này cho thấy mô hình đã học được các đặc trưng của dữ liệu.

**CNN:**Mô hình tốt nhất có siêu tham số :Optimizer:Adam ,Batch size: 32, Learning rate:0.001

Biểu đồ các chỉ số loss, test loss và accuracy sau mỗi epoch



Kết quả cuối cùng của mô hình này: Train Loss: 0.7015, Test Loss: 0.8513, Accuracy: 72.59%

Đánh giá:

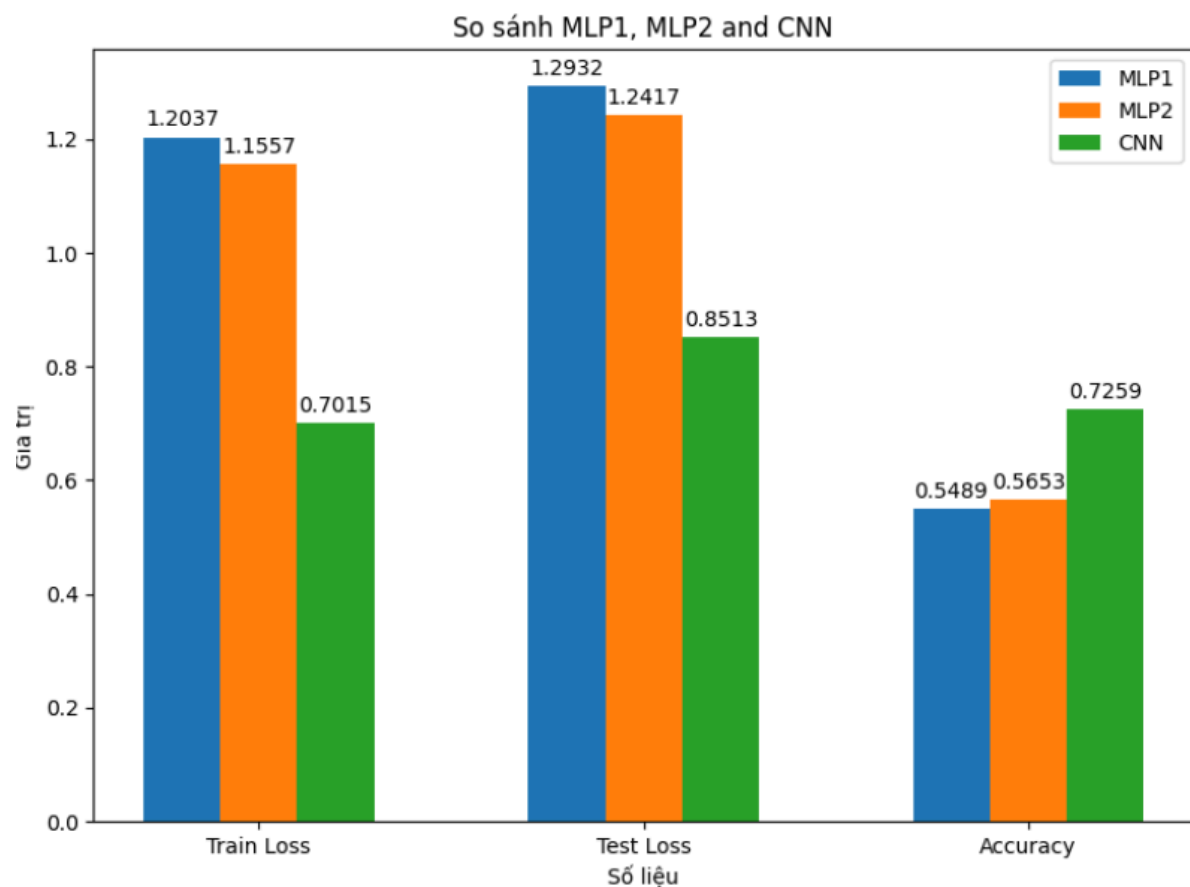
+) Mô hình đang hoạt động ổn định.

+) Train Loss (đường xanh) và Test Loss (đường cam) đều giảm dần qua các epoch, cho thấy mô hình đang học tốt. Sau khoảng 15-20 epoch, cả hai đường Loss đều có xu hướng hội tụ, cho thấy việc huấn luyện đã dần ổn định. Khoảng cách giữa Train Loss và Test Loss khá nhỏ, nên mô hình đang học một cách ổn định và có thể đang hoạt động tốt trên cả tập huấn luyện và kiểm tra.

+) Độ chính xác tăng nhanh trong các epoch đầu tiên và dần giảm tốc độ tăng, đạt khoảng 72% sau 30 epoch, điều này cho thấy mô hình đã học được các đặc trưng của dữ liệu.

## So sánh các mô hình

Biểu đồ so sánh các mô hình MLP 1 lớp ẩn, MLP 2 lớp ẩn và CNN



Đánh giá:



+)Train loss:CNN có giá trị Loss thấp nhất (0.7015), thể hiện khả năng học tốt hơn so với MLP1 (1.2037) và MLP2 (1.1557).

+)Test loss:CNN tiếp tục có giá trị Test Loss thấp nhất (0.8613), cho thấy nó tổng quát hóa tốt hơn so với MLP1 (1.2932) và MLP2 (1.2417).

+)Accuracy:CNN đạt độ chính xác cao nhất (72.59%), vượt trội so với MLP1 (54.89%) và MLP2 (56.53%).

+)Mô hình CNN có hiệu suất vượt trội hơn cả trên các chỉ số Train Loss, Test Loss và Accuracy, điều này phù hợp với tính chất của CNN trong việc xử lý dữ liệu hình ảnh.

+)MLP2 mang lại hiệu suất tốt hơn chút so với MLP1, việc tăng số lớp ẩn trong mạng MLP cải thiện kết quả mô hình.

### **Kết quả phụ:**

Kết quả việc điều chỉnh siêu tham số:

#### **MLP1:**

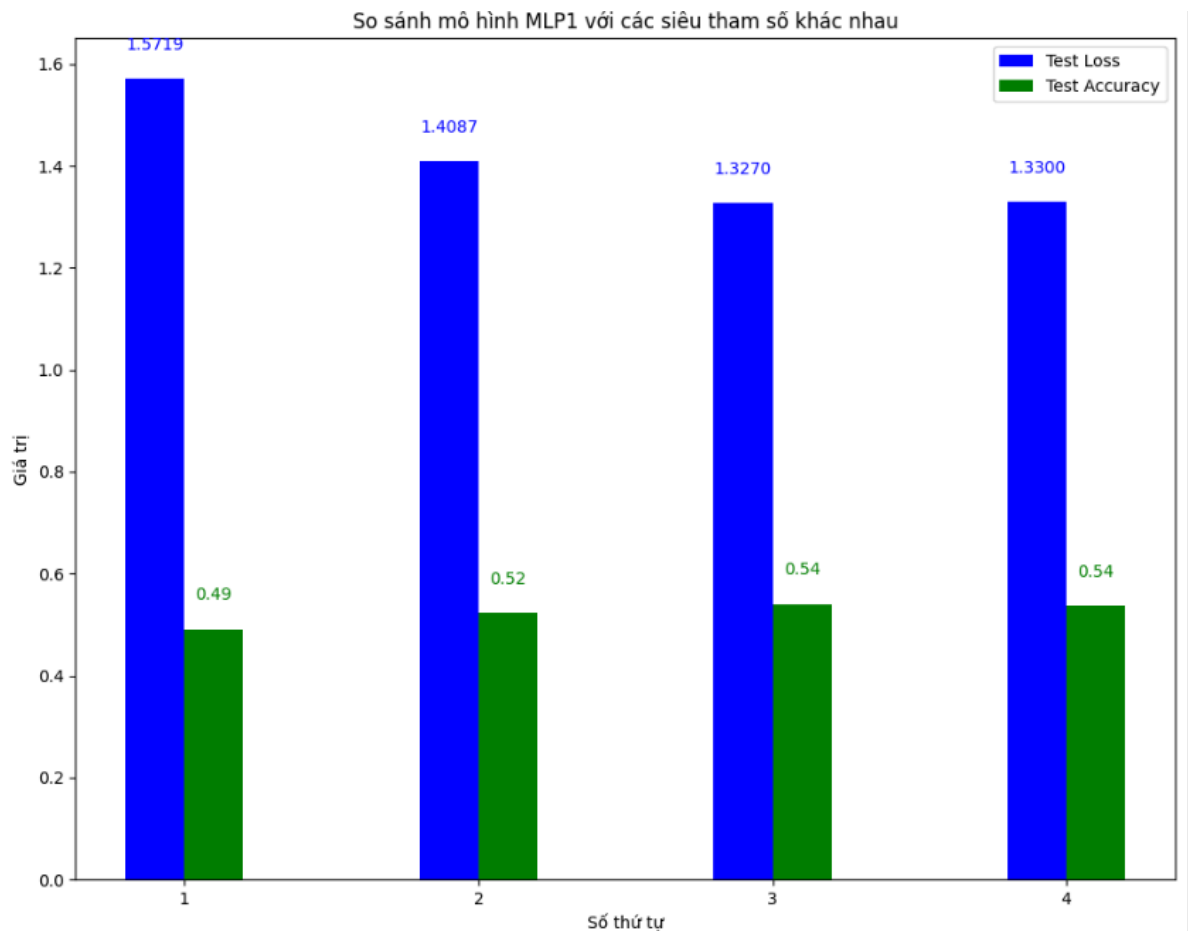
Training log : training\_log1.txt.

Optimizer: Adam,SGD

Batch size: 32,64

Learning rate:0.001,0.01

Biểu đồ so sánh hiệu suất mô hình MLP1 với các siêu tham số khác nhau:



1:Optimizer = Adam,learning rate = 0.001,batch size = 32

2:Optimizer = Adam,learning rate = 0.001,batch size = 64

3:Optimizer = SGD,learning rate = 0.01,batch size = 64

4:Optimizer = SGD,learning rate = 0.01,batch size = 32

Đánh giá :

+)Các cấu hình sử dụng SGD tốt hơn một chút so với sử dụng Adam.

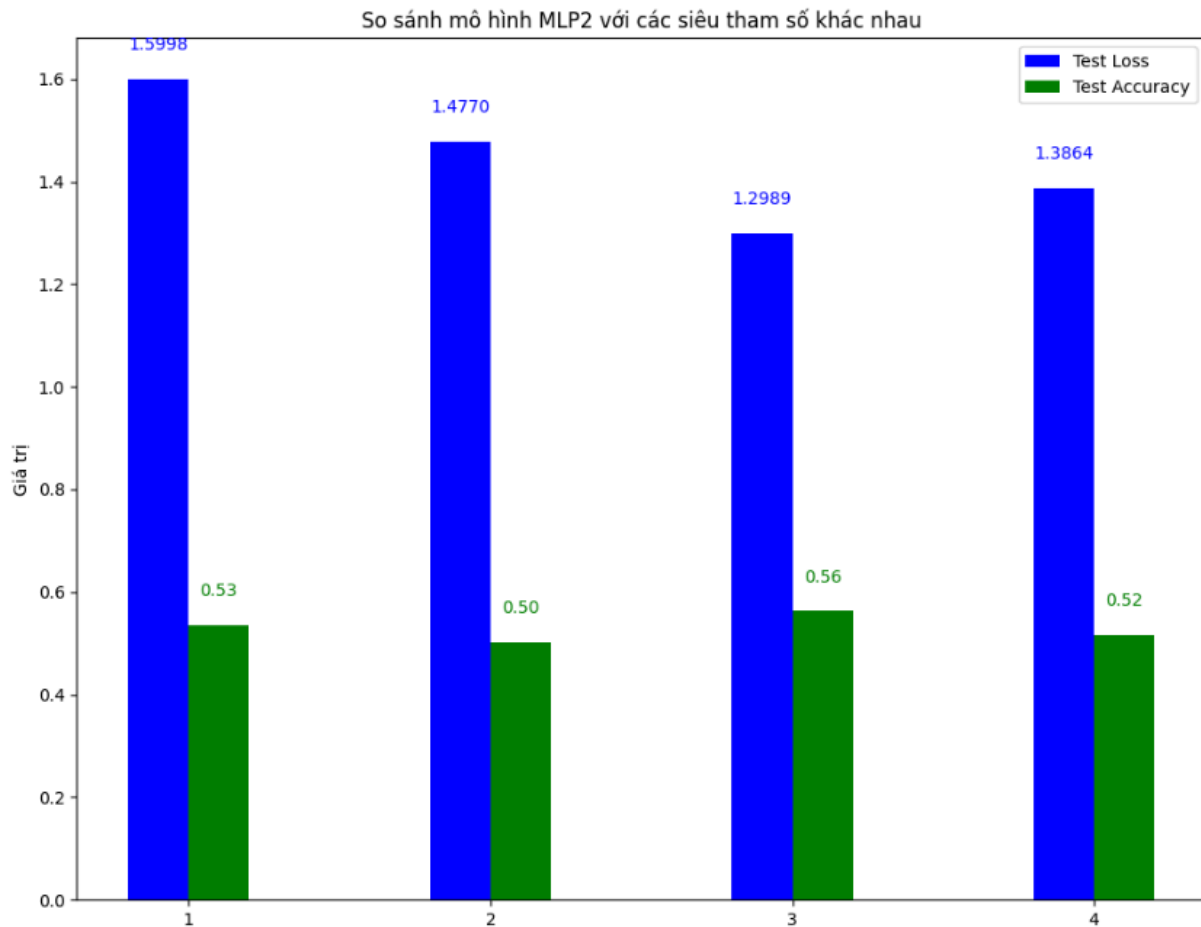
+)Với các batch size khác nhau(32 và 64) thì mô hình có độ chênh lệch hiệu quả khá nhỏ.

+)Mô hình có kết quả tốt nhất là mô hình có siêu tham số Optimizer: SGD, Batch size: 64, Learning rate: 0.01 với test loss = 1.327 và test accuracy =54%

## MLP2:

Training log : training\_log2.txt.

Biểu đồ so sánh hiệu suất mô hình MLP2 với các siêu tham số khác nhau:



1:Optimizer = Adam,learning rate = 0.001,batch size = 64

2:Optimizer = Adam,learning rate = 0.001,batch size = 32

3:Optimizer = SGD,learning rate = 0.01,batch size = 32

4:Optimizer = SGD,learning rate = 0.01,batch size = 64

Đánh giá :

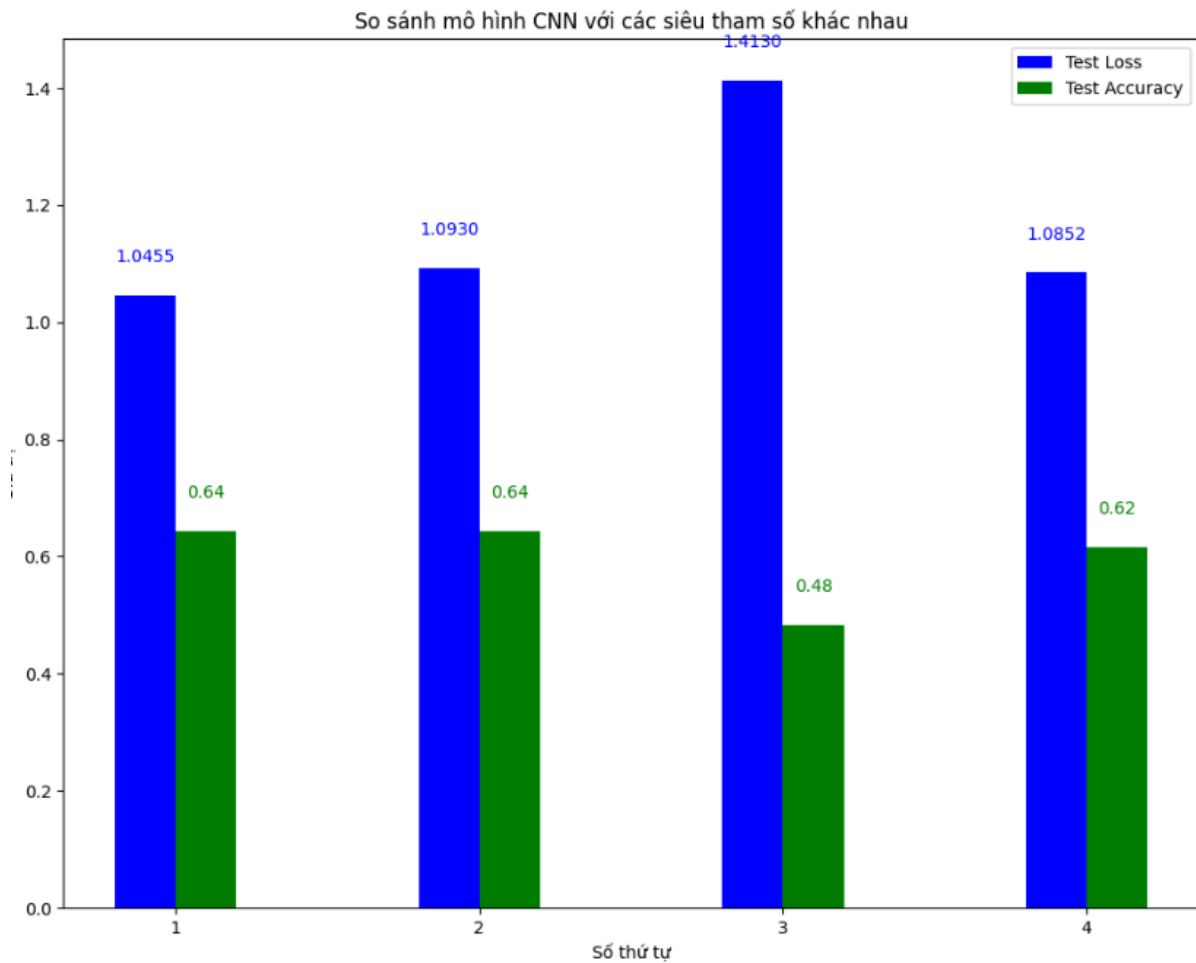
+)Mô hình tốt nhất có cấu hình 3(SGD, learning rate = 0.01,batch size = 32) với Train Loss:0.9023 , Test Loss: 1.2989, Accuracy: 56.24%.

+)Các cấu hình có độ chênh lệch hiệu suất không cao từ 1%-6% và độ chênh lệch Test Loss cũng thế.Việc sử dụng Adam,SGD với các chỉ số khác nhau thì mô hình có sự thay đổi không quá đáng kể.

## CNN

Training log : training\_logCNN.txt.

Biểu đồ so sánh hiệu suất mô hình CNN với các siêu tham số khác nhau:



1:Optimizer = Adam,learning rate =0.001,batch size = 32

2:Optimizer = Adam,learning rate =0.001,batch size = 64

3:Optimizer = SGD,learning rate =0.01,batch size = 64

4:Optimizer = SGD,learning rate =0.01,batch size = 32

Đánh giá :

+)Adam với learning rate thấp và batch size 32 hoặc 64 mang lại kết quả ổn định và tốt.

+)Với learning rate cao hơn và sử dụng optimizer SGD, mô hình gặp khó khăn trong việc hội tụ hoặc tối ưu hóa.

+)Test Loss:Adam hoạt động hiệu quả hơn trong việc giảm lỗi so với SGD. Việc sử dụng SGD với learning rate cao có thể dẫn đến mất mát lớn và khó tối ưu hóa.

+)Ảnh hưởng của batch size:Khi sử dụng Adam, thay đổi batch size giữa 32 và 64 không ảnh hưởng nhiều đến Test Accuracy (vẫn đạt 0.64) hoặc Test Loss.Khi sử dụng SGD, batch size lớn hơn (64) có cải thiện một chút so với batch size nhỏ hơn (32), tuy nhiên vẫn kém hơn so với Adam.

+)Mô hình tốt nhất có cấu hình 1(Adam, learning rate = 0.001,batch size = 32) với Train Loss: 0.9213, Test Loss: 1.0455, Accuracy: 64.33%.

## **Limitations and future work**

### **1. Hạn chế**

+)Các kiến trúc được sử dụng trong dự án:MLP và CNN,tương đối đơn giản, hiệu suất không cao.

+)Xử lý bộ dữ liệu một cách đơn giản.

+)Khó khăn trong việc thử nghiệm nhiều siêu tham số:giới hạn tài nguyên tính toán,thời gian thử nghiệm.

+)Các mô hình có xu hướng overfitting,khiến cho hiệu quả mô hình không tăng cao được.

### **2. Công việc tương lai**

+)Chỉnh sửa và tối ưu mô hình hiện tại,khắc phục các hạn chế.

+)Thử nghiệm với các siêu tham số khác nhau để tìm phương án tốt nhất.

+)Sử dụng với nhiều tập dữ liệu khác nhau.

+)Thử nghiệm với mô hình phức tạp hơn:để cải thiện độ chính xác và hiệu suất tổng thể.

+)Áp dụng thêm các kỹ thuật để tối ưu hóa mô hình.

## **Conclusion**

Trong bài tập này, chúng tôi đã thực hiện thành công bài toán phân loại ảnh trên bộ dữ liệu CIFAR-10, sử dụng hai mô hình chính: MLP và CNN. Kết quả cho thấy mô hình CNN vượt trội hơn về độ chính xác so với MLP nhờ khả năng trích xuất đặc trưng tốt hơn từ dữ liệu hình ảnh. Quá trình huấn luyện đã bao gồm việc tinh chỉnh các siêu tham số và áp dụng các kỹ thuật giảm thiểu overfitting, mang lại hiệu quả khá tốt.

Tuy nhiên, vẫn còn một số hạn chế như việc sử dụng mô hình đơn giản, thiếu các kỹ thuật tiền xử lý dữ liệu nâng cao, và giới hạn về tài nguyên tính toán. Trong tương lai, chúng tôi dự định mở rộng nghiên cứu bằng cách triển khai các mô hình tiên tiến hơn, thử nghiệm trên các bộ dữ liệu lớn hơn, và tích hợp mô hình vào các ứng dụng thực tế.

## References

1. CIFAR - 10 Dataset(2009).
2. [Training a Classifier — PyTorch Tutorials 2.5.0+cu124 documentation](#)
3. [pdf-slides - Google Drive](#)

## Danh sách thành viên

Trần Đức Hùng - 22022639

**Github:**[hungtran1210/Final-project-DL](https://github.com/hungtran1210/Final-project-DL)