Sử dụng các mô hình học sâu cho phân loại ảnh

Cao Tuấn Anh[1], Đinh Duy Bách[2], Lê Hữu Đức[3], Lý Quốc An[4], and Bùi Thế Long[5]

[1]*22022562@vnu.edu.vn*
[2]*22022531@vnu.edu.vn*
[3]*22022535@vnu.edu.vn*
[4]*22022660@vnu.edu.vn*
[5]*22022647@vnu.edu.vn*

Mục lục

Danh sách hình vẽ

Danh sách bảng

Sử dụng các mô hình học sâu cho phân loại ảnh

***Tóm tắt nội dung*—Dự án của chúng tôi tập chung vào việc sử dụng 3 mô hình học sâu hiện đại là CNN, MLPs và Transformer trong việc phân loại hình ảnh, sau đó so sánh và đánh giá hiệu suất của các mô hình với nhau.**

I. Giới thiệu

Học sâu là một tập con của học máy, tập trung vào việc xây dựng và huấn luyện mạng nơ ron nhiều lớp (DNN) để chúng có thể tự học, hiểu dữ liệu, mô phỏng khả năng quyết định của con người.

Học sâu đã và đang trở thành một trong những lĩnh vực quan trọng của Trí tuệ nhân tạo, và ngoài ra còn được ứng dụng rất nhiều vào mọi mặt của cuộc sống như y tế, kinh tế, xã hội, ... Các doanh nghiệp hiện nay đã và đang áp dụng các mô hình học sâu tiên tiến và hiện đại vào quá trình sản xuất và kinh doanh nhằm tối ưu hóa lợi nhuận của họ.

Ở trong xã hội hiện nay, ta thấy rằng nhiều thông tin được lưu trữ dưới dạng dữ liệu số, dữ liệu văn bản, dữ liệu hình ảnh, ... Trong đó dữ liệu hình ảnh chiếm một phần quan trọng. Và bài toán xử lý dữ liệu ảnh hiện nay không còn là một bài toán quá mới mẻ nữa, mà đã được đầu tư nghiên cứu rất nhiều để đưa ra giải pháp tốt nhất. Một trong những giải pháp được áp dụng vào vã đã chứng minh được sự thành công đó là các mô hình mạng học sâu có khả năng học hỏi, từ đó thực hiện các tác vụ như dự đoán, phân loại, ...

Trong đề tài này chúng tôi xin phép trình bày về việc ứng dụng các mô hình học sâu trong việc phân loại hình ảnh, cụ thể là trong tập dữ liệu tiny-imageNet.

II. Các nghiên cứu liên quan

1. Comparative Analysis of Image Classification Algorithms Based on Traditional Machine Learning and Deep Learning: Bài báo tập trung vào việc so sánh độ chính xác cho tác vụ phân loại hình ảnh với 2 thuật toán: SVM đại diện cho các phương pháp học máy truyền thống còn CNN đại diện cho phương pháp học sâu trên tập dữ liệu MNIST. [1]

2.Deep Learning Models Based on Image Classification: A Review: Bài báo tập trung so sánh các mô hình học sâu hiện nay trên 2 tập dữ liệu CIFAR-10 và CIFAR-100 [2]

3.Deep learning for biological image classification: Bài báo tập chung vào việc so sánh các mô hình học sâu và học máy trong tác vụ phân loại ảnh trong y tế. [3]

III. Thông tin về bộ dữ liệu

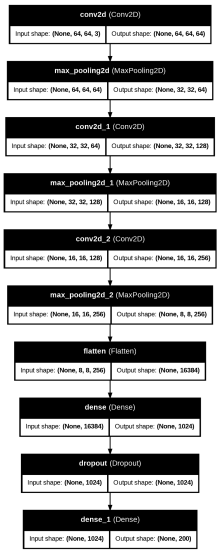
Bộ dữ liệu mà nhóm sử dụng cho bài tập lớn là tiny-ImageNet với 100000 ảnh dùng cho mục đích training,10000 ảnh dùng cho mục đích testing và 200 lớp cần phân loại. Trong một lớp cần phân loại thì có 500 ảnh sử dụng cho mục đích training, 50 ảnh cho mục đích validation.

Hình 1. Bộ dữ liệu ImageNet

IV. Các mô hình sử dụng

A. CNN

[7] Một mạng tích chập hay còn được gọi là CNN hoặc



Hình 2. Mô hình CNN

ConvNet, là một lớp củ các mạng thần kinh chuyên biệt trong xử lý dữ liệu dạng topo lưới, chẳng hạn như hình ảnh. Một bức ảnh kỹ thuật số là dạng biểu diễn nhị phân của dữ liệu ảo. Nó chứa một chuỗi các điểm ảnh ở dạng lưới mà chứa các giá trị điểm ảnh để biểu thị độ sáng và màu sắc của mỗi điểm.

Một mạng CNN tiêu biểu thường có 3 lớp: 1 lớp Convolutional, 1 lớp Pooling và 1 lớp fully-connected.

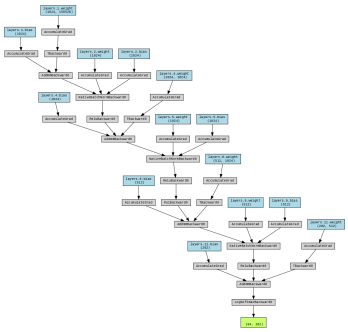
- Lớp Convolutional: Lớp này thực hiện phép nhân chấm giữa hai ma trận, trong đó một ma trận là tập hợp các tham số có thể học được, còn được gọi là kernel (hạt nhân), và ma trận kia là phần giới hạn của vùng tiếp

nhận. Kernel có kích thước không gian nhỏ hơn một hình ảnh nhưng có độ sâu lớn hơn. Điều này có nghĩa là, nếu hình ảnh được tạo thành từ ba kênh (RGB), chiều cao và chiều rộng của kernel sẽ nhỏ về mặt không gian, nhưng độ sâu của nó sẽ mở rộng đến cả ba kênh.

- Lớp Pooling: Lớp pooling thay thế đầu ra của mạng tại một số vị trí nhất định bằng cách lấy một thống kê tóm tắt từ các đầu ra lân cận. Điều này giúp giảm kích thước không gian của biểu diễn, từ đó giảm lượng tính toán và số lượng tham số cần thiết. Hoạt động pooling được thực hiện trên từng phần riêng lẻ của biểu diễn.
- Lớp Fully-connected: Các neuron trong lớp này có kết nối đầy đủ với tất cả các neuron ở lớp trước và lớp sau, giống như trong mạng nơ-ron kết nối đầy đủ thông thường (FCNN). Do đó, nó có thể được tính toán như bình thường bằng phép nhân ma trận, sau đó là tác động của hệ số bias.Lớp FC (Fully Connected) giúp ánh xạ biểu diễn giữa đầu vào và đầu ra.

B. MLPs

[8]

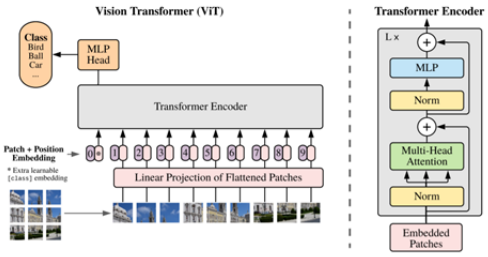


Hình 3. Mô hình MLPs

- Multilayer Perceptron là một thuật toán học máy có giám sát (Machine learning) thuộc lớp mạng nơ ron nhân tạo, là tập hợp các perceptron được chia làm nhiều lớp là một layer.
- Một MLPs thường có cấu trúc là một lớp đầu vào (input layer), lớp ẩn (hidden layer) và lớp đầu ra (output layer).
- MLP có tính phi tuyến nên nên có thể học các hàm phi tuyến phức tạp và phân biệt dữ liệu không phân tách tuyến tính. Tuy nhiên MLP có một số nhược điểm như hàm bị phân tách cho lớp ẩn dẫn đến vấn đề tối ưu hóa không lồi và tồn tại cực tiểu cục bộ. Ngoài ra MLP còn sử dụng 1 số siêu tham số dẫn đến tiêu tốn về thời gian và nguồn lực, hơn nữa còn nhạy cảm với việc mở rộng.

C. ViT

- Với mỗi ảnh được đưa vào huấn luyện với ViT, mô hình sẽ chia ảnh ra làm các patch, với vai trò tương tự như các token trong NLP. Tuy nhiên, một patch được tạo thành bởi một ma trận các pixel, không có cách embedding dễ dàng như token ngôn ngữ vì mỗi patch của mỗi ảnh đều khác nhau. Do đó, với mỗi patch, một mạng thần kinh



Hình 4. Mô hình ViT

sẽ được sử dụng để chuyển mỗi patch ảnh thành một embedding. Tiếp dó, tương tự như với NLP, mỗi một patch ảnh sẽ được cộng thêm một position embedding để biểu diễn thông tin về vị trí của patch trong ảnh.

- Bên cạnh embedding từ các patch của ảnh đầu vào, một embedding đặc biệt là [CLS] (classification) được thêm vào nhằm phục vụ cho quá trình phân biệt các lớp đầu ra. Vector embedding [CLS] được khởi tạo ngẫu nhiên, và sau đó được huấn luyện và được sử dụng ở các phần sau của mô hình, vector này có thể được coi như là một biểu diễn toàn cục về nội dung của ảnh.
- Sau đó, các vector embedding sẽ được đưa vào lớp transformer encoder. Ở mỗi lớp encoder, thông tin được đưa vào xử lý ở các attention head, được dùng để chia sẻ thông tin giữa các input vector. Mỗi attention head sẽ thực hiện song song quá trình biến đổi vector đầu vào bằng các ma trận Key, Query và Value. Sau đó, đầu ra từ các output sẽ được tổng hợp và biến đổi để có chiều cần thiết. Sau khi mỗi vector đi qua lớp attention, đầu ra sẽ được normalize bởi một lớp LayerNorm. Cuối cùng ở mỗi lớp Transformer Encoder, vector sẽ được đưa qua một lớp feed-forward. Vector đầu ra được tạo ra từ input của token [CLS] sẽ được sử dụng để phân loại ảnh.
- Độ phức tạp tính toán của bước attention là $O(n^2)$ với n là độ dài của chuỗi đầu vào. Để giảm khối lượng tính toán mà vẫn giữ được hiểu quả, các ma trận attention Key và Value được chiếu xuống chiều thấp hơn với rank là k (k là hyperparameter của mô hình). Ma trận bậc thấp với chiều k được dùng để ước lượng ma trận Key và Value. k có chiều càng lớn thì càng thể hiện được đúng hơn ma trận key và value nhưng sẽ tốn nhiều tài nguyên tính toán hơn.

V. So sánh các mô hình pretrained

Với bộ dữ liệu tiny-ImageNet, ta sử dụng các mô hình pretrained GoogleNet [4],AlexNet [6] và Resnet50 [5] để training độc lập. Sau đó so sánh kết quả training của các mô hình trên với cùng 1 điều kiện như nhau.

Với số lượng epoch = 1, sử dụng hàm optimizer là Adam cùng với learning rate = 0.01, cùng hàm criterion là Cross Entropy Loss, được training trên CPU ta được kết quả như bảng 1:

Trong cùng một điều kiện huấn luyện thì mô hình GoogleNet tỏ ra chiếm ưu thế hơn với tỷ lệ chính xác xấp xỉ 3(%) cùng với chỉ số Loss dưới 5, thể hiện rằng mô hình có khả năng

	Kết quả training	
Chỉ số	Độ chính xác (%)	Loss
GoogleNet	2.83	4.969
AlexNet	0.06	10.496
Resnet50	0.47	5.307

Bảng I. BẢNG SO SÁNH ĐỘ KẾT QUẢ TRAING

học được nhiều đặc trưng quan trọng so với 2 mô hình còn lại, đồng thời có độ sai số trong huấn luyện là nhỏ nhất. Mô hình có các chỉ số tệ nhất là AlexNet với tỷ lệ chính xác chỉ là 0.06(%) cùng với độ Loss lớn hơn 10, chứng tỏ mô hình không quá phù hợp trong quá trình training bộ dữ liệu trên. Mô hình Resnet50 có độ Loss là 5.307 tuy nhiên độ chính xác thấp hơn khá nhiều so với mô hình GoogleNet.

VI. THỰC NGHIỆM

A. Tiền xử lý dữ liệu

Tiny ImageNet là một dataset phổ biến chứa 200 lớp hình ảnh với kích thước nhỏ (64x64 pixel), cho nên nhóm chúng tôi đã tiền xử lý bộ dữ liệu này như sau:

- Sử dụng các bộ thư viện sẵn có như torch, datasets và torchvision để lấy dữ liệu từ HuggingFace về. Chia tập dữ liệu đã tải về thành 2 tập chính là tập training và tập validation.
- Áp dụng các kỹ thuật tiền xử lý để xử lý các loại ảnh là RBG và Grayscale. Các ảnh ta cùng đưa về kích thước 256 * 256, sau đó cắt ảnh ngẫu nhiên với kích thước 224 * 224 với các ảnh trong tập training, cắt ảnh trung tâm với kích thước 224 * 224 với các ảnh trong tập test. Với tập rgb-train thì ta áp dụng các biến đổi sau với xác suất 0.5: lật ngang hình ảnh ngẫu nhiên, xoay ảnh ngẫu nhiên trong khoảng từ -15 đến 15 độ, thay đổi các thuộc tính màu sắc của ảnh. Với tập grayscale-train thì ta thay đổi màu sắc của các ảnh và chuyển đổi ảnh grayscale thành ảnh rgb. Với tập grayscale-test, ta chuyển đổi thành 3 kênh rbg giả lập. Sau khi xử lý xong từng tập, ta chuyển các ảnh trên thành các Tensor.
- Các nhãn được chuyển dưới dạng one-hot encoding để tương thích với yêu cầu của mô hình.
- Tập dữ liệu training ban đầu được chia ra thành 2 tập dữ liệu là training set và test set với tỷ lệ 0.8:0.2 và tạo các DataLoader phục vụ cho mục đích huấn luyện dữ liệu.

B. Mô tả các tham số

1) CNN: Mô hình SimpleCNN gồm:

- Feature extractor bao gồm ba khối tích chập liên tiếp. Khối đầu tiên áp dụng một tầng Conv2D với 64 bộ lọc, kernel 3×3, và padding=1 để trích xuất các đặc trưng cục bộ. Sau đó, hàm kích hoạt ReLU được áp dụng để thêm phi tuyến tính vào đầu ra. Tiếp theo là tầng MaxPooling với kernel 2×2 và stride=2, giúp giảm kích thước không gian của đầu ra đi 2.
- Khối thứ hai sử dụng một tầng Conv2D với 128 bộ lọc và các thông số tương tự như khối đầu tiên, tiếp theo là ReLU và MaxPooling.

```python
class SimpleCNN(nn.Module):
    def __init__(self, num_classes=200):
        super(SimpleCNN, self).__init__()
        self.features = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Conv2d(64, 128, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=2, stride=2),
            nn.Conv2d(128, 256, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=2, stride=2)
        )
        self.classifier = nn.Sequential(
            nn.Linear(256 * 8 * 8, 1024),
            nn.ReLU(inplace=True),
            nn.Dropout(0.5),
            nn.Linear(1024, num_classes)
        )

    def forward(self, x):
        x = self.features(x)
        x = x.view(x.size(0), -1)
        x = self.classifier(x)
        return x
```

Hình 5. Các tham số trong mô hình CNN

```python
class MLP(nn.Module):
    def __init__(self, input_dim, hidden_dim, num_classes=200):
        super(MLP, self).__init__()
        self.layers = nn.Sequential(
            nn.Flatten(),
            nn.Linear(input_dim, hidden_dim),
            nn.BatchNorm1d(hidden_dim),
            nn.ReLU(),
            nn.Linear(hidden_dim, 1024),
            nn.BatchNorm1d(1024),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(1024, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(),
            nn.Linear(512, num_classes),
            nn.LogSoftmax(dim=1)
        )
```

Hình 6. Các tham số trong mô hình MLPs

- Khối thứ ba áp dụng một tầng Conv2D với 256 bộ lọc, tiếp theo là ReLU và MaxPooling. Đầu ra cuối cùng từ phần feature extractor là một tensor kích thước (8,8,256).
- Classifier nhận đầu vào từ phần feature extractor sau khi được làm phẳng bằng thành một vector 1 chiều. Tầng đầu tiên trong classifier là một Dense Layer với 1024 nơ-ron và hàm kích hoạt ReLU. Dropout với tỷ lệ 0.5 được sử dụng sau đó để giảm nguy cơ overfitting. Cuối cùng, một tầng Dense Layer với số nơ-ron bằng số lớp 200 trả về kết quả phân loại.
- Trong forward, dữ liệu đầu vào đi qua phần feature extractor, được flatten, và sau đó đi qua classifier để trả về đầu ra cuối cùng.

2) MLPs: Mạng MLP được xây dựng gồm 4 lớp:

- Lớp đầu vào: Lớp đầu tiên là lớp đầu vào được làm phẳng bằng flatten để chuyển thành một vector một chiều cho các lớp linear.
- Lớp Linear 1 (nn.Linear(input-dim, hidden-dim)): Sau khi dữ liệu được chuyển hóa thành vector ,Lớp fully connected đầu tiên kết nối dữ liệu đầu vào input-dim được đưa vào hidden layer với hidden-dim neuron. Ta dùng Batch Normalization để chuẩn hóa đầu ra của lớp này , đảm bảo rằng mỗi batch có phân phối ổn định, đồng thời sử dụng hàm kích hoạt ReLU để huấn luyện.
- Lớp Linear 2 (nn.Linear(hidden-dim, 1024)): Dữ liệu từ lớp trước được chuyển qua lớp fully connected thứ hai, với 1024 đơn vị (neuron) Batch Normalization và ReLU

```
efficient_transformer = Linformer(
    dim=128,
    seq_len= 64 + 1,  # 8x8 patches + 1 cls-token
    depth=12,
    heads=8,
    k=32
)

model = ViT(
    dim=128,
    image_size=64,
    patch_size=8,
    num_classes=200,
    transformer=efficient_transformer,
    channels=3,
).cuda()
```

Hình 7. Các tham số trong mô hình ViT

tiếp tục được sử dụng để ổn định quá trình huấn luyện và kích hoạt các đặc trưng. Dropout 50% các neuron trong lớp này để tránh overfiting.

- Lớp thứ 3 (nn.Linear(1024, 512)): Dùng Linear kết nối lớp có 1024 neuron đến lớp có 512 neuron Batch Normalization và ReLU tiếp tục được sử dụng để ổn định quá trình huấn luyện và kích hoạt các đặc trưng.
- Lớp thứ 4 (nn.Linear(512, num-classes)): Lớp fully connected cuối cùng chuyển từ không gian ẩn (512 neuron) đến không gian đầu ra, với số lượng neuron là num-classes(200 lớp).Dùng LogSoftmax làm hàm kích hoạt để tính toán xác suất logarit.

3) ViT: Mô hình ViT nhóm sử dụng được huấn luyện trên bộ dữ liệu Tiny-ImageNet gồm 100,000 ảnh thuộc 200 lớp. Trong đó, mô hình sử dụng Linear Transformer (Linformer) với k=32. Ở đây, các ma trận attention sẽ được chiếu xuống chiều thấp hơn để tăng tốc độ tính toán. Mỗi đầu vào được biểu diễn bởi một vector 128 chiều (dim=128), mỗi ảnh 3x64x64 được chia làm 64 batch 8x8. Ở seq-len, ta thấy 1 token [CLS] được thêm vào bên cạnh 64 patch vector embedding. Có 12 layer attention được sử dụng, với 8 attention heads.

C. Thiết lập môi trường huấn luyện

- Với mô hình MLP, ta dùng hàm criterion là Cross entropy loss, hàm tối ưu là Adam với learning rate là 0.0001 và weights-decay là 0.001.
- Với mô hình SimpleCNN, ta dùng hàm criterion là Cross entropy loss, hàm tối ưu là Adam với learning rate là 0.001 và weights=decay là 0.0001.
- Với mô hình ViT, ta dùng hàm criterion là Cross entropy loss, hàm tối ưu là Adam cùng với learning rate là 0.001, gamma là 0.7.

D. Kết quả huấn luyện

- Sau hơn 10 lần chạy với mô hình MLPs, độ chính xác dao động từ khoảng 0.5% đến 8% với cả hai tập training và validation, đồng thời chỉ số loss dao động trong khoảng từ 4.5 đến 5.4 trên cùng 2 tập dữ liệu,
- Với 20 epochs training trên mô hình SimpleCNN, độ chính xác dao động trong khoảng từ 7% đến xấp xỉ 52% trên tập training và từ 15% đên 27% trên tập validation , đồng thời chỉ số loss dao động trong khoảng từ 1.8 đến 4.6 trên tập training và 3.18 đến 3.93 trên tập validation.

	Độ chính xác (%)	
Mô hình	Tập train	Tập validation
SimpleCNN	51.60	27.36
MLP	7.55	7.6
ViT	51.37	25.87

Bảng II. BẢNG SO SÁNH ĐỘ CHÍNH XÁC

	Độ chính xác (%)	
Mô hình	Tập train	Tập validation
SimpleCNN	1.8136	3.1858
MLP	4.5448	4.7429
ViT	1.8781	3.3915

Bảng III. BẢNG SO SÁNH CHỈ SỐ LOSS

- Với 20 epochs training trên mô hình ViT, độ chính xác dao động từ 4.9% đên xấp xỉ 52% trên tập training, 7.7% đến 25.5% trên tập validation, đồng thời chỉ số loss dao động từ 1.87 đến 4.79 trên tập training và 3.45 đến 4.5 trên tập validation
-

VII. SO SÁNH, ĐÁNH GIÁ

Trước hết ta đánh giá độ chính xác cao nhất của 3 mô hình như bảng 2

Nhìn vào bảng dữ liệu trên ta thấy rằng 2 mô hình CNN và ViT có độ chính xác trên tập training và validation là ngang nhau, trong khi đó mô hình MLP tỏ ra yếu thế hơn với chỉ 7.55% độ chính xác, trong khi đó mô hình CNN nhỉnh hơn mô hình ViT ở tập validation 1 chút. Qua đây ta thấy rằng 2 mô hình CNN và ViT tỏ ra tốt hơn hẳn so với MLP ở khả năng dự đoán.

Sau đó là so sánh chỉ số loss thấp nhất của các mô hình như bảng 3

Ta thấy rằng độ thất thoát trong dự đoán của mô hình CNN là thấp nhất trong các mô hình, tiếp theo đó là mô hình ViT với xấp xỉ 1.9 trên tập training và 3.4 trên tập validation. Mô hình MLP tỏ ra yếu hơn hẳn với chỉ số loss khá cao, chứng tỏ độ mất mát trong tính toán khá lớn.

VIII. HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN

Các mô hình được triển khai đều có khả năng thực thi trên tập dữ liệu ImageNet. Tuy nhiên, kết quả hiện tại còn gặp một số hạn chế đáng kể. Thứ nhất, độ chính xác của các mô hình chưa cao, nguyên nhân chủ yếu xuất phát từ hạn chế về tài nguyên phần cứng, đặc biệt là bộ xử lý GPU mà nhóm đang sử dụng. Thứ hai, tập dữ liệu ImageNet có đặc điểm khá phức tạp với số lượng lớn hình ảnh và nhiều nhãn phân loại, đòi hỏi thời gian huấn luyện kéo dài và tiêu tốn nhiều tài nguyên tính toán. Ngoài ra, việc tối ưu hóa siêu tham số của các mô hình cũng chưa được thực hiện một cách triệt để, dẫn đến kết quả chưa đạt được như kỳ vọng.

Để khắc phục những hạn chế trên và cải thiện kết quả bài toán trong tương lai, nhóm chúng tôi đề xuất một số hướng phát triển như sau:

- Sử dụng các bộ xử lý mạnh mẽ hơn như GPU hoặc TPU để đẩy nhanh tốc độ huấn luyện và tăng khả năng thử nghiệm nhiều mô hình phức tạp hơn.
- Áp dụng các kỹ thuật như Grid Search, Random Search hoặc Bayesian Optimization để tìm ra bộ siêu tham số tối ưu nhất cho từng mô hình.
- Áp dụng các kỹ thuật tiền xử lý dữ liệu như chuẩn hóa, tăng cường dữ liệu (Data Augmentation) nhằm cải thiện chất lượng đầu vào và giúp mô hình học được nhiều đặc trưng hơn. Thực hiện huấn luyện lại (Fine-tuning) trên các mô hình đã được tiền huấn luyện (Pre-trained Models).
- Nghiên cứu và thử nghiệm các mô hình tiên tiến như Transformers, Hybrid CNN-Transformer, hoặc các phiên bản cải tiến của ViT nhằm tận dụng được khả năng học đặc trưng toàn cục của ảnh.
- Ngoài ImageNetra , nhóm sẽ thử nghiệm trên các tập dữ liệu khác như CIFAR-10, CIFAR-100, Tiny ImageNet hoặc COCO để đánh giá khả năng tổng quát của các mô hình.

Bằng việc kết hợp các giải pháp trên, chúng tôi kỳ vọng có thể cải thiện đáng kể độ chính xác và hiệu suất của các mô hình, đồng thời rút ngắn thời gian huấn luyện và mở rộng khả năng ứng dụng vào thực tế.

IX. Kết luận

Bài toán phân loại ảnh là một bài toán khá là hay, luôn được đầu tư để cải thiện và kết quả bài toán có ứng dụng rất nhiều với đời sống. Trong khuôn khổ nghiên cứu này, nhóm chúng tôi đã triển khai và so sánh các mô hình học sâu bao gồm CNN (Convolutional Neural Networks), MLPs (Multi-Layer Perceptrons) và ViT (Vision Transformers) trên tập dữ liệu ImageNet. Mặc dù còn một số hạn chế về tài nguyên phần cứng và thời gian huấn luyện, các mô hình đã thể hiện được khả năng học và phân loại hình ảnh tương đối hiệu quả.

Kết quả thực nghiệm cho thấy:

- Các mô hình CNN có khả năng học đặc trưng cục bộ tốt, phù hợp với các tập dữ liệu hình ảnh lớn.
- Mô hình MLPs tuy đơn giản nhưng vẫn có thể hoạt động hiệu quả khi được tối ưu hóa đúng cách.
- ViT là một hướng tiếp cận mới mẻ, có tiềm năng lớn trong việc học các đặc trưng toàn cục của hình ảnh, đặc biệt trên các tập dữ liệu lớn như ImageNet.

Trong tương lai, nhóm chúng tôi sẽ tập trung vào việc cải thiện kết quả bài toán bằng cách nâng cấp tài nguyên phần cứng, tối ưu hóa các siêu tham số, sử dụng các mô hình tiên tiến hơn và áp dụng các kỹ thuật tăng cường dữ liệu. Hy vọng với sự cố gắng và kiên trì thì các mô hình sẽ hoàn thiện hơn nữa trong việc xử lý bài toán.

Tài liệu

[1] Wang, Pin, En Fan, and Peng Wang. "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning."Pattern recognition letters 141 (2021): 61-67.https://www.sciencedirect.com/science/article/abs/pii/S0167865520302981

[2] Deep Learning Models Based on Image Classification: A Review.

[3] Deep learning for biological image classification. https://www.sciencedirect.com/science/article/abs/pii/S0957417417303627

[4] Szegedy Christian, et al. "Going deeper with convolutions."Proceedings of the IEEE conference on computer vision and pattern recognition.https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf

[5] He, Kaiming, et al. "Deep residual learning for image recognition."Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

[6] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge."International journal of computer vision 115 (2015): 211-252. https://link.springer.com/article/10.1007/s11263-015-0816-y

[7] Convolutional Neural Networks, Explained.https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939.

[8] Phân loại chữ số viết tay với Perceptron nhiều lớp (MLP).https://product.vinbigdata.org/phan-loai-chu-so-viet-tay-voi-perceptron-nhieu-lop-mlp/