Course Title: CSE 297F Graduate Individual Study
Professor: Cihang Xie
Name: Xi Wen
Student ID: 2005917
Date: 08/30/2024

Capstone Project Report
Optimizing CLIP for Fashion Classification: Direct Image Classification vs. Image-Text Similarity Learning

1. Introduction

Computer Vision (CV) has witnessed significant advancements in recent years, largely driven by deep learning techniques. These advancements have revolutionized various domains, such as object detection, image classification, and image generation, enabling machines to interpret and understand visual data with increasing accuracy. Among these, image classification has become a cornerstone task in CV, where the goal is to assign a label to an image based on its visual content. This task is particularly critical in domains such as fashion, where accurate classification of clothing items is essential for applications like e-commerce, inventory management, and personalized recommendations.

Traditional approaches to image classification have primarily relied on convolutional neural networks (CNNs) trained on large labeled datasets. While these methods have achieved state-of-the-art performance in many cases, they often require extensive computational resources and large amounts of labeled data. Moreover, these models are typically specialized for specific tasks, lacking the flexibility to generalize across different domains or incorporate multimodal information.

Recent developments in multimodal models, particularly those that bridge the gap between vision and language, offer a promising alternative. One such model is CLIP (Contrastive Language-Image Pretraining), which leverages large-scale pretraining on image-text pairs to learn a joint embedding space for both modalities. By training on a vast corpus of images and their corresponding textual descriptions, CLIP can capture the semantic relationships between visual and textual data, enabling zero-shot transfer learning and robust performance across various tasks.

In this report, we explore the application of CLIP for the task of fashion image classification. Specifically, we investigate two approaches for fine-tuning the CLIP model on a custom fashion dataset. The first approach involves adding a softmax layer after the CLIP model and fine-tuning it using only the image data. This method leverages the pre-trained image encoder while adapting it to the specific classification task. The second approach involves fine-tuning the CLIP model with both image and text data. In this method, we compute the similarity between all possible labels and the generated text embeddings to predict the class of the label. This approach exploits the inherent multimodal capabilities of CLIP, potentially improving classification performance by incorporating textual information.

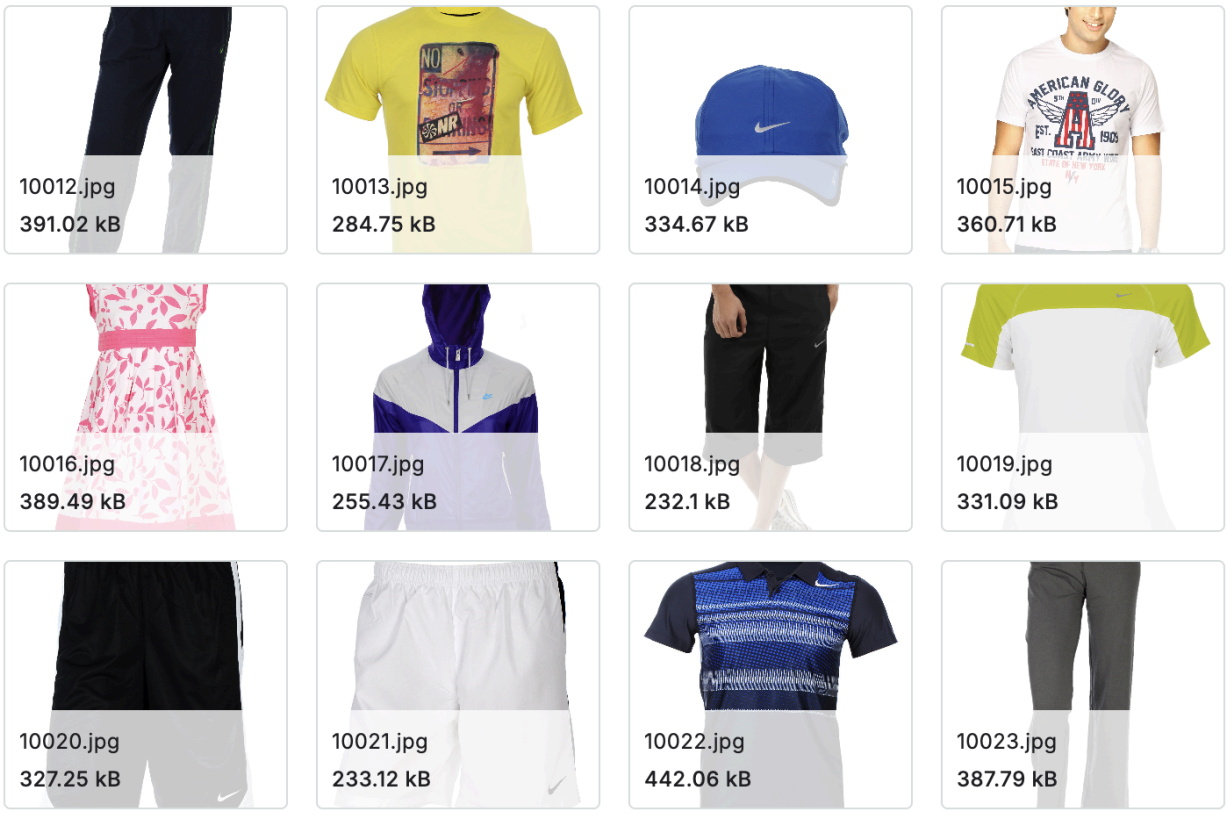
The following sections will detail the methodologies, experiments, and results of these two approaches, highlighting the effectiveness of CLIP in the context of fashion image classification.

2. Dataset

This study utilizes the Fashion Product Images Dataset from Kaggle. The dataset contains 35G high-resolution (1200 x 1080) images of fashion products, each linked to detailed metadata, including multiple label attributes and descriptive text. This study uses the images and their subcategory information as the classification label.

The subcategories have 45 different labels, some example labels are 'Accessories', 'Apparel Set', 'Bags', 'Bath and Body', 'Beauty Accessories', 'Belts', 'Bottomwear', 'Cufflinks', 'Dress', 'Eyes', 'Eyewear', 'Flip Flops', 'Fragrance', 'Free Gifts', 'Gloves', 'Hair', 'Headwear', 'Home Furnishing', etc.

Some sample images are like the following.



3. Experiment Details

A. Machine: A100_80G

B. Data Preprocessing: Preprocess from OpenAI CLIP Library

C. Data Split: Used 10% of the data, of which 80% for training, and 20% for validation.

D. Model: CLIP Model ViT-B/32

E. Optimizer: AdamW

F. Parameters:

- Batch Size: 32
- Learning Rate: 0.0001
- Epoch: 30 & 50

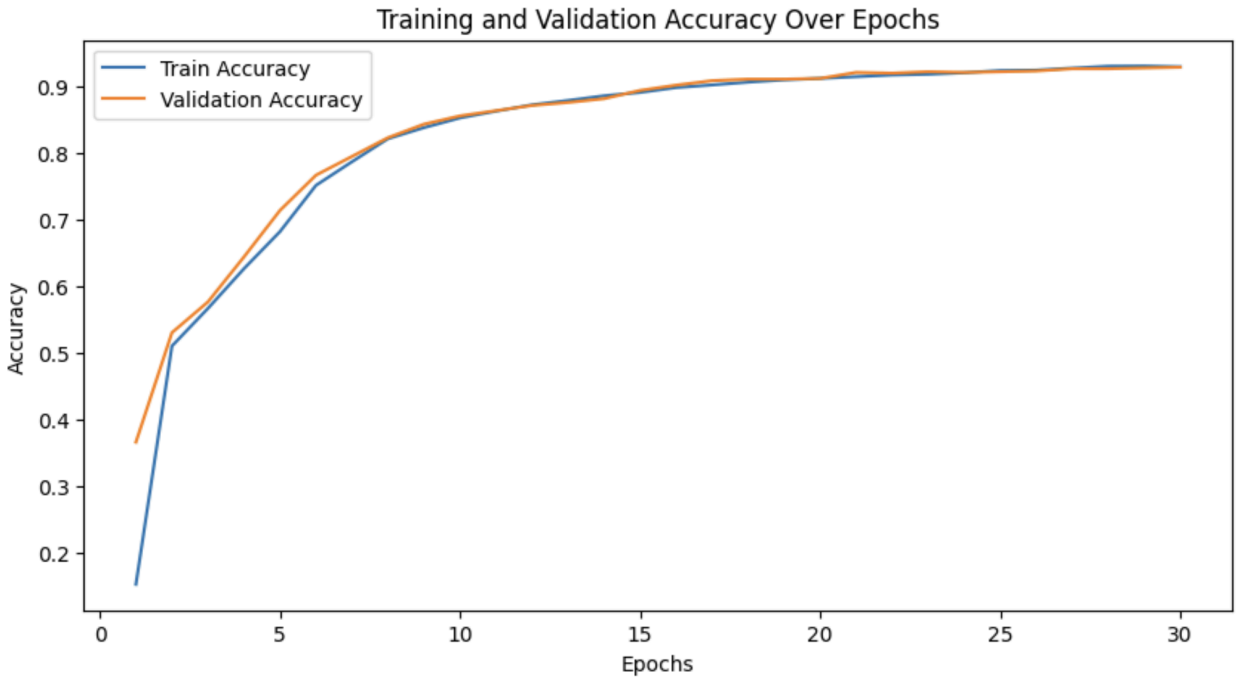
4. Experiment Result

Experiment 1:

Training Loss: 0.0094, Accuracy: 92.97%;

Validation Loss: 0.0096, Accuracy: 92.91%.

```
Epoch [1/30], Train Loss: 0.1058, Train Accuracy: 0.1525, Val Loss: 0.0893, Val Accuracy: 0.3660
Epoch [2/30], Train Loss: 0.0777, Train Accuracy: 0.5103, Val Loss: 0.0676, Val Accuracy: 0.5304
Epoch [3/30], Train Loss: 0.0604, Train Accuracy: 0.5668, Val Loss: 0.0549, Val Accuracy: 0.5766
Epoch [4/30], Train Loss: 0.0502, Train Accuracy: 0.6264, Val Loss: 0.0467, Val Accuracy: 0.6441
Epoch [5/30], Train Loss: 0.0430, Train Accuracy: 0.6821, Val Loss: 0.0406, Val Accuracy: 0.7140
Epoch [6/30], Train Loss: 0.0377, Train Accuracy: 0.7516, Val Loss: 0.0358, Val Accuracy: 0.7669
Epoch [7/30], Train Loss: 0.0336, Train Accuracy: 0.7871, Val Loss: 0.0318, Val Accuracy: 0.7950
Epoch [8/30], Train Loss: 0.0300, Train Accuracy: 0.8217, Val Loss: 0.0286, Val Accuracy: 0.8232
Epoch [9/30], Train Loss: 0.0271, Train Accuracy: 0.8383, Val Loss: 0.0259, Val Accuracy: 0.8435
Epoch [10/30], Train Loss: 0.0248, Train Accuracy: 0.8529, Val Loss: 0.0237, Val Accuracy: 0.8559
Epoch [11/30], Train Loss: 0.0229, Train Accuracy: 0.8630, Val Loss: 0.0218, Val Accuracy: 0.8637
Epoch [12/30], Train Loss: 0.0209, Train Accuracy: 0.8726, Val Loss: 0.0202, Val Accuracy: 0.8716
Epoch [13/30], Train Loss: 0.0195, Train Accuracy: 0.8790, Val Loss: 0.0188, Val Accuracy: 0.8761
Epoch [14/30], Train Loss: 0.0183, Train Accuracy: 0.8861, Val Loss: 0.0176, Val Accuracy: 0.8818
Epoch [15/30], Train Loss: 0.0171, Train Accuracy: 0.8911, Val Loss: 0.0166, Val Accuracy: 0.8941
Epoch [16/30], Train Loss: 0.0163, Train Accuracy: 0.8985, Val Loss: 0.0157, Val Accuracy: 0.9020
Epoch [17/30], Train Loss: 0.0153, Train Accuracy: 0.9024, Val Loss: 0.0149, Val Accuracy: 0.9088
Epoch [18/30], Train Loss: 0.0145, Train Accuracy: 0.9066, Val Loss: 0.0142, Val Accuracy: 0.9110
Epoch [19/30], Train Loss: 0.0139, Train Accuracy: 0.9100, Val Loss: 0.0136, Val Accuracy: 0.9110
Epoch [20/30], Train Loss: 0.0137, Train Accuracy: 0.9122, Val Loss: 0.0130, Val Accuracy: 0.9122
Epoch [21/30], Train Loss: 0.0128, Train Accuracy: 0.9148, Val Loss: 0.0126, Val Accuracy: 0.9212
Epoch [22/30], Train Loss: 0.0123, Train Accuracy: 0.9173, Val Loss: 0.0121, Val Accuracy: 0.9200
Epoch [23/30], Train Loss: 0.0118, Train Accuracy: 0.9184, Val Loss: 0.0117, Val Accuracy: 0.9223
Epoch [24/30], Train Loss: 0.0114, Train Accuracy: 0.9207, Val Loss: 0.0113, Val Accuracy: 0.9212
Epoch [25/30], Train Loss: 0.0114, Train Accuracy: 0.9241, Val Loss: 0.0109, Val Accuracy: 0.9223
Epoch [26/30], Train Loss: 0.0108, Train Accuracy: 0.9246, Val Loss: 0.0107, Val Accuracy: 0.9234
Epoch [27/30], Train Loss: 0.0103, Train Accuracy: 0.9277, Val Loss: 0.0104, Val Accuracy: 0.9268
Epoch [28/30], Train Loss: 0.0101, Train Accuracy: 0.9305, Val Loss: 0.0101, Val Accuracy: 0.9268
Epoch [29/30], Train Loss: 0.0097, Train Accuracy: 0.9308, Val Loss: 0.0098, Val Accuracy: 0.9279
Epoch [30/30], Train Loss: 0.0094, Train Accuracy: 0.9297, Val Loss: 0.0096, Val Accuracy: 0.9291
Training complete.
```



Experiment 2:

Training Loss: 0.0487;

Validation Loss: 0.0656, Accuracy: 73.20%.

```
Epoch [1/50], Train Loss: 0.1060, Val Loss: 0.1128, Val Accuracy: 0.1768
Epoch [2/50], Train Loss: 0.0920, Val Loss: 0.0871, Val Accuracy: 0.3818
Epoch [3/50], Train Loss: 0.0815, Val Loss: 0.0773, Val Accuracy: 0.3570
Epoch [4/50], Train Loss: 0.0735, Val Loss: 0.0741, Val Accuracy: 0.3435
Epoch [5/50], Train Loss: 0.0697, Val Loss: 0.0689, Val Accuracy: 0.5473
Epoch [6/50], Train Loss: 0.0658, Val Loss: 0.0670, Val Accuracy: 0.6813
Epoch [7/50], Train Loss: 0.0639, Val Loss: 0.0679, Val Accuracy: 0.6070
Epoch [8/50], Train Loss: 0.0604, Val Loss: 0.0675, Val Accuracy: 0.6351
Epoch [9/50], Train Loss: 0.0584, Val Loss: 0.0693, Val Accuracy: 0.5124
Epoch [10/50], Train Loss: 0.0573, Val Loss: 0.0694, Val Accuracy: 0.6971
Epoch [11/50], Train Loss: 0.0554, Val Loss: 0.0672, Val Accuracy: 0.6115
Epoch [12/50], Train Loss: 0.0541, Val Loss: 0.0650, Val Accuracy: 0.6318
Epoch [13/50], Train Loss: 0.0522, Val Loss: 0.0773, Val Accuracy: 0.5349
Epoch [14/50], Train Loss: 0.0540, Val Loss: 0.0689, Val Accuracy: 0.6881
Epoch [15/50], Train Loss: 0.0522, Val Loss: 0.0766, Val Accuracy: 0.6014
Epoch [16/50], Train Loss: 0.0504, Val Loss: 0.0712, Val Accuracy: 0.6149
Epoch [17/50], Train Loss: 0.0503, Val Loss: 0.0749, Val Accuracy: 0.6498
Epoch [18/50], Train Loss: 0.0503, Val Loss: 0.0687, Val Accuracy: 0.6678
Epoch [19/50], Train Loss: 0.0491, Val Loss: 0.0708, Val Accuracy: 0.6678
Epoch [20/50], Train Loss: 0.0489, Val Loss: 0.0697, Val Accuracy: 0.6734
Epoch [21/50], Train Loss: 0.0495, Val Loss: 0.0694, Val Accuracy: 0.7095
Epoch [22/50], Train Loss: 0.0482, Val Loss: 0.0746, Val Accuracy: 0.7061
Epoch [23/50], Train Loss: 0.0493, Val Loss: 0.0712, Val Accuracy: 0.7331
Epoch [24/50], Train Loss: 0.0480, Val Loss: 0.0693, Val Accuracy: 0.7252
Epoch [25/50], Train Loss: 0.0488, Val Loss: 0.0708, Val Accuracy: 0.7095
...
Epoch [48/50], Train Loss: 0.0470, Val Loss: 0.0722, Val Accuracy: 0.7331
Epoch [49/50], Train Loss: 0.0479, Val Loss: 0.0724, Val Accuracy: 0.7106
Epoch [50/50], Train Loss: 0.0472, Val Loss: 0.0743, Val Accuracy: 0.7477
```

5. Conclusion and Lessons Learned

In this project, we explored two approaches to fine-tuning the CLIP model for fashion image classification. The first experiment, which involved adding a softmax layer after the CLIP model and fine-tuning it with only image data, yielded a high accuracy of 92% after 30 epochs. This result demonstrates the effectiveness of directly optimizing the classification layer for the task.

The second experiment utilized both image and text data, leveraging CLIP's multimodal capabilities by computing the similarity between text embeddings and images. However, this approach resulted in a lower accuracy of approximately 73%.

The disparity in performance between the two experiments highlights the strengths and limitations of each method. While direct optimization of classification-specific layers (as in the first experiment) proved to be more effective for this task, the second experiment showed the potential of integrating multimodal data, albeit with more challenges in achieving similar performance. Ultimately, the experiments show that while image-based classification benefits from clear supervision, multimodal learning introduces additional complexity, which can require further tuning for optimal results.

Some lessons learned are as follows.

Direct Optimization Yields Better Classification Results: The first experiment's high accuracy suggests that directly optimizing the final classification layer allows the model to quickly adapt to the task, especially when fine-tuning with task-specific data. This approach leverages the pre-trained image encoder effectively and can provide fast convergence in a classification task.

Multimodal Learning is Complex: The second experiment, which involved learning image-text similarities, did not perform as well. This highlights the challenges of training models with both image and text data, where alignment of embeddings across modalities can be difficult. Learning a shared representation for both image and text requires careful tuning and a better understanding of how each modality contributes to the classification task.

Classification Loss Provides Stronger Task-Specific Signals: The direct classification loss in the first experiment gives the model clear, strong signals to differentiate between categories. In contrast, the second experiment's contrastive-style loss function was more indirect, focusing on similarities between images and text rather than directly optimizing classification performance.

Potential for Hybrid Approaches: A hybrid approach, where the model initially learns image-text similarities and is then fine-tuned with a task-specific classification layer, might provide the best of both worlds—leveraging the richness of multimodal data while benefiting from direct classification optimization.

In future work, experimenting with different training techniques, fine-tuning strategies, and improving data quality could further improve the performance of multimodal models in complex classification tasks.

7. Reference and Appendix

1. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. and Krueger, G., 2021, July. Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PMLR.

2. Source Code on Github: CLIP-FP

3. Dataset: Fashion Product Images Dataset