Gradient Descent into Style: An Investigation into Fashion Parsing, Image Segmentation, and Generalized Boundaries between Fashion Styles

Lucas Papadopoulos

Kennesaw State University College of Computing and Software Engineering, CS 4732 Mr. Maxwell Bradley, Marietta, GA, 12/10/25 {lpapadop@students.kennesaw.edu}

Abstract. Recent advances in computer vision have significantly improved automated fashion identification, yet most existing models rely on supervised learning with predefined category labels that fail to capture the nuanced diversity and evolving nature of fashion styles. This work investigates an unsupervised approach to fashion categorization through adaptive clustering.The proposed architecture employs a frozen EfficientNet-B3 backbone pretrained on ImageNet as a feature extractor, followed by a trainable two-layer projection head that produces 256-dimensional embeddings optimized for clustering. These embeddings feed into an adaptive MiniBatchKMeans clustering module with novelty detection capabilities. The system dynamically expands the number of clusters (k) when samples consistently fail to fit existing categories, as determined by dual-threshold novelty detection combining distance-based (mu + 2*sigma) and confidence-based (less than 0.6) criteria. The model alternates between cluster reassignment via k-means and supervised learning of cluster predictions via cross-entropy loss.Evaluated on the DeepFashion2 dataset (40,940 training images, 25,988 validation images), the model gave a training accuracy of 51.40% (predicting self-discovered cluster assignments across 17 categories) and a silhouette score of 0.1493, indicating moderate cluster cohesion with room for improvement. While these metrics suggest weak to moderate cluster separation, they represent a meaningful baseline for unsupervised fashion categorization on this scale.

Keywords: Fashion Identification · Computer Vision · EfficientNet-B3 · Adaptive K-Means Clustering

1 Introduction

For everyday dress, individuals often operate within constrained stylistic frameworks shaped by cultural norms. Fashion can be understood as the collective set of clothing compositions deemed typical or desirable within prevailing trends and customs. Characteristically, the Fashion set excludes large sets of clothing schema based on unspoken or subjective metrics. If a party should choose to wear something that is not included in the current Fashion set for a given year,

that party risks social alienation, professional exclusion, and diminished social belonging. Therefore, it is the imperative of researchers to categorize, metricize, and classify compositions of clothing that map to the grand Fashion set. Providing an ontology and classification system for identifying similar fashion compositions, fashion genres would enable clarity of artistic expression, cultural preservation of aesthetic trends, and improved edge case identification of clothing styles. It was this researcher's goal to provide a functional tool that not only classifies fashion genres in real time but also aligns them with prevailing fashion genres within a generalized subset of the broader Fashion set. A proper fashion genre classification model would be able to ingest photos of people's clothing of any type or composition, and identify what general group the clothes fall under.

1.1 Differences From M1/M2 Methodology

My original proposal described a fairly complex architecture that extracted features using EfficientNet-B3, fed them into a GNN to learn relationships, and output a probabilistic style vector to be shown on a Grad-Cam overlay. The fundamental problem with this architecture lies with the concept: any ontology that I could develop and impose on DeepFashion2 would provide an incomplete understanding of fashion genres. In fact, the utility provided would only be as useful as the public consensus on the general ontology.

Fashion datasets are not stable datasets. New classes can be introduced into the overall corpus of fashion styles with no precedent, no foundation in any other style. Similar styles may be classified according to similar texture, composition, orientation, etc., but they are not explicitly beholden to a generalized class structure according to a strict rule set. It was folly to expect to create some ontology and impose it onto a dataset without care as to how much it models reality. It was similarly useless to try to take utility out of a snapshot of the correlative features in a dataset: they provide extremely limited insights about how the corpus is structured (or how it evolves). Even in combination (EDA/Clustering + CNN classification), the overall utility of these models boils down to a ruleset with subjective value. It's not set in stone; there are no ground truths in fashion. You could publish rules on how clothing should be organized, and it can be rejected based on cultural values.

But, if there isn't a statistical analysis or at least some analysis into the quality of the classifications we have now, we're just guessing. That's why adaptive clustering analysis is useful here. The intent of the new approach was to create a system that could potentially accept new continuous fashion data streams and detect novel (unassociated with any other genre) classes that come into the stack. If we can discover clothing feature contributions and create new groupings as they're introduced into the population, we can get a better picture of what people are wearing.

2 Personal Value Statement

I thought this would be a valuable project to build because it was cool and I was passionate about it. Making this model had been a dream of mine for a couple of years. It was the first idea I came up with when my friends and I founded the AI Club here at KSU. Completing this project was my New Year's resolution (along with losing weight). I really think there is a moral imperative for people to have creative clarity and security in their own sense of fashion. It is unfair that people have to consult influencers, read stupid magazines, and listen to unreliable/uninformed friends to discern clothing aesthetics. In addition, this topic is generally hot in research and commercial applications. All of my sources are from the last five years. The domain is sexy, and the fashion is sexy. In the most professional sense, I think that this project has sexual appeal. For those reasons, I wanted to develop this project.

Overall, the results of this project do not speak to the value that the project provides. This project will help to highlight to employers that I was not afraid to pivot off a flawed idea (although I pivoted way too late). I was able to identify the underlying problem with imposing an ontology onto data: no one told me why my original idea was flawed or what could be changed about it. I came to these conclusions independently. I tried a new strategy to achieve my goal. It was a failure, but I know what needs to be improved in order to continue with the project. Starting from nothing, I think I did a pretty good job. Now I have a baseline to work off if I want to develop this idea in the future.

3 Literature Review

This section is meant to give a brief history of the progress made in fashion identification and explain the major research papers that will be used (directly or indirectly) for project development. After reading this review, my objective is that you will understand the call for further research on this topic and understand the current techniques used for fashion identification.

3.1 Comprehensive Domain Overview: Fashion Meets Computer Vision

Fashion Meets Computer Vision: A Survey[3] provides a comprehensive survey of research at the intersection of fashion and computer vision. Cheng et al. broadly define all of the interrelated fashion identification sub-fields and describe the common deep learning methodologies prevalent within them. My proposed fashion identification model falls into two main subfields of fashion identification: fashion parsing and fashion analysis (style learning in fashion analysis). Both fields are relatively mature in the scope of image identification research. The following subsections describe these two fields.

Fashion Parsing Fashion parsing is a form of "semantic segmentation". It involves doing "human parsing with clothes classes". Fashion parsing models seek to partition images of people into finely labeled semantic clothing regions. Fashion parsers use image segmentation techniques focusing on pixel-label predictions or color category predictions. According to Cheng et al., three main requirements distinguish fashion parsing from regular object/scene segmentation: it requires "higher level judgment based on the semantics of clothing, the deforming structure within an image, and a potentially large number of classes." The hardest of these requirements to model around is the deformative structure of clothes. Clothing is weird. There will be a lot of intra-class variance (things look similar to each other). These models often leverage some type of context, like pose or texture. One strategy is introducing a definable ontology in the form of a tree-like structure of "parselets", introduced by the **Deformable Mixture Parsing Model (DMPM)**[4]. That model compacts "semantically meaningful, often locally homogeneous segments" into nodes in a big tree, then "assigns geometric differences (distance, angle, scale) between parent and child nodes" (and then aggregates their scores).

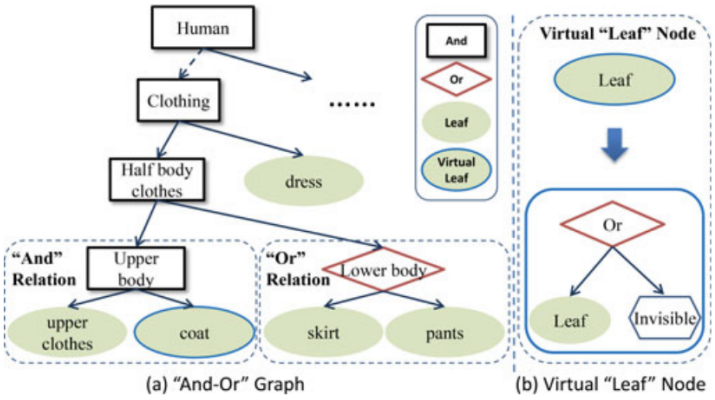

Fig. 1: AND/OR Structure of the DMPM. Great inspiration for both Fashion Parsing and Fashion Analysis

GCNNs, or really any GNN-type representations of pixels, are super common for this type of problem. A graph post-processing model that is used a lot is a CRF, a **Conditional Relation Field**. CRFs are good at modeling "spatial and contextual relationships in pixel-wise semantic segmentation". But perhaps the most common strategy among all of these fashion parser pipelines is using **K-Nearest Neighbor (KNN) matching** as a heuristic baseline for matching clothes (usually Euclidean or Cosine distance). Creating decision boundaries using SVM or KNN is a common theme across all methodologies for fashion parsing.

Fashion Analysis Fashion Analysis involves revealing insights from what people choose to wear using deep learning techniques. It's broken into three major

sub-sub fields: attribute recognition, style learning, and popularity prediction. While attribute learning, determining which elements of clothing are associated with attributes among a set of n attributes, may play a pivotal part in this project, I am most concerned with style learning. Style Learning concerns itself with **analyzing "discriminative features for different styles"**.

How Much Did I Actually Use the Literature The idea to extract features using a robust CNN backbone and then feed these features into some classification structure came from the Sportive Fashion Trend Model, in general. I didn't end up using KNN for clustering, as was common in many of the fashion parsing papers. I used the DeepFashion2 Dataset from the Sportive paper. That's it: most fashion analysis is concerned with fashion segmentation or fashion prediction based on datasets that are unrepresentative of contemporaneous fashion. A Vogue fashion dataset or even FashionCLIP is useful for specific domains. DeepFashion and DeepFashion2 were the only datasets that represented the diversity and structure of fashion on the street (without the label of "fast fashion").

3.2 Sportive Fashion Trend Model

Sportive Fashion Trend Reports: A Hybrid Style Analysis Based on Deep Learning Techniques[1] is a cool paper which leverages a "Multi-Label Graph Convolutional Networks (ML-GCN)" to identify fashion general fashion styles. This particular model identifies co-occurence patterns among overlapping styles. It was a great paper because it aligned perfectly with my overall goal of assigning clothes to a pre-determined ontology using multi-label classification techniques. Too bad that I couldn't actually use the model in which this paper was based off. I should wonder if anyone has reproduced their results.

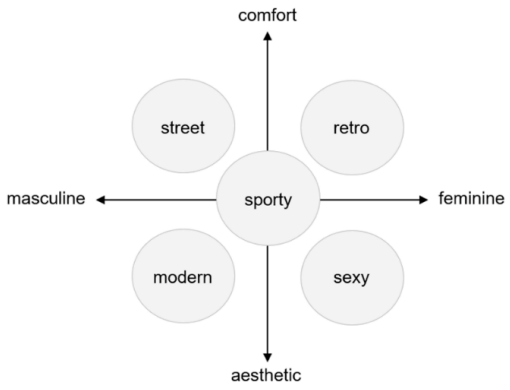

Figure 1. Theoretical framework for the hybrid style in sportive fashion.

Fig. 2: An Example of a Basic Ontological Framework

For label identification, GNN's seem to be the way to go. They got really impressive results from this type of model. Sportive's Top-1 classification accuracy

was 73.23 with ML-GCN and outperformed CNN results. The model captures the probabilistic expected output that I really wanted, too. However, as stated previously, the labels they have created are largely useless as identifiers for a broader classification system.

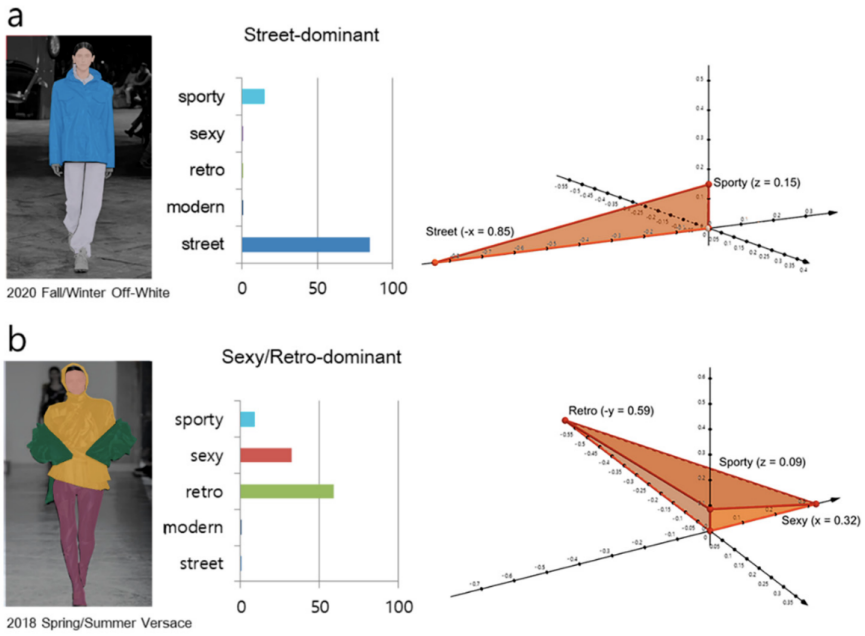

Fig. 3: Realization of the Sportive Ontology for Classification. Pretty impressive!

3.3 DeepFashion and DeepFashion2

DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images[5] This dataset is comprised of different street shots of Koreans, as well as of just random clothes being laid out.

Fig. 4: Some data points from the DeepFashion dataset

Fig. 5: Example 1 of Typical Datapoint

Fig. 6: Example 2: Just Clothes Laid Out

4 Technical Description/Methodology

4.1 General Architecture

Images are preprocessed through data augmentation transforms (resize, horizontal flip, color jitter, rotation for training; resize only for validation) and normalized using ImageNet statistics. These preprocessed images are fed into a pretrained, frozen EfficientNet-B3 backbone (trained on ImageNet. Check the docs for EfficientNet-B3). From there, the feature embeddings gathered from EfficientNet are feed into a projection head that transforms the embeddings into clusterable inputs. From there, we cluster and adapt for training using the DeepCluster approach [2]

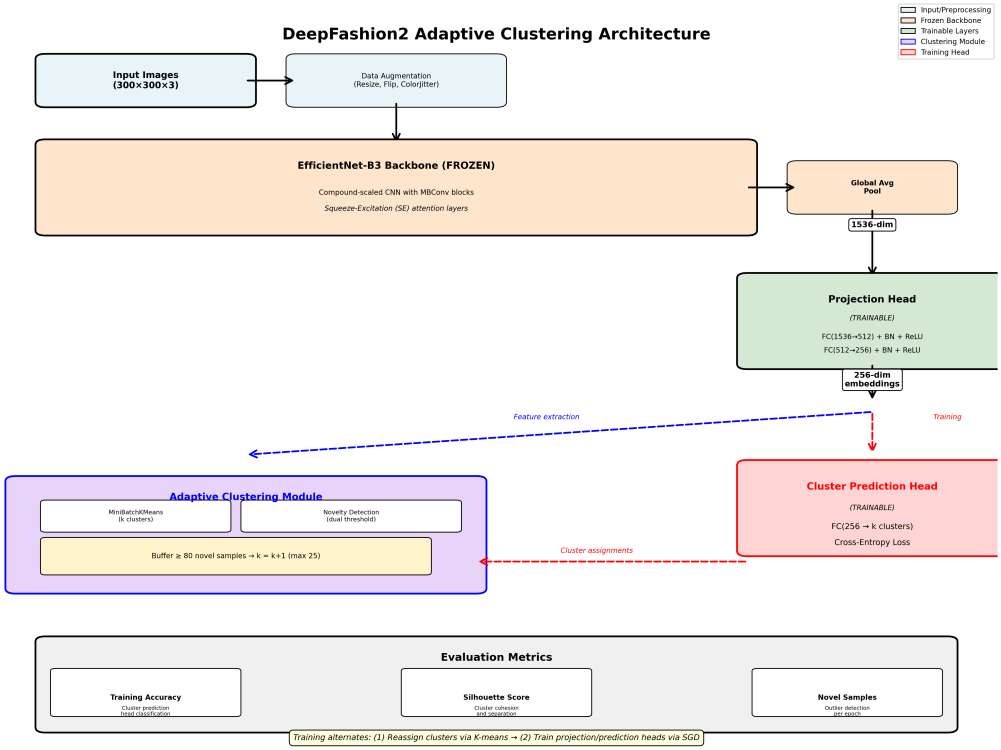

Fig. 7: General Architecture (To be Explained)

4.2 Data Considerations

I didn't use any other dataset than the DeepFashion2 dataset. I trained at **40940** samples and used **25988** in my validation set. The quality of the images was variable; at least they all had pretty decent resolution. I standardized the images though before taking them in. Here are the various transforms I did on the data:

```
train_transform = transforms.Compose([
    transforms.Resize((config.IMAGE_SIZE, config.IMAGE_SIZE)),
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1),
    transforms.RandomRotation(15),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])

val_transform = transforms.Compose([
    transforms.Resize((config.IMAGE_SIZE, config.IMAGE_SIZE)),
    transforms.ToTensor(),
```

```
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

4.3 Feature Extraction Backbone

The backbone extracts hierarchical convolutional features through EfficientNet-B3's compound-scaled architecture, which employs Mobile Inverted Bottleneck (MBConv) blocks with Squeeze-and-Excitation (SE) layers. SE layers provide channel-wise attention by learning channel interdependencies, allowing the network to emphasize informative features and suppress less useful ones. This is distinct from Vision Transformer (ViT) architectures - EfficientNet-B3 is a pure CNN with no self-attention mechanisms.

The convolutional feature maps undergo global average pooling, reducing spatial dimensions to produce 1536-dimensional feature vectors. The EfficientNet-B3 backbone remains frozen throughout training, serving as a fixed feature extractor pretrained on ImageNet.

4.4 Projection Head

The 1536-dimensional features pass through a trainable projection head: a two-layer multilayer perceptron (MLP) with the following architecture:

FC($1536 \to 512$) $\to$ BatchNorm $\to$ ReLU $\to$ Dropout(0.2) $\to$ FC($512 \to 256$) $\to$ BatchNorm $\to$ ReLU

This produces 256-dimensional embeddings that are optimized specifically for clustering. The projection head is crucial as it learns a representation space where semantic similarities are more clearly expressed, making clustering more effective.

4.5 Adaptive Clustering Module

The 256-dimensional embeddings feed into an adaptive clustering pipeline using MiniBatchKMeans. MiniBatchKMeans is a scalable variant of k-means that processes data in mini-batches (batch_size=512).

Starting Configuration:

- Initial k = 10 clusters
- Maximum k = 25 clusters
- Testing progressed through 17 clusters

The clustering module performs standard k-means optimization: assigning each sample to its nearest centroid and iteratively updating centroids to minimize intra-cluster variance. This is fairly typical for a K-means approach

4.6 Novelty Classes (Important to Know!)

After each clustering iteration, the system performs novelty detection using a dual-threshold approach:

1. **Distance-Based Threshold**: Samples are flagged if their distance to the nearest centroid exceeds: threshold = mu + 2*sigma where mu is the mean distance and sigma is the standard deviation across all samples. This is typical for the normalization step in K-means.

2. **Confidence-Based Threshold**: Confidence is calculated for each sample as:

```
confidence = 1 - (min_distance / max_distance)
```

Where min_distance = Euclidean distance to the nearest centroid and max_distance = Maximum distance observed across all sample-centroid pairs. High confidence ($\rightarrow$ 1.0) means the sample IS very close to its assigned centroid relative to the overall distance scale. Low confidence ($\rightarrow$ 0.0) means the sample is far from its nearest centroid, even at the scale of the entire feature space; it's a potential outlier. The threshold (0.6) I am using flags samples with confidence ¡ 0.6 as novel.

Average confidence across all samples provides a global metric of cluster quality. Declining average confidence suggests the current k-value may be inadequate for capturing the data's structure.

A sample is marked as novel if EITHER condition is met:

```
novel = (distance > mu + 2*sigma) OR (confidence < 0.6)
```

4.7 Adaptive Clustering and Training

Novel sample indices accumulate in a buffer. When the buffer reaches 80 samples, the system expands by one cluster (k $\rightarrow$ k+1) and retrains on all features with the new k value. The buffer is then reset. This allows the model to dynamically adapt to discover new fashion categories as patterns emerge that don't fit existing clusters. We go until k=25 or there are no more novel samples detected (or until I get tired of running all this on google collab).

Then, we use that DeepCluster strategy we talked about before.

Every reassign frequency epoch, we

- Extract 256-dim embeddings for all training images (with frozen backbone)
- Run MiniBatchKMeans clustering on embeddings
- Perform novelty detection and potentially expand k
- Assign each image a pseudo-label based on its cluster assignment

Then, using the clusters we made, we make supervised predictions.

- A cluster prediction head (FC layer: 256 $\rightarrow$ k) maps embeddings to k logits
- Cross-entropy loss trains the projection head and prediction head to classify images according to their cluster pseudo-labels
- We use an ADAM optimizer because ADAM is OP for hyper-params. We use it with learning rate 1e-4 and weight decay 1e-5

– The EfficientNet backbone remains frozen throughout

The projection head is intended to learn discriminative features that facilitate better clustering, while the clustering assignments guide the learning process. The model gradually discovers and refines fashion categories in an unsupervised manner.

5 Evaluation

For evaluation, I used three main metrics: the accuracy, silhouette score, and the number of data points that fit into the novelty class.

Let us now discuss the metrics used for this report. First, training accuracy. The accuracy measures how accurately the cluster prediction head assigns samples to their k-means cluster labels. Higher accuracy indicates that the projection head has learned to produce embeddings that match cluster structure. It also indicates that cluster assignments are consistent and learnable It's important to note that **this is NOT measuring classification accuracy on ground-truth labels, but rather the model's ability to predict self-discovered cluster assignments**.

Next, the silhouette score represents the cohesiveness of the created clusters (how close data points are to centroids in K-means) compared with data points in other clusters. It quantifies cluster quality by measuring intra-cluster **cohesion**: how close samples are to others in their cluster and **separation**: how far samples are from samples in other clusters. Silhouette scores greater than 0.5 mean that we're getting strong, well-separated clusters. Scores from 0.2 to 0.5 mean that we have a reasonable cluster structure. Scores less than 0.2 mean that we have weak or overlapping clusters. Finally, scores less than 0 mean that samples may be assigned to wrong clusters. These metrics evaluate the utility of the model's boundary creation.

Finally, we record novel classes to understand how many samples don't fit well into existing clusters: this is pivotal for understanding how robust the model is in general. The novel class number tracks the number of samples flagged as novel per epoch. It tracks whether the model needs to expand to accommodate new patterns. It also tracks the stability of the current cluster configuration. A consistently high novel sample count suggests k may need to increase, while declining counts indicate the model is converging to a stable configuration.

The goal of this architecture is to:

1. **Maximize training accuracy**: Indicating the projection head effectively learns cluster-discriminative features
2. **Maximize silhouette scores**: Ensuring discovered clusters are cohesive and well-separated, representing genuine fashion categories
3. **Adaptively adjust k clusters**: Using novelty detection to discover the appropriate number of fashion categories present in the dataset, neither over-fragmenting nor over-generalizing the data

4. **Enable unsupervised fashion discovery**: Automatically identifying fashion categories without requiring labeled data, allowing the model to adapt to new fashion trends and styles as they appear in the dataset

5.1 Quantitative

The model successfully expanded from k=10 to k=18 clusters over 8 epochs, achieving a peak silhouette score of 0.1493 at k=17 and training accuracy of 51.40% at k=14. While the silhouette score of 0.15 indicates weak cluster separation typical of complex unsupervised fashion categorization, the average confidence of 0.82 demonstrates that most samples exhibit clear cluster membership. That 0.82 numbers means that, on average, samples are only 17.7% of max distance from their nearest centroid. Most samples fit reasonably well into their assigned clusters at 0.82. The model's ability to achieve 51% accuracy across 17 self-discovered categories (9× better than random) without ground truth labels validates the adaptive clustering approach.

But, you know, it still kind of sucks. That accuracy is disappointing. And the clusters need to be better

Fig. 8: Best Silhouette Score. Confidence Still at 0.80

5.2 Qualitative

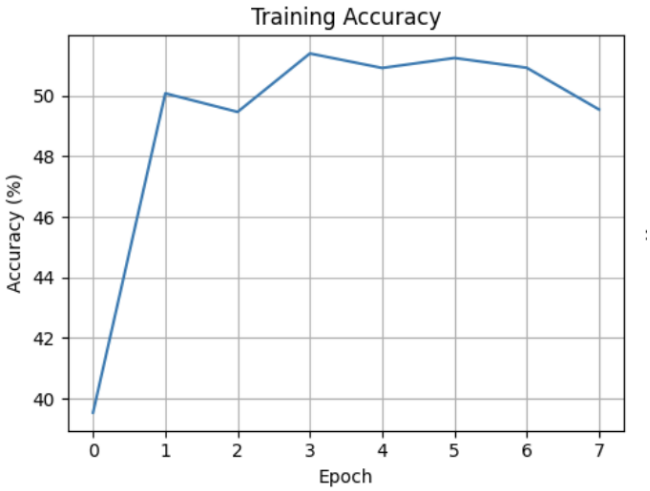

Fig. 9: Training Accuracy over 8 Epochs

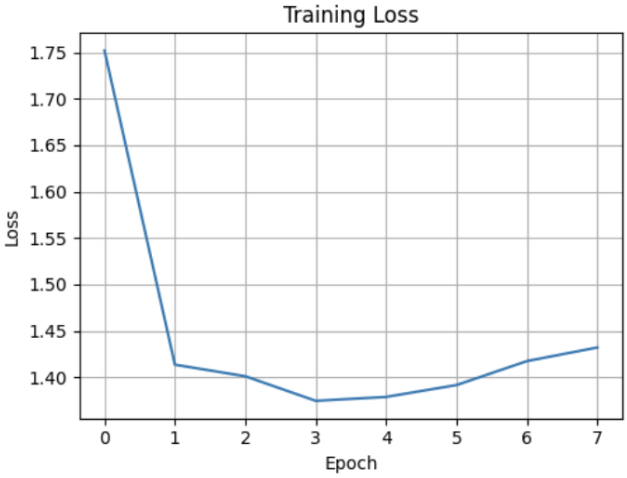

Fig. 10: Training Loss over 8 Epochs

6 Conclusion

6.1 Result Implications and Future Work

The observed cluster overlap (silhouette score of 0.15) suggests that fashion categories exist on a continuum rather than in discrete partitions. That's what I suspected; it was why I needed to pivot in the first place. Future work could leverage these results to inform a Graph Neural Network (GNN) approach that explicitly models inter-cluster relationships rather than enforcing hard boundaries. Specifically, the 256-dimensional embeddings could be encoded as nodes

in a graph, with edge weights representing pairwise similarities between samples or cluster centroids. A GNN architecture (e.g., Graph Convolutional Network or Graph Attention Network) could then learn soft cluster membership, feature importance (identifying which embedding dimensions contribute most to discriminating between overlapping fashion categories through attention mechanisms) and hierarchical structure (overarching meta-clusters).

Of course, prior to GNN construction, refinement of the novelty detection thresholds or additional clustering iterations could reduce the 4% outlier rate, ensuring the graph structure is built primarily on stable, well-clustered samples rather than noise.

But once we hammer down a good cluster structure, a GNN could learn these relational structures. The discovered cluster relationships could inform targeted manual labeling efforts. Rather than labeling all 40,000+ samples, annotators could focus on cluster centroids and boundary samples where multiple clusters intersect, using the GNN's learned similarities to propagate labels efficiently. This semi-supervised approach would convert unsupervised cluster discoveries into interpretable fashion taxonomies while preserving the nuanced relationships the model identified. This is WAY better than just manually labeling based on some stupid ontology, or just binning based on "expert opinions".

Beyond specifics though, here's where all of this can improve:

- EfficientNet-B3 is trained on ImageNet. We need a better backbone trained on segmenting clothing specifically. Fine-tuning the EfficientNet backbone to learn fashion-specific features rather than relying solely on ImageNet representations would be an obvious improvement
- We need to look into the images that are close to a centroid, and start to make some inferences. Qualitative analysis of discovered clusters is needed to understand their semantic coherence.
- We need to explore alternative embedding dimensions and projection architectures. Maybe this one just sucked in general.
- We might need to investigate hierarchical clustering approaches to capture both coarse-grained and fine-grained fashion distinctions. Maybe the data is way more hierarchical than even manual labelers could find out.

6.2 The Wrap-Up

Here I will answer some more general questions (not relating to the model). This is at the behest of Mr. Maxwell Bradley, instructor at KSU.
Q: What's the big takeaway, the main message of your work?
A: Don't try to impose your own ontology onto a dataset, especially if the ontology is made basically ex-nihilo. It doesn't matter how much research you do into underlying patterns. You **SHOULD NEVER** use computer science as some authority, some ethos for forcing a rule set down people's throats. That would make you a HACK. Your set of beliefs about aesthetics are not superior to all other beliefs about aesthetics. Even the most perfect rule set will become

obsolete. It's a hard pill to swallow; I would've needed therapy to understand this concept before this semester.
Also, this work shows that there are learnable patterns to the ways that we dress. There are underlying connections between different fashion genres. This is a valuable starting point for more research.
Q: Why did things turn out this way? Why isn't there more progress?
A: This project was way too ambitious. I got really bogged down into trying to understand the underlying rules of fashion before actually doing any computer science work. Of course, the benefit of the research (and of the stalling) is that I was able to identify the problems with my original architecture before getting any results. The downside of all this is that there's barely any real progress. I had too much on my plate, and I should've been training in September. For that, I apologize. I should've had the foresight to recognize that this was going to be extremely computationally expensive. I didn't know what steps needed to be taken, and that's on me.
Q: If you had another month, what one specific thing would you fix or add?
A: More training, more clustering. More samples. The 40,000-20,000 sample split is unacceptable. There are around 400,000 samples in the DeepFashion2 dataset. There needs to be more training! And I need to progressively keep storing the features and going slow with it.
Q: Would you want to continue with this work?
A: Hell yeah I would. SCREW THE INFLUENCERS. I'M NOT DONE WITH THIS.

7 Reproducibility

The dataset is huge. It can be found at `https://www.kaggle.com/datasets/thusharanair/deepfashion2-original-with-dataframes`. It took like 8 hours of doing Google Drive syncing to get the 60,000 images for the model. A version of the Jupyter lab notebook used for this investigation should be linked alongside the final report in D2L. Please contact me if it is not.

References

1. An, H., Kim, S., Choi, Y.: Sportive fashion trend reports: A hybrid style analysis based on deep learning techniques. Sustainability **13**(17) (2021). https://doi.org/10.3390/su13179530, `https://www.mdpi.com/2071-1050/13/17/9530`
2. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features (2019), `https://arxiv.org/abs/1807.05520`
3. Cheng, W., Song, S., Chen, C., Hidayati, S.C., Liu, J.: Fashion meets computer vision: A survey. CoRR **abs/2003.13988** (2020), `https://arxiv.org/abs/2003.13988`
4. Dong, J., Chen, Q., Huang, Z., Yang, J., Yan, S.: Parsing based on parselets: A unified deformable mixture model for human parsing. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(1), 88–101 (2016). https://doi.org/10.1109/TPAMI.2015.2420563

5. Ge, Y., Zhang, R., Wu, L., Wang, X., Tang, X., Luo, P.: Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images (2019), https://arxiv.org/abs/1901.07973