Fashion Intelligence through Multitask Learning: A Unified Model for Product Classification and Retrieval

By

Elene Zuroshvili

Submitted to
Central European University Private University
Department of Undergraduate Studies

In partial fulfillment of the requirements for the Bachelor of Arts and the Bachelor of Science in Data Science and Society
Major: Data Science and Economics

Supervisor: Professor Marton Posfai

Vienna, Austria
2025

Copyright Notice

Author’s Declaration

I, the undersigned, **Elene Zuroshvili**, candidate for the Bachelor of Art and Bachelor of Science in Data Science and Society, declare herewith that the present thesis is exclusively my own work, based on my research and only such external information as properly credited in notes and bibliography. I declare that no unidentified and illegitimate use was made of the work of others, and no part of the thesis infringes on any person’s or institution’s copyright. I also declare that no part of the thesis has been submitted in this form to any other institution of higher education for an academic degree.

Vienna, 26 May 2025

Elene Zuroshvili

Abstract

Online fashion platforms increasingly rely on automated systems to organize and retrieve product information. This thesis develops a multitask deep learning model that jointly performs category classification, attribute prediction, and image-based retrieval using the DeepFashion dataset. The model combines these tasks into a unified ResNet-50-based architecture with separate output heads, trained using cross-entropy and triplet loss. Five configurations are compared: single-task classifiers for category and attributes, a standalone retrieval model, a dual-head classification model, and a full multitask model incorporating all three objectives. While multitask learning introduces challenges in balancing competing losses, a staged training pipeline improves stability and leads to performance comparable to single-task baselines. Beyond model performance, the thesis discusses how such systems can help reduce search frictions and improve product discoverability in online fashion markets.

Acknowledgements

I am extremely grateful to Bojan Evkovski, PhD candidate at the Department of Network and Data Science (DNDS) at CEU, for his invaluable help throughout this project. His enthusiasm in brainstorming solutions, and his patient explanations of complex concepts have not only accelerated my research but also enriched my understanding of the subject.

I'd like to extend my heartfelt gratitude to my supervisor, Professor Marton Posfai, for his expert guidance and unwavering support. His feedback on experimental design, his patience in helping me refine my methods, and his constant encouragement have been invaluable both in completing this thesis and in helping me grow as a researcher.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the age of platform economies, digital infrastructure increasingly shapes how products are discovered, evaluated, and purchased. Nowhere is this more apparent than in the fashion industry, where visual presentation, branding, and search functionality all affect consumer behavior. As online fashion marketplaces expand, both in product scale and user diversity, the importance of intelligent systems that can organize, tag, and retrieve clothing items efficiently has grown. Yet, the quality and structure of product data often lag behind, especially in user-facing features such as search and recommendation.

Fashion is a uniquely visual domain. Unlike many other product categories, the way a clothing item looks is often more important to a buyer than its textual description or brand. Whether users are browsing casually or searching with a specific item in mind, visual information plays a critical role in shaping search behavior and purchase decisions [1]. However, most fashion platforms still rely on manual tagging, brand-centric filters, or natural language search—methods that can fail when metadata is incomplete or when users seek items based on appearance rather than keywords[2].

Recent developments in computer vision offer promising solutions to this problem. Deep learning architectures, especially Convolutional Neural Networks (CNNs) like ResNet [3], have shown strong results in both image classification and visual similarity tasks. These models can automatically tag clothing items by categories, or attributes, or retrieve similar-looking

[1] Hsiao and Grauman (2017)

[2] Deldjoo et al. (2023)

[3] He et al. (2016)

items from a large catalog. In commercial applications, such tools are used to enhance recommendation engines, support visual search on mobile devices, or reduce the burden of manual data entry. However, in most research and production systems, classification and retrieval are treated as separate tasks—trained on different models, using different objectives, and optimized independently.

This thesis explores an alternative approach: *joint training of fashion classification and visual retrieval within a single multitask architecture*. The idea is to use shared feature representations between tasks to improve both performance and efficiency. By combining these capabilities into one system, platforms could streamline how they process, tag, and index visual content, particularly as product catalogs grow larger and more visually diverse. Using the DeepFashion dataset [4], a widely-used benchmark in fashion recognition, I build and evaluate five models: a CNN-based classifier for categories, a CNN-based classifier for multilabel attributes, a triplet-loss retrieval model, a unified multitask classifier model for both categories and attributes, and a unified multitask model that performs all three tasks—category classification, attribute classification, and image retrieval—simultaneously. All five use a shared ResNet backbone for image embedding.

While the technical contributions of this thesis lie in deep learning and computer vision, the work is also informed by broader questions in *digital economics*. Online fashion marketplaces often suffer from search frictions caused by missing or inconsistent metadata[5]. Tools that improve automated tagging and visual similarity retrieval can help reduce these frictions, making it easier for consumers to find relevant items and for sellers to gain product exposure. Although this thesis does not empirically evaluate user behavior, it draws on theoretical work in economics to frame these systems as part of the digital infrastructure that shapes platform efficiency. The next chapter reviews related work in both computer vision and economic theory, setting the stage for the technical models and experiments that follow.

[4] Liu et al. (2016)
[5] Ngwe et al. (2019)

Chapter 2

Literature Review

Image classification and image retrieval are two core tasks in computer vision. While traditionally studied separately, they are increasingly explored together, particularly in fashion, where visual similarity and fine-grained detail are crucial. This review surveys each task individually, with an emphasis on fashion-specific challenges, followed by a discussion of multitask learning and its application to combining classification and retrieval. Finally, the review covers the economic theory that is used to connect the technicality of this thesis to the broader realm of digital economics.

2.1 Image Classification

Image Classification is a foundational task in vision recognition and computer vision. It involves categorizing images by assigning a class label. At its core, image classification seeks to answer a deceptively simple question: What is in this picture? This ease, however, masks a complex visual understanding process involving pattern recognition, context interpretation, and prior knowledge. Deep learning architectures, especially Convolutional Neural Networks (CNNs), have revolutionized this task since the introduction of AlexNet[1], which enabled large-scale image classification on datasets like ImageNet. Successive models such as VGG[2], ResNet[3], and

[1] Krizhevsky et al. (2012)

[2] Simonyan and Zisserman (2014)

[3] He et al. (2016)

EfficientNet[4] have further improved accuracy and efficiency through architectural innovations such as deeper networks, residual connections, and parameter optimization.

Fashion image classification is particularly challenging due to subtle visual differences between classes and significant variability within them. Datasets such as Fashion-MNIST[5] and DeepFashion[6] have enabled targeted research in this area. Jia et al.[7] in their research used the DeepFashion database, and tried to adapt Faster R-CNN for attribute recognition across online and runway images, highlighting the need for detailed annotation and multi-domain handling. Other studies focus on architectural and optimization improvements to CNNs for fashion tasks. Shin et al.[8] and Vijayaraj et al.[9] both demonstrate how tuning CNN architectures and hyperparameters, such as learning rate schedules and batch normalization, can significantly improve classification performance using the Fashion-MNIST dataset.

Amin et al.[10] contribute to the discussion by introducing a modular multitask pipeline (FSCAP) that combines YOLO, Faster R-CNN, and EfficientNet for person detection, subcategory classification, and attribute prediction. Their work underscores the importance of flexible, attribute-aware models in fashion contexts, aligning closely with this thesis's focus on developing unified systems.

Together, these studies illustrate a trajectory from basic classification models to sophisticated, domain-aware pipelines capable of handling the nuances of fashion imagery. This thesis builds on these foundations by proposing a multitask model that simultaneously addresses category, attribute, and similarity-based tasks.

2.2 Image Retrieval

While classification answers what an item is, image retrieval focuses on finding visually similar items given a query image. Image retrieval is the process of finding specific images from a database or dataset that match a given query, whether that query is another image, a text descrip-

[4] Tan and Le (2019)
[5] Xiao et al. (2017)
[6] Liu et al. (2016)
[7] Jia et al. (2018)
[8] Shin et al. (2023)
[9] Alwarsamy et al. (2022)
[10] Amin et al. (2022)

tion, or specific visual features[11]. Traditional methods based on bag-of-visual-words have been largely replaced by embedding-based approaches using deep learning[12]. Loss functions like triplet loss[13], contrastive loss[14], and proxy-based losses[15] are commonly used to train models that map images into a discriminative embedding space. Evaluation typically involves metrics like RecallK and mean average precision (mAP).

Fashion retrieval systems must account for subjective notions of similarity and style. Park et al. [16] explore deep retrieval networks specifically tuned for clothing and demonstrate that performance improvements are not solely the result of complex architectures, but can be significantly boosted through thoughtful combinations of loss functions and training strategies. More recently, Tian et al.[17] advance retrieval further with a multimodal system (AACL) that allows users to refine image queries using text-based edits. Their transformer-based model achieves state-of-the-art results across several fashion benchmarks, demonstrating the shift from static to interactive, adaptive retrieval.

The growing success of such compositional models signals an important shift from static retrieval to more intelligent, adaptive systems—one this work continues by proposing a model that integrates both classification and retrieval as joint learning objectives.

2.3 Multitask Learning

Multi-task learning (MTL) is a machine learning approach where a single model is trained to perform multiple related tasks at the same time, rather than learning them in isolation. Early empirical work by Caruana [18] showed that MTL could reduce overfitting by acting as a form of inductive bias: tasks with shared structure can help each other avoid memorizing noise. In deep learning, MTL usually involves a shared backbone with task-specific heads. Combined losses guide training, and recent work has explored automatic loss weighting to balance task

[11] Datta et al. (2008)

[12] Babenko et al. (2014)

[13] Schroff et al. (2015)

[14] Hadsell et al. (2006)

[15] Movshovitz-Attias et al. (2017)

[16] Park et al. (2019)

[17] Tian et al. (2023)

[18] Caruana (1997)

performance[19]. MTL is especially promising when tasks are structurally related, such as classification and retrieval.

A key challenge in MTL is managing conflicting gradients across tasks. Zhao et al.[20] address this by introducing a modulation module that dynamically adjusts feature sharing, improving performance on multi-domain retrieval tasks. Such architecture-aware approaches are essential in balancing classification and retrieval objectives.

Some researchers explore frameworks that unify classification and retrieval. Xie et al.[21] treat each image instance as its own class, framing retrieval as nearest-neighbor classification. Noh et al.[22] introduce DELF, which uses attention-based keypoints trained with classification labels to support fine-grained retrieval, suggesting that category supervision alone can produce useful retrieval embeddings.

Several works demonstrate multitask systems in specific domains. Zhang et al.[23] propose I-Net and DC-I-Net, which combine detection and retrieval using Siamese networks and task-specific branches. Yang et al.[24] merge emotion classification and sentiment retrieval in a shared embedding space. In food and storefront recognition, Wei and Wang[25] and Mafla et al.[26] show that shared architectures can improve both retrieval and classification.

Across these efforts, a common insight emerges: integrating classification and retrieval fosters more flexible and semantically rich representations. This thesis applies that insight to fashion, aiming to build a unified model that handles multiple visual recognition tasks simultaneously.

2.4 Economic Perspectives on Consumer Search

While the previous section focused on the technical aspects of deep learning models for classification and image retrieval, these systems also have broader implications for consumer behavior

[19] Cipolla et al. (2018)
[20] Zhao et al. (2018)
[21] Xie et al. (2015)
[22] Noh et al. (2017)
[23] Zhang et al. (2019)
[24] Yang et al. (2018)
[25] Wei and Wang (2020)
[26] Mafla et al. (2020)

and platform efficiency. In particular, tools that enhance product discovery—whether through visual similarity, tagging, or personalization—can reduce the time and effort required to find relevant items. This reduction in search costs connects directly to long-standing insights from economic theory.

One of the most influential papers in this area is by Diamond [27], who shows that even very small search costs can lead to surprising market outcomes. In his model, consumers may stop searching after seeing just one option, which allows firms to charge higher prices, similar to a monopoly. This idea highlights how even minor frictions can prevent markets from working efficiently. These issues are especially relevant in digital markets. Ellison and Ellison[28] explore how online retailers sometimes take advantage of limited consumer attention. They find that firms often use "obfuscation"—deliberately hiding important details like fees or using vague product descriptions—to make price comparisons harder.

In more applied work, Koulayev[29] uses real-world data to estimate how much effort consumers are willing to put into searching across different products. He shows that when platforms make it easier to search, consumers benefit: they find better options, and markets allocate goods more effectively. A recent paper by Wan et al.[30] add experimental evidence, showing that recommendations help users discover more suitable and cost-effective products. Together, these studies reveal how even minor search frictions shape pricing, competition, and consumer welfare.

Beyond individual behavior, some researchers focus on how platforms themselves shape search. Athey and Luca [31] argue that platforms act as market designers: they influence what people see, how products are organized, and what tools are available for discovery. Things like recommendation algorithms, product tags, and visual search tools are not just technical features—they change how markets work. As Athey and Luca write, "platforms structure markets through their design choices."

While economic theory highlights how search frictions reduce market efficiency, recent

[27] Diamond (1982)

[28] Ellison and Ellison (2009)

[29] Koulayev (2014)

[30] Wan et al. (2024)

[31] Athey and Luca (2019)

work in computer science and information systems has begun to explore how machine learning tools—particularly those based on deep learning—can help reduce these frictions and improve consumer experience. For example, Dagan et al.[32] analyze millions of search sessions on an e-commerce platform and find that visual search helps users discover items that are often missed in text-based search. Chaube et al.[33] take this further by developing a multimodal deep learning system that predicts the success of new product listings using both visual and textual features. In a separate but related domain, Zhang et al.[34] study how image quality affects rental outcomes on Airbnb, and find that better image quality boosts the bookings. Finally, Zheng et al.[35] show that query recommendations in a digital marketplace can increase both sales volume and product diversity. Their randomized experiment finds that suggesting search terms leads users to discover new sellers and categories—improving both market reach and consumer welfare.

Together, these studies show that deep learning systems—whether for visual retrieval, classification, or recommendation—can meaningfully reduce search frictions and improve outcomes for both consumers and sellers.

2.4.1 Research Gap and Contribution

Despite promising results, most of these systems are implemented in isolation, handling classification, retrieval, or recommendation separately. Few studies connect these technical tools to economic theory or explore unified architectures. This thesis addresses that gap by proposing a multitask model tailored to fashion platforms, where search frictions are acute due to inconsistent metadata and image variability.

The contribution is twofold: technically, the model integrates classification and retrieval into a shared architecture trained on DeepFashion; conceptually, it frames the system as part of digital market infrastructure, helping reduce information asymmetries and improve product discovery. To the best of my knowledge, this is the first study to explicitly link multitask visual learning in fashion to economic theories of consumer search.

[32] Dagan et al. (2023)

[33] Chaube et al. (2025)

[34] Zhang et al. (2021)

[35] Zheng et al. (2023)

Chapter 3

Data and Preprocessing

This chapter introduces the dataset used in this study and details the steps taken to preprocess and structure the data for training classification, retrieval, and joint multitask models.

3.1 Dataset: DeepFashion

The dataset used throughout this thesis is the DeepFashion dataset[1], a large-scale benchmark developed by the Multimedia Lab at the Chinese University of Hong Kong. Designed to facilitate a wide range of fashion recognition tasks, DeepFashion includes over 800,000 images of diverse clothing items, annotated with rich metadata and fine-grained labels.

The DeepFashion dataset is especially suitable for this project because it provides a unified structure for both classification (category and attributes) and visual retrieval (in-shop image pairs), which enables direct experimentation with both unitask and multitask architectures. All images in DeepFashion are high-resolution and reflect real-world variability in lighting, pose, background clutter, and garment deformation, making it an ideal candidate for evaluating the vision models. The dataset subsets used in this thesis include 289,222 images for the classification and attribute prediction tasks, and 52,712 images for the in-shop retrieval benchmark.

[1] Liu et al. (2016)

3.2 Category and Attribute Labels

The category classification task involves assigning a fashion image to one of 50 predefined classes, such as blouse, dress, pants, or jacket. These labels are mutually exclusive, and the task is treated as a standard multiclass classification problem. The attribute prediction task involves predicting the presence of multiple binary attributes from a pool of 26 tags. These include visual descriptors such as long-sleeved, striped, v-neck, or floral. Each image may be annotated with zero or more attributes, and the task is framed as multi-label classification using binary cross-entropy loss. A complete list of the attribute and category labels used in the model is provided in Appendix 7.

To better understand the distribution of labels in the dataset, Figure 3.1 shows the prevalence of each visual attribute in the data set. Certain attributes are underrepresented, indicating an imbalance that may affect model training.

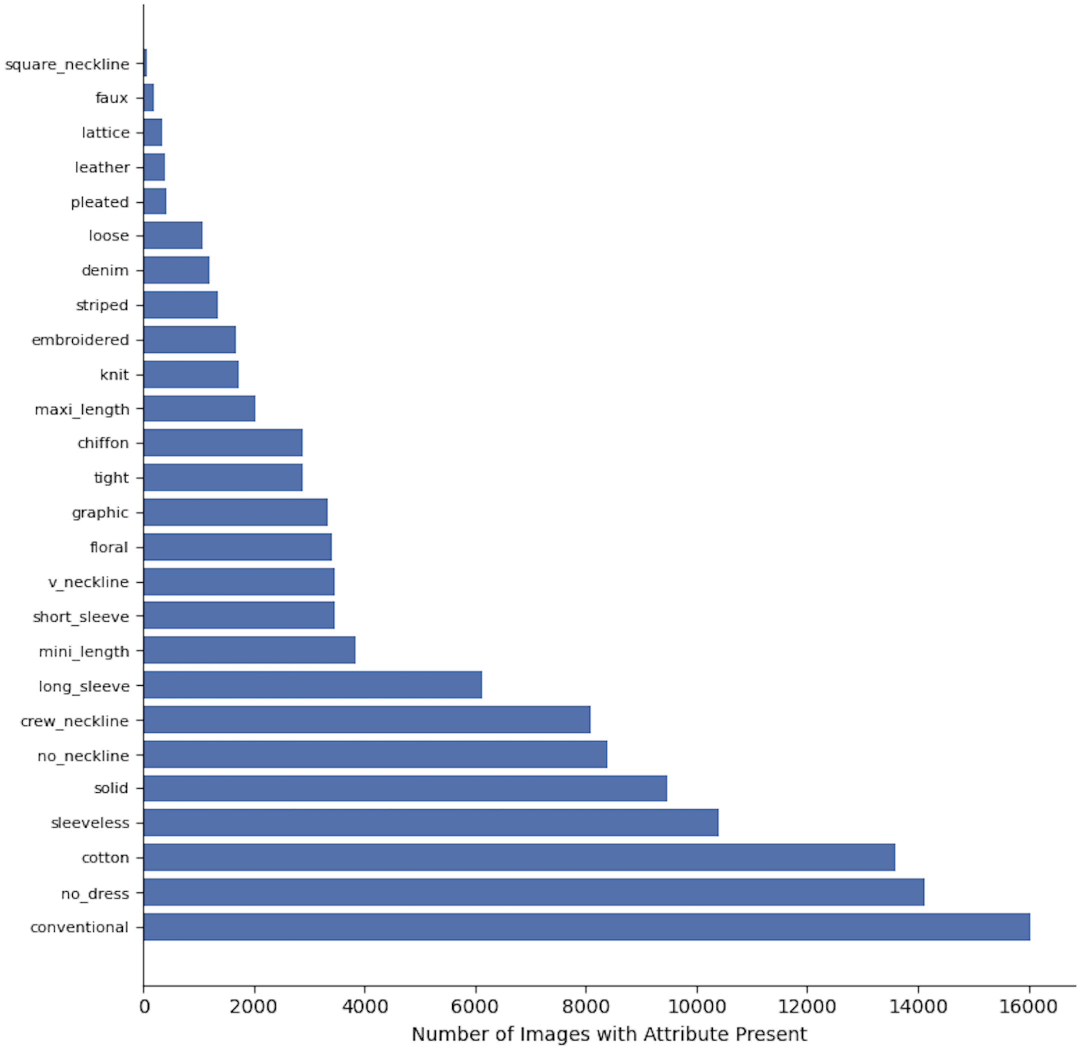

Figure 3.1: Attribute Prevalence: Number of images annotated with each attribute.

Similarly, Figure 3.2 displays the distribution of category labels on a logarithmic scale. The highly heterogeneous distribution underscores the challenges of learning accurate classifiers for rare categories.

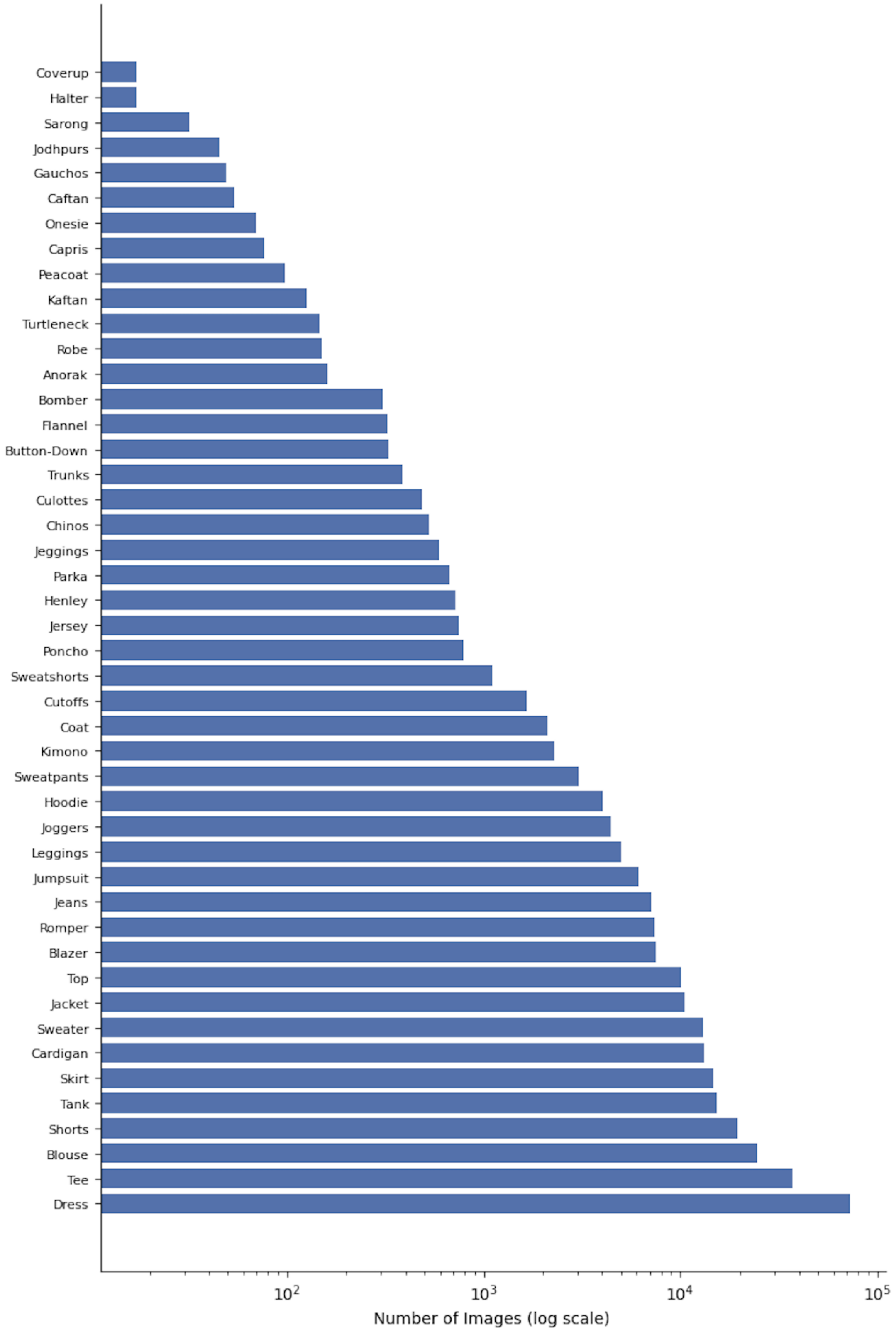

Figure 3.2: Category Distribution: Number of images per category label (log scale).

3.3 In-Shop Retrieval

The In-Shop Clothes Retrieval benchmark focuses on retrieving instances of the same clothing item from a large gallery of professional shop images. All images in this benchmark are high-quality catalog photos taken in controlled conditions. Each clothing item is represented by multiple views or variations, and the retrieval task involves identifying the correct match for a given query image from a gallery of visually similar but distinct items.

3.4 Image Preprocessing

To standardize input across tasks, all images are resized to 224×224 pixels. During training, a series of augmentations are applied to improve generalization and mitigate overfitting. These include random horizontal flipping with probability of 50%, random rotations of up to 10 degrees, and color jittering that alters brightness, contrast, and saturation levels. Additional transformations such as random resized cropping and random erasing further enhance the variability in the training data. All images are normalized using the ImageNet mean and standard deviation values.

3.5 Label Encoding

Categories are encoded as integer indices ranging from 0 to $C - 1$, where C is the number of categories retained (for example, 20). Attributes are represented using multihot vectors of length A (e.g. 100), with each dimension indicating the presence or absence of a particular attribute. For the retrieval task, each image is mapped to a unique product ID, which is used to construct training triplets for embedding supervision. To ensure consistency, item IDs are mapped to numerical labels, and class-balanced samplers are used in retrieval to ensure each mini-batch contains multiple examples per identity.

3.6 Dataset Splits

Following DeepFashion's official partitioning, the classification and attribute data are divided into a training set of 209 222 images, a validation set of 40 000 images, and a held-out test set of 40 000 images. For the in-shop retrieval benchmark, the training gallery comprises 25 882 images, while evaluation uses a separate query set of 14 218 images and a gallery set of 12 612 images. Care was taken to ensure that no image or product identifier appears in more than one partition, neither across the train, validation, and test for classification, nor between query and gallery for retrieval, to prevent data leakage.

Chapter 4

Methodology

This chapter presents the core methodological components of the thesis, including the design of model architectures, the formulation of loss functions, training strategies, and evaluation procedures. I began by developing three foundational single-task models: one focused on category classification, one focused on attribute classification, and the other on image-based retrieval. These models were trained independently to better understand the demands of each task and to establish performance baselines. Their design and behavior informed the construction of the multitask architectures presented in the later sections.

4.1 Category-Only Model

The category prediction model serves as the first single-task baseline, framing the problem as a standard multi-class classification over $C = 48$ clothing categories. It is built upon a ResNet-50 backbone pretrained on ImageNet, with early convolutional layers (conv1, bn1, and layer1) frozen to preserve low-level feature representations and reduce training time. The original fully connected head is removed, and the remaining feature map is global-pooled and fed into a three-layer multilayer perceptron (MLP) head. Concretely, after an adaptive average pooling to a 1×1 spatial map, the feature vector of dimension 2048 is passed through two hidden layers of sizes 1024 and 512, each followed by Batch Normalization, ReLU activation, and Dropout (rate 0.3), before a final linear layer projects to the 48 category logits.

$$
\begin{aligned}
h_1 &= \mathrm{ReLU}\big(\mathrm{BN}(W_1\, f + b_1)\big), & h_2 &= \mathrm{ReLU}\big(\mathrm{BN}(W_2\, h_1 + b_2)\big),\\
\ell &= W_3\, h_2 + b_3, & \hat{y}_i &= \frac{\exp(\ell_i)}{\sum_{j=1}^{C} \exp(\ell_j)},
\end{aligned}
$$

where $f \in \mathbb{R}^{2048}$ is the pooled feature, $W_{1,2,3}$, $b_{1,2,3}$ are learned parameters, and $\hat{y}$ is the predicted category distribution.

Given the long-tailed distribution of categories (Figure 3.2), I employ both sampling- and loss-based strategies to mitigate class imbalance. I compute per-class inverse-frequency weights,

$$
w_c = \frac{1}{\max\big(\mathrm{count}_c,\ \epsilon\big)},
$$

then clip $w_c \in [0.5, 5.0]$ for stability and normalize to mean 1.0. These weights serve two roles: first, as sampling weights in a WeightedRandomSampler to oversample rare classes during training; second, as the class weight vector in the cross-entropy loss,

$$
\mathcal{L}_{\mathrm{cat}} = -\sum_{c=1}^{C} w_c\, y_c\, \log \hat{y}_c.
$$

I train with the AdamW optimizer (learning rate 1×10^{-3}, weight decay 1×10^{-4}) and a cosine annealing learning rate scheduler ($\mathrm{T}_{\mathrm{max}}$=10, $\eta_{\mathrm{min}} = 10^{-5}$). Batches of size 64 are drawn using the weighted sampler; mixed precision is enabled via PyTorch AMP for efficiency. I run for up to 10 epochs with early stopping (patience=5) on the validation macro-F1 score. Training and validation images are preprocessed with random horizontal flip ($p = 0.5$), random rotation (±10°), color jitter (brightness/contrast/saturation=0.2, hue=0.1), random erasing ($p = 0.5$), and normalization to ImageNet statistics; test images use only resize to 224×224 and normalization. Finally, evaluation is done using top-1 accuracy, macro-F1 (to equally weight all classes), and weighted-F1.

4.2 Attribute-Only Model

The attribute prediction model treats inference of $A = 26$ binary visual attributes as a multi-label classification problem. I employ the same ResNet-50 backbone used for category prediction, removing its final fully connected layer and replacing it with a multi-layer perceptron head that outputs A logits. Specifically, after global average pooling, the backbone feature vector of dimension 2048 is passed through two hidden layers of sizes 1024 and 512—each followed by Batch Normalization, ReLU activation, and Dropout at a rate of 0.3—and finally projected via a linear layer to A outputs:

$$\hat{y}_a = \sigma\big(W_a\, h_2 + b_a\big), \quad a = 1, \dots, A,$$

where σ denotes the sigmoid activation and $h_2 \in \mathbb{R}^{512}$ is the output of the second hidden layer.

To address the severe class imbalance among attributes, I compute per-attribute positive-class weights from the training labels. Denoting by pos_a and neg_a the counts of positive and negative examples for attribute a, the weight for positives is

$$w_a^+ = \frac{\text{neg}_a}{\text{pos}_a + \epsilon},$$

with $\epsilon = 10^{-6}$ to avoid division by zero. These weights are incorporated into the binary cross-entropy loss:

$$\mathcal{L}_{\text{attr}} = -\sum_{a=1}^{A} \Big[w_a^+\, y_a\, \log \hat{y}_a + (1 - y_a)\, \log(1 - \hat{y}_a) \Big].$$

I train the model using the AdamW optimizer with a learning rate of 10^{-3} and weight decay of 10^{-4}, coupled with a CosineAnnealingLR scheduler ($T_{\text{max}} = 30$, $\eta_{\text{min}} = 10^{-5}$). Training is performed for up to 10 epochs with a batch size of 64, using PyTorch's Automatic Mixed Precision to improve throughput. During each epoch, I evaluate the model on the validation set using two metrics:

- Hamming Loss, defined as the proportion of attribute-bits incorrectly predicted:

$$\text{Hamming} = \frac{1}{N\,A} \sum_{i=1}^{N} \sum_{a=1}^{A} \big[\mathbb{I}(\hat{y}_{i,a} > 0.5) \neq y_{i,a} \big].$$

- Micro-F1 Score, computed by aggregating true positives, false positives, and false negatives across all attribute predictions and taking the harmonic mean of precision and recall.

The data preprocessing and augmentation pipelines mirror those used for category classification, ensuring consistent input resolution and statistical normalization across models.

4.3 Category and Attribute Prediction Multitask Model

Next step in the pipline was to extend the single-task classifiers into a unified multitask network that predicts both clothing category and visual attributes from a shared feature representation. I wanted to perform this step before diving into the three head multitask to see how the two classification tasks would interact with each other. The logical flow was to take the single architectures, using all the same parameters, and see if the shared backbone would affect training and results. As before, the convolutional backbone is a ResNet-50 pretrained on ImageNet, with all layers up through layer1 frozen to preserve low-level filters. I replace the final pooling and classification layers with a *shared representation* module followed by two separate heads:

- Shared representation: after adaptive average pooling to a 1×1 feature map, the 2048-dimensional vector is flattened and passed through two fully connected layers of sizes 1024 and 512. Each layer is followed by Batch Normalization, ReLU activation, and Dropout (rate 0.3).

- Category head: a linear layer mapping from 512 to C categories, trained with cross-entropy loss weighted by the inverse class frequencies (clipped to $[0.5, 5.0]$ and normalized to mean 1.0) and oversampled via a WeightedRandomSampler.

- Attribute head: a linear layer mapping from 512 to $A = 26$ attributes, trained with binary cross-entropy loss incorporating a per-attribute positive weight $w_a^+ = \frac{\text{neg}_a}{\text{pos}_a + \epsilon}$ to counteract label imbalance.

Formally, denoting by $f(x) \in \mathbb{R}^{2048}$ the pooled backbone features, the shared module computes

$$h_1 = \mathrm{ReLU}\big(\mathrm{BN}(W_1\, f(x) + b_1)\big), \quad h_2 = \mathrm{ReLU}\big(\mathrm{BN}(W_2\, h_1 + b_2)\big),$$

and the two heads produce

$$\ell^{\mathrm{cat}} = W_{\mathrm{cat}}\, h_2 + b_{\mathrm{cat}}, \quad \ell^{\mathrm{attr}} = W_{\mathrm{attr}}\, h_2 + b_{\mathrm{attr}}.$$

The category probabilities $\hat{y}^{\mathrm{cat}}$ follow a softmax over ℓ^{cat}, and the attribute probabilities $\hat{y}^{\mathrm{attr}}$ follow a sigmoid applied element-wise to ℓ^{attr}.

4.3.1 Loss Functions and Class Imbalance Handling

I train the multitask model by minimizing the sum of the two task losses:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{cat}} + \mathcal{L}_{\mathrm{attr}}.$$

The category loss is weighted cross-entropy,

$$\mathcal{L}_{\mathrm{cat}} = -\sum_{c=1}^{C} w_c\, y_c\, \log \hat{y}_c^{\mathrm{cat}},$$

where w_c are the normalized inverse-frequency weights. The attribute loss is weighted binary cross-entropy,

$$\mathcal{L}_{\mathrm{attr}} = -\sum_{a=1}^{A} \Big[w_a^{+}\, y_a\, \log \hat{y}_a^{\mathrm{attr}} + (1 - y_a)\, \log(1 - \hat{y}_a^{\mathrm{attr}}) \Big],$$

with $w_a^{+} = \mathrm{neg}_a / (\mathrm{pos}_a + \epsilon)$.

4.3.2 Training Setup and Evaluation

The training setup mirrors the two single-task configurations. The optimizer (AdamW with learning rate 10^{-3}, weight decay 10^{-4}), learning rate scheduler (cosine annealing with $T_{\max} = 30$, $\eta_{\min} = 10^{-5}$), batch size (64), and use of mixed precision (via PyTorch AMP) remain unchanged. Data augmentation, preprocessing, and early stopping are applied identically, ensuring consistency across experiments.

The only difference lies in evaluation: during validation and testing, I compute all five metrics jointly—top-1 accuracy, macro-F1, and weighted-F1 for category prediction, along with Hamming loss and micro-F1 for attribute prediction. These allow for a direct performance comparison with the single-task models and help assess whether the multitask setup introduces trade-offs or synergies between the two objectives.

4.4 Image Retrieval Model

4.4.1 Architecture Design

The retrieval model is similarly built upon a ResNet-50 backbone but is modified to support deep metric learning objectives. The final classification layer of the original network is removed, and the resulting feature map is processed through a series of transformations to generate embeddings suitable for similarity-based retrieval.

First, the output feature map is pooled using Adaptive Average Pooling to produce a fixed-size feature vector, regardless of the original input dimensions. This vector is then passed through a projection head, implemented as a multilayer perceptron (MLP). The projection head consists of a sequence of Batch Normalization, ReLU activation, and Dropout with a rate of 0.3, followed by a final linear transformation that maps the features to a 256-dimensional embedding space. Finally, the output embeddings are L2-normalized to lie on the unit hypersphere, enabling effective comparison via cosine similarity.

Formally, the model learns an embedding function $f(x) \in \mathbb{R}^{256}$ such that items with the same clothing ID (i.e., semantically similar products) have embeddings located close together in the feature space, while dissimilar items are pushed farther apart.

4.4.2 Triplet Loss with Batch-Hard Mining

The retrieval model is trained using margin-based triplet loss, a widely used objective in deep metric learning that encourages the model to position similar items close together and dissimilar items farther apart in the embedding space. Each training sample is structured as a triplet: an anchor image (reference), a positive image (of the same item), and a negative image (of a different item).

The training objective is to minimize the distance between the anchor and the positive embedding while ensuring that the distance to the negative embedding exceeds it by at least a predefined margin. This teaches the model to "pull" semantically similar items closer and "push" dissimilar ones apart, which is particularly effective in fine-grained retrieval tasks such as fashion search.

The triplet loss is formally defined as:

$$\mathcal{L}_{\text{triplet}} = \sum_{i=1}^{N} \left[\| f(x_i^a) - f(x_i^p) \|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + \alpha \right]_+$$

where x_i^a is the anchor image, x_i^p is a positive image with the same item ID, x_i^n is a negative image from a different item, and $\alpha = 0.3$ is the margin enforcing a minimum separation. The hinge function $[\cdot]_+$ ensures that only triplets violating the margin contribute to the loss. The margin was selected based on standard practice and empirical tuning, offering a balance between separation strength and training stability.

To enhance training efficiency and encourage more discriminative learning, I used batch-hard mining, a strategy that dynamically selects the hardest triplets within each mini-batch. For each anchor, the hardest positive (the farthest correct match) and the hardest negative (the closest incorrect match) are selected. This ensures that training focuses on the most challenging examples, which accelerates convergence and improves embedding quality. To support this, each mini-batch is constructed to include multiple examples per class, ensuring that valid anchor-positive pairs are always present.

Finally, all embeddings are L2-normalized, allowing similarity to be computed via cosine distance. This normalization ensures that the learned distances are consistent and that retrieval performance reflects angular similarity in the latent space.

4.4.3 Training Setup

The retrieval model is trained for 20 epochs using the AdamW optimizer with an initial learning rate of 1×10^{-4}. A StepLR scheduler reduces the learning rate by a factor of two every five epochs, allowing for more stable convergence as training progresses. To improve computational efficiency and reduce memory consumption, mixed precision training is enabled via PyTorch's Automatic Mixed Precision (AMP) module.

Sampling plays a key role in triplet-based training. To ensure each mini-batch contains a sufficient number of anchor-positive pairs, I use the MPerClassSampler from the PyTorch Metric Learning library with $M = 4$. This guarantees that each batch includes multiple instances of the same class, which is essential for effective batch-hard mining and meaningful gradient updates.

4.4.4 Evaluation Metrics

The retrieval model is evaluated using standard Recall@K metrics, which assess how well the model ranks the correct match among the retrieved candidates. Recall@1 indicates the proportion of queries for which the correct item appears as the top-ranked result, while Recall@5 measures the percentage of queries where the correct item is found within the top five retrieved results. Together, these metrics provide a direct measure of the model's ability to embed semantically similar items close together in the learned feature space.All evaluations are performed using cosine similarity to compare query embeddings against gallery embeddings.

4.5 Multitask Architecture for Joint Classification and Retrieval

The central goal of this thesis is to develop a multitask deep learning system capable of simultaneously performing fashion classification and image-based retrieval. I began by implementing a unified architecture that optimizes classification and retrieval objectives simultaneously within a single-stage training process. However, early experiments with this approach produced mediocre results—suggesting that the competing objectives were interfering with one another during joint optimization.

This led to a reconsideration of the training strategy. Inspired by the idea that retrieval and classification may benefit from learning complementary but differently prioritized representations, I designed a staged training pipeline. In this second variant, the model is first trained for retrieval using triplet loss, then for classification, and finally jointly fine-tuned. This structure proved more effective in practice.

The remainder of this section outlines both multitask variants. While they share a common architectural foundation and loss formulation, they differ in how training is structured and how the two objectives are balanced throughout the learning process.

4.5.1 Model Architecture

In designing the multitask model, I built on the same ResNet-50 convolutional backbone used in earlier experiments. The final classification head of ResNet is removed, and the architecture then splits into two distinct branches after a shared feature extraction pipeline—one for classification and one for retrieval.

The classification head is responsible for predicting both the clothing category (as a single-label classification task) and multiple binary visual attributes (as a multi-label task). This branch consists of two fully connected layers, followed by separate linear layers for category and attribute predictions. The retrieval head, on the other hand, maps the shared feature representation into a 256-dimensional embedding space, which is optimized for deep metric learning using triplet loss.

To improve regularization and stabilize training, I incorporated batch normalization, ReLU activation, and dropout (set at a rate of 0.3) after each fully connected layer in both branches. This design choice was informed by earlier tuning in the classification-only model, where these components helped reduce overfitting and accelerated convergence.

Finally, during multitask training, all convolutional layers up to ResNet's layer2 block are frozen. This was a practical decision to preserve low-level feature representations while reducing training time and computational load, especially given the added complexity of jointly optimizing multiple tasks.

4.5.2 Loss Function

Training the multitask model involves jointly optimizing both classification and retrieval objectives. These are combined into a single loss function that balances category prediction, attribute recognition, and embedding learning:

$$\mathcal{L}_{\text{total}} = \lambda_{cls}(\mathcal{L}_{cat} + \mathcal{L}_{attr}) + \lambda_{ret}\mathcal{L}_{triplet}$$

Here, $\mathcal{L}_{cat}$ denotes the cross-entropy loss used for single-label category prediction, while $\mathcal{L}_{attr}$ refers to the binary cross-entropy loss applied to multi-label attribute classification. The retrieval component, $\mathcal{L}_{triplet}$, is the same margin-based triplet loss described in Section for the triplet loss.

Since this formulation introduces competing objectives, selecting appropriate weights for λ_{cls} and λ_{ret} was a key part of model tuning. I experimented with several configurations before arriving at a balanced setting that ensured neither task dominated the optimization. In particular, I found that an underweighted retrieval loss could result in poor embedding structure, while overemphasizing it would degrade classification performance. Thus, finally I chose to use λ_{cls} of 0.2 and λ_{ret} of 1.0.

4.5.3 Training Strategies

After defining the multitask architecture and loss function, I experimented with two distinct training strategies: a unified single-stage approach and a structured three-stage pipeline. These strategies represent different philosophies of multitask learning—whether to jointly optimize from the start or allow each task to specialize before merging.

Unified Multitask Training

The first approach involved training both the classification and retrieval components together from the very beginning. In this setup, batches from the classification and retrieval datasets were interleaved during each epoch, and gradients from all loss components were backpropagated simultaneously. A cosine annealing scheduler was used to gradually reduce the learning rate, and all model parameters—including the shared backbone—were updated in each optimization step.

While this approach was straightforward to implement and conceptually elegant, it quickly became clear that the two objectives interfered with one another during early training. The model struggled to improve on either task, likely due to conflicting gradient signals from the classification and retrieval heads. This observation motivated a shift in strategy.

Three-Stage Pipeline

In response to the challenges of joint training, I implemented a staged optimization pipeline that allowed each task to develop more independently before unifying them. This variant involved three sequential phases:

1. **Stage 1: Retrieval Pretraining** — The retrieval head and shared feature extractor were trained using triplet loss, with the classification layers frozen.

2. **Stage 2: Classification Pretraining** — The classification head was trained using cross-entropy and binary cross-entropy losses, while the retrieval branch was frozen.

3. **Stage 3: Joint Fine-Tuning** — All layers were unfrozen, and the model was fine-tuned using the full multitask loss. Loss weights were rebalanced to $\lambda_{cls} = 0.2$ and $\lambda_{ret} = 1.0$ to prioritize retrieval performance during this final phase.

This structured approach proved significantly more effective. By allowing each task to first learn in isolation, the model was better able to integrate both objectives during joint fine-tuning. In practice, this strategy improved overall convergence and led to a more stable training process, while also yielding better performance on both classification and retrieval tasks.

4.5.4 Evaluation Metrics

To evaluate the multitask model's effectiveness, I assessed its performance along the two dimensions with metrics that were used in the previous models. For the classification branch, I report top-1 accuracy, macro-F1, and weighted-F1; for the attribute branch, I use Hamming loss and micro-F1. For the retrieval task, I used Recall@K, again specifically Recall@1 and Recall@5.

All evaluations were conducted on held-out validation sets. To track generalization and detect potential overfitting, I performed validation at the end of every epoch and used these metrics to guide early stopping and checkpoint selection. After the training and validation was complete, I once again evaluated the model on the held out test set as the final metric.

Chapter 5

Results

This section reports the results for the five models described above. It begins with training and validation performance, highlighting convergence behavior and learning stability. The final section presents a summary of the evaluation metrics on the held-out test set, enabling a direct assessment of model generalization across tasks.

5.1 Category Classification Performance

I begin by examining the learning dynamics and final metrics of the category-only model on the validation set. Figure 5.1 plots the average training loss alongside validation top-1 accuracy and macro-F1 over the ten training epochs. From the very first epoch, validation accuracy jumps from 50.95% to 57.80%, and macro-F1 climbs from 0.1635 to 0.2383, reflecting rapid initial learning. After epoch 3 the gains slow down but remain steady, peaking at epoch 9 with 63.80% accuracy and a macro-F1 of 0.3420. A slight dip in epoch 10 suggests the onset of overfitting, which informed the early-stopping criterion.

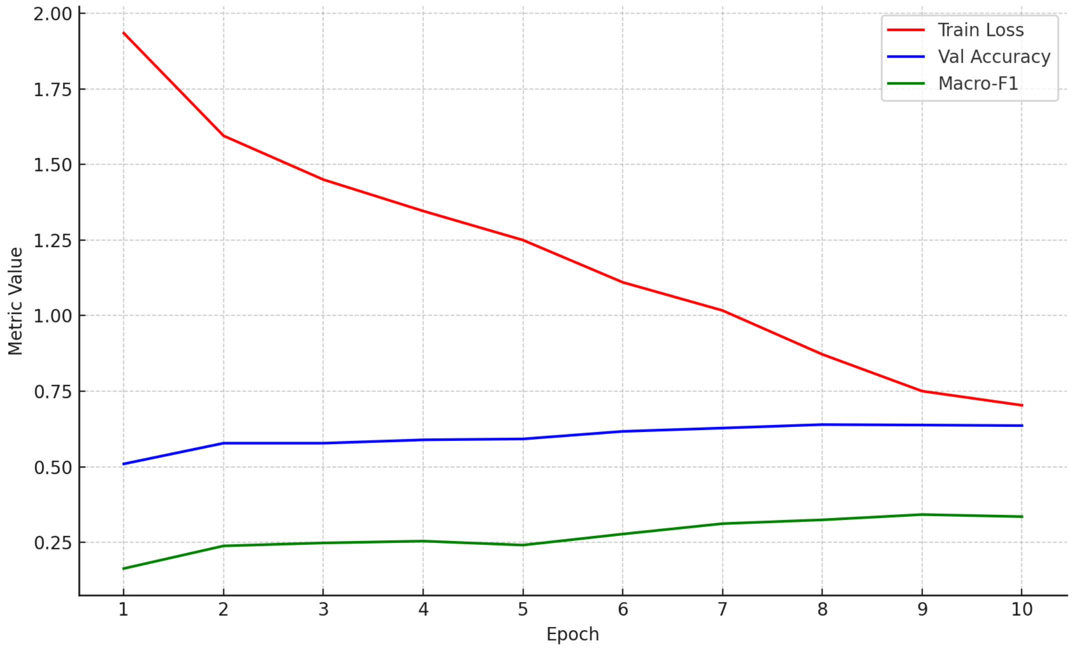

Figure 5.1: Training loss, validation top-1 accuracy, and validation macro-F1 over epochs for the category-only model.

The detailed summary of the validation metrics at the each epoch can be found in Table 3 in the appendix.

To illustrate class-specific behavior, Figure 5.2 shows a horizontal bar chart of per-class F1 scores (for classes 0–47) on the validation set at the best checkpoint (epoch 9). This visualization highlights strong performance on common categories and the long-tail drop-off on rarer ones.

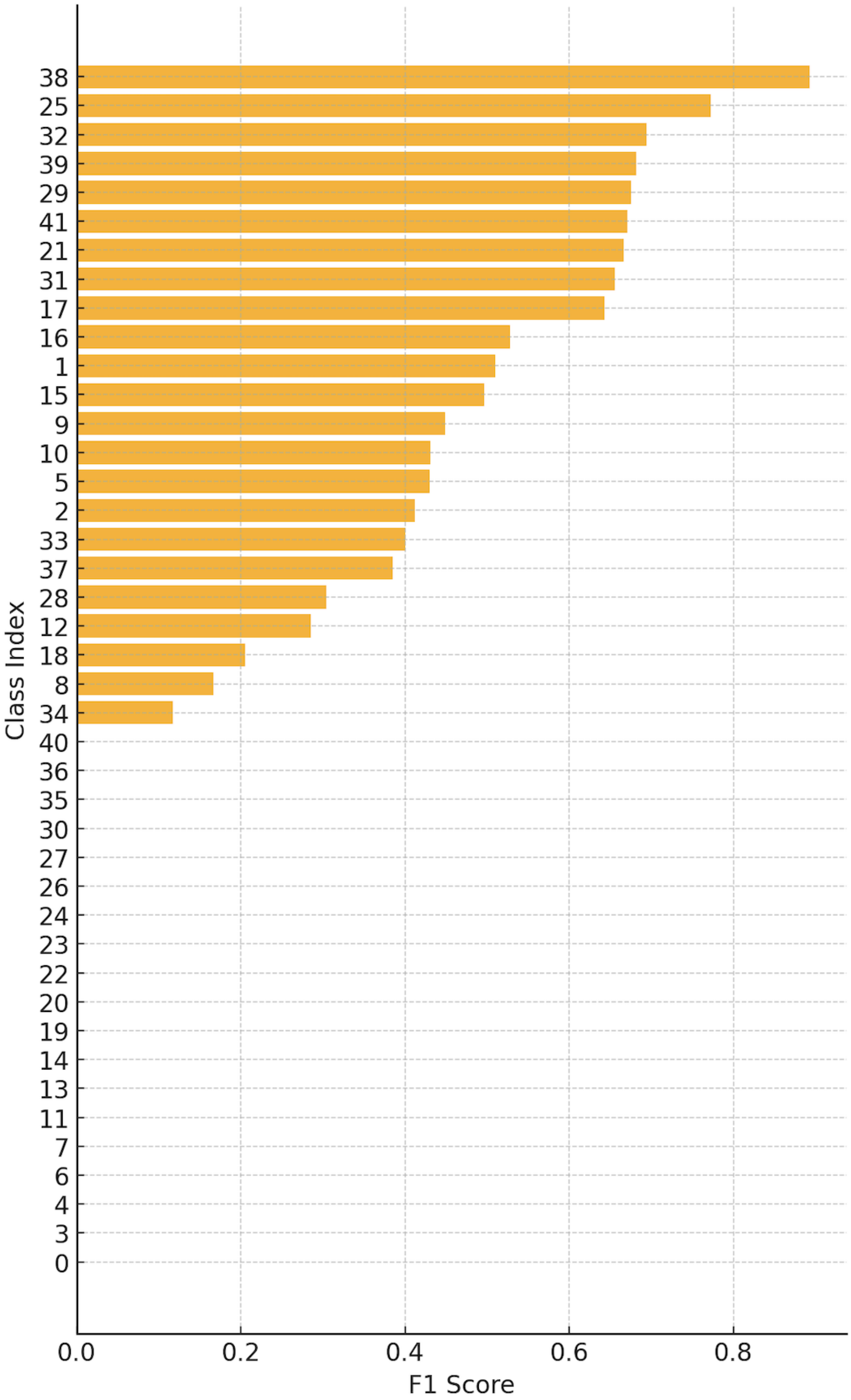

Figure 5.2: Per-class F1 scores on the validation set for the category-only model (epoch 9).

Finally, Table 5.1 provides a truncated classification report for a selection of categories. Full per-class precision, recall, and F1-scores are available in Appendix 7.

Table 5.1: Selected per-class precision, recall, and F1-score (validation set, epoch 9).

Class	Precision	Recall	F1-Score
Blouse	0.5556	0.4717	0.5102
Dress	0.7103	0.6786	0.6941
Pants	0.5234	0.5333	0.5283
Tee	0.8622	0.9134	0.8871
Skirt	0.8125	0.7358	0.7723

5.2 Attribute Prediction Performance

Next, I evaluate the attribute-only model on the validation set. Figure 5.3 plots validation Hamming loss and micro-F1 across the ten epochs, illustrating rapid convergence up to epoch 6 and subsequent metric fluctuations.

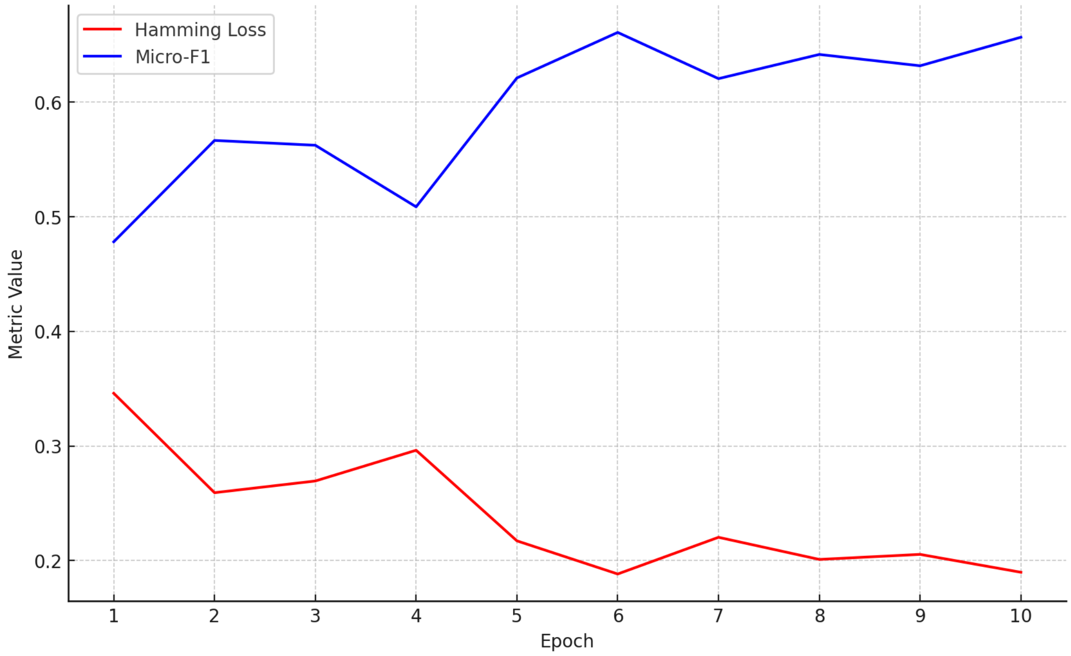

Figure 5.3: Validation Hamming loss and micro-F1 over epochs for the attribute-only model.

The detailed summary of the validation metrics at the each epoch can be found in Table 4 in the appendix.

5.3 Multitask Model: Category and Attribute Recognition

The classification model was trained for 30 epochs. The evolution of both category and attribute metrics on the validation set is shown in Figure 5.4. The best snapshot was saved at epoch 23, yielding category accuracy of 63.65 %, macro-F_1 = 0.3959, weighted-F_1 = 0.6286, attribute Hamming loss = 0.1110, and attribute micro-F_1 = 0.7870.

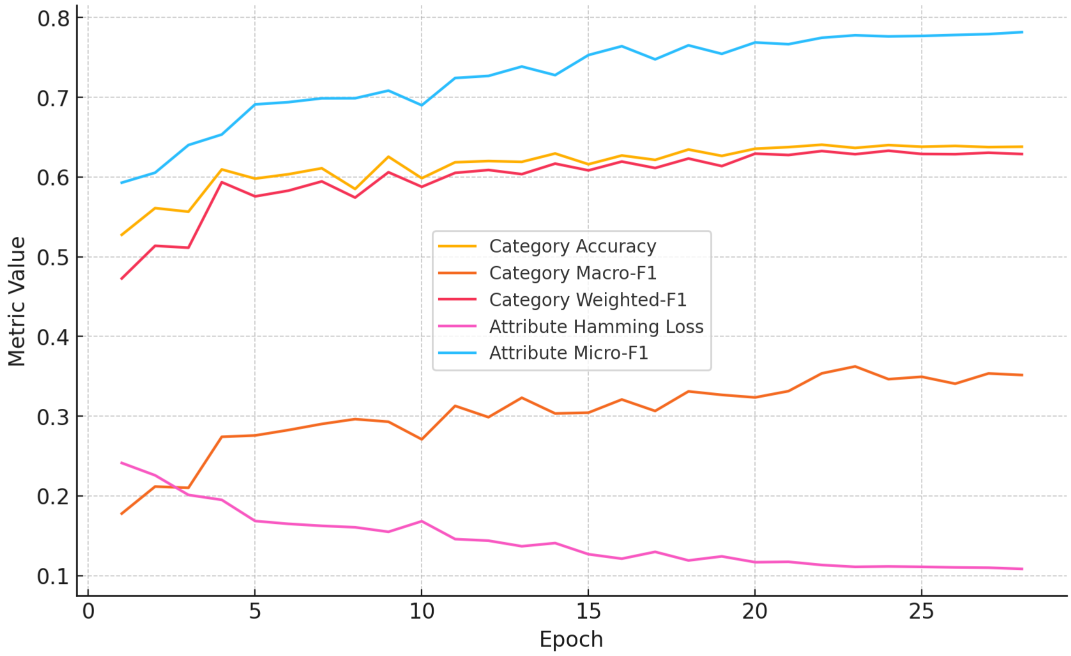

Figure 5.4: Validation metrics over 28 epochs for the multitask category+attribute model: category accuracy, category macro-F_1, category weighted-F_1, attribute Hamming loss, and attribute micro-F_1.

In addition, Figure 5.5 shows that the model successfully minimized training loss while keeping validation loss stable, indicating good generalization without overfitting.

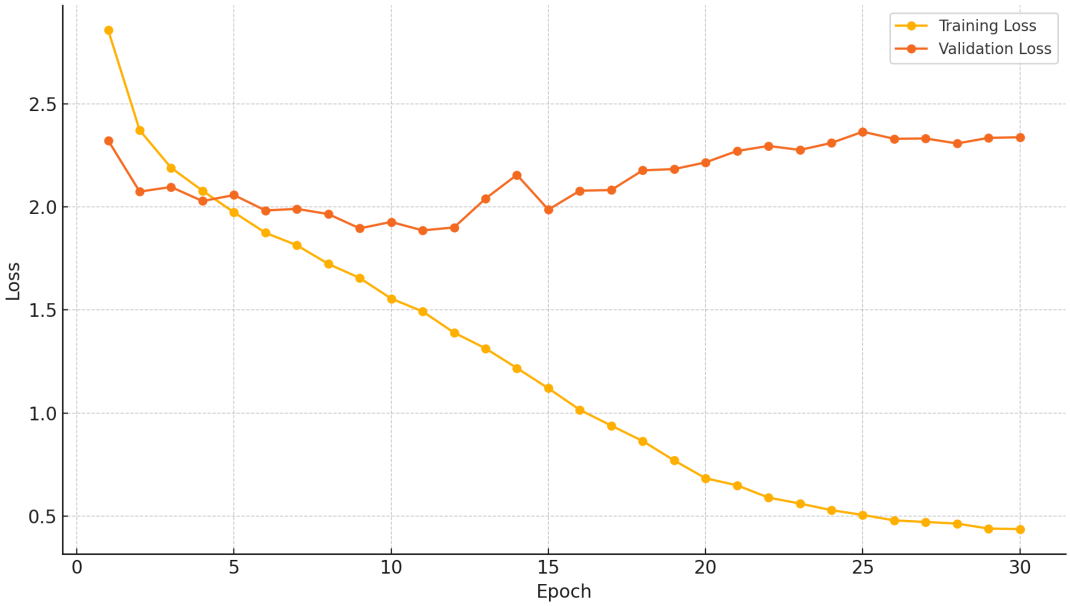

Figure 5.5: Training and validation loss over 30 epochs for the final classification model.

5.4 Retrieval Performance

5.4.1 Model Iterations and Improvements

The primary evaluation metric for retrieval was Recall@K, with K set to 1 and 5.

Model 1: Triplet Loss with Random Sampling The initial retrieval model was trained using random sampling for the triplets. On the evaluation set, the model achieved a Recall@1 of 36.37% and a Recall@5 of 58.01%.

Model 2: Hard Triplet Mining (Final Retrieval Model) To improve embedding quality, I incorporated batch-hard mining into the training process (see Section 4.4.2). The addition of batch-hard mining resulted in a substantial performance boost, with Recall@1 increasing to 72.67% and Recall@5 reaching 87.42%. These results highlighted the critical role of mining strategies in deep metric learning for visual similarity tasks.

Figure 5.6 shows the evolution of the average triplet loss across training epochs for the final retrieval model. The steady decrease in loss indicates effective optimization and convergence of the embedding space.

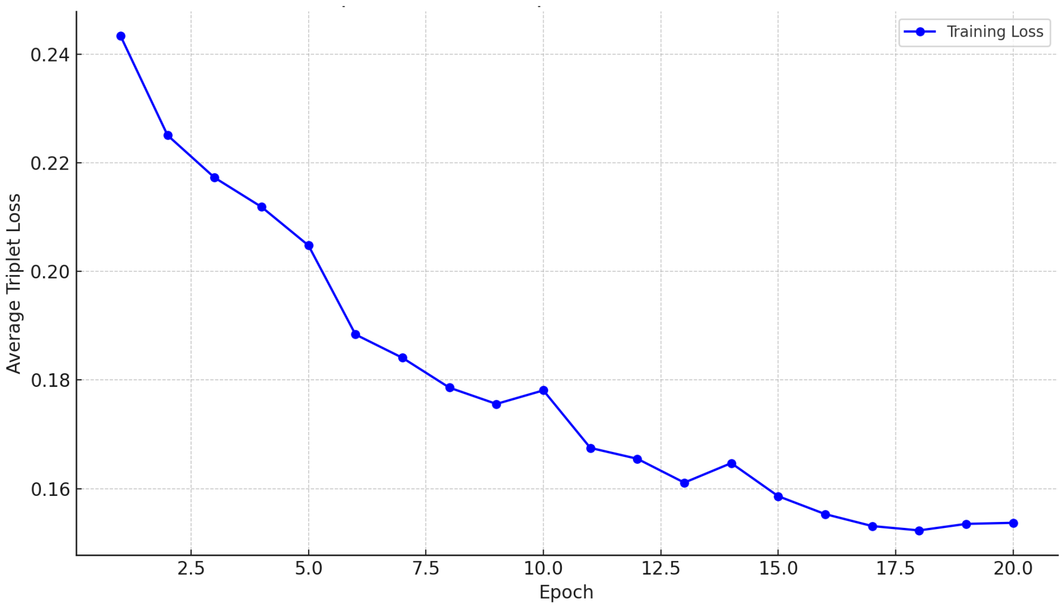

Figure 5.6: Average triplet loss over 20 epochs during training of the final retrieval model.

5.5 Multitask Model Performance

5.5.1 Multitask Model 1: Unified Training

The first multitask model was trained using a unified strategy in which classification and retrieval losses were optimized simultaneously throughout the training process. The classification component combined categorical cross-entropy loss for category prediction and binary cross-entropy for multi-label attribute recognition. For retrieval, a triplet margin loss was applied using batch-hard mining.

On the validation set, the best classification performance was achieved at epoch 11, with a top accuracy of 62.35% and a macro F1-score of 0.3040 (weighted F1 = 0.6130). Attribute recognition peaked with a lowest Hamming loss of 0.1369 (epoch 18) and a highest attribute micro-F1 of 0.7371 (epoch 18). Retrieval performance reached its maximum at epoch 18, with Recall@1 = 15.77% and Recall@5 = 31.00%.

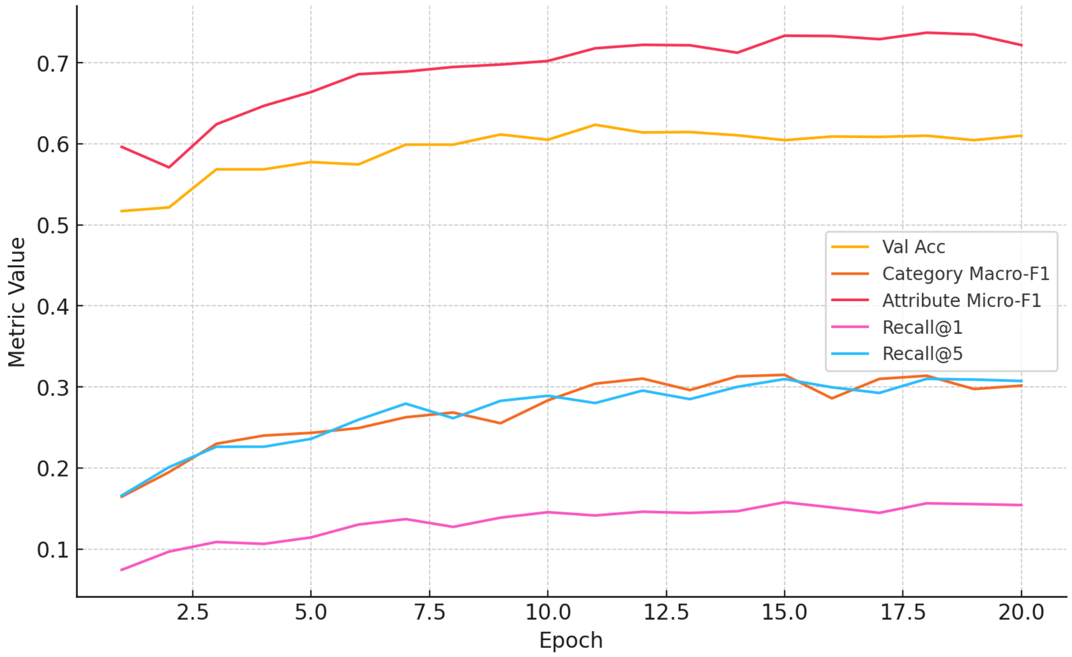

Figure 5.7: Validation metrics over 20 epochs for the unified multitask model: category accuracy, category macro-F1, attribute micro-F1, and retrieval Recall@1 and Recall@5.

5.5.2 Multitask Model 2: Staged Training Pipeline

The second multitask model was trained in three sequential phases: retrieval pretraining, classification pretraining, and joint fine-tuning.

Stage 1: Retrieval Pretraining The retrieval head was trained independently for five epochs using triplet margin loss with batch-hard mining. Recall@1 improved from 40.8% to 62.3%, and Recall@5 from 60.5% to 80.9%.

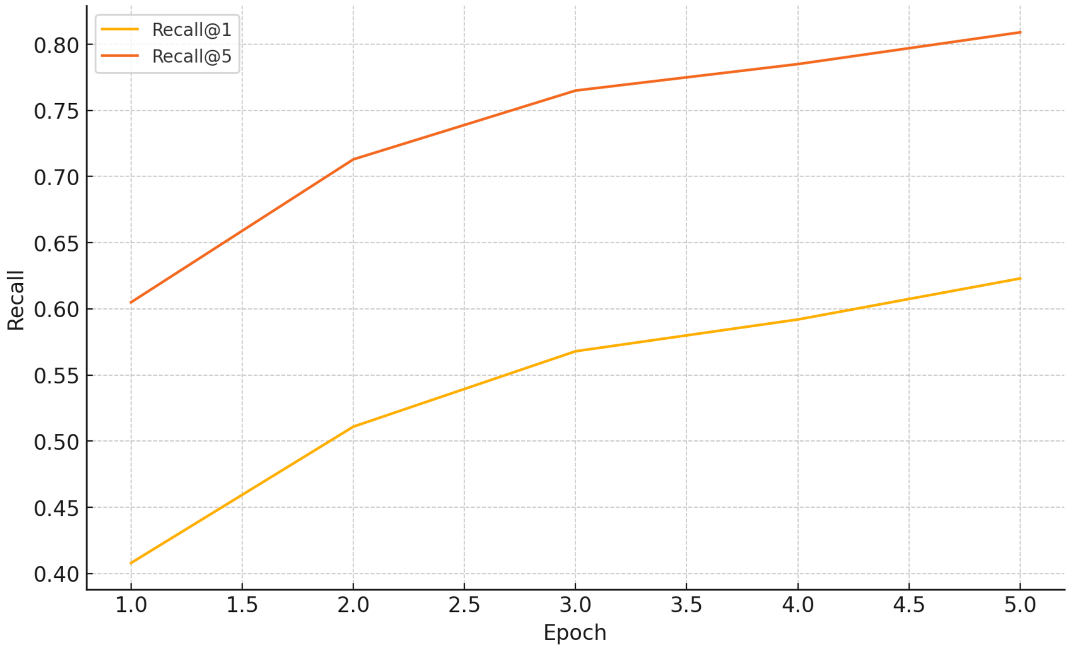

Figure 5.8: Retrieval performance (Recall@1, Recall@5) over 5 epochs during Stage 1 pretraining.

Stage 2: Classification Pretraining Next, the classification branch was trained for five epochs with categorical and binary cross-entropy losses. Validation accuracy rose from 62.10% to 66.30%, and macro F1 from 0.2587 to 0.3205 (weighted F1 = 0.6445). Attribute Hamming loss decreased to 0.1515, with micro-F1 reaching 0.7157.

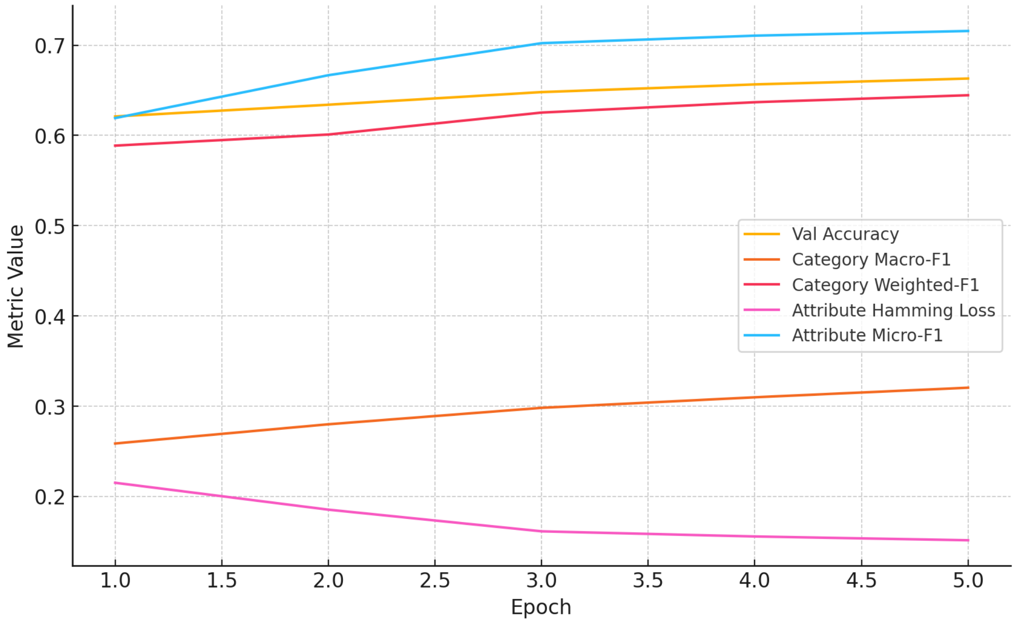

Figure 5.9: Stage 2 classification pretraining: combined validation metrics over 5 epochs, including category accuracy, category macro-F1, category weighted-F1, attribute Hamming loss, and attribute micro-F1.

Stage 3: Joint Fine-tuning Finally, both branches were fine-tuned together over 20 epochs. The best validation performance was observed at epoch 14, with category accuracy reaching 65.20 %, a category macro-F_1 of 0.3633 (weighted-F_1 = 0.6395), and attribute Hamming loss dropping to 0.1289 (epoch 11) alongside an attribute micro-F_1 of 0.7509 (epoch 11). Retrieval performance peaked with Recall@1 = 68.34 % at epoch 14 and Recall@5 = 86.28 % at epoch 17.

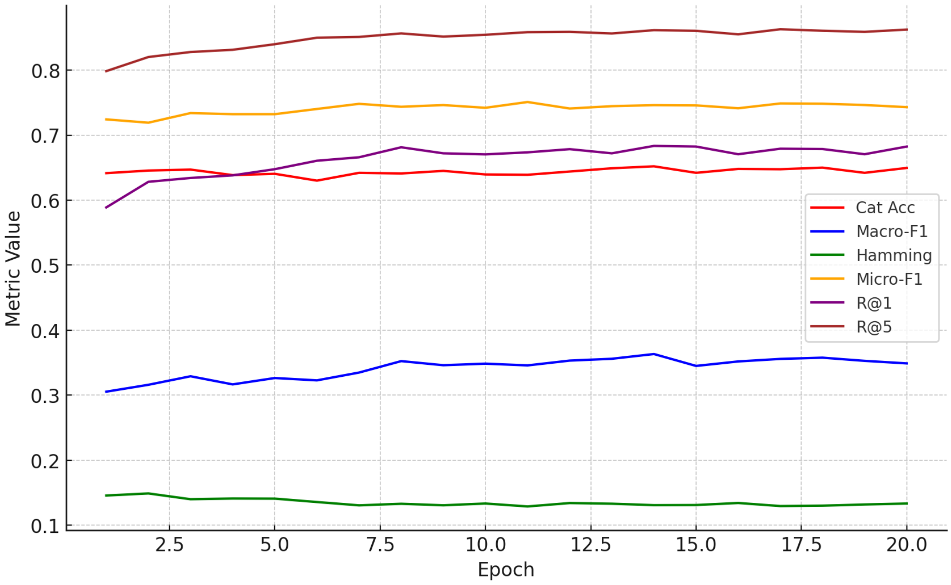

Figure 5.10: Validation metrics over 20 epochs for the staged multitask model during joint fine-tuning: category accuracy, category macro-F1, attribute hamming loss, attribute micro-f1, retrieval Recall@1 and Recall@5.

The full table of detailed validation metrics for each task over each epoch can be seen in Table 6 (in Appendix 7).

5.6 Final Evaluation on Test Set

Finally, I am in the position to systematically compare the five models, which represents the main result. In order to assess the generalization performance of each approach, I evaluate all models on the held-out test set and summarize their results in Table 5.2. This unified comparison highlights how multitask training strategies impact both classification and retrieval performance under identical test conditions.

Table 5.2: Performance comparison of single-task and multi-task models across all metrics.

Model	**Acc**	**Macro-F1**	**W-F1**	**Ham.**	**Micro-F1**	**R@1**	**R@5**
Single classification only	62.78	0.2965	0.6119	—	—	—	—
Single attribute only	—	—	—	0.1870	0.6636	—	—
Multi classification + attribute	61.75	0.3008	0.6059	0.1124	0.8876	—	—
Single retrieval only	—	—	—	—	—	0.7289	0.8768
Unified multitask	57.80	0.2940	0.5679	0.1463	0.7196	0.1542	0.3073
Staged multitask	62.12	0.2978	0.6062	0.1305	0.7468	0.6834	0.8623

Chapter 6

Discussion

The experiments in this thesis reveal a multifaceted picture of how single-task and multitask models behave on fashion categorization and visual retrieval, and what practical trade-offs they entail.

6.1 Model Performance and Metric Behavior

The category-only classifier achieved a top-1 accuracy of about 64% and a macro-F_1 of 0.34. These moderate scores are largely driven by the long-tailed distribution of fashion categories: very common items like “tee” or “dress” are learned easily, while rare items (for example “coverup” or “jumpsuit”) remain difficult, dragging down the macro average. The ResNet-50 backbone fine-tuned with class-balanced sampling helps mitigate the imbalance, but cannot eliminate it entirely without additional data augmentation or oversampling strategies.

The attribute-only model converged faster (around epoch 6) and reached a micro-F_1 of 0.66 with Hamming loss near 0.19. Here, frequent attributes—solid colors, long sleeves—achieved high precision and recall, whereas low-frequency attributes (e.g., square necklines, fringe details) lagged behind. This suggests that, for sparse labels, simple per-attribute loss weighting can only go so far; more sophisticated imbalance remedies (such as focal loss or attribute-specific data augmentation) might further improve rare-attribute recall.

For retrieval alone, batch-hard triplet mining nearly doubled Recall@1 compared with uniform sampling (from 36% to 73%). By focusing each mini-batch on the most challenging positive and negative pairs, the model learns a tighter embedding space where visually similar items cluster more reliably. This underlines that, in fine-grained image search, mining strategy has a larger impact than network depth or margin tuning.

6.2 Multitask Classification: Category and Attribute

Before exploring the joint challenge of classification and retrieval, I first examine multitask learning that combines category and attribute prediction. In this setup, a single ResNet-50 backbone feeds two heads: one for discrete category labels and another for multi-label attribute outputs. On held-out tests, the category-only model achieved 62.78 % top-1 accuracy, a macro-F1 of 0.2965, and a weighted-F1 of 0.6119. The attribute-only model reached a Hamming loss of 0.1870 and micro-F1 of 0.6636.

When the backbone is trained jointly, performance shifts: category accuracy falls slightly to 61.75 %, macro-F1 rises to 0.3008, and weighted-F1 dips to 0.6059, while attribute performance improves dramatically with Hamming loss dropping to 0.1124 and micro-F1 climbing to 0.8876. These results indicate that shared feature learning acts as an implicit regularizer. By forcing the network to capture fine-grained texture, shape, and color cues for attribute detection, it also enhances discrimination of infrequent category classes (hence the slight macro-F1 gain), even though the overall category boundary precision softens marginally.

From a practical standpoint, this multitask approach halves storage and inference requirements compared to running two separate networks. Joint training consumed only about 1.2× the compute time of a single task rather than the 2× that two independent trainings would require, thanks to parallel gradient updates. Thus, multitask classification offers a compelling trade-off: a modest 1.03 % drop in top-1 accuracy and minor shifts in F1 scores are more than offset by a dramatic 33 % reduction in Hamming loss for attributes, along with significant savings in deployment cost and latency.

6.3 Multitask Learning: Classification, Attribute and Retrieval

In extending the multitask framework to include retrieval alongside category and attribute prediction, I evaluated two training regimes—unified (simultaneous) multitask and staged multitask—against the single-task baselines. The unified model, trained from scratch on all three objectives, suffered severe negative transfer: top-1 accuracy dropped to 57.80%, macro-F1 to 0.2940 and weighted-F1 to 0.5679. Attribute performance (Hamming loss 0.1463, micro-F1 0.7196) and retrieval (Recall@1 = 0.1542, Recall@5 = 0.3073) both fell dramatically compared to their dedicated counterparts. This collapse can be attributed to conflicting gradient signals: the cross-entropy loss pulls the backbone toward distinct category clusters, the multi-label loss pushes for fine-grained feature sensitivity, and the triplet loss drives continuous embedding geometry. Without explicit balancing or gradient decorrelation, the network converges to representations that satisfy none of the tasks particularly well.

By contrast, the staged multitask model recovered nearly all single-task performance by decoupling learning into phases. After pretraining the classification and attribute heads separately, I fine-tuned the entire network jointly. The result was a unified model achieving 62.12% accuracy, macro-F1 = 0.2978, weighted-F1 = 0.6062; Hamming loss = 0.1305, micro-F1 = 0.7468; and Recall@1 = 0.6834, Recall@5 = 0.8623. Sequential pretraining preserves specialized features—discriminative edges for categories, fine-grained textures for attributes, and metric structure for retrieval—before blending them, thereby mitigating destructive interference.

When comparing the staged multitask model to all alternatives, several patterns emerge. Classification accuracy and F1 scores nearly match the single-task classifier (62.12% vs. 62.78%; macro-F1 0.2978 vs. 0.2965), while attribute micro-F1 jumps to 0.7468 (versus 0.6636) and retrieval Recall@1 reaches 0.6834 (versus 0.7289). The small 0.66% drop in retrieval performance is outweighed by gains in attribute and classification synergy. Against the simpler classification+attribute multitask (61.75% accuracy, macro-F1 = 0.3008, micro-F1 = 0.8876), adding retrieval preserves category metrics and yields an embedding capable of visual search. In other words, the staged triple-task model offers a balanced solution: it performs nearly as well as specialized networks on each task, while sharing a single backbone and incurring only a modest 1.3× overhead over one task's training time.

Overall, staged multitask learning demonstrates that with careful sequencing, a single architecture can deliver strong categorization, fine-grained attribute tagging, and high-quality retrieval embeddings. The practical implication is clear: a unified deployment supports automated labeling and visual search in one footprint, reducing storage, inference latency, and maintenance costs—all without sacrificing the predictive power of dedicated models.

6.4 Resource Efficiency and Economic Implications

6.4.1 Computational Efficiency

Beyond accuracy metrics, multitask models offer practical efficiency gains in training time, storage, and deployment complexity. Training two separate ResNet-50 models roughly doubles GPU hours compared to a single shared backbone. Similarly, serving two independent models doubles inference latency and memory usage. In contrast, the staged multitask model developed here consolidates resources: fine-tuning jointly takes approximately 1.3× the time of a single-task model (including both pretraining phases), and adds only marginal overhead during inference. These technical efficiencies could translate into reduced compute costs and faster response times in real-world applications.

6.4.2 Economic Impact on Digital Marketplaces

One of the additional takeaways from my results is that multitask models—especially when trained in a staged way—don't just make technical sense, they also support the broader goal of improving how users interact with online fashion platforms. My best-performing model doesn't just predict categories or attributes or find similar images, it does all three reasonably well in one pipeline. And while this seems like a machine learning achievement, it also ties back to something more fundamental: making it easier for people to find what they're looking for.

This matters because search frictions like missing tags, poor metadata, or low-quality listings can make users give up early or settle for products that aren't quite right. Economic theory shows that even small barriers like these can distort competition and lead to worse outcomes for both buyers and sellers.[1] In my case, adding automated visual tagging and retrieval seems to reduce some of that friction. For example, the sharp improvement in attribute micro-F1 and solid retrieval Recall@1 scores suggest that the model can surface more detailed, visually relevant matches—potentially helping users discover items they might have missed with keyword search alone.

This is especially relevant for secondhand platforms, where listing quality is uneven and many sellers don't label items properly[2]. A system like the one built in this thesis could potentially fill in those gaps behind the scenes: suggesting tags, clustering similar products, and enabling more visual exploration—all of which make the search process smoother. That's not just helpful technically; it supports better matches and broader exposure for sellers who might otherwise be overlooked.

And if we think of platforms as more than just marketplaces—as designers of the user experience—then this kind of model is part of that design. Research shows that when platforms change how products are organized or surfaced, it changes what users see and buy.[3] While I don't test user outcomes here, my results support the idea that smarter visual systems can help structure the market more effectively—giving users more relevant results, and giving sellers a fairer shot at being found.

6.5 Limitations and Future Directions

This thesis faced several methodological and practical limitations that constrain the generality of its conclusions. First, the evaluation relies on a single train/validation/test split; no cross-validation was performed. While the dataset is large, this may conceal variance in performance, and future work should include k-fold validation to estimate stability more robustly. Second, the architecture search was limited to a ResNet-50 backbone. While effective and widely used, newer models such as vision transformers or more efficient CNNs may yield different trade-offs in accuracy, training time, and deployment cost.

[1] Diamond (1982); Ellison and Ellison (2009); Koulayev (2014)

[2] Deldjoo et al. (2023)

[3] Athey and Luca (2019); Dagan et al. (2023); Zheng et al. (2023)

Training multitask models also introduced significant optimization challenges. Balancing classification and retrieval losses required manual tuning, and performance was sensitive to hyperparameter settings. Future research could explore dynamic loss-weighting strategies—such as uncertainty-based methods—to automate this process and potentially improve convergence. Additionally, the dataset's long-tail class distribution points to further directions, including few-shot learning for rare categories and attribute hierarchies to improve macro-level metrics.

In terms of data, the DeepFashion dataset provides high-quality, curated images that may not reflect the variability seen in real-world secondhand marketplaces, where images are user-uploaded and often have poor lighting or cluttered backgrounds. As such, the results achieved here may not fully generalize to production environments without further domain adaptation or robustness testing.

Finally, the economic analysis presented in this thesis is conceptual and not empirically validated. While the model is motivated by its potential to reduce search frictions and improve product discoverability, no user studies, A/B tests, or platform-integrated evaluations were conducted. Likewise, important commercial factors such as price, brand perception, or user personalization were outside the scope of this study.

Despite these limitations, this work serves as a foundation for future development of unified classification and retrieval systems in fashion. With further tuning, broader validation, and eventual deployment testing, such systems could meaningfully improve both platform efficiency and user experience in visually driven marketplaces.

Chapter 7

Conclusion

This thesis explored the development of a multitask learning model for fashion product classification and image retrieval using the DeepFashion dataset. The aim was to integrate category prediction, attribute recognition, and image-based retrieval into a single system, with potential applications for improving product discovery on online fashion platforms.

Technically, the individual models for each task performed reasonably well. The multitask models, particularly those trained in staged phases, showed some promise but also highlighted the difficulties of optimizing competing objectives within a shared architecture. The results suggest that while joint training is feasible, careful design and tuning are essential to avoid performance trade-offs between tasks. Future work could build on this foundation by experimenting with alternative loss functions, improved sampling strategies, or more diverse datasets to better support real-world applications.

From an economic perspective, this work connects visual recognition tools to broader issues in digital marketplaces—namely, search frictions and inefficient product discovery. In contexts like fashion platforms, where metadata is often incomplete or inconsistent, models that support visual tagging and similarity-based retrieval can help lower the effort required for consumers to find relevant items. While this thesis does not evaluate market outcomes directly, it frames machine learning tools as part of the digital infrastructure that shapes buyer experience and platform efficiency.

Overall, the project provides a technical and conceptual foundation for future research at the intersection of computer vision and market design. While the results are preliminary, they point toward a broader role for multitask models in supporting more accessible, searchable, and efficient fashion resale platforms.

Appendix

Model Label Definitions

Attribute Labels

Attribute Name	Type	Attribute Name	Type
floral	1	long_sleeve	2
graphic	1	short_sleeve	2
striped	1	sleeveless	2
embroidered	1	maxi_length	3
pleated	1	mini_length	3
solid	1	no_dress	3
lattice	1	crew_neckline	4
denim	5	v_neckline	4
chiffon	5	square_neckline	4
cotton	5	no_neckline	4
leather	5	tight	6
faux	5	loose	6
knit	5	conventional	6

Table 1: Fashion attributes grouped by type.

Category Labels

Category Name	Type	Category Name	Type
Anorak	1	Cape	3
Blazer	1	Coat	3
Blouse	1	Coverup	3
Bomber	1	Dress	3
Button-Down	1	Jumpsuit	3
Cardigan	1	Kaftan	3
Flannel	1	Kimono	3
Halter	1	Nightdress	3
Henley	1	Onesie	3
Hoodie	1	Robe	3
Jacket	1	Romper	3
Jersey	1	Shirtdress	3
Parka	1	Sundress	3
Peacoat	1	Capris	2
Poncho	1	Chinos	2
Sweater	1	Culottes	2
Tank	1	Cutoffs	2
Tee	1	Gauchos	2
Top	1	Jeans	2
Turtleneck	1	Jeggings	2
Caftan	3	Jodhpurs	2
		Joggers	2
		Leggings	2
		Sarong	2
		Shorts	2
		Skirt	2
		Sweatpants	2
		Sweatshorts	2
		Trunks	2

Table 2: Fashion categories grouped by type.

Additional Results

Table 3: Validation metrics for category classification at epochs with new best macro-F1

Epoch	Avg Train Loss	Val Acc	Val Macro-F1
1	1.9341	0.5095	0.1635
2	1.5946	0.5780	0.2383
3	1.4497	0.5780	0.2481
4	1.3459	0.5890	0.2541
6	1.1101	0.6170	0.2775
7	1.0169	0.6280	0.3119
8	0.8718	0.6395	0.3244
9	0.7502	0.6380	0.3420

Table 4: Validation Hamming loss and micro-F1 across all training epochs for the attribute prediction model

Epoch	Hamming Loss	Micro-F1
1	0.3459	0.4780
2	0.2592	0.5664
3	0.2694	0.5622
4	0.2962	0.5085
5	0.2172	0.6209
6	**0.1883**	**0.6606**
7	0.2203	0.6203
8	0.2010	0.6414
9	0.2054	0.6315
10	0.1898	0.6564

Full Classification Report

Table 5: Per-class precision, recall, F1-score, and support on the validation set (category-only model, epoch 9).

Class Index	Precision	Recall	F1-Score	Support
0	0.0000	0.0000	0.0000	1
1	0.5556	0.4717	0.5102	53
2	0.4444	0.3846	0.4124	156
3	0.0000	0.0000	0.0000	3
4	0.0000	0.0000	0.0000	0
5	0.3960	0.4706	0.4301	85
6	0.0000	0.0000	0.0000	0
7	0.0000	0.0000	0.0000	0
8	0.2500	0.1250	0.1667	8
9	0.4400	0.4583	0.4490	24
10	0.4000	0.4667	0.4308	60
11	0.0000	0.0000	0.0000	4
12	0.5000	0.2000	0.2857	5
13	0.0000	0.0000	0.0000	1
14	0.0000	0.0000	0.0000	9
15	0.4872	0.5067	0.4967	75
16	0.5234	0.5333	0.5283	105
17	0.6164	0.6714	0.6427	213
18	0.2113	0.2000	0.2055	75
19	0.0000	0.0000	0.0000	1
20	0.0000	0.0000	0.0000	0
21	0.7500	0.6000	0.6667	5
22	0.0000	0.0000	0.0000	3
23	0.0000	0.0000	0.0000	10
24	0.0000	0.0000	0.0000	0
25	0.8125	0.7358	0.7723	53
26	0.0000	0.0000	0.0000	3
27	0.0000	0.0000	0.0000	0
28	0.3182	0.2917	0.3043	24
29	0.5952	0.7812	0.6757	32
30	0.0000	0.0000	0.0000	0
31	0.6212	0.6949	0.6560	118
32	0.7103	0.6786	0.6941	112
33	0.3889	0.4118	0.4000	17
34	0.1667	0.0909	0.1176	11
35	0.0000	0.0000	0.0000	4
36	0.0000	0.0000	0.0000	0
37	0.0000	0.0000	0.0000	0
38	0.5556	0.2941	0.3846	17

Continued on next page

Table 5 – continued from previous page

Class Index	Precision	Recall	F1-Score	Support
39	0.0000	0.0000	0.0000	0
40	0.8622	0.9134	0.8871	589
41	0.7045	0.6596	0.6813	47
42	0.0000	0.0000	0.0000	0
43	0.3125	0.2381	0.2703	21
44	0.0000	0.0000	0.0000	0
45	0.0000	0.0000	0.0000	0
46	0.0000	0.0000	0.0000	2
47	0.6731	0.6481	0.6604	54

Full Multitask validation Metrics

Epoch	Cat Acc	Macro-F_1	Hamming	Micro-F_1	R@1	R@5
1	0.6415	0.3054	0.1457	0.7242	0.5888	0.7984
2	0.6455	0.3160	0.1489	0.7191	0.6282	0.8201
3	0.6470	0.3291	0.1400	0.7339	0.6341	0.8278
4	0.6385	0.3166	0.1411	0.7322	0.6381	0.8312
5	0.6405	0.3264	0.1409	0.7322	0.6476	0.8398
6	0.6300	0.3228	0.1357	0.7402	0.6606	0.8498
7	0.6420	0.3348	0.1306	0.7481	0.6659	0.8510
8	0.6410	0.3524	0.1330	0.7436	0.6813	0.8564
9	0.6450	0.3461	0.1307	0.7462	0.6720	0.8515
10	0.6395	0.3485	0.1334	0.7420	0.6704	0.8543
11	0.6390	0.3458	0.1289	0.7509	0.6735	0.8583
12	0.6440	0.3532	0.1341	0.7409	0.6784	0.8588
13	0.6490	0.3560	0.1331	0.7445	0.6720	0.8563
14	0.6520	0.3633	0.1309	0.7461	0.6834	0.8614
15	0.6420	0.3450	0.1311	0.7457	0.6824	0.8604
16	0.6480	0.3519	0.1342	0.7413	0.6706	0.8550
17	0.6475	0.3558	0.1295	0.7487	0.6791	0.8628
18	0.6500	0.3577	0.1301	0.7483	0.6786	0.8605
19	0.6420	0.3528	0.1319	0.7463	0.6706	0.8588
20	0.6495	0.3489	0.1334	0.7430	0.6824	0.8623

Table 6: Stage-3 joint training metrics over 20 epochs

Bibliography

A.Vijayaraj Alwarsamy, Vasanth Pt, Jebakumar Rethnaraj, P. Senthilvel, N. Kumar, R. Kumar, and R. Dhanagopal. Deep learning image classification for fashion design. *Wireless Communications and Mobile Computing*, 2022:1–13, 06 2022. doi: 10.1155/2022/7549397.

Muhammad Amin, Changbo Wang, and Summaira Jabeen. Fashion sub-categories and attributes prediction model using deep learning. *The Visual Computer*, 39, 06 2022. doi: 10.1007/s00371-022-02520-3.

Susan Athey and Michael Luca. The economics of technology and the media. In *Handbook of Industrial Organization*. Elsevier, 2019.

Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. volume 8689, 04 2014. ISBN 978-3-319-10589-5. doi: 10.1007/978-3-319-10590-1_38.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Suryanaman Chaube, Rijula Kar, Sneh Gupta, and Mayank Kant. Multimodal ai framework for the prediction of high-potential product listings in e-commerce: Navigating the cold-start challenge. *Expert Systems with Applications*, 282:127524, 04 2025. doi: 10.1016/j.eswa.2025.127524.

Roberto Cipolla, Yarin Gal, and Alex Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. pages 7482–7491, 06 2018. doi: 10.1109/CVPR.2018.00781.

Arnon Dagan, Ido Guy, and Slava Novgorodov. Shop by image: characterizing visual search in e-commerce. *Information Retrieval Journal*, 26, 03 2023. doi: 10.1007/s10791-023-09418-1.

Ritendra Datta, Dhiraj Joshi, Jia Li, and James Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40, 01 2008.

Yashar Deldjoo, Fatemeh Nazary, Arnau Ramisa, Julian McAuley, Giovanni Pellegrini, Alejandro Bellogín, and Tommaso Di Noia. A review of modern fashion recommender systems. *ACM Computing Surveys*, 56, 09 2023. doi: 10.1145/3624733.

Peter A. Diamond. Aggregate demand management in search equilibrium. *Journal of Political Economy*, 90(5):881–894, 1982.

Glenn Ellison and Sara Fisher Ellison. Search, obfuscation, and price elasticities on the internet. *Econometrica*, 77(2):427–452, 2009.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742. IEEE, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016. doi: 10.1109/CVPR.2016.90.

Wei-Hong Hsiao and Kristen Grauman. Learning the latent "look": Unsupervised discovery of a style-coherent embedding from fashion images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4213–4221, 2017.

Menglin Jia, Yichen Zhou, Mengyun Shi, and Bharath Hariharan. A deep-learning-based fashion attributes detection model, 10 2018.

Sergei Koulayev. Search for differentiated products: Identification and estimation. *The RAND Journal of Economics*, 45(3):553–575, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012. doi: 10.1145/3065386.

Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Alan Mafla, Juan C Valenzuela, Sebastian Rios, Cristian Parra, Cristian Rodriguez, Cristian Calderon, and Cristian Lagos. Fine-grained image classification and retrieval by combining visual and locally aggregated textual features. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1741–1750, 2020.

Yair Movshovitz-Attias, Alexander Toshev, Thomas Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, pages 360–368, 2017.

D. Ngwe, K. J. Ferreira, and T. Teixeira. The impact of increasing search frictions on online shopping behavior: Evidence from a field experiment. *Journal of Marketing Research*, 56 (6):944–959, 2019. doi: 10.1177/0022243719865516. Original work published 2019.

Hyeonwoo Noh, Andre Araujo, Joonhyung Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3456–3465, 2017.

Sanghyuk Park, Minchul Shin, Sungho Ham, Seungkwon Choe, and Yoohoon Kang. Study on fashion image retrieval methods for efficient fashion visual search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823. IEEE, 2015.

Seong-Yoon Shin, Gwanghyun Jo, and Guangxing Wang. A novel method for fashion clothing image classification based on deep learning. *Journal of Information and Communication Technology*, 22:127–148, 01 2023. doi: 10.32890/jict2023.22.1.6.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ICML*, 2019.

Yuxin Tian, Shawn Newsam, and Kofi Boakye. Fashion image retrieval with text feedback by additive attention compositional learning. pages 1011–1021, 01 2023. doi: 10.1109/WACV 56688.2023.00107.

Yao Wan, Vineet Kumar, and Xitong Li. How do product recommendations help consumers search? evidence from a field experiment. *Management Science*, 70(3):1344–1368, 2024. doi: 10.1287/mnsc.2023.4951.

Jian Wei and Zhaoqiang Wang. Food image recognition by using convolutional neural networks with multi-task learning. *Journal of Electronic Imaging*, 29(1):013021, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. 08 2017. doi: 10.48550/arXiv.1708.07747.

Lingxi Xie, Jingdong Wang, and Qi Tian. Image classification and retrieval are one. In *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR)*, pages 3–10, 2015.

Yuanhao Yang, Jia Wu, Haifeng Lin, Ming Li, and Yanyan Li. Retrieving and classifying affective images using deep metric learning with texture-based sentiment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Shunyuan Zhang, Dokyun Lee, Param Singh, and Kannan Srinivasan. What makes a good image? airbnb demand analytics leveraging interpretable image features. *Management Science*, 68, 12 2021. doi: 10.1287/mnsc.2021.4175.

Yichao Zhang, Qijie Zhao, Zhipeng Sun, Liwei Zhang, Yang Song, Yifan Liu, Yunchao Liu, and Junchi Yan. Tasks integrated networks for joint object detection and retrieval. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1538–1546, 2019.

Shuai Zhao, Yiqun Liu, Min Zhang, and Shaoping Ma. Modulation for task-aware feature sharing in multi-task learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 401–416, 2018.

Shuang Zheng, Siliang Tong, Hyeokkoo Kwon, Gordon Burtch, and Xianneng Li. Recommending what to search: Sales volume and consumption diversity effects of a query recommender system. *SSRN Electronic Journal*, 01 2023. doi: 10.2139/ssrn.4667778.