Classifying Fashion Products with ResNet-50 and Confidence-Based Fallback Mechanisms

Aswathy Gopalakrishnan
Department of AI Engineering
Jönköping University
goas24ee@student.ju.se

Princy Bindhu
Department of AI Engineering
Jönköping University
bipr24za@student.ju.se

Yonge Li
Department of AI Engineering
Jönköping University
liyo23ac@student.ju.se

***Abstract*—This report details the development and evaluation of a hierarchical fashion image classification system designed for the SlowFashion platform, a company committed to sustainable and circular fashion. Leveraging a modified ResNet-50 architecture with dual classification heads, the system categorizes images into broad classes such as clothing, shoes, and bags, and further refines them into 9 specific subcategories. Dataset balancing, stratified sampling, and weighted loss functions were employed to address class imbalance, which significantly enhanced the model's ability to recognize both common and rare categories. The training process involved advanced augmentation techniques, adaptive learning rate scheduling, and robust data handling strategies. The model's performance was comprehensively evaluated using multiple metrics, including accuracy, precision, recall, confusion matrices, and ROC curves, across different training phases. Additionally, a practical inference prototype was developed for real-time predictions, incorporating confidence-based fallback strategies to manage uncertainty and out-of-scope images. The results demonstrate a highly accurate, robust, and scalable solution aligned with SlowFashion's objectives to provide trustworthy and efficient fashion item classification, supporting sustainable e-commerce practices. Future work aims to further improve rare class detection and out-of-distribution handling for enhanced real-world deployment.**

***Index Terms*—Fashion Image Classification, Hierarchical Classification, ResNet-50, CNN, Transfer Learning, Confidence Threshold**

I. Introduction

SLOWFASHION, a company committed to sustainable and circular fashion, have created a platform that makes buying and selling high quality luxury products effortless and trustworthy. In order to further support this initiative, and to enhance the user experience, an AI based classification system has been developed to recognize and categorize the fashion items within images.

Given the hierarchical nature of fashion where main categories such as 'Clothing' or ''shoes' encompass more specific subcategories such as 'Dresses' or 'Flats', a hierarchical model has been chosen. This approach allows for precise identification of the most specific categories, while allowing space for broader categorization in cases of uncertainty. When the system cannot confidently assign a specific category, it defaults to a general label, "unknown', thereby maintaining the accuracy of the model. The layered classification strategy ensures that the model can deliver practical and reliable results, supporting Slow fashion's goal to make the listing trustworthy.

The model was initially trained with the DeepFashion dataset and later the performance was tested on the saved model using the company provided dataset. The initial results demonstrate that hierarchical model effectively distinguishes between major categories like clothing and shoes, even when the subcategories are ambiguous. Fine tuning the system further improved the classification accuracy into the subcategories as well, but not as accurate as the main categories. However, the fallback mechanism to broader group categories proved useful in handling uncertain cases ensuring consistent outputs. Overall, the evaluation metrics and results of the model indicate that hierarchical approach offers a robust solution in classification tasks, and it aligned well with Slow fashion's objectives.

II. Related Works

Several recent studies have addressed the fashion image classification problem, and the effectiveness hierarchical modelling and fine tuning on the model. Here, we discuss the papers that have motivated us in the chosen approach.

The first paper, "Towards Fashion image Annotation: A Clothing Category Recognition procedure [1] explores methodologies for automating fashion image annotation using deep learning to improve the efficiency of clothing category recognition. The authors have used the DeepFashion and I Materialist datasets, and investigated two well established architectures, VGG and ResNet, along with a variation of ResNet. Their strong comparison of models provided valuable insights, influencing our choice of ResNet architecture for its proven efficacy. The paper's demonstration of how transfer learning and fine tuning can effectively be deployed further impacted our approach.

The paper, "Fashion Image Classification Using Deep Convolution Neural Network [2]", investigates the use of convolutional neural networks (CNNs) for fashion data classification. The paper focuses on enhancing the test accuracy using the Adam optimizer which has been adapted in our approach to improve the performance and efficiency of the model. This paper reinforces our conviction that using CNN architectures yield high accuracy, particularly in feature extraction, making it well suited for the problem in hand.

"Deep Residual Learning for Image Recognition [3]" is a paper that details the exploration of ResNet architectures and their ability to overcome the degradation problem usually observed in deep networks. This helped the team understand the pre-trained ResNet frameworks, enabling us to customize and extend it to the needs of the specific task of classifying fashion images, using two additional classification heads and uncertainty handling.

III. Background

The project utilizes a range of machine learning and image processing techniques to develop the classification tool tailored for Slowfashion's requirements.

Deep Learning and Convolutional Neural Networks (CNNs): CNNs form the foundation of the model as it excels at automatically learning hierarchical features from raw pixel data. A pre-trained ResNet-50 model was selected for its proven efficacy in image recognition and ability to handle complex visual patterns.

Hierarchical Classification: Given the nature of the fashion categories and the requirements of the project, a hierarchical classification approach was chosen. This involved modifying ResNet-50 architecture with two classification heads.

- **Main classifier**: Predicts the main class (Clothing, shoes, Bags)
- **Sub-Classifier**: Predicts specific subcategory within the assigned main class (Dresses, High heels, Shoulder bags).

This allows the model to first categorize into a broader group and then refine the classification to a specific subcategory.

Balanced Data Loading: Since the dataset consists of varying number of images for each category, a balanced data loader was implemented to ensure fair representation across all categories of data. Each clothing category was limited to a maximum of 2000 samples to prevent bias towards overrepresented categories. Stratified sampling was also used to ensure consistent class distributions in training and validation sets.

Adam optimizer: The adaptive Moment estimation algorithm was chosen for its efficiency in deep learning models. Adam adapts the learning rates for individual parameters based on their historical gradients. This particularly advantageous in the case of ResNet-50 model where different layers and parameters may require varying degrees of adjustments during fine tuning. By this, Adam helps to accelerate convergence, prevent oscillations, and handle complex parameters effectively.

ReduceLROnPlateau Scheduler: In order to reduce overfitting and ensure optimal training, a ReduceLROnPlateau scheduler was utilized. This monitors the validation accuracy during training and dynamically adjusts the learning rate based on the performance. In cases where the accuracy plateaus for a set number of epochs, it reduces the learning rate by a predefined factor, this helps prevent stagnation and allows model to fine tune the parameters precisely as it approaches convergence.

Weighted Cross-Entropy Loss: To address potential imbalance issues, a weighted cross-entropy loss function is used. Cross entropy is a standard measure of the difference between predicted and actual class distributions. By assigning different weights to each class, the model can be penalized heavily for misclassification of underrepresented classes. This encourages it to learn more robust and balanced representations.

Others/Unknown Class and Confidence thresholding: To mitigate the misclassification images outside the 9 categories an 'Others/unknown' class has been introduced. A confidence threshold has been set, and if the confidence score for a predicted class falls below the defined threshold, the image is assigned to 'Others/unknown' class.

These methods, collectively form a robust and adaptable framework for fashion image classification.

IV. Data Exploration

During the data exploration stage, the process was typically divided into three main sections – Data collection, data understanding, data cleaning, and data augmentation.

A. Data Collection

A crucial step in the project was the thorough analysis of the dataset used for training and evaluation to uncover potential issues, and guide subsequent data preprocessing and augmentation steps.

The primary dataset used was DeepFashion2, which includes images across 13 clothing categories. To expand the dataset for categories like shoes and bags, which were underrepresented or missing, supplementary data was collected using a custom web crawler called iCrawler. An initial attempt to use images from the MNIST dataset for bags was made but was found to be of poor quality, prompting a shift to collecting images directly from the web. This process resulted in a final dataset comprising over 39,000 images for dresses, as well as images for outerwear, skirts, and various subcategories of shoes and bags such as boots, flats, high heels, clutches, shoulder bags, and tote bags.

The resulting dataset has the following composition: Clothing:

- Dresses: 39186
- Outerwear: 3713
- Skirts: 1921

Shoes:

- Boots: 308
- Flats: 484
- High Heels: 362

Bags:

- Clutches: 439
- Shoulder Bags: 306
- Tote Bags: 430

B. Data Organization

A custom script was developed to identify each image's category_id and organize the images into category-specific folders. The original 13 categories from DeepFashion2 were mapped and consolidated into main categories - Clothing, Shoes, and Bags, and further subdivided into relevant subcategories like Dresses, Outerwear, Skirts, etc. Some categories were merged based on relevance, while others were discarded to align with the project's focus. This organization simplified data handling and prepared the dataset for training.

C. Data Cleaning

During the cleaning phase, images were inspected, and categorization was verified to ensure labels matched visual content. Categories that were irrelevant or poorly labelled were merged or discarded. Supplementary images from the web crawler were reviewed for quality and consistency before inclusion. Due to the poor quality and high 'Unknown' categorization, the slow fashion dataset provided by the client was not used directly for training, though it was reviewed for potential future cleaning or annotation improvements. Overall, the cleaning process aimed to ensure that the dataset was well-organized, relevant, and of consistent quality for model training.

D. Data Augmentation

To improve the model's generalization capabilities, a series of data augmentation techniques are applied during training. These include random resizing and cropping with 'RandomResizedCrop' to produce input images of size 224×224, which helps the model learn from various scales and compositions. Geometric transformations are also used, such as horizontal and vertical flipping (RandomHorizontalFlip, RandomVerticalFlip) and rotation (RandomRotation), to simulate different viewing angles and orientations. Additionally, colour jittering (ColorJitter) to introduce variations in brightness, contrast, saturation, and hue, aiding the model in handling different lighting and colour conditions. Finally, images are normalized using the Normalize function with ImageNet mean and standard deviation values to standardize pixel intensities. For validation, resizing followed by centre cropping ensures that input images are consistent in size and appearance, providing a reliable evaluation setting.

V. Approach Description

This study proposes a hierarchical image classification framework designed to accurately identify both broad categories and their corresponding subcategories within fashion images. By integrating data balancing techniques, a specially designed model architecture, and comprehensive training and evaluation strategies, the approach aims to overcome challenges posed by class imbalance and data scarcity. The following sections detail the step-by-step processes involved in data handling, model development, and model architecture, illustrating how each component contributes to building a robust and practical classification system capable of functioning effectively in real-world applications.

A. Dataset Structuring and Imbalance Handling

As already discussed, the dataset was organized into a hierarchical folder structure, where each main class directory (e.g., Clothing, Bags, Shoes) contained several subclass subdirectories (e.g., Dresses, Skirts, Clutches). To maintain this hierarchy and address class imbalance, we implemented a stratified data pipeline that supports both structure traversal and controlled sampling.

The key function, 'stratified_split_dataset()', performs the following tasks:

- Constructs a mapping from each subclass to its corresponding main class
- Generates a subclass-to-index dictionary for label encoding
- Limits the number of samples per subclass based on a configurable threshold (e.g., max 2,000 samples for Dresses)
- Performs stratified sampling to split the dataset into training and validation sets, ensuring consistent subclass distribution

This strategy allows us to mitigate data imbalance, particularly between overrepresented categories like Dresses and underrepresented ones like Clutches or Boots, while preserving the hierarchical semantics required by the model.

B. Loss Function Design

To incorporate class imbalance directly into the learning process, we assign a higher penalty to misclassified samples from minority classes by using weighted cross-entropy losses. Specifically, we compute class weights for both the main and subclass prediction tasks using scikit-learn's compute_class_weight() function, which generates weights inversely proportional to class frequencies in the training set.

Given a batch of N samples, the weighted cross-entropy loss is formulated as:

Given a batch of N samples, the weighted cross-entropy loss is formulated as:

$$L_{\text{weighted}} = -\frac{1}{N}\sum_{i=1}^{N} w_{y_i} \cdot \log(p_{i,y_i})$$

where:

- $y_i \in \{1, \ldots, C\}$ is the ground-truth class label for the i-th sample,
- p_{i,y_i} is the predicted probability for the true class y_i,
- w_{y_i} is the weight associated with class y_i, derived from label frequencies.

This loss formulation is applied independently to both the main class head and the subclass head. The total objective function used during training is the sum of the two:

$$L_{\text{total}} = L_{\text{main}} + L_{\text{sub}}$$

By penalizing errors from underrepresented categories more heavily, this approach improves the model's ability to learn meaningful decision boundaries across all classes, regardless of frequency.

C. Model Architecture

In this project, a modified ResNet-50 backbone with a dual-head output structure was adopted as the model architecture.

*1) **ResNet-50**:* ResNet-50, short for Residual Network with 50 layers, is particularly suited for deep learning tasks due to its use of residual connections, which alleviate the vanishing gradient problem and enable stable training of very deep architectures.

ResNet-50 is composed of stacked residual blocks, where each block allows activations to bypass intermediate layers via shortcut paths. This structure facilitates more efficient gradient flow during backpropagation, making it easier to train deep models without degradation. Each residual block typically includes multiple convolutional layers followed by batch normalization and ReLU activation functions. Instead of relying on fully connected layers, ResNet-50 often uses global average pooling, which compresses feature maps into compact representations suitable for classification.

*2) **Hierachical Model Architecture**:* We leverage the pretrained ResNet-50 model (trained on ImageNet) and modify its architecture by removing the final classification layer. The extracted feature embeddings are then passed through two additional classifier heads.

- Main Classifier Head: Predicts one of three main categories (Clothing, Bags, Shoes).
- Subclass Classifier Head: Predicts one of nine fine-grained subcategories (Dresses, Skirts, Outerwear, Shoulder Bags, Tote Bags, Clutches, Flats, Boots, High Heels).

The shared ResNet backbone extracts high-level features from images. These are fed into two parallel fully connected layers for independent classification. This structure allows the model to learn both general concepts and fine-grained subcategories at the same time.

VI. Experimental Design

This section outlines the comprehensive process employed to train and evaluate the hierarchical classification model.

A. Training Strategy

The training strategy focuses on optimizing model performance through a combination of class imbalance mitigation techniques, adaptive learning rates, and robust data handling. To optimize training efficiency and address class imbalance, the following strategies were employed:

*1) **Weighted CrossEntropy Loss**:* Weighted CrossEntropyLoss was applied independently to both the main and subclass classification heads. As discussed earlier, this approach compensates for data imbalance by assigning higher loss weights to underrepresented classes, ensuring that the model pays proportionally more attention to minority categories during backpropagation

*2) **Differential Learning Rates with Adam Optimizer**:* Training employs the Adam optimizer with differential learning rates: a reduced rate (lr / 10) for the pretrained backbone to preserve learned features, and a higher rate for the classifier heads which are trained from scratch. This separation allows the backbone to adapt slowly while the new classifier layers learn rapidly and effectively from the task-specific dataset.

*3) **Adaptive Learning Rate Scheduling**:* To enhance convergence and mitigate overfitting, the ReduceLROnPlateau scheduler has been used. The learning rate is halved if the validation accuracy plateaus for three consecutive epochs. This scheduling strategy allows the model to fine-tune more effectively by escaping local minima and dynamically adjusting the learning pace as training stabilizes.

This separation of learning rates preserves pretrained features while encouraging fast learning in new classifier layers.

*4) **Robust Batching and Data Integrity Handling**:* A custom collate_fn function is incorporated into the data loading pipeline. This mechanism discards corrupted, unreadable, or misformatted images before they enter the training loop, ensuring training stability and preventing runtime errors. This step is essential for maintaining batch consistency and avoiding interruption in large-scale training environments with potentially noisy datasets.

B. Training Procedure

The model was trained using a ResNet-50 backbone pretrained on ImageNet. The training process was conducted in two phases, each utilizing different datasets and configurations. In Phase 1, training was performed on the 'train_raw' dataset (detailed in TABLE II), while in Phase 2, the 'train' dataset (detailed in TABLE II II) was used. An 80/20 stratified split was applied to divide the data into training and validation sets.

TABLE I: Phase 1 Dataset Overview

Main Class	**Subclass**	**Images Used**
Clothing	Dresses	2000
Clothing	Outerwear	2000
Clothing	Skirts	1921
Bags	Shoulder Bags	16
Bags	Tote Bags	16
Bags	Clutches	16
Shoes	Flats	16
Shoes	Boots	16
Shoes	High Heels	16

TABLE II: Phase 2 Dataset Overview

Main Class	**Subclass**	**Images Used**
Clothing	Dresses	2000
Clothing	Outerwear	2000
Clothing	Skirts	1921
Bags	Shoulder Bags	306
Bags	Tote Bags	430
Bags	Clutches	439
Shoes	Flats	484
Shoes	Boots	308
Shoes	High Heels	362

For batch sizes, Phase 1 used a size of 4, whereas Phase 2 experimented with larger batch sizes of 16 and 32. The training epochs varied across phases, with Phase 1 training for 5, 10, and 15 epochs, and Phase 2 extending up to 20, 35, 48, and 60 epochs. To enhance model robustness, data augmentation techniques such as random cropping, flipping (horizontal and vertical), colour jittering, and rotation were employed during training.

The optimization was carried out with the Adam optimizer, with different learning rates assigned to distinct components: a learning rate of 1e-4 was used for the ResNet backbone to preserve pretrained weights, while a higher learning rate of 1e-3 was applied to the classifier heads to facilitate rapid learning. An adaptive learning rate schedule, ReduceLROnPlateau, monitored validation performance, halving the learning rate after three consecutive epochs without improvement, with an initial learning rate of 0.001. This approach helped fine-tune the training process and prevent overfitting, ensuring effective convergence.

C. Evaluation Strategy

To comprehensively evaluate model performance across both classification levels, a multi-metric evaluation strategy was adopted, and applied to both training and validation phases.

The metrics used in this experiment are as follows:

- **Accuracy**: Measures the overall correctness of the model by computing the proportion of correctly predicted labels over all predictions.
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
where TP = true positives, TN = true negatives, FP = false positives, and FN = false negatives.
- **Precision**: Indicates the proportion of true positive predictions among all instances classified as positive. It is particularly useful when the cost of false positives is high.
$$\text{Precision} = \frac{TP}{TP + FP}$$
- **Recall**: Measures the proportion of actual positive cases that are correctly identified by the model. High recall is critical when missing positive instances is costly.
$$\text{Recall} = \frac{TP}{TP + FN}$$
- **F1-Score**: The harmonic mean of precision and recall, providing a balanced metric that is useful when the data is imbalanced.
$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$
- **Macro Average**: Averages the metric (e.g., precision, recall, F1-score) equally across all classes, regardless of class frequency.
$$\text{Macro-Avg} = \frac{1}{C} \sum_{i=1}^{C} M_i$$
where C is the number of classes and M_i is the metric score for class i.
- **Weighted Average**: Averages the metric across classes while taking into account the number of instances in each class.
$$\text{Weighted-Avg} = \sum_{i=1}^{C} \frac{n_i}{N} M_i$$

where n_i is the number of samples in class i, N is the total number of samples, and M_i is the metric for class i.

- **Confusion Matrix**: A matrix layout showing the distribution of predicted vs. true labels.

These metrics provide a comprehensive assessment of both hierarchical correctness and class-wise balance.

D. Metrics Implementation

*1) **Multi-Level Accuracy Metrics:*** The following metrics were computed for each epoch:

- **Main Class Accuracy**: Proportion of samples with correctly predicted main category (e.g., Clothing, Bags, Shoes).
- **Subclass Accuracy**: Proportion of correct subclass predictions, conditioned on the main class prediction being correct.
- **Overall Accuracy**: Proportion of samples for which both the main and subclass predictions are correct.
- **Total Loss**: The sum of weighted cross-entropy losses from both classification heads.

All metrics are logged and plotted over epochs to monitor learning dynamics and convergence behaviours. To ensure fair subclass evaluation, subclass predictions are masked when the corresponding main class prediction is incorrect. This avoids attributing subclass errors to the model when it has already failed at the higher level of classification.

*2) **Confusion Matrices and ROC Curves:*** To analyze class-wise behavior, the following are generated:

- Confusion matrices - for both main and subclass levels, visualizing misclassification patterns.
- ROC curves - for the main class head using a one-vs-rest strategy, evaluating discriminative capacity across categories.

These visual diagnostics provide valuable insights, especially for identifying residual bias toward majority classes. They also allow us to verify the effectiveness of imbalance mitigation strategies such as class weighting and data resampling.

All metrics and visualizations are computed independently for both classification tasks, ensuring that each level of the hierarchy is evaluated rigorously and consistently.

*3) **Precision, Recall, and F1-score Reports:*** Beyond top-level accuracy, we also compute **per-class precision, recall, and F1-scores** for both main and subclass predictions. These metrics are calculated using scikit-learn's classification_report() function and provide a detailed view of the model's behavior across classes.

In the subclass evaluation, scores are computed **only for valid predictions**, i.e., where the main class was correctly predicted. We report:

- Macro average: treats all classes equally, highlighting performance on minority classes
- Weighted average: accounts for class frequency, reflecting real-world performance balance

This level of analysis is crucial in imbalanced settings, where accuracy alone may obscure underperformance on rare classes. For example, while Dresses and Skirts show high recall, minority subclasses such as Clutches and Shoulder Bags also achieved F1-scores above 0.88, indicating successful generalization despite fewer samples.

*E. **Fallback Strategies for Prediction Confidence***

To improve the reliability and robustness of the classification system, specific fallback rules are implemented to handle uncertain or ambiguous predictions, ensuring consistent and trustworthy results in real-world scenarios.

- If the confidence for the main class prediction drops below 0.7, the system defaults to returning ”Unknown/Others.”
- If subclass confidence is high but the subclass does not belong to the predicted main class, it falls back to main class.
- These fallback strategies enhance the system's robustness and reliability in handling uncertain or ambiguous cases.

The fallback mechanism not only enhances interpretability but also reduces misclassification, especially in ambiguous cases. It encourages “abstention” during high uncertainty, providing the option for user-in-the-loop corrections in real deployments. This approach highlights the practical applicability of the hierarchical model and opens pathways for further improvements, such as out-of-distribution detection.

*F. **Inference Evaluation***

Since the slow fashion images did not meet the quality standards, the team decided against using them for transfer learning. Instead, the trained model was saved and later tested in a Jupyter notebook environment with uploaded SLOWFASHION images to evaluate its real-world performance.

A lightweight, browser-based inference prototype was developed using ipywidgets within a Jupyter Notebook environment. This interface allows users to upload images and receive real-time classification results, including confidence scores for both main and subcategory predictions. The system employs the defined fallback strategies. This ensures clarity and interpretability, even on incomplete or ambiguous inputs.

The predictions generated by the model were manually reviewed to assess their relevance and accuracy, with special attention given to ambiguous or challenging categories. This process provided insights into the model's practical applicability and highlighted potential areas for further improvement.

VII. Result Analysis

The result analysis compares results from different training phases, highlighting improvements achieved through dataset balancing, augmentation, and model fine-tuning. Additionally, it discusses the model's robustness in real-world scenarios and its ability to accurately classify both common and rare categories, providing a thorough understanding of its strengths and limitations.

A. Overall Performance

The hierarchical classification model demonstrated strong performance across both phases of training.

TABLE III: Phase 1 - Main Classes

Batch Size	Epochs	LR	Acc	Macro avg	Weighted avg	Support
4	5	1e-3	0.98	0.46	0.97	1209
4	10	5e-4	0.98	0.33	0.97	1209
4	15	1e-4	0.97	0.55	0.97	1209
4	30	2e-5	0.98	0.43	0.97	1209

TABLE IV: Phase 1 - Subclasses (Valid Predictions)

Batch Size	Epochs	LR	Acc	Macro avg	Weighted avg	Support
4	5	1e-3	0.72	0.47	0.66	340
4	10	5e-4	0.71	0.32	0.68	314
4	15	1e-4	0.85	0.44	0.85	805
4	30	2e-5	0.69	0.66	0.68	194

In Phase 1, with limited subclass data, the main class accuracy reached approximately 98%, while subclass accuracy was around 85% trained over 15 epochs and also support more samples than other epochs, which are seen from TABLE III and TABLE IV.

TABLE V: Phase 2 - Main Classes

Batch Size	Epochs	LR	Acc	Macro avg	Weighted avg	Support	Best Val Acc
16	20	2.5e-4	0.97	0.73	0.98	1658	-
16	35	6e-5	0.98	0.73	0.98	1658	-
16	48	1e-5	0.98	0.73	0.99	1658	0.9825
16	60	0	0.98	0.73	0.98	1658	-
32	20	2.5e-4	0.98	0.73	0.99	1658	-
32	35	6e-5	0.99	0.74	0.99	1658	0.9897
32	48	1e-5	0.99	0.74	0.99	1658	-
32	60	0	0.98	0.74	0.99	1658	-

After dataset expansion in Phase 2, achieved through balancing and additional data collection, the main class accuracy remained near 99%, with subclass accuracy significantly improving to over 93%. The best results were observed with a batch size of 32, trained over 35 epochs, achieving a main class accuracy of 99% and subclass accuracy of 93%, corroborated by high macro and weighted averages indicating reliable overall performance, shown on TABLE V and TABLE VI.

TABLE VI: Phase 2 - Subclasses (Valid Predictions)

Batch Size	Epochs	LR	Acc	Macro avg	Weighted avg	Support
16	20	2.5e-4	0.92	0.93	0.92	1368
16	35	6e-5	0.92	0.92	0.92	1449
16	48	1e-5	0.93	0.94	0.93	1448
16	60	0	0.94	0.94	0.94	1440
32	20	2.5e-4	0.93	0.94	0.93	1421
32	35	6e-5	0.92	0.93	0.93	1479
32	48	1e-5	0.92	0.93	0.92	1492
32	60	0	0.93	0.93	0.93	1492

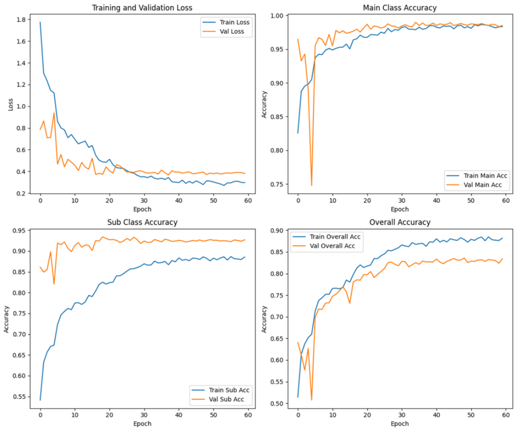


Fig. 1: phase2 training curves

This graph Fig 1 illustrates that the loss decreased with increasing epochs and the use of ReducePlateauonLR visibly helped in stabilizing the initial spikes as seen in the graphs.

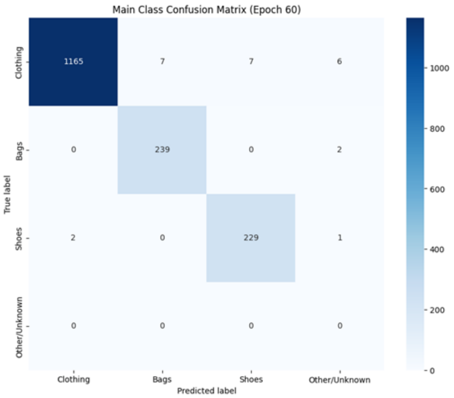


Fig. 2: main class confusion matrix

From the confusion matrix Fig 2 and Fig 3, it is easily understandable that the mainclass classification happens with a greater accuracy. There are some discrepancies when it comes to subclass classification. However, the misclassification is happening within the main class category; for instance, heels being classified as flats, or totes being classified as handbags.

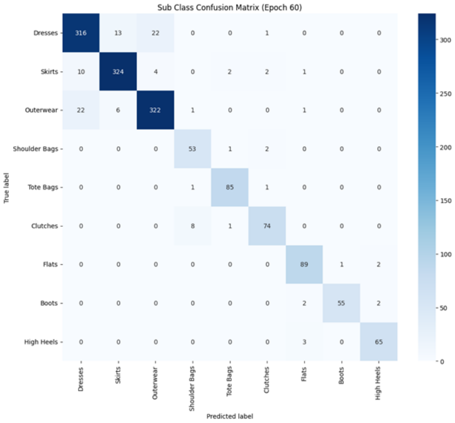


Fig. 3: sub class confusion matrix

B. Impact of Dataset Balancing

TABLE VII: Rare Subclass Recall:Phase 1 vs Phase 2

Subclass	Supp. (P1)	Rec. (P1)	Supp. (P2)	Rec. (P2)
Clutches	1	0.00	84	0.89
Tote Bags	0	0.00	85	0.98
Shoulder Bags	2	0.33	57	0.91
High Heels	1	0.00	68	0.99
Boots	2	0.00	59	0.93
Flats	4	0.31	93	0.95

The extension of the dataset in Phase 2 had a notable impact, especially on rare subclasses like Clutches, Boots, and High Heels, shown from table VII and Fig 4. Previously underrepresented, these categories saw a dramatic increase in recall, from zero in early phases to near-perfect scores, highlighting the importance of sufficient training data for the model's hierarchical classification capabilities.

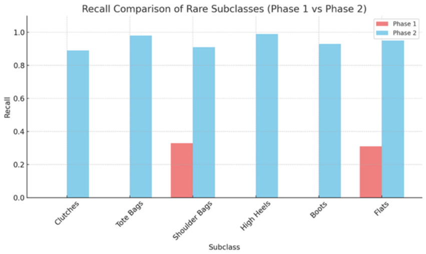


Fig. 4: recall comparison of rare subclass

C. Comparison Between Phases

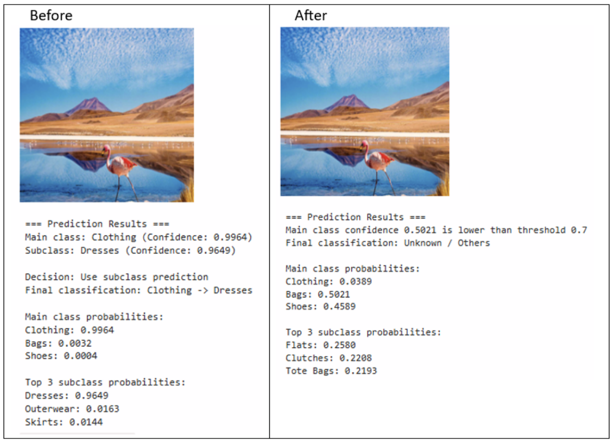
The comparison underscores that balancing strategies and dataset expansion effectively improved subclass recall and overall robustness. While initial training suffered from data imbalance, the more balanced dataset in Phase 2 resulted in improved precision and recall, especially for difficult and rare categories.

TABLE VIII: Comparison of two phase Results

Phase	Main Acc	Sub Acc	Comments
1	∼0.98	∼0.68	Limited subclass data
2	∼0.99	∼0.93	Balanced data, better performance

D. Handling Unknown and Out-of-Distribution Inputs

To prevent overconfident misclassifications of out-of-scope images, a confidence threshold of 0.7 for the main class was implemented. Images with lower confidence are labelled as "Unknown/Others," effectively rejecting uncertain predictions, one example shows the performance by Fig 5 before and after handling unknown strategee. Testing showed that this rejection mechanism significantly reduces misclassification errors, achieving up to 80% correct rejection of irrelevant images such as landscapes.



```
=== Prediction Results ===
Main class: Clothing (Confidence: 0.9964)
Subclass: Dresses (Confidence: 0.9649)

Decision: Use subclass prediction
Final classification: Clothing -> Dresses

Main class probabilities:
Clothing: 0.9964
Bags: 0.0032
Shoes: 0.0004

Top 3 subclass probabilities:
Dresses: 0.9649
Outerwear: 0.0163
Skirts: 0.0144
```

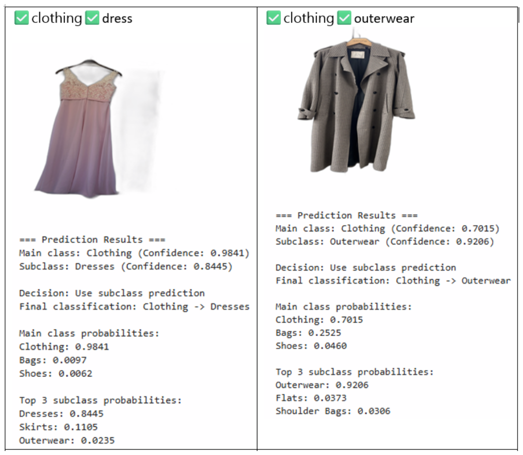
```
=== Prediction Results ===
Main class confidence 0.5021 is lower than threshold 0.7
Final classification: Unknown / Others

Main class probabilities:
Clothing: 0.0389
Bags: 0.5021
Shoes: 0.4589

Top 3 subclass probabilities:
Flats: 0.2580
Clutches: 0.2208
Tote Bags: 0.2193
```

Fig. 5: before and after handling unknown

E. Robustness to Occlusion and Partial Inputs

The model performance was evaluated on all 9 sub classes, and strong results were observed consistently in each case(see Fig 678910 for detailed results).

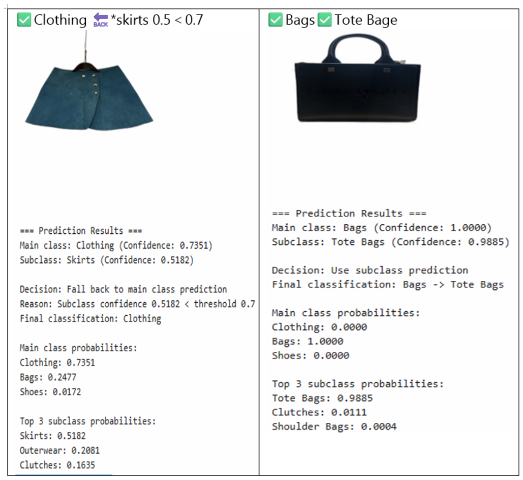
The model was also evaluated on occluded and partial images Fig11 common in real-world scenarios. Results indicated that it maintains high confidence in main class predictions despite low subclass confidence, demonstrating the effectiveness of the fallback logic in enhancing robustness and user trust.



```
=== Prediction Results ===
Main class: Clothing (Confidence: 0.9841)
Subclass: Dresses (Confidence: 0.8445)

Decision: Use subclass prediction
Final classification: Clothing -> Dresses

Main class probabilities:
Clothing: 0.9841
Bags: 0.0097
Shoes: 0.0062

Top 3 subclass probabilities:
Dresses: 0.8445
Skirts: 0.1105
Outerwear: 0.0235
```

```
=== Prediction Results ===
Main class: Clothing (Confidence: 0.7015)
Subclass: Outerwear (Confidence: 0.9206)

Decision: Use subclass prediction
Final classification: Clothing -> Outerwear

Main class probabilities:
Clothing: 0.7015
Bags: 0.2525
Shoes: 0.0460

Top 3 subclass probabilities:
Outerwear: 0.9206
Flats: 0.0373
Shoulder Bags: 0.0306
```

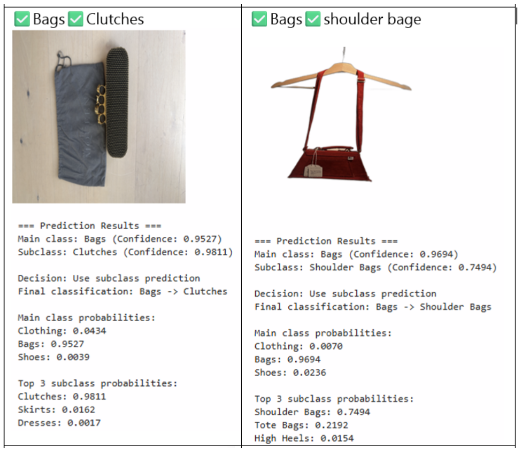
Fig. 6: Correctly classified: dress and outerwear



```
=== Prediction Results ===
Main class: Clothing (Confidence: 0.7351)
Subclass: Skirts (Confidence: 0.5182)

Decision: Fall back to main class prediction
Reason: Subclass confidence 0.5182 < threshold 0.7
Final classification: Clothing

Main class probabilities:
Clothing: 0.7351
Bags: 0.2477
Shoes: 0.0172

Top 3 subclass probabilities:
Skirts: 0.5182
Outerwear: 0.2081
Clutches: 0.1635
```

```
=== Prediction Results ===
Main class: Bags (Confidence: 1.0000)
Subclass: Tote Bags (Confidence: 0.9885)

Decision: Use subclass prediction
Final classification: Bags -> Tote Bags

Main class probabilities:
Clothing: 0.0000
Bags: 1.0000
Shoes: 0.0000

Top 3 subclass probabilities:
Tote Bags: 0.9885
Clutches: 0.0111
Shoulder Bags: 0.0004
```

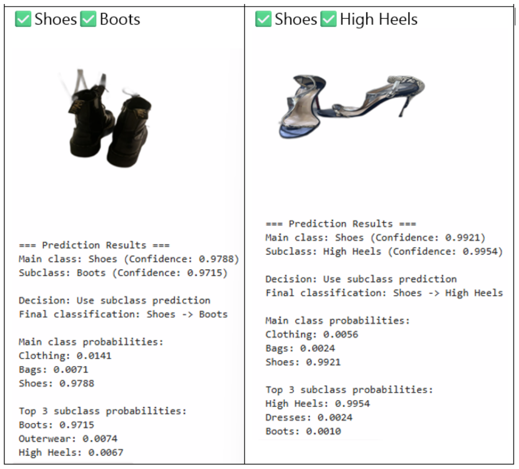
Fig. 7: Skirt fallback to clothing; tote bag correct



```
=== Prediction Results ===
Main class: Bags (Confidence: 0.9527)
Subclass: Clutches (Confidence: 0.9811)

Decision: Use subclass prediction
Final classification: Bags -> Clutches

Main class probabilities:
Clothing: 0.0434
Bags: 0.9527
Shoes: 0.0039

Top 3 subclass probabilities:
Clutches: 0.9811
Skirts: 0.0162
Dresses: 0.0017
```

```
=== Prediction Results ===
Main class: Bags (Confidence: 0.9694)
Subclass: Shoulder Bags (Confidence: 0.7494)

Decision: Use subclass prediction
Final classification: Bags -> Shoulder Bags

Main class probabilities:
Clothing: 0.0070
Bags: 0.9694
Shoes: 0.0236

Top 3 subclass probabilities:
Shoulder Bags: 0.7494
Tote Bags: 0.2192
High Heels: 0.0154
```

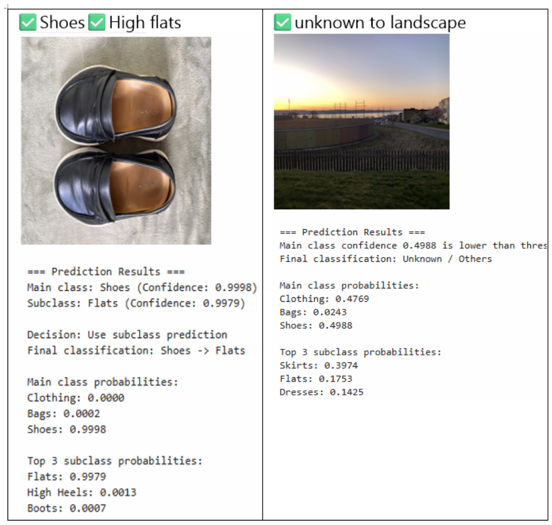
Fig. 8: Correctly classified: clutch and shoulder bag



```
=== Prediction Results ===
Main class: Shoes (Confidence: 0.9788)
Subclass: Boots (Confidence: 0.9715)

Decision: Use subclass prediction
Final classification: Shoes -> Boots

Main class probabilities:
Clothing: 0.0141
Bags: 0.0071
Shoes: 0.9788

Top 3 subclass probabilities:
Boots: 0.9715
Outerwear: 0.0074
High Heels: 0.0067
```

```
=== Prediction Results ===
Main class: Shoes (Confidence: 0.9921)
Subclass: High Heels (Confidence: 0.9954)

Decision: Use subclass prediction
Final classification: Shoes -> High Heels

Main class probabilities:
Clothing: 0.0056
Bags: 0.0024
Shoes: 0.9921

Top 3 subclass probabilities:
High Heels: 0.9954
Dresses: 0.0024
Boots: 0.0010
```

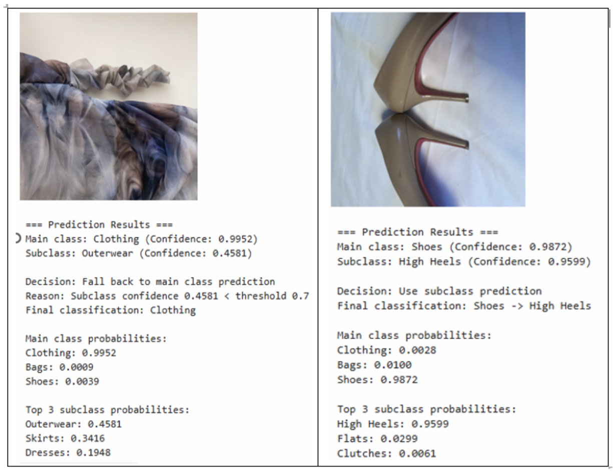
Fig. 9: Correctly classified: boots and high heels



```
=== Prediction Results ===
Main class: Shoes (Confidence: 0.9998)
Subclass: Flats (Confidence: 0.9979)

Decision: Use subclass prediction
Final classification: Shoes -> Flats

Main class probabilities:
Clothing: 0.0000
Bags: 0.0002
Shoes: 0.9998

Top 3 subclass probabilities:
Flats: 0.9979
High Heels: 0.0013
Boots: 0.0007
```

```
=== Prediction Results ===
Main class confidence 0.4988 is lower than thres
Final classification: Unknown / Others

Main class probabilities:
Clothing: 0.4769
Bags: 0.0243
Shoes: 0.4988

Top 3 subclass probabilities:
Skirts: 0.3974
Flats: 0.1753
Dresses: 0.1425
```

Fig. 10: Correctly classified: flats and others



```
=== Prediction Results ===
Main class: Clothing (Confidence: 0.9952)
Subclass: Outerwear (Confidence: 0.4581)

Decision: Fall back to main class prediction
Reason: Subclass confidence 0.4581 < threshold 0.7
Final classification: Clothing

Main class probabilities:
Clothing: 0.9952
Bags: 0.0009
Shoes: 0.0039

Top 3 subclass probabilities:
Outerwear: 0.4581
Skirts: 0.3416
Dresses: 0.1948
```

```
=== Prediction Results ===
Main class: Shoes (Confidence: 0.9872)
Subclass: High Heels (Confidence: 0.9599)

Decision: Use subclass prediction
Final classification: Shoes -> High Heels

Main class probabilities:
Clothing: 0.0028
Bags: 0.0100
Shoes: 0.9872

Top 3 subclass probabilities:
High Heels: 0.9599
Flats: 0.0299
Clutches: 0.0061
```

Fig. 11: Robustness to Occlusion and Partial Inputs

VIII. Conclusion

The hierarchical classification framework developed for the SlowFashion platform has demonstrated strong robustness and accuracy across multiple evaluation stages. Results from the two training phases highlight the importance of dataset balancing and augmentation techniques, which significantly improved the system's ability to accurately classify both broad categories and fine-grained subcategories. In particular, the model achieved over 99% accuracy in main class predictions and more than 93% in subclass predictions after dataset expansion, emphasizing the effectiveness of the balancing strategies in addressing data imbalance.

The introduction of a fallback mechanism further enhanced model reliability, especially in handling uncertain or ambiguous cases. This approach proved instrumental in maintaining consistent outputs, reducing misclassifications, and supporting practical deployment scenarios through confidence thresholds and out-of-scope rejection strategies. The model also proved resilient when tested on occluded or partial images, reinforcing its suitability for real-world applications.

Key insights from this study underscore that a hierarchical, balanced approach, combined with adaptive training techniques, enables robust and scalable fashion image classification. This approach aligns with SlowFashion's goal of providing trustworthy listings, supporting both user confidence and platform integrity. Future improvements could focus on further enhancing rare category detection and out-of-distribution recognition, fostering even greater practical applicability.

IX. Acknowledgment

We sincerely thank Dr. Florian Westphal for his continuous support and insightful guidance, which greatly contributed to the successful completion of our project. We also deeply appreciate Ludvig from SlowFashion for the opportunity to collaborate and for sharing valuable industry perspectives. Our gratitude also goes to Simon Sleman at Science Park for his assistance in making this collaboration possible.

References

[1] T.-R. Tzikas, A.-C. Kyprianidis, M. Kotouza, S.-F. Tsarouchis, A. Chrysopoulos, and P. Mitkas, "Towards Fashion Image Annotation: A Clothing Category Recognition Procedure," in *Proc. AI4FASHION 2020*, Athens, Greece, Sep. 2020, pp. 1–8. [Online]. Available: https://ceur-ws.org/Vol-2844/fashion2.pdf

[2] S. Saranya and S. Geetha, "Fashion Image Classification Using Deep Convolution Neural Network," in *Proc. Int. Conf. Comput. Commun. Signal Process. (ICCCSP)*, Chennai, India, Mar. 2022, pp. 105–112. [Online]. Available: https://doi.org/10.1007/978-3-031-11633-9_10

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint*, Dec. 2015. [Online]. Available: https://doi.org/10.48550/arXiv.1512.03385