Fashion Image Classification Using Convolutional Neural Networks

Shitai Zhao, Yuming Huang, Ece Yildiz, Raagini Tyagi

✦

Abstract—In this paper, we present a multi-faceted fashion image classification using Convolutional Neural Networks (CNNs). We implement and evaluate two popular CNN architectures—ResNet18 and EfficientNetB0—to classify fashion product images across multiple taxonomic levels: master category, subcategory, and season. Our experiments were conducted on both the original Fashion Product Images Dataset and a smaller version of this same dataset to evaluate model performance across different data scales. The models were trained using transfer learning with pre-trained weights on ImageNet. Results show that both models achieve high accuracy on all classification tasks, with EfficientNetB0 slightly outperforming ResNet18 on the larger dataset. Similar performance patterns were observed for master category and season classification. On the smaller dataset, both models performed similarly with F1 scores around 77% for subcategory classification. Our work demonstrates the effectiveness of CNN-based approaches for multi-level fashion image classification, which can significantly improve product categorization efficiency in e-commerce platforms and inventory management systems.

1 Introduction

Fashion image classification is a critical task in the e-commerce industry, where millions of products need to be accurately categorized to improve search functionality, recommendations, and inventory management. Manual classification of these products at different granularity levels (master category, subcategory, season, etc.) is time-consuming, inconsistent, and error-prone, making automated solutions increasingly necessary for large-scale operations.

Convolutional Neural Networks (CNNs) have demonstrated remarkable performance in image classification tasks in recent years. Their ability to automatically learn hierarchical features from images makes them particularly suitable for fashion product categorization, where visual attributes play a crucial role in determining product categories.

In this work, we focus on the task of classifying fashion product images across multiple classification dimensions:

1) **Master Category**: Broad product categories (e.g., Apparel, Accessories, Footwear)
2) **Subcategory**: More specific categories (e.g., Topwear, Bottomwear, Watches)
3) **Season**: The appropriate season for the product (e.g., Summer, Winter, Fall, Spring)

We evaluate two prominent CNN architectures—ResNet18 and EfficientNetB0—on both a large-scale dataset and a smaller version to compare performance and assess their suitability for deployment in real-world scenarios.

The contributions of this paper are:

1) A multi-level classification approach for fashion images covering master category, subcategory, and season
2) An evaluation of ResNet18 and EfficientNetB0 across these classification tasks
3) Performance analysis on different dataset sizes (original vs. small dataset)
4) A comparative study of model accuracy, F1 scores, and computational efficiency across different classification problems

Our findings provide valuable insights for practitioners looking to implement automated fashion image classification systems in commercial applications.

2 Related Work

Image classification has been a central problem in computer vision, with significant advances in recent years due to the development of deep learning techniques. In the fashion domain specifically, several approaches have been proposed to tackle various classification tasks.

2.1 CNN Architectures

Convolutional Neural Networks have become the standard approach for image classification tasks. LeNet, introduced by LeCun et al. [1], was among the first successful CNN architectures. Later, AlexNet [2] popularized CNNs by winning the ImageNet competition in 2012. Since then, several architectures have been proposed, including VGG [3], GoogLeNet [4], and ResNet [5].

ResNet, introduced by He et al. [5], addressed the vanishing gradient problem in deep networks through the use of residual connections. This innovation allowed for training much deeper networks, resulting in improved performance on various vision tasks. ResNet18, the variant used in our study, consists of 18 layers and provides a good balance between computational efficiency and accuracy.

EfficientNet, proposed by Tan and Le [6], introduced a compound scaling method that uniformly scales network width, depth, and resolution with a fixed set of scaling coefficients. This approach led to state-of-the-art performance on ImageNet while being more efficient than previous architectures. EfficientNetB0, the baseline model in the EfficientNet family, offers competitive performance with fewer parameters compared to models like ResNet.

2.2 Fashion Image Classification

Fashion image classification has gained significant attention due to its commercial applications. Liu et al. [7] introduced DeepFashion, a large-scale clothing dataset with annotations for various tasks, including category classification. Xiao et al. [8] proposed a multi-task approach that jointly performs classification and attribute prediction. Zou et al. [9] explored weak supervision techniques for fashion image classification to reduce the need for extensive labeled data.

Multi-level classification of fashion items has been explored by several researchers. Inoue et al. [10] proposed a multi-task learning approach to simultaneously predict category, season, and other attributes. Similarly, Hadi Kiapour et al. [11] developed systems for fine-grained categorization of fashion items across multiple taxonomic levels.

Transfer learning has been widely adopted in fashion image classification due to the limited availability of large-scale fashion datasets. In this approach, models pre-trained on large datasets like ImageNet are fine-tuned on fashion-specific datasets, leveraging the general features learned from diverse images [12].

3 Dataset

In our study, we used the Fashion Product Images Dataset [13], which contains approximately 44,000 fashion product images along with their corresponding metadata. The dataset provides rich information about each product, including gender, master category, subcategory, article type, color, season, and usage.

We focused on classifying images into three different levels:

1) **Master Category**: Including 7 broad categories such as Apparel, Accessories, Footwear, etc.
2) **Subcategory**: Including 45 more specific categories such as Topwear, Bottomwear, Watches, Shoes, Bags, etc.
3) **Season**: Including 4 categories - Summer, Winter, Fall, and Spring

The dataset is structured with images stored as individual JPEG files, and the categorical information is provided in a CSV file named ”styles.csv”.

We experimented with two versions of the dataset:

1) **Full-sized dataset**: Contains high-resolution images with the complete set of products.
2) **Small-sized dataset**: A compressed version with lower resolution images but maintaining the same number of products.

For both datasets, we performed a standard train-validation-test split with a ratio of 70%/15%/15% to ensure proper evaluation. We used a fixed random seed to ensure reproducibility and to maintain consistent splits across different experiments.

4 Methodology

4.1 Data Preprocessing

We applied several preprocessing steps to prepare the data for our models:

1) **Image Resizing**: All images were resized to 224×224 pixels to match the input requirements of both ResNet18 and EfficientNetB0.
2) **Normalization**: Images were normalized using the mean and standard deviation values from ImageNet (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]) to leverage the pre-trained weights effectively.
3) **Data Organization**: For each classification task (master category, subcategory, season), we organized images into folders according to their respective labels to facilitate the use of PyTorch’s ImageFolder dataset class.

4.2 Model Architecture

We implemented and evaluated two CNN architectures for each classification task:

4.2.1 ResNet18

ResNet18 is an 18-layer deep residual network that uses skip connections to address the vanishing gradient problem in deep neural networks. The skip connections allow gradients to flow through the network more easily during backpropagation, enabling the training of deeper networks.

We used the pre-trained ResNet18 model from PyTorch’s model zoo and modified the final fully connected layer to output the appropriate number of classes for each task:

- 7 classes for master category
- 45 classes for subcategory
- 4 classes for season

4.2.2 EfficientNetB0

EfficientNetB0 is a CNN architecture designed for improved efficiency through a compound scaling method that balances network depth, width, and resolution. It achieves competitive performance with significantly fewer parameters compared to other models.

We used the pre-trained EfficientNetB0 from the timm library and adapted it for our classification tasks by modifying the final classification layer accordingly for each task.

4.3 Training Procedure

Both models were trained using the following configuration for each classification task:

1) **Loss Function**: Cross-Entropy Loss
2) **Optimizer**: Adam with a learning rate of 1e-4
3) **Batch Size**: 16
4) **Epochs**: 10

We implemented a custom training function that includes validation steps after each epoch and saves the best model based on the F1 score on the validation set.

4.4 Evaluation Metrics

We evaluated the models using the following metrics:

1) **Accuracy**: The proportion of correctly classified instances.
2) **F1 Score (Macro)**: The harmonic mean of precision and recall, calculated across all classes with equal weight. This metric is particularly important for imbalanced datasets, as it ensures that performance on minority classes contributes equally to the overall score.

These metrics were calculated on the test set after training was completed for each classification task.

5 Experiment

5.1 Experimental Design

We designed our experiments to answer the following research questions:

1) How do ResNet18 and EfficientNetB0 compare in their ability to classify fashion images across different taxonomic levels (master category, subcategory, and season)?
2) What is the impact of dataset size and image resolution on model performance?
3) Which model architecture offers the best balance between accuracy and computational efficiency?

To answer these questions, we conducted a series of experiments with a factorial design considering:

- Two model architectures (ResNet18 and EfficientNetB0)
- Two dataset versions (full-sized and small)
- Three classification tasks (master category, subcategory, and season)

For each combination of factors, we trained the models from scratch using the same hyperparameters to ensure a fair comparison.

5.2 Implementation Details

All experiments were conducted using PyTorch. The models were trained on NVIDIA GPUs to accelerate training.

To handle class imbalance, particularly in the subcategory classification task, we used the macro-averaged F1 score as our primary evaluation metric since it gives equal weight to all classes regardless of their frequency in the dataset.

5.3 Evaluation Protocol

We evaluated model performance using the following protocol:

1) **Dataset Splitting**: The dataset was split into training (70%), validation (15%), and test (15%) sets. The same splits were used across all experiments to ensure comparable results.
2) **Model Selection**: For each experiment, we saved the model checkpoint that achieved the highest macro F1 score on the validation set during training.
3) **Evaluation Metrics**: We evaluated the selected models on the test set using:
    - Accuracy: The proportion of correctly classified instances
    - Macro F1 Score: The harmonic mean of precision and recall, calculated for each class and then averaged
    - Weighted F1 Score: Similar to macro F1, but weighted by the class frequencies
4) **Confusion Matrix Analysis**: We generated confusion matrices to identify particular classes that were challenging for the models to distinguish.
5) **Statistical Significance**: We performed statistical tests (McNemar’s test) to determine if the performance differences between models were statistically significant.

5.4 Computational Resources

The training and evaluation were conducted on Google Colaboratory with the following specifications:

- For small dataset experiments: NVIDIA T4 GPU with 16GB memory
- For full-sized dataset experiments: NVIDIA A100 GPU with 40GB memory

Models for the subcategory, master category, and season category classifications can also be found on HuggingFace.

Subcategory: https://huggingface.co/spaces/KiritoYH/6140_Sub

Master category: https://huggingface.co/spaces/KiritoYH/6140_Mas

Season category: https://huggingface.co/spaces/KiritoYH/6140_Season

6 Results

We present the results of our experiments with both ResNet18 and EfficientNetB0 on the small and large datasets for each classification task.

6.1 Subcategory Classification

6.1.1 Small Dataset Results

The results on the small dataset for subcategory classification are summarized in Table 1:

TABLE 1
Performance metrics on the small dataset (Subcategory)

Model	Accuracy	F1 Score (Macro)
ResNet18	0.9649	0.7766
EfficientNetB0	0.9655	0.7763

Both models achieved similar performance on the small dataset, with ResNet18 and EfficientNetB0 having F1 scores of 77.66% and 77.63%, respectively.

6.1.2 Large Dataset Results

The results on the large dataset for subcategory classification are summarized in Table 2:

On the large dataset, EfficientNetB0 outperformed ResNet18 with an F1 score of 81.64% compared to 79.73%.

TABLE 2
Performance metrics on the large dataset (Subcategory)

Model	Accuracy	F1 Score (Macro)
ResNet18	0.9626	0.7973
EfficientNetB0	0.9682	0.8164

6.2 Master Category Classification

6.2.1 Small Dataset Results

The results on the small dataset for master category classification are summarized in Table 3:

TABLE 3
Performance metrics on the small dataset (Master Category)

Model	Accuracy	F1 Score (Macro)
ResNet18	0.9701	0.8245
EfficientNetB0	0.9712	0.8264

Both models performed well on the master category classification task, with EfficientNetB0 slightly outperforming ResNet18.

6.2.2 Large Dataset Results

The results on the large dataset for master category classification are summarized in Table 4:

TABLE 4
Performance metrics on the large dataset (Master Category)

Model	Accuracy	F1 Score (Macro)
ResNet18	0.9735	0.8517
EfficientNetB0	0.9768	0.8689

EfficientNetB0 showed better performance on the large dataset for master category classification, consistent with the pattern observed in subcategory classification.

6.3 Season Classification

6.3.1 Small Dataset Results

The results on the small dataset for season classification are summarized in Table 5:

TABLE 5
Performance metrics on the small dataset (Season)

Model	Accuracy	F1 Score (Macro)
ResNet18	0.8523	0.8415
EfficientNetB0	0.8547	0.8432

Both models achieved good performance on the season classification task, with similar F1 scores.

6.3.2 Large Dataset Results

The results on the large dataset for season classification are summarized in Table 6:

EfficientNetB0 slightly outperformed ResNet18 on the season classification task with the large dataset.

TABLE 6
Performance metrics on the large dataset (Season)

Model	Accuracy	F1 Score (Macro)
ResNet18	0.8732	0.8653
EfficientNetB0	0.8795	0.8738

7 Conclusion

Our experiments on both small and large versions of the Fashion Product Images Dataset showed that:

1) **Dataset Size Impact**: Both models showed improved performance on the larger dataset across all classification tasks. Higher resolution images provide more discriminative features for classification.
2) **Model Comparison**: While the models performed similarly on the small dataset, EfficientNetB0 consistently showed a more substantial improvement on the large dataset. This may be attributed to EfficientNetB0's compound scaling approach, which effectively balances network depth, width, and resolution.
3) **Task Complexity**: As expected, the models achieved higher performance on tasks with fewer classes (master category with 7 classes and season with 4 classes) compared to the more fine-grained subcategory classification task with 45 classes.
4) **Class Imbalance**: The difference between accuracy and F1 scores highlights the class imbalance in the dataset, particularly for the subcategory classification task. While the overall accuracy is high (¿96%), the macro F1 scores are lower, indicating that some classes have poorer performance than others.
5) **Computational Efficiency**: Although not explicitly measured in our experiments, it's worth noting that EfficientNetB0 typically requires fewer parameters and FLOPs compared to ResNet18, potentially offering better efficiency for deployment.

These results demonstrate that both ResNet18 and EfficientNetB0 are effective for multi-level fashion image classification, with EfficientNetB0 being slightly more effective, particularly on larger datasets.

These findings have practical implications for e-commerce platforms and inventory management systems that rely on accurate product categorization. Automated fashion image classification can significantly reduce the time and resources required for manual categorization while maintaining high accuracy across multiple classification dimensions.

Future work could explore more advanced architectures, such as incorporating the more performant EfficientNet into a multimodal model where we use descriptions of the item, reviews of the item, etc., to conduct product recommendations.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ”Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ”ImageNet classification with deep convolutional neural networks,” Advances in Neural Information Processing Systems, pp. 1097-1105, 2012.

[3] K. Simonyan and A. Zisserman, ”Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[4] C. Szegedy et al., ”Going deeper with convolutions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, ”Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[6] M. Tan and Q. V. Le, ”EfficientNet: Rethinking model scaling for convolutional neural networks,” in International Conference on Machine Learning, pp. 6105-6114, 2019.

[7] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, ”DeepFashion: Powering robust clothes recognition and retrieval with rich annotations,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1096-1104, 2016.

[8] H. Xiao, K. Rasul, and R. Vollgraf, ”Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms,” arXiv preprint arXiv:1708.07747, 2017.

[9] M. Zou, S. Y. Xia, and M. Campanella, ”Fashion classification with weakly annotated data,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1865-1872, 2019.

[10] N. Inoue, E. Simo-Serra, T. Yamasaki, and H. Ishikawa, ”Multi-label fashion image classification with minimal human supervision,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 2261-2267, 2017.

[11] M. Hadi Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, ”Where to buy it: Matching street clothing photos in online shops,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 3343-3351, 2015.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, ”How transferable are features in deep neural networks?,” Advances in Neural Information Processing Systems, pp. 3320-3328, 2014.

[13] P. Aggarwal, ”Fashion product images dataset,” Kaggle, 2019. [Online]. Available: https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-dataset