# StyleNet: Unraveling Designer Signatures in Fashion with Advanced CNN Architectures

Authors: Jinanshi Mehta & Kelly Huang

## Introduction

In the rapidly evolving world of fashion, the aesthetic and thematic elements of designer collections serve as a mirror to societal trends, technological advancements, and cultural shifts. What one chooses to wear reflects a lot about their identity. Therefore, different brands have different identities customers want to associate themselves with. For instance, Burberry conveys minimalism, class, and sophistication, whereas Versace reflects bold, extravagant, and glamorous.
Therefore, establishing uniqueness in their brand through their signature style becomes essential. With Chanel's iconic use of pearls, classic tweed suits, and quilted handbags, it can be immediately recognized. However, Stella McCartney, often celebrated for its sustainable initiatives, lacks a widely recognizable logo or signature style making it harder to identify. This subtlety underscores the need to identify the brand correctly to retain its customers' projected identities as well as preserve the integrity of the brand.

This project leverages the power of Convolutional Neural Networks (CNNs) to analyze the bodies of work of notable designers– Alexander McQueen, Burberry, Chanel, Giorgio Armani, Stella McCartney, Versace and Yohji Yamamoto– **majority of who do not have notable tokens that make them easily recognizable.**

## Dataset Description (refer to Appendix)

The customized dataset is curated from the Vogue Fashion website, encompassing fashion runway imagery of Ready-to-Wear collections from the 7 designers from 2021 to 2024, focusing on Spring and Fall Collections. This choice of a single source ensures consistency in image style, quality, and editorial perspective, thereby providing a homogeneous dataset ideal for systematic analysis. This temporal and seasonal range is strategically chosen to capture the dynamic evolution of fashion trends during a period marked by significant global changes in the fashion industry.

## Data Pre-processing

All images in our dataset were initially in WEBP format, necessitating a conversion to JPEG to ensure compatibility with our image processing tools. Using Python's PIL library, we scripted a function to automate this conversion. In the initial stages of our dataset preparation, the presence of complex backgrounds in the fashion imagery sourced from Vogue's runway shows was a significant challenge. These backgrounds posed a risk of confounding the neural network, which might learn to recognize brands based on background cues rather than the fashion items themselves. For instance, the Neural Network might pick up on the background to identify these outfits as Alexander McQueen’s instead of analyzing their outfits.


To address this issue, we initially employed an automated background removal tool using the Deeplabv3 model with a ResNet101 backbone, sourced from PyTorch's model hub. Although this method promised efficiency by leveraging a pre-trained state-of-the-art semantic segmentation model, it fell short in terms of accuracy and processing time, often failing to completely isolate the fashion elements from varied runway backgrounds in the training dataset.

Recognizing the critical need for high-quality, clean images to ensure the reliability of our neural network training, we opted to redo the background removal process manually. Using tools like Adobe Photoshop and Canva, we meticulously edited each image, ensuring that all backgrounds were uniformly removed. This manual intervention allowed us to maintain a consistent quality across our dataset, thereby eliminating any inadvertent bias or learning errors that might arise from inconsistent image backgrounds. The decision to switch to manual processing, while time-consuming, was pivotal in preserving the integrity and quality of our training dataset, ensuring that the subsequent machine learning processes could focus solely on the fashion content without distraction from extraneous background elements.

Furthermore, we implemented a validation step to ensure the integrity of the image files prior to training our neural network. Utilizing the Python Imaging Library (PIL), we created a function, validate_images, to check each image file in the dataset for corruption. This function iterates through all the images in a specified directory, attempting to open and verify each file as a legitimate image. If an image file fails to open or is detected as corrupted—indicated by catching exceptions such as IOError or SyntaxError—the file name is recorded and printed. This process not only helped us in identifying and excluding corrupted files that could potentially skew the training process but also maintained the overall quality of the dataset. Our dataset is then normalized, along with changes in rotation, width shift, height shift, shear range and zoom range to ensure consistency.

In the dataset preparation for our study, we established a systematic approach to data partitioning to ensure robust training, validation, and testing of our neural network model. We defined split ratios of 70% for training (1756 images), 15% for validation, and 15% for testing with a total of 2666 images, applied uniformly across all classes. This means, for example, because the Burberry class contained 310 images, 217 images (70%) were allocated to the training set. Similarly, since Giorgio Armani had 605 images, 423 (70%) images were included in the training dataset. This consistent distribution methodology across different brands ensures that each set—training, validation, and testing—reflects a proportionate representation of all classes, thereby facilitating a balanced approach to model training and evaluation. This balanced distribution is crucial for avoiding model bias towards more frequently represented classes and for ensuring that the model's performance accurately reflects its generalization capabilities across diverse fashion styles.

## Methodology

Convolutional Neural Networks are extremely suitable for image recognition and classification tasks due to their ability to automatically detect important features without any human supervision. "CNNs are able to automatically learn spatial hierarchies of features, starting with simple patterns such as edges and moving on to more complex patterns as the layers get deeper."[1] It reduces an image to a form that is easier to process, without losing important features that might be useful for prediction. This quality makes CNN the perfect choice for the dataset chosen where a traditional Neural Network might falter due to dimensionality or size of the data.

## CNN Architecture Decisions

The objective was to deploy a model that effectively classifies fashion images into predefined categories, based on their style, texture, and design patterns. To achieve this, we experimented with various architectures and configurations of CNNs to determine the optimal model for our specific dataset.

[1] https://datagen.tech/guides/image-classification/image-classification-using-cnn/#

The initial phase of our methodology involved developing a CNN from scratch. This initial model included three dense layers, designed to classify images into various fashion categories based on learned features such as style and texture. The purpose of starting with a basic CNN model was to establish a baseline performance, which would help in understanding the complexity of the task and the effectiveness of simpler neural network architectures in handling it.
As the complexity of fashion images led to challenges in feature extraction and classification accuracy with our basic model, we decided to explore more sophisticated architectures.

**Transition to Pre-trained Models**

1. **InceptionV3**: Known for its inception modules that perform convolutions at multiple scales simultaneously, InceptionV3 was considered to potentially improve our model's ability to handle the diverse scales and styles present in fashion imagery.
2. **ResNet-50**: With its deep residual networks that use skip connections to allow gradients to flow through very deep networks, ResNet-50 was tested to overcome the challenges of vanishing gradients, a common issue in deep neural networks.

The performance of each model configuration was evaluated based on accuracy, computational efficiency, and their propensity to overfit using a separate validation set. These experiments revealed that while both InceptionV3 and ResNet-50 offered significant improvements over our initial model, compared to VGG-16, their accuracies turned out to be lower.

Ultimately, after rigorous testing, we settled on the VGG-16 model. Despite trying various sophisticated models, VGG-16's straightforward, sequential architecture—when supplemented with our two dense layers—provided a balance between accuracy, training stability, and computational efficiency. This model not only improved upon the baseline set by our original CNN (approximately 50% training accuracy) but also demonstrated superior generalization capabilities in our validation assessments.

**Model Elements:**

1. **VGG-16 Pre-Trained Base:** The VGG16 network, known for its simplicity and effectiveness in image classification tasks is the base of the model. VGG16 includes multiple convolutional layers followed by max-pooling layers, repeated in blocks. By using include_top=False, the fully connected layers at the top of the network are excluded, which are originally designed for 1000 classes in ImageNet. The input shape for the images is set to 224x224x3, which is standard for VGG16, indicating the height, width, and depth (color channels) of the input images.

2. **Layer Freezing**: All the layers in the VGG16 base are frozen (layer.trainable = False), meaning their weights will not update during training. This is done to preserve the features that VGG16 has already learned from ImageNet, which are generally useful for many image recognition tasks.
3. **Flatten Layer**: After the convolutional base, there's a Flatten layer, which transforms the 2D feature maps into a 1D vector. This layer is necessary to transition from the convolutional layers (which handle spatial data) to dense layers (which handle 1D data).
4. **Dropout Layer**: A Dropout layer follows, with a dropout rate of 0.5. This means randomly 50% of the input units to the next layer are set to zero at each step during training, which helps in reducing overfitting by preventing complex co-adaptations on training data.
5. **Dense Layers:** The first Dense layer has 256 neurons and uses ReLU (rectified linear unit) activation. The second Dense layer follows with 128 neurons, also with ReLU activation. This continuation of dense layers with a decreasing number of neurons helps in refining the model's decision-making ability, narrowing down from broader features to more specific ones as it prepares for the final classification.
6. **Output Layer:** The final layer is a Dense layer with 7 neurons, corresponding to the 7 classes of the classification task. It uses the softmax activation function, which is standard for multi-class classification tasks because it outputs the probabilities of each class, with all probabilities summing up to 1.

The model is compiled with the Adam optimizer and categorical cross entropy as the loss function. Adam is chosen for its effectiveness in handling sparse gradients and non-stationary objectives and Categorical cross entropy is chosen because of its suitability for multi-class classification problems where each target is a one-hot encoded vector. The model is then trained with 50 epochs for best accuracy after experimenting with different numbers of epochs.

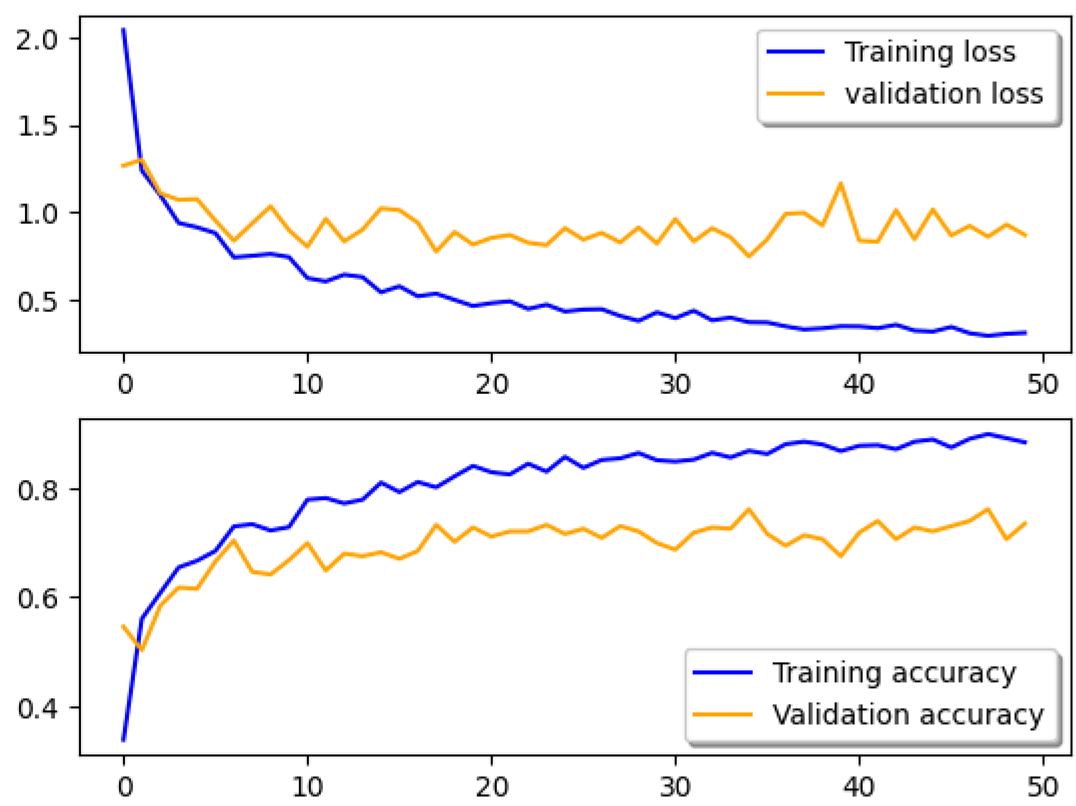
In the development phase of our convolutional neural network using the VGG-16 architecture, we opted not to utilize optimization techniques such as early stopping, model checkpointing, or dynamic learning rate adjustments. This decision was driven by the intent to fully assess the model's behavior over a complete training cycle, establishing a comprehensive baseline performance without interruptions. This approach allowed us to gain clear insights into the model’s capacity and generalization ability across all epochs, simplifying the training process and avoiding the complexity of tuning callback parameters.

**Results:**

Over the course of the training period spanning 50 epochs, the convolutional neural network (CNN) demonstrated a consistent improvement in its ability to classify fashion images from seven distinct designers. The final epochs of the training phase highlighted significant developments and challenges:

- Epoch 48: Achieved a training accuracy of 90.02% and a loss of 0.2908, indicating a strong ability of the model to generalize from the training data. However, the validation accuracy was observed at 76.20% with a validation loss of 0.8581, suggesting some overfitting to the training data as the model did not perform equally well on unseen data.
- Epoch 49: Recorded a slight decrease in training accuracy to 89.27% with a corresponding increase in loss to 0.3020. The validation accuracy further dropped to 70.67% with an increase in validation loss to 0.9277, reinforcing concerns regarding the model's ability to generalize.
- Epoch 50: The final epoch showed a training accuracy of 88.52% and a loss of 0.3078. The validation accuracy was 73.56% with a loss of 0.8685, indicating a stabilization in performance metrics but still reflecting a gap between training and validation outcomes.

Despite these challenges, the model maintained a relatively high training accuracy throughout the experiment, consistently staying above 88%. This level of performance on the training set demonstrates the model's effective learning and adaptation to the features of the dataset within the constraints of its training environment.




The provided training and validation curves over 50 epochs suggest that while the model is learning effectively, as indicated by the steadily improving training accuracy and decreasing training loss, it is not generalizing as well to new data, as evidenced by the higher validation loss and lower validation accuracy. The persistent gap between training and validation performance is indicative of overfitting, where the model is performing well on training data but less so on unseen data. The flattening of the validation accuracy curve points to a plateau in learning, suggesting that without properly addressing overfitting, further training is unlikely to result in meaningful improvements in model generalization.

The performance of the CNN model on the validation set highlighted several challenges in accurately classifying fashion images from seven distinct designers. Key performance metrics indicated significant room for improvement: precision was notably low at 0.1265, suggesting a high number of false positives where the model often mislabeled the images. Similarly, recall was marginally better at 0.1293, indicating only a small proportion of actual positives were correctly identified, with an F1 score of 0.1255 further confirming difficulties in achieving high accuracy in prediction and retrieval of relevant instances. The metrics collectively underscore the need for refining the model’s architecture, training process, or perhaps incorporating more nuanced features to improve its discriminative power.

**Conclusion**

The model's current iteration presents several challenges that impede its generalization capabilities, most notably reflected in the low validation and testing accuracies (with background: 45%). It indicates a performance discrepancy that underscores the model's difficulty in adapting to new, unseen data, which is further exacerbated when fashion images are introduced with varied backgrounds. This suggests that the model's feature extraction is significantly influenced by background noise, leading to reduced accuracy in more realistic or complex scenarios. Additionally, the model may struggle with the high intra-class variation and low inter-class variation inherent in fashion design images, where different designers may have similar elements in their styles. These challenges highlight the necessity for further refinement in both data preprocessing and model complexity, such as improved background segmentation techniques, enhanced regularization, and possibly the integration of attention mechanisms to focus on relevant features and mitigate the influence of confounding factors.

## Appendix

Chanel:
https://www.vogue.com/fashion-shows/spring-2024-ready-to-wear/chanel (2024)
https://www.vogue.com/fashion-shows/fall-2024-ready-to-wear/chanel (2024)
https://www.vogue.com/fashion-shows/fall-2023-ready-to-wear/chanel (2023)
https://www.vogue.com/fashion-shows/spring-2023-ready-to-wear/chanel (2023)
https://www.vogue.com/fashion-shows/fall-2022-ready-to-wear/chanel (2022)
https://www.vogue.com/fashion-shows/spring-2022-ready-to-wear/chanel (2022)
https://www.vogue.com/fashion-shows/fall-2021-ready-to-wear/chanel (2021)
https://www.vogue.com/fashion-shows/spring-2021-ready-to-wear/chanel (2021)

Alexander McQueen:
https://www.vogue.com/fashion-shows/spring-2024-ready-to-wear/alexander-mcqueen (2024)
https://www.vogue.com/fashion-shows/fall-2024-ready-to-wear/alexander-mcqueen (2024)
https://www.vogue.com/fashion-shows/spring-2023-ready-to-wear/alexander-mcqueen (2023)
https://www.vogue.com/fashion-shows/fall-2023-ready-to-wear/alexander-mcqueen (2023)
https://www.vogue.com/fashion-shows/spring-2022-ready-to-wear/alexander-mcqueen (2022)
https://www.vogue.com/fashion-shows/fall-2022-ready-to-wear/alexander-mcqueen (2022)
https://www.vogue.com/fashion-shows/spring-2021-ready-to-wear/alexander-mcqueen (2021)
https://www.vogue.com/fashion-shows/fall-2021-ready-to-wear/alexander-mcqueen (2021)

Giorgio Armani:
https://www.vogue.com/fashion-shows/spring-2024-ready-to-wear/giorgio-armani (2024)
https://www.vogue.com/fashion-shows/fall-2024-ready-to-wear/giorgio-armani (2024)
https://www.vogue.com/fashion-shows/spring-2023-ready-to-wear/giorgio-armani (2023)
https://www.vogue.com/fashion-shows/fall-2023-ready-to-wear/giorgio-armani (2023)
https://www.vogue.com/fashion-shows/spring-2022-ready-to-wear/giorgio-armani (2022)
https://www.vogue.com/fashion-shows/fall-2022-ready-to-wear/giorgio-armani (2022)
https://www.vogue.com/fashion-shows/spring-2021-ready-to-wear/giorgio-armani (2021)
https://www.vogue.com/fashion-shows/fall-2021-ready-to-wear/giorgio-armani (2021)

Versace:
https://www.vogue.com/fashion-shows/spring-2024-ready-to-wear/versace (2024)
https://www.vogue.com/fashion-shows/fall-2024-ready-to-wear/versace (2024)
https://www.vogue.com/fashion-shows/spring-2023-ready-to-wear/versace (2023)
https://www.vogue.com/fashion-shows/fall-2023-ready-to-wear/versace (2023)
https://www.vogue.com/fashion-shows/spring-2022-ready-to-wear/versace (2022)
https://www.vogue.com/fashion-shows/fall-2022-ready-to-wear/versace (2022)
https://www.vogue.com/fashion-shows/spring-2021-ready-to-wear/versace (2021)

https://www.vogue.com/fashion-shows/fall-2021-ready-to-wear/versace (2021)

Yohji Yamamoto:
https://www.vogue.com/fashion-shows/spring-2024-ready-to-wear/yohji-yamamoto (2024)
https://www.vogue.com/fashion-shows/fall-2024-ready-to-wear/yohji-yamamoto (2024)
https://www.vogue.com/fashion-shows/spring-2023-ready-to-wear/yohji-yamamoto (2023)
https://www.vogue.com/fashion-shows/fall-2023-ready-to-wear/yohji-yamamoto (2023)
https://www.vogue.com/fashion-shows/spring-2022-ready-to-wear/yohji-yamamoto (2022)
https://www.vogue.com/fashion-shows/fall-2022-ready-to-wear/yohji-yamamoto (2022)
https://www.vogue.com/fashion-shows/spring-2021-ready-to-wear/yohji-yamamoto (2021)
https://www.vogue.com/fashion-shows/fall-2021-ready-to-wear/yohji-yamamoto (2021)

Stella McCartney:
https://www.vogue.com/fashion-shows/spring-2024-ready-to-wear/stella-mccartney (2024)
https://www.vogue.com/fashion-shows/fall-2024-ready-to-wear/stella-mccartney (2024)
https://www.vogue.com/fashion-shows/spring-2023-ready-to-wear/stella-mccartney (2023)
https://www.vogue.com/fashion-shows/fall-2023-ready-to-wear/stella-mccartney (2023)
https://www.vogue.com/fashion-shows/spring-2022-ready-to-wear/stella-mccartney (2022)
https://www.vogue.com/fashion-shows/fall-2022-ready-to-wear/stella-mccartney (2022)
https://www.vogue.com/fashion-shows/spring-2021-ready-to-wear/stella-mccartney (2021)
https://www.vogue.com/fashion-shows/fall-2021-ready-to-wear/stella-mccartney (2021)

Burberry:
https://www.vogue.com/fashion-shows/spring-2024-ready-to-wear/burberry-prorsum (2024)
https://www.vogue.com/fashion-shows/fall-2024-ready-to-wear/burberry-prorsum (2024)
https://www.vogue.com/fashion-shows/spring-2023-ready-to-wear/burberry-prorsum (2023)
https://www.vogue.com/fashion-shows/fall-2023-ready-to-wear/burberry-prorsum (2023)
https://www.vogue.com/fashion-shows/spring-2022-ready-to-wear/burberry-prorsum (2022)
https://www.vogue.com/fashion-shows/fall-2022-ready-to-wear/burberry-prorsum (2022)
https://www.vogue.com/fashion-shows/spring-2021-ready-to-wear/burberry-prorsum (2021)
https://www.vogue.com/fashion-shows/fall-2021-ready-to-wear/burberry-prorsum (2021)