DATA612 - DEEP LEARNING

Summer 2024

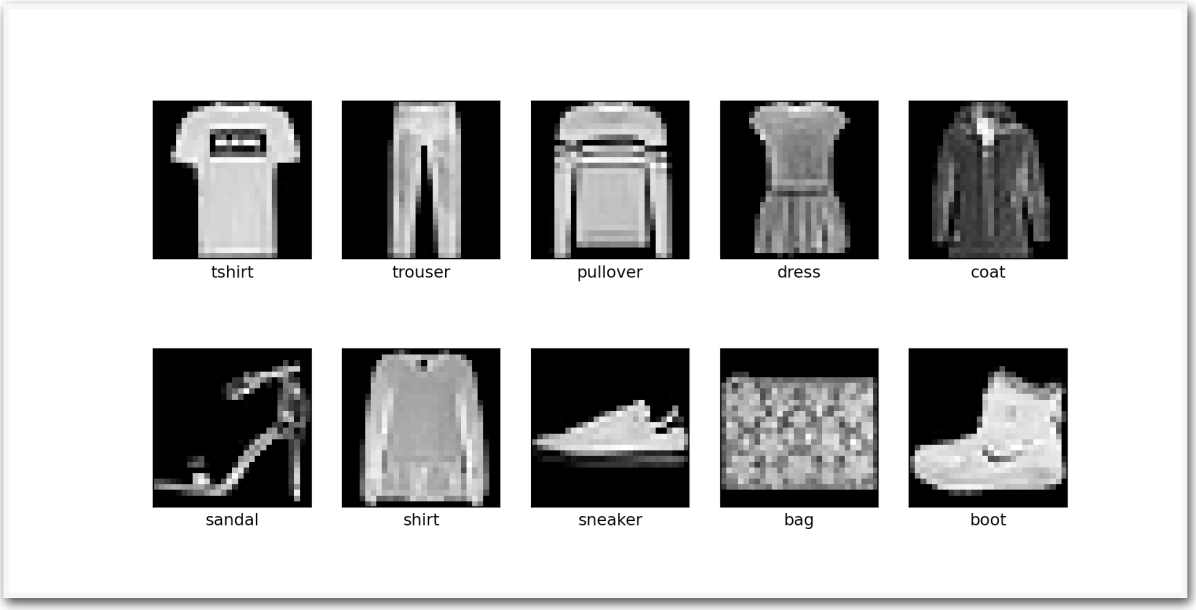
Name: Swathi Baskaran
UID: 120428326

Objective:

In this project, I will utilize the ResNet18 classic Convolutional Neural Network (CNN), pre-trained on ImageNet, as a feature extractor for the FashionMNIST dataset. The goal is to extract features from a specific layer of the ResNet model and visualize these features using t-SNE to analyze their distribution. I will quantify the intra-class and inter-class variances of the extracted features to understand their effectiveness in distinguishing between different classes. This project involves adapting the FashionMNIST images to three channels for CNN input, resizing images as needed, and reporting on the process, challenges, and resolutions encountered.

Dataset:

I chose FashionMNIST as my dataset. It consists of 70,000 grayscale images, each 28x28 pixels in size, divided into a training set of 60,000 images and a test set of 10,000 images. The dataset includes images from the following 10 classes, each corresponding to a different type of clothing or accessory: T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. Each category is represented by 7,000 images, ensuring a balanced distribution across the dataset. The images are labeled with integers ranging from 0 to 9, corresponding to the respective fashion categories.



Despite being a single-channel dataset, Fashion MNIST images can be easily adapted to three channels by repeating the single channel three times, making them compatible with pre-trained models like ResNet. This adaptability, combined with its balanced classes and standardized format, makes Fashion MNIST an excellent choice for evaluating machine learning models and their ability to generalize across different types of image data.

Why Fashion MNIST?

I was contemplating choosing between the Fashion MNIST and CIFAR-10 datasets. I chose the Fashion MNIST dataset over CIFAR-10 for this project primarily due to its simplicity in preprocessing and its suitability for focused feature extraction. Firstly, Fashion MNIST images are grayscale and uniformly 28x28 pixels, simplifying the preprocessing pipeline. Converting these images to three channels required by the ResNet model is straightforward, avoiding the complexity of handling and normalizing colored images in CIFAR-10. Grayscale images are computationally less intensive compared to the RGB images. Secondly, Fashion MNIST's domain-specific content, consisting of various clothing items, allows for a more targeted analysis of the CNN's performance in recognizing patterns related to fashion. This specificity facilitates a clearer understanding of intra-class and inter-class variances, making the extracted features more interpretable and relevant for applications in fashion-related tasks.

What is CNN model?

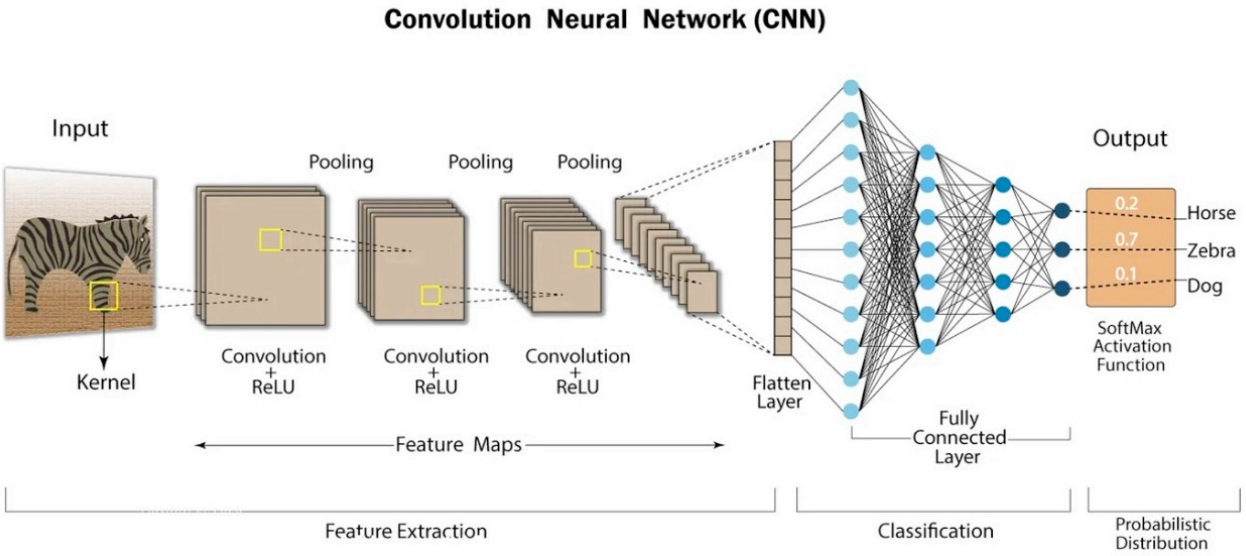
A convolutional neural network (CNN or ConvNet) is a network architecture for deep learningthat learns directly from data. CNNs are particularly useful for finding patterns in images to recognize objects, classes, and categories. They can also be quite effective for classifying audio, time-series, and signal data.

A CNN can have tens or hundreds of layers that each learn to detect different features of an image. Filters are applied to each training image at different resolutions, and the output of each convolved image is used as the input to the next layer. The filters can start as very simple features, such as brightness and edges, and increase in complexity to features that uniquely define the object.

A CNN is composed of an input layer, an output layer, and many hidden layers in between. These layers perform operations that alter the data with the intent of learning features specific to the data. Three of the most common layers are convolution, activation or ReLU, and pooling.

- **Convolution** puts the input images through a set of convolutional filters, each of which activates certain features from the images.
- **Rectified linear unit (ReLU)** allows for faster and more effective training by mapping negative values to zero and maintaining positive values. This is sometimes referred to as *activation*, because only the activated features are carried forward into the next layer.
- **Pooling** simplifies the output by performing nonlinear downsampling, reducing the number of parameters that the network needs to learn.

These operations are repeated over tens or hundreds of layers, with each layer learning to identify different features. Given below is a simple diagram of CNN model.



There are pre-trained models available for various architectures, including VGG, ResNet, Inception, and MobileNet, among others. Fashion MNIST images are grayscale, so we convert them to 3 channels and resize them to 224x224 pixels to match the input requirements of ResNet and VGG. I ran my code on both VGG and ResNet models. VGG took about 107 mins to run and ResNet took about 35mins to run. Considering the computational complexity and the availability of resources I decided to go with ResNet model. I chose ResNet on the basis of the below reasons:

VGG vs ResNet Models

ResNet18 models are relatively lightweight and provide a good balance of depth and performance. They are well-suited for the Fashion MNIST dataset and can produce high-quality feature representations.

VGG16: If you prefer a simpler architecture and have sufficient computational resources, VGG16 is also a viable option. It has been widely used for feature extraction and can perform well on the Fashion MNIST dataset.

Depth and Performance:

- ResNet models can be deeper than VGG models while maintaining good performance due to the residual connections. This allows for more complex feature representations.
- For a dataset like Fashion MNIST, which has relatively simple images compared to other datasets like ImageNet, ResNet18 can provide a good balance between complexity and computational efficiency.

Training Efficiency:

- ResNet models are generally easier to train due to the residual connections that help mitigate the vanishing gradient problem.
- This results in faster convergence and potentially better feature representations for downstream tasks.

Computational Efficiency:

- ResNet models can be more computationally efficient and have fewer parameters compared to VGG models with similar performance levels.
- This can be advantageous when working with limited computational resources or when processing a large number of images.

Feature Quality:

- The skip connections in ResNet help in learning better feature representations, which can be crucial for tasks like t-SNE visualization and variance analysis.

Which feature to extract?

Feature extraction is a critical process in computer vision, especially in CNNs. It involves identifying and isolating essential patterns and information from visual data, enabling the network to make sense of the input.

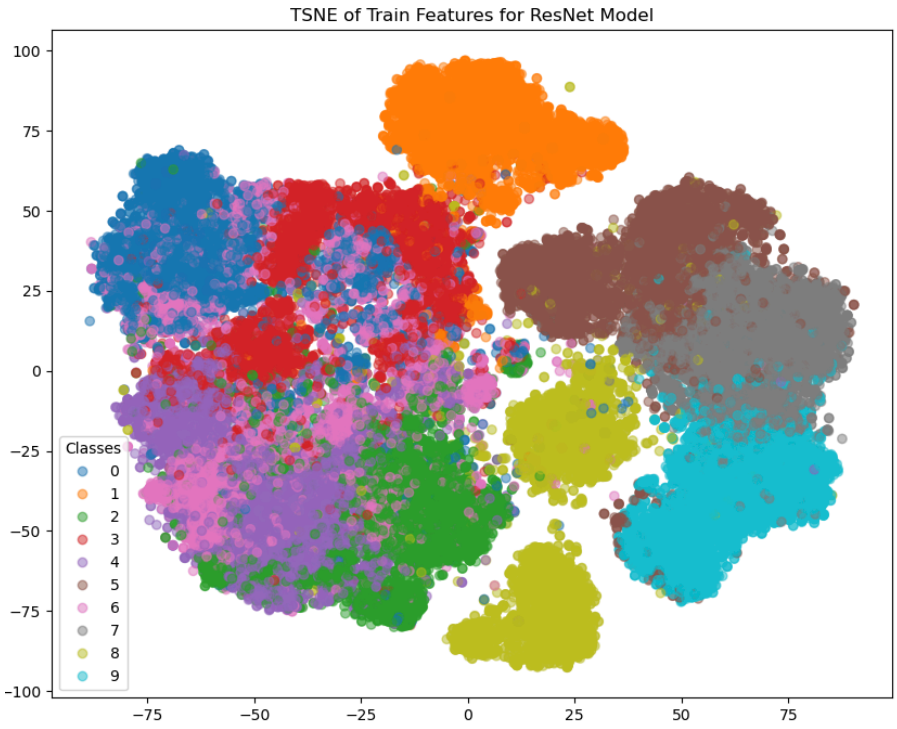
For the best visualization of features using t-SNE (t-Distributed Stochastic Neighbor Embedding), it's generally beneficial to extract features from a layer that captures high-level representations of the input images. These high-level features are more discriminative and can reveal interesting patterns when visualized.

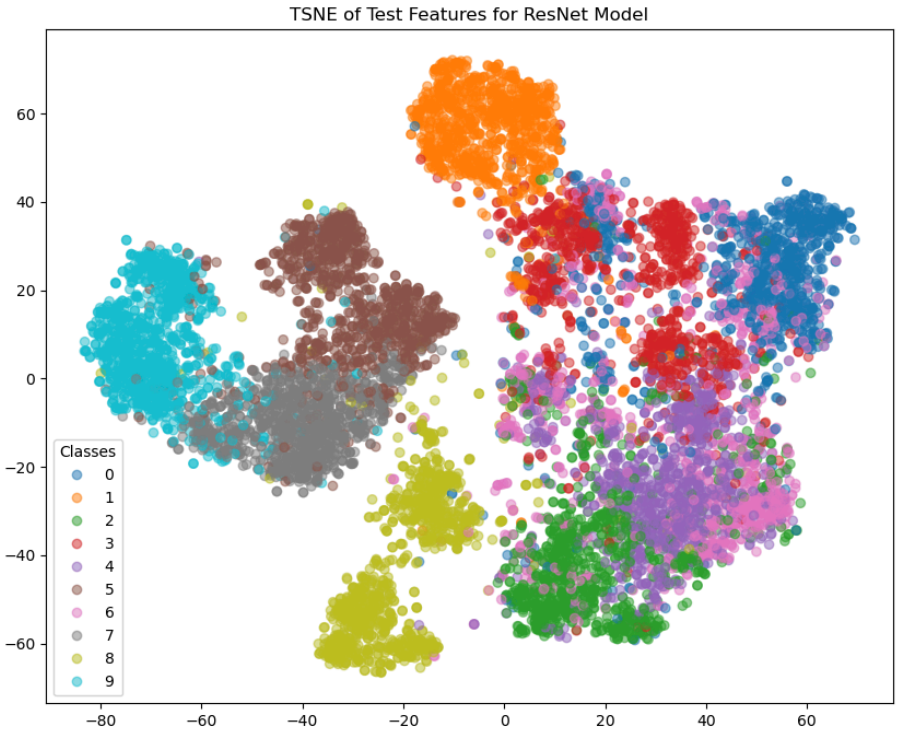
In the context of using a pre-trained ResNet18 model, the best layer for extracting features would typically be one of the last layers before the final classification layer. Specifically, the features from the layer right before the fully connected (FC) layer are usually the most suitable for t-SNE visualization. So, I chose the **last conv layer**, before the fully connected layer as my feature. This layer captures rich, high-level information about the images, making it ideal for visualization tasks.

TSNE Visualisation

The t-SNE visualization of the last convolutional layer of the ResNet model to the Fashion MNIST dataset shows how the model clusters the features of different classes. In the visualization of the training features, we observe that distinct classes form relatively compact and

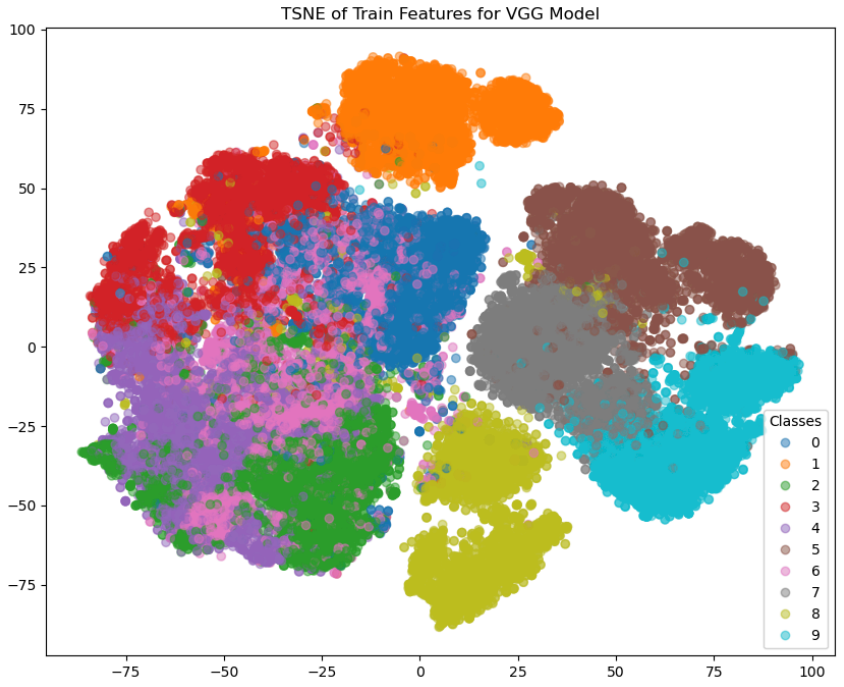
separate clusters, indicating that the CNN has effectively learned to differentiate between the various types of fashion items. Classes such as 0 (T-shirt/top), 2 (Pullover), and 6 (Shirt) show well-defined clusters, while some overlap is present among classes that share similar visual characteristics, such as classes 2 (Pullover) and 4 (Coat).

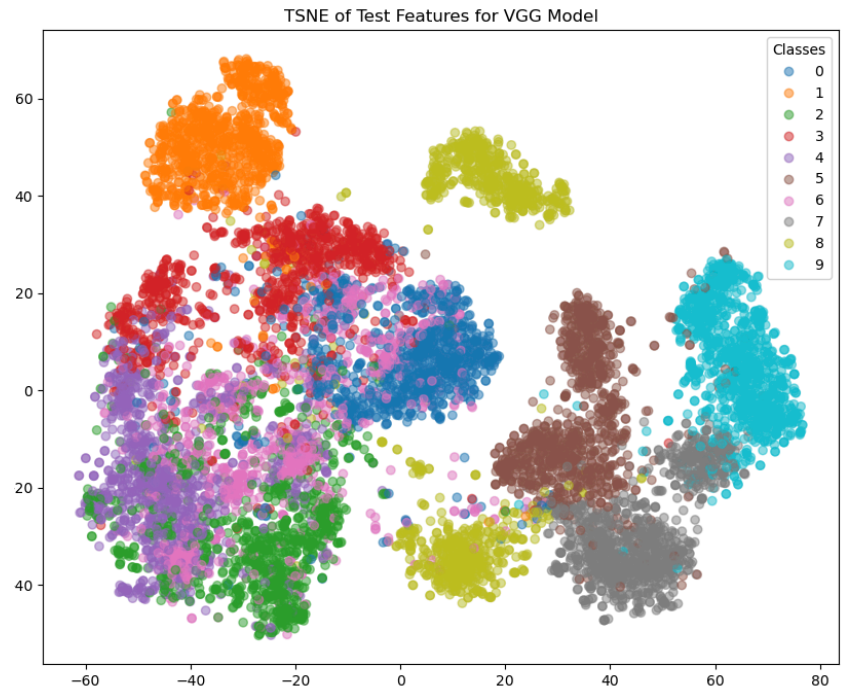




In the test features visualization, the clustering behavior is similar, though slightly less distinct, which is typical as the test data represents unseen data for the model. Here too, most classes form distinct groups, but there is noticeable overlap among certain classes, reflecting the inherent challenge of distinguishing between visually similar items. This overlap suggests that while the model generalizes well, there are still areas for improvement, particularly in enhancing the separability of similar classes. The compactness of some clusters, such as classes 5 (Sandal) and 7 (Sneaker), indicates strong model performance in recognizing these items.

Here are the plots from the VGG model, features from the last conv layer.





In the VGG model's t-SNE plots, the plots shows a more scattered and overlapping clustering of different classes. This suggests that the VGG model, while capable of extracting distinct features, might not separate the classes as clearly as desired. Some clusters, like those for Class 4 (Coat) and Class 1 (Trouser), are more distinct, but there is significant overlap among other classes.

In contrast, the ResNet model's t-SNE visualizations demonstrate better-separated clusters for both the training and test sets. Classes such as 0 (T-shirt/top) and 9 (Ankle boot) are clearly separated from others, reflecting the model's effectiveness in feature extraction.

Comparing the two models, the ResNet model shows superior performance in feature extraction and class separation. The t-SNE plots indicate that ResNet's features are more distinct, leading to better intra-class compactness and inter-class separation.

Intra-Class Variance

Intra-class variance measures the variability within each class. It quantifies how much the features of samples within the same class differ from each other. It is the average variance within each class. It is calculated by taking the variance of the features for each class separately and then averaging these variances across all classes. Given below is the intra class variance of the 10 classes from the dataset using ResNet model:

```
Intra-class Variances for Training Set:
Class 0: 453.983
Class 1: 214.494
Class 2: 388.850
Class 3: 424.731
Class 4: 401.174
Class 5: 287.407
Class 6: 754.865
Class 7: 166.507
Class 8: 575.319
Class 9: 204.719
```

```
Intra-class Variances for Test Set:
Class 0: 292.235
Class 1: 114.587
Class 2: 231.755
Class 3: 227.353
Class 4: 171.047
Class 5: 118.399
Class 6: 501.118
Class 7: 84.634
Class 8: 223.437
Class 9: 101.578
```

The intra-class variance results provide insight into how well the features are clustered within each class. For the training set, we observe that Class 6 (Shirt) has the highest intra-class variance at 754.865, indicating a broader spread of feature representations within this class. This could be due to the diverse styles and designs of shirts that make them harder to cluster tightly. Conversely, Class 7 (Sneaker) has the lowest intra-class variance at 166.507, suggesting that the features of sneakers are more consistently captured by the model, leading to tighter clusters. The test set shows a similar pattern with Class 6 having the highest variance at 501.118 and Class 7 the lowest at 84.634. The lower variances in the test set compared to the training set suggest that the model generalizes well to unseen data, but still struggles with the diversity within certain classes. Good standard to evaluate how strong the features are: intra is significantly smaller than inter.

Inter-Class Variance

Inter-class variance measures the variability between different classes. It quantifies how much the features of samples from different classes differ from each other. It is a bit more complex. It involves the variance of the means of the different classes. Essentially, it measures how much the mean feature vector of each class differs from the overall mean feature vector of the dataset.

```
Inter-class Variances for Training Set (pairwise):
Class pair (0, 1): 992.556
Class pair (0, 2): 1328.456
Class pair (0, 3): 530.535
Class pair (0, 4): 1010.479
Class pair (0, 5): 1493.629
Class pair (0, 6): 856.782
Class pair (0, 7): 2263.672
Class pair (0, 8): 2033.266
Class pair (0, 9): 2686.361
Class pair (1, 2): 2034.485
```

```
Inter-class Variances for Training Set:
Class pair (2, 6): 780.220
```

```
Inter-class Variances for Test Set (pairwise):
Class pair (0, 1): 637.086
Class pair (0, 2): 802.285
Class pair (0, 3): 319.053
Class pair (0, 4): 509.479
Class pair (0, 5): 932.506
Class pair (0, 6): 528.126
Class pair (0, 7): 1254.444
Class pair (0, 8): 1302.119
Class pair (0, 9): 1844.348
Class pair (1, 2): 1309.717
```

```
Inter-class Variances for Test Set:
Class pair (2, 6): 505.205
```

When examining the inter-class variances, the values indicate how well-separated the feature representations of different class pairs are. I pooled 2 classes together and calculate the pairwise variance, because there were too many classes, I calculated a subset of pairwise variances. For the training set, the pair (0, 9) exhibits the highest inter-class variance at 2686.361, suggesting a significant difference in the feature space between Class 0 (T-shirt/top) and Class 9 (Ankle boot). This high variance indicates that these classes are well-separated by the model. In contrast, the pair (0, 3) has a lower inter-class variance of 530.535, indicating that Classes 0 and 3 (Dress) are more closely related in the feature space, reflecting potential challenges in distinguishing between these two types of clothing. The test set follows a similar trend, with high variances for pairs like (0, 9) and lower variances for pairs such as (0, 3), supporting the observations from the training set.

These two types of variances are useful for understanding the spread and separation of features in the dataset. As we can see it is usually hard to differentiate between a pullover and shirt, I decided to check the variance. Classes 2 (Pullover) and 6 (Shirt), the training set shows a variance of 780.220, while the test set shows a variance of 505.205. These values are relatively high, indicating that there is a noticeable difference in the feature space between these two classes. However, the lower variance in the test set suggests that the model finds it more challenging to distinguish between pullovers and shirts in unseen data compared to the training data. This could be due to the visual similarities between these two types of clothing, which may not be as pronounced in the training set.

Intra-class vs Inter-class variance

When intra-class variance is significantly smaller than inter-class variance, I implies that the features within each class are tight and consistent, while features between classes are distinct and

well-separated. This balance leads to high classification accuracy because the model can clearly differentiate between classes while maintaining consistency within each class. Low intra-class variance is desirable because it indicates that the features for each class are consistent and tightly clustered. This tight clustering means that the model has learned a reliable and stable representation for each class, making it easier to distinguish between different classes. High inter-class variance is desirable because it shows that the features of different classes are well separated in the feature space. This separation means that the model can easily differentiate between different classes, leading to better classification performance.

Interpreting these variance values in the context of the t-SNE visualizations, we can see that the intra-class variances correspond to the spread of points within each class cluster. For example, the larger spread for Class 6 in the t-SNE plots aligns with its higher intra-class variance, indicating a diverse set of features within this class. Similarly, the tighter clustering of Class 7 points in the t-SNE plots reflects its lower intra-class variance. The inter-class variances are also evident in the t-SNE visualizations, where well-separated clusters, such as those for Class 0 and Class 9, correspond to higher inter-class variances. Conversely, overlapping or closely situated clusters, such as those for Class 0 and Class 3, are reflected in the lower inter-class variances. These visual and quantitative analyses together provide a comprehensive understanding of the model's feature extraction capabilities and the inherent complexities of the Fashion MNIST dataset.