CS464 Introduction to Machine Learning

Section 2

Fall 2024-25

Homework 2

1.12.2024

Furkan Komaç

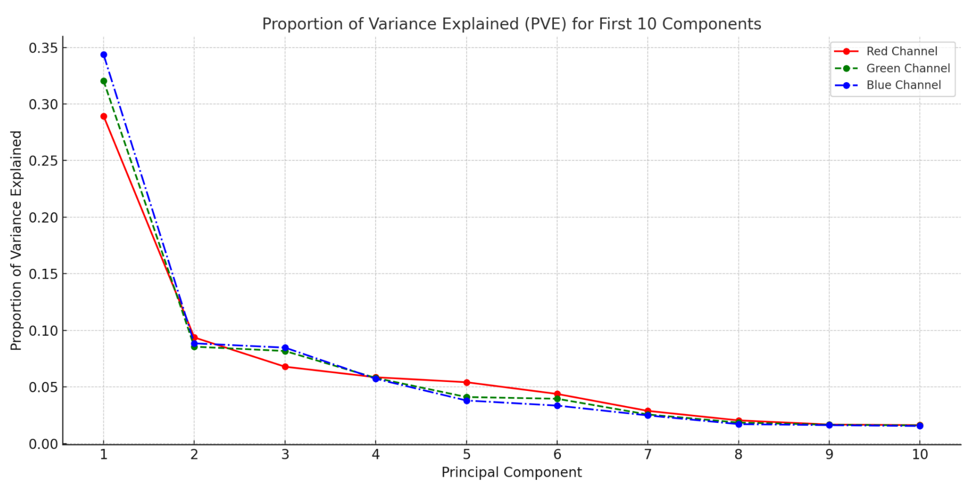
22102165

1 PCA Analysis

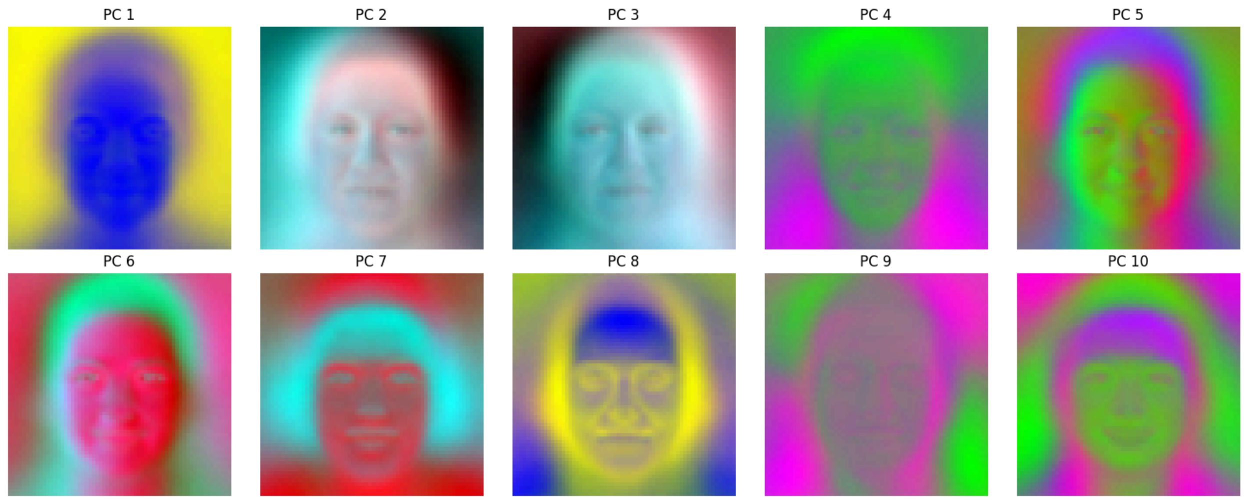
Question 1.1

After applying PCA on a given dataset, PVEs for the first ten principal components, each color (red, green, blue), are given in the table below. PVE of the first ten principal components shows that PCA1 captures most of the variance across the color channels: 29%, 32%, and 34% for red, green, and blue, respectively, underlining that the first component summarizes the major features dominating the dataset. This percentage dropped dramatically in subsequent components, with the contribution of PCA2 being around 9% for all channels and PCA3 at approximately 7-8%. Beyond PCA3, there is a much larger decrease in explained variance, tending towards a much less explained variance. This is because a bigger eigenvalue means the related vector covers more variance. Thus, we sorted the eigenvalues according to their related eigenvalues; the PVEs are also in descending order, as shown in the graph. The minimum number of principal components required to obtain at least 70% PVE is 11, 10, and 9 for red, green, and blue, respectively.

Principal Component	PVE (Red Channel)	PVE (Green Channel)	PVE (Blue Channel)
PCA1	28.93%	32.04%	34.36%
PCA2	9.37%	8.56%	8.85%
PCA3	6.79%	8.18%	8.48%
PCA4	5.86%	5.8%	5.72%
PCA5	5.42%	4.11%	3.81%
PCA6	4.38%	3.96%	3.35%
PCA7	2.89%	2.57%	2.49%
PCA8	2.05%	1.86%	1.72%
PCA9	1.68%	1.66%	1.62%
PCA10	1.63%	1.6%	1.55%

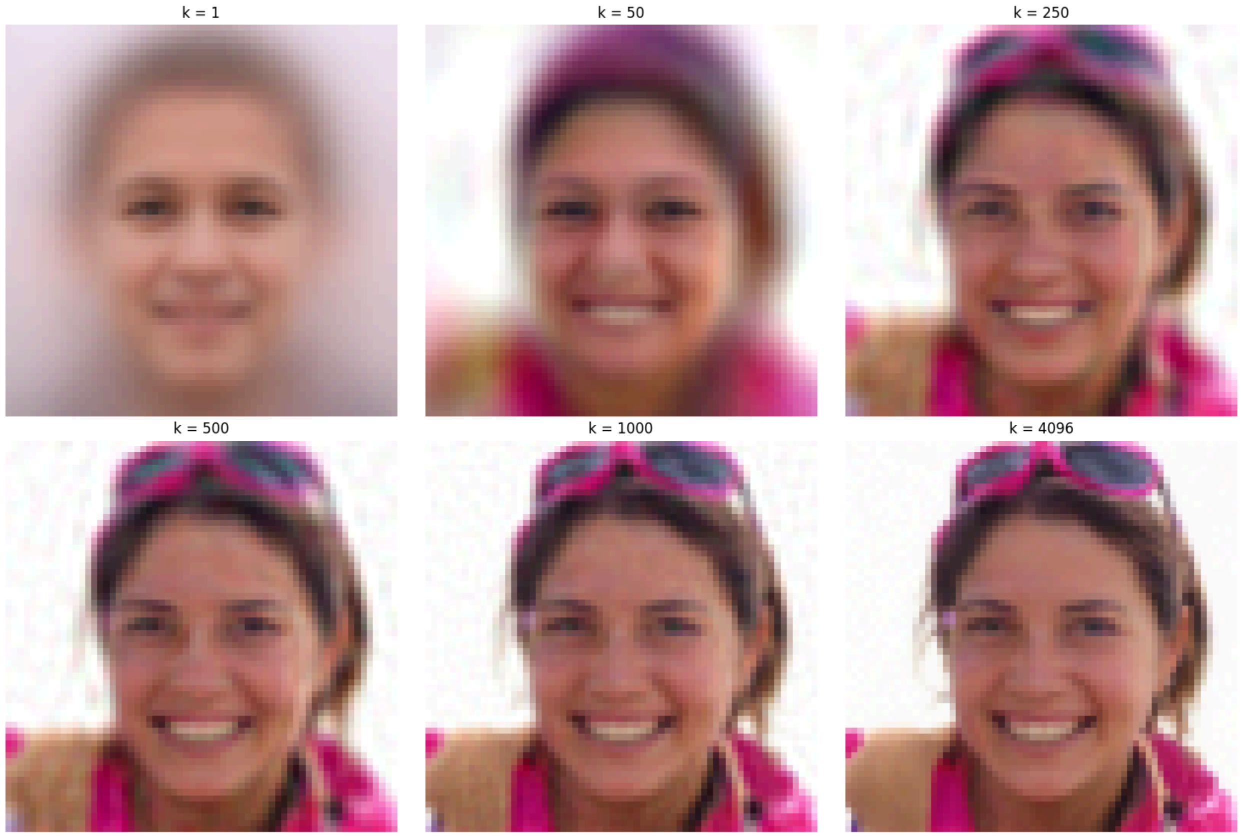


Question 1.2



The visualization of the first 10 principal components is given in the figure above. Results show that how earlier principal components, like PC1 and PC2, represent broad structural and global aspects related to facial outlines and general facial symmetry, while the later components -like PC9 and PC10- catch minor, localized variations, and as such texture or very much distinct features of a different kind, like those that may define eyes, lip structure, or hairline characteristics.

Question 1.3

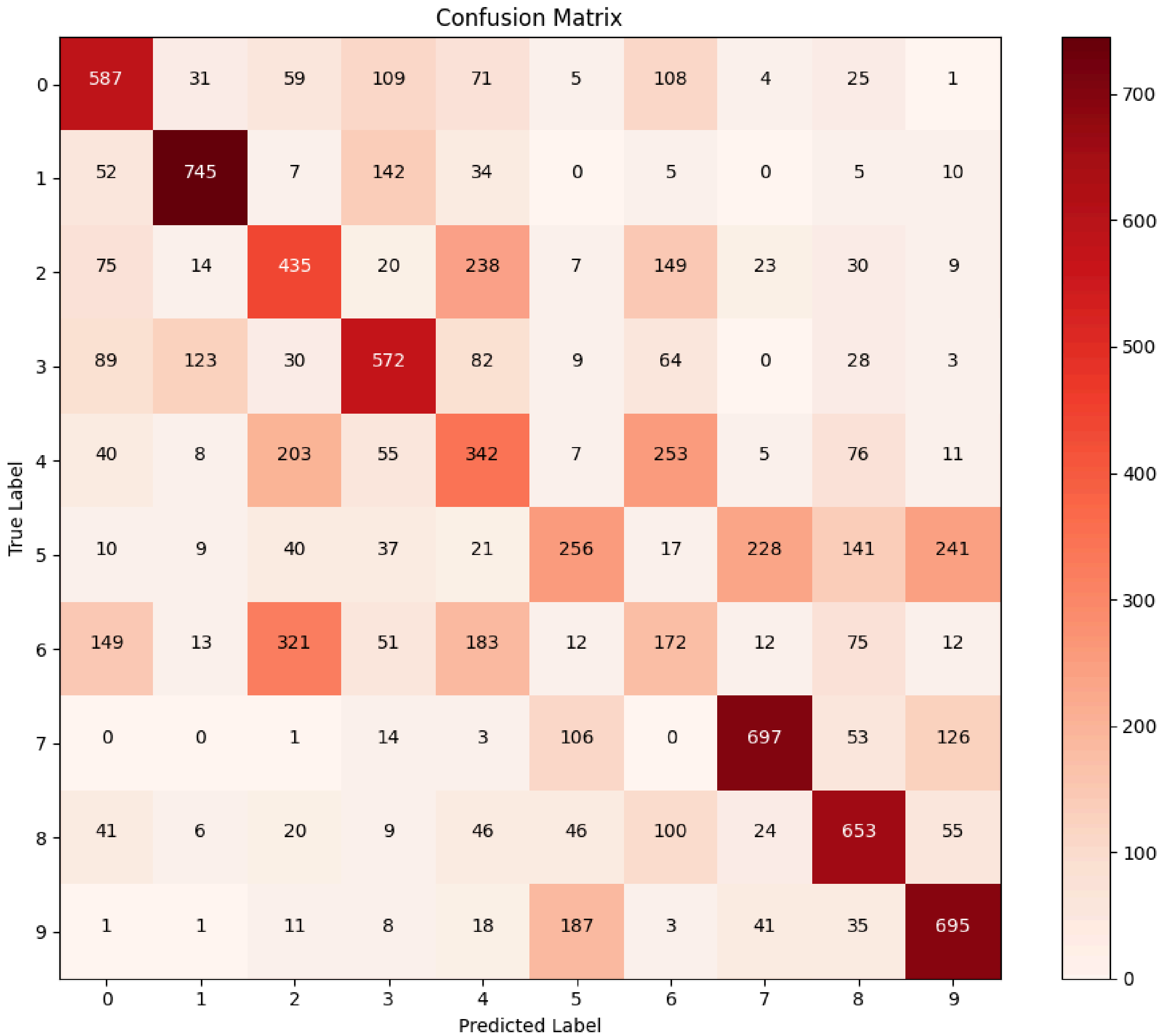


The reconstructed photo uses the first k principal components to analyze and reconstruct the first image in the dataset where $k \in \{1, 50, 250, 500, 1000, 4096\}$ is given above. Result shows that as the k increases the reconstructed image becomes more clear and more like the original image. Also it can be said the increases in smaller k's are more important and have more effect on output, because the crucial variances are captured in first PC's with bigger eigenvalues.

2 Logistic Regression

Question 2.1

Accuracy of the model is %51.5 with following confusion matrix:



Question 2.2

Following results are obtained for different hyperparameters:

Batch Size	`Test accuracy with batch size 1: 0.8236`
	`Test accuracy with batch size 64: 0.6199`
	`Test accuracy with batch size 3000: 0.1671`

Weight Initialization Method	`Test accuracy with weight initialization method zero: 0.7677`
	`Test accuracy with weight initialization method uniform: 0.595`
	`Test accuracy with weight initialization method normal: 0.5023`

Learning Rate	`Test accuracy with learning rate 0.01: 0.7513`
	`Test accuracy with learning rate 0.001: 0.5976`
	`Test accuracy with learning rate 0.0001: 0.2573`
	`Test accuracy with learning rate 1e-05: 0.0787`

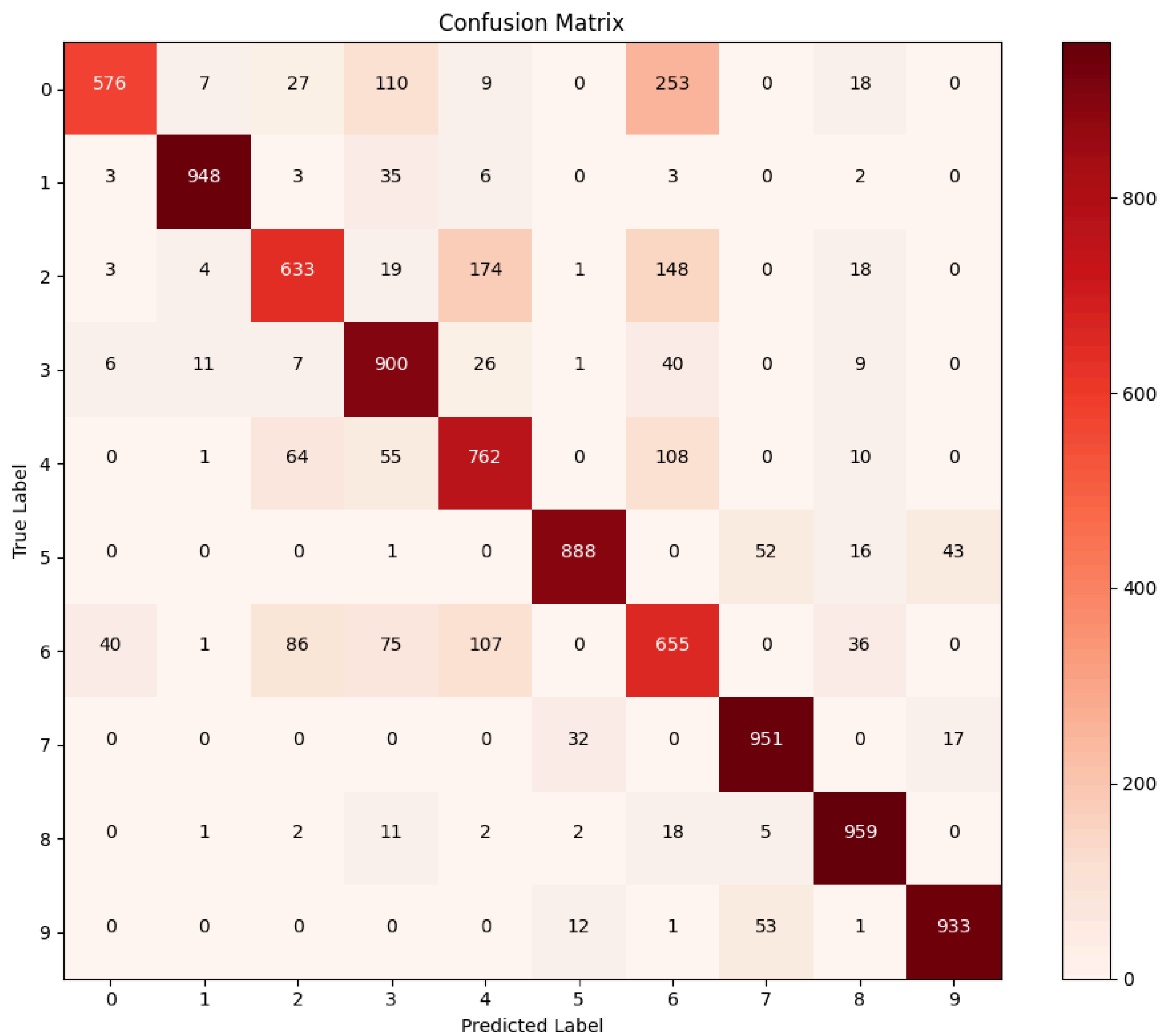
Regularization Coefficient (λ)	`Test accuracy with regularization coefficient (lambda) 0.01: 0.5232`
	`Test accuracy with regularization coefficient (lambda) 0.0001: 0.5244`
	`Test accuracy with regularization coefficient (lambda) 1e-09: 0.5131`

So for the optimal model I selected:

Batch Size: `1`
Weight Initialization Method: `Zero initialization`
Learning Rate: `0.01`
Regularization Coefficient (λ): `0.01`

Question 2.3

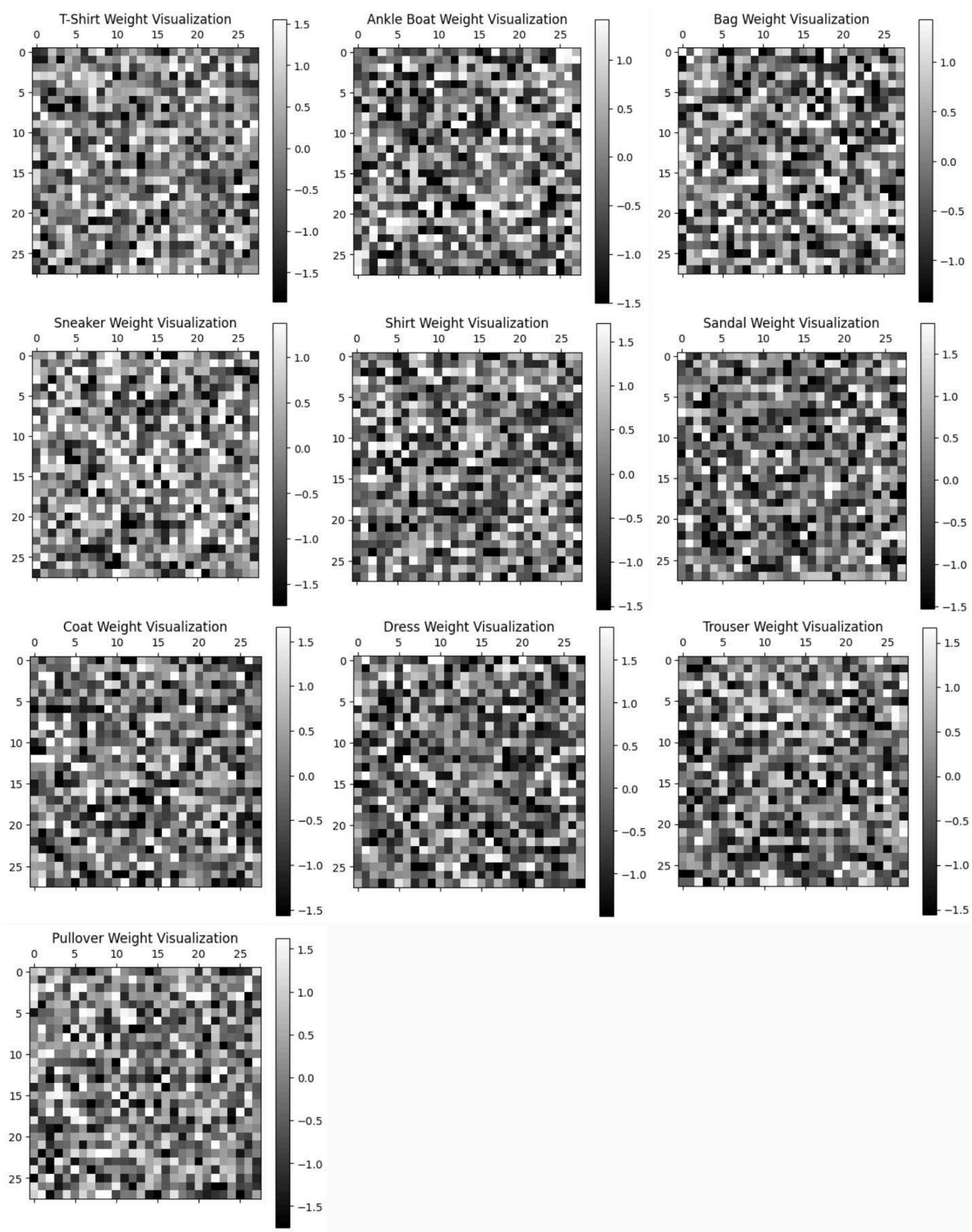
Accuracy of the optimal model is %82.05 with following confusion matrix:



Optimal model increased the accuracy from %51.8 to %82.05. Which shows us how important it is to decide hyperparameters correctly.

Question 2.4

Visualization for weight vector of each class:



This data shows different weight visualization patterns across different classes, reflecting the model's focus for different image regions. Classes such as sneakers and bags are easier to differentiate which is related to the high rate of scores shown in Question 2.5. In comparison, classes such as 'Shirts' and 'Sweaters' appear to have scattered and non-obvious patterns. This is probably linked to low performance scores and indicates some difficulties in clearly distinguishing their characteristics. On the other hand, in classes such as 'Pants', there is a clear contrast between lighter and darker regions, suggesting that the model is more focused on these specific features. In conclusion, these differences in weight distributions seem to reveal how well the model can or cannot distinguish between certain classes.

Question 2.5

Performance metrics for each class:

```
Class T-Shirt:    Precision=0.92, Recall=0.58, F1 Score=0.71, F2 Score=0.62

Class Trouser:    Precision=0.97, Recall=0.95, F1 Score=0.96, F2 Score=0.95

Class Pullover:   Precision=0.77, Recall=0.63, F1 Score=0.69, F2 Score=0.66

Class Dress:      Precision=0.75, Recall=0.90, F1 Score=0.82, F2 Score=0.86

Class Coat:       Precision=0.70, Recall=0.76, F1 Score=0.73, F2 Score=0.75

Class Sandal:     Precision=0.95, Recall=0.89, F1 Score=0.92, F2 Score=0.90

Class Shirt:      Precision=0.53, Recall=0.66, F1 Score=0.59, F2 Score=0.63

Class Sneaker:    Precision=0.90, Recall=0.95, F1 Score=0.92, F2 Score=0.94

Class Bag:        Precision=0.90, Recall=0.96, F1 Score=0.93, F2 Score=0.95

Class Ankle Boat: Precision=0.94, Recall=0.93, F1 Score=0.94, F2 Score=0.93
```

The results highlight differences in model performance across classes. For example, “Trousers” scores high on all metrics, indicating strong performance, while “Shirt” has relatively low Sensitivity and F1 scores, indicating difficulties in accurately discriminating it. The F2 score emphasizing Recall suggests that recall is prioritized in certain cases, such as “Dress”, where it outperforms the F1 score. These observations indicate class imbalance or overlap in feature representation and provide insights into potential areas for improvement, such as data augmentation or feature engineering.