Ahmet Sayan

21903426

1 PCA Analysis

Image data is uploaded into separate arrays for Red, Green and Blue then they flattened as pre-processing. Total data is 10000x4096 for each color channel.

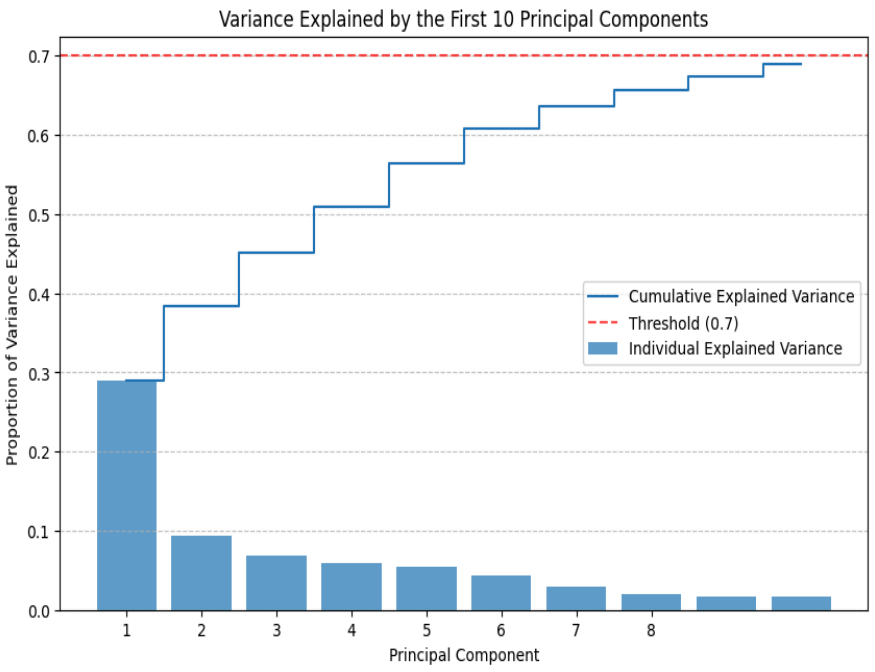
Question 1.1

I created PCA function to calculate Principal Components and find how many Principal Components I need to explain 70% variance of the data. This function calculates covariance matrix of the standardize data for channels and find eigenvector/eigenvalues for covariance matrix. Then returns the eigenvector and eigenvalues of the sorted eigenvalues.

Second function is PVE_plot. This function calculates PVE by using eigenvalues calculated previously then plot explained and cumulative variance by principal components. Then prints the number of the PC is needed to obtain 70% PVE.

To capture 70% of the variance, the Red, Green, and Blue channels require **11**, **10**, and **9** principal components, respectively. Most of the time first Principle Component should be higher but I couldn't get higher results. Therefore found the number of PCs higher to explain 70% of the variance.

Red Channel



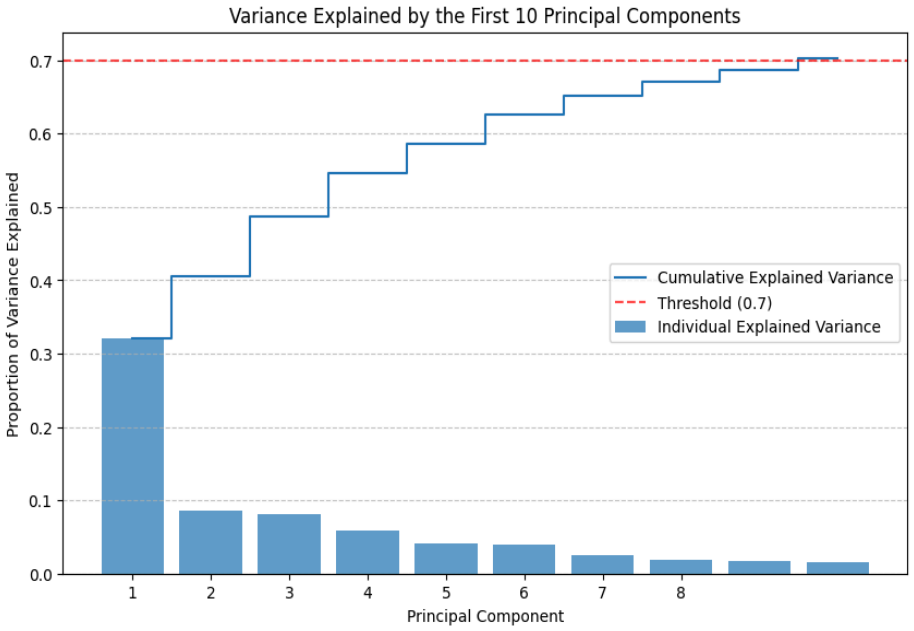
PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
28.929%	9.3737%	6.788%	5.859%	5.419%	4.384%	2.887%	2.049%	1.168%	1.163%

Explained Variance Ratio for Red Channel respect to PCs

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
28.929%	38.30%	45.091%	50.95%	56.37%	60.75%	63.64%	65.69%	67.37%	69%

Explained Cumulative Variance Ratio

Green Channel



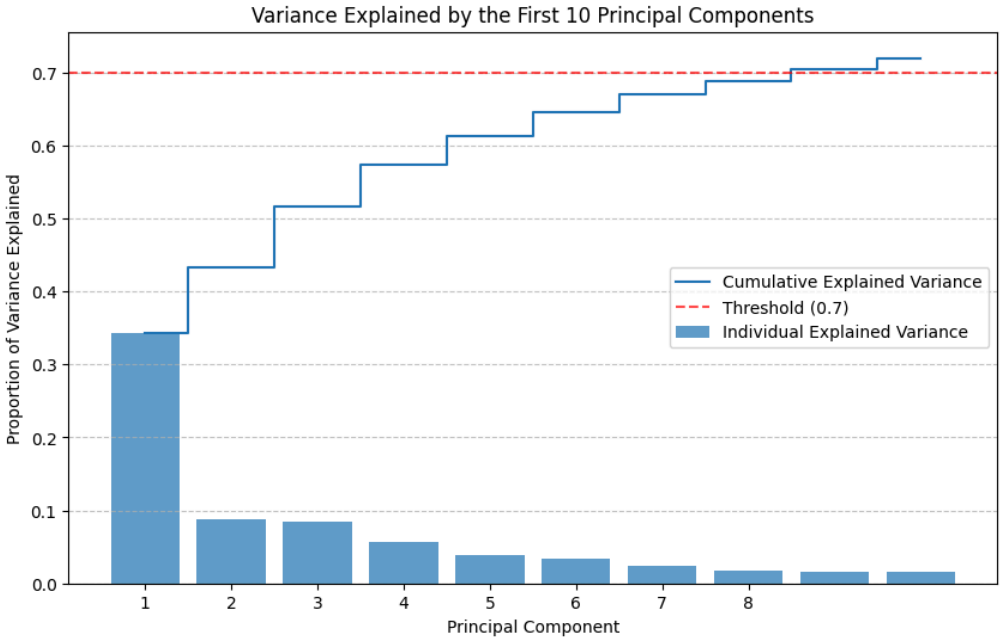
PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
32.041%	8.559%	8.177%	5.796%	4.110%	3.958%	2.568%	1.855%	1.663%	1.604%

Explained Variance Ratio for Red Channel respect to PCs

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
32.041%	40.60%	48.777%	54.57%	58.68%	62.64%	65.21%	67.06%	68.73%	70.33%

Explained Cumulative Variance Ratio

Blue Channel



PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
34.357%	8.854%	8.477%	5.722%	3.805%	3.354%	2.486%	1.717%	1.624%	1.553%

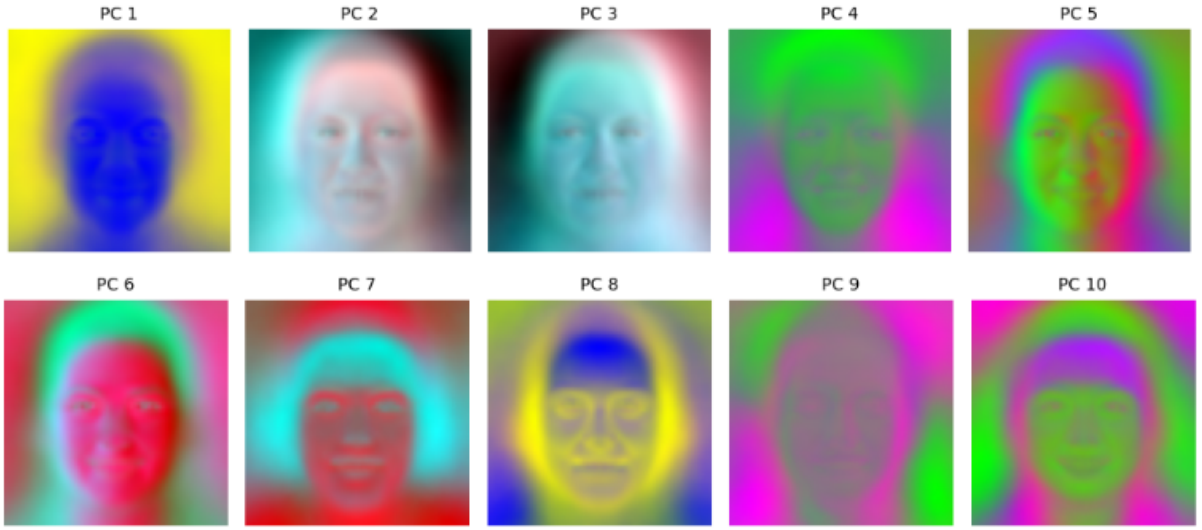
Explained Variance Ratio for Red Channel respect to PCs

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
33.357%	43.21%	51.689%	57.41%	61.21%	64.57%	67.05%	68.77%	70.40%	71.95%

Explained Cumulative Variance Ratio

Question 1.2

I stacked first 10 Principal Components for each color and created images of them after normalized them.

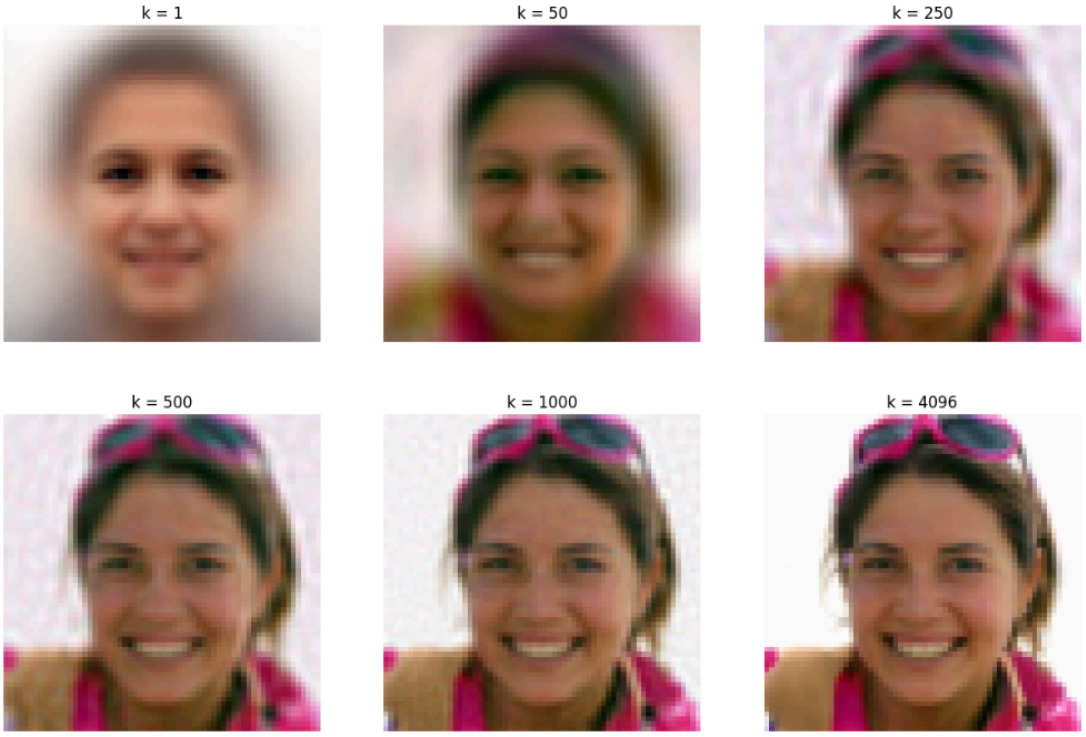


Visuals of Eigenvectors

Created images resemble to human faces and capture the dominant features of the dataset like face and hair shapes, teeth, noses, eyes and lips. Each principal component highlights the different patterns.

Question 1.3

I used $k\epsilon\{1,\ 50,\ 250,\ 500,\ 1000,\ 4096\}$ principal component to reconstruct the image.



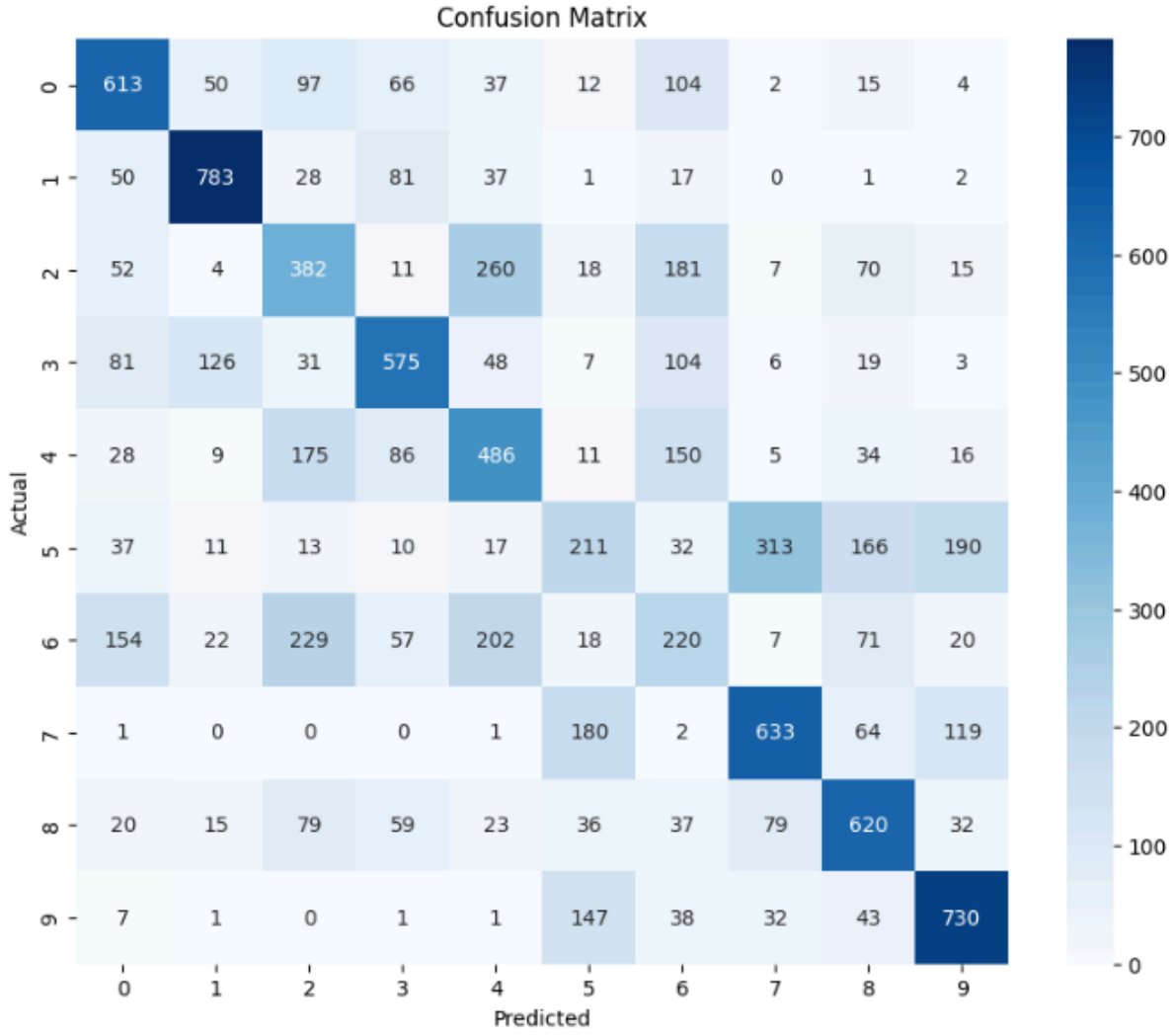
First image only gives the dominant features from the dataset since we used only one Principal Component. The more we used Principal Component, image gets better since explanation of the variance is increased so it's gets better to reconstruct the image. Using 4096 PC is exactly match the original image since my images 64x64 meaning that 4096 feature. Even though using fewer principal components is computationally efficient, we lost the details of the image. However, k=500 and k=1000 looks good for this trade off.

2 Logistic Regression

I uploaded and pre-processed the data with function provided. Then I created gradient descent and softmax functions to implement Logistic Regression Classifier.

Question 2.1

I trained my model by using functions I created. These functions optimize the weights to minimize loss function and give the best accuracy for train data. I trained default model with parameters provided and get **0.53** accuracy.

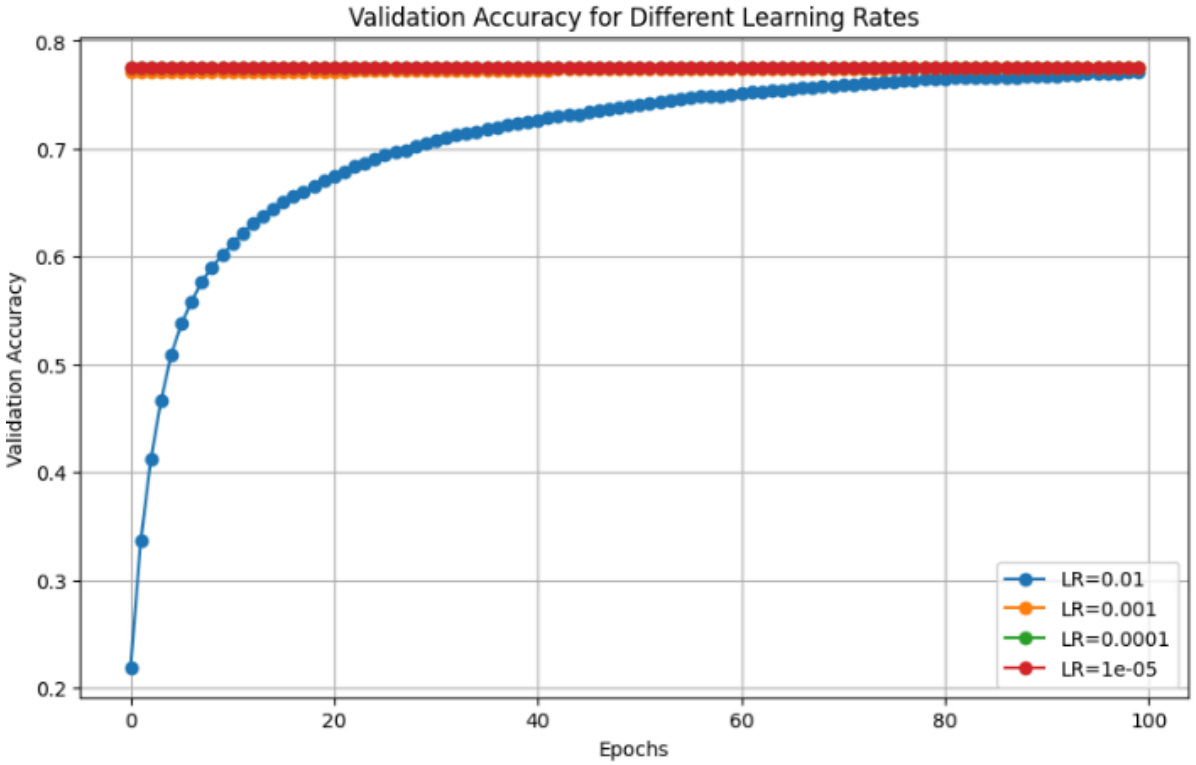


Confusion Matrix

Question 2.1

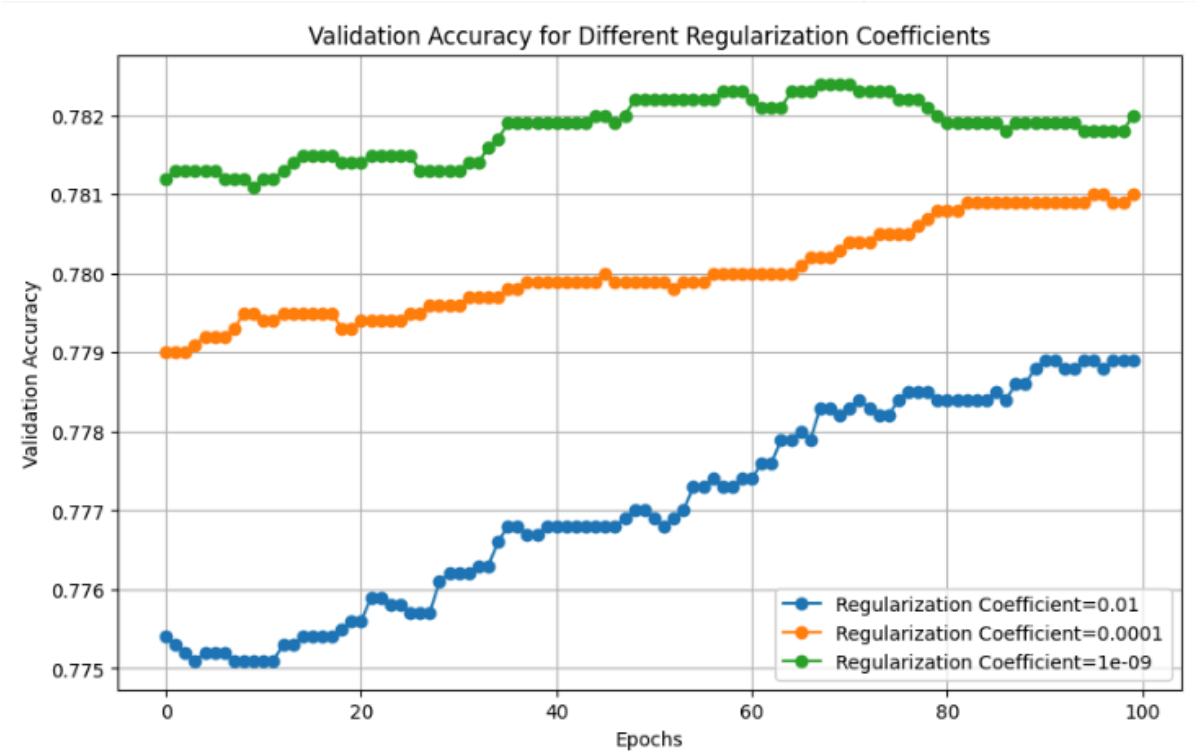
I divided my train data into train and validation for that part to find best hyperparameters.

Learning Rate



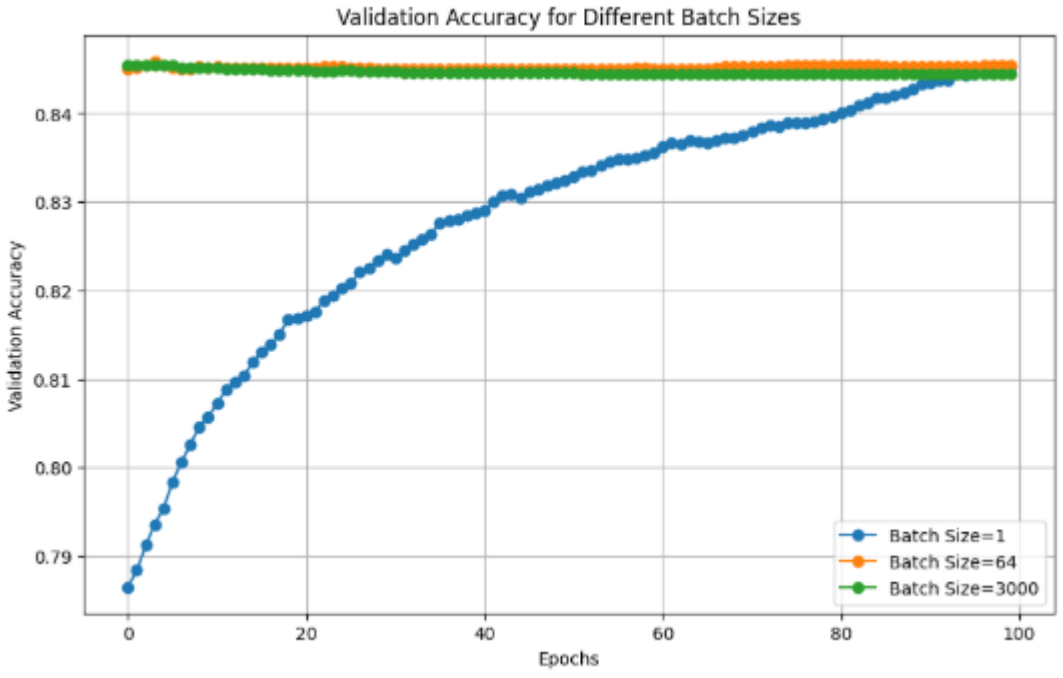
All learning rates give similar accuracies at the end of these epochs. Since large learning rates cause to missing optimal points. I choose 0.0001 learning rate for my best model to avoid long computation time.

Regularization Coefficient



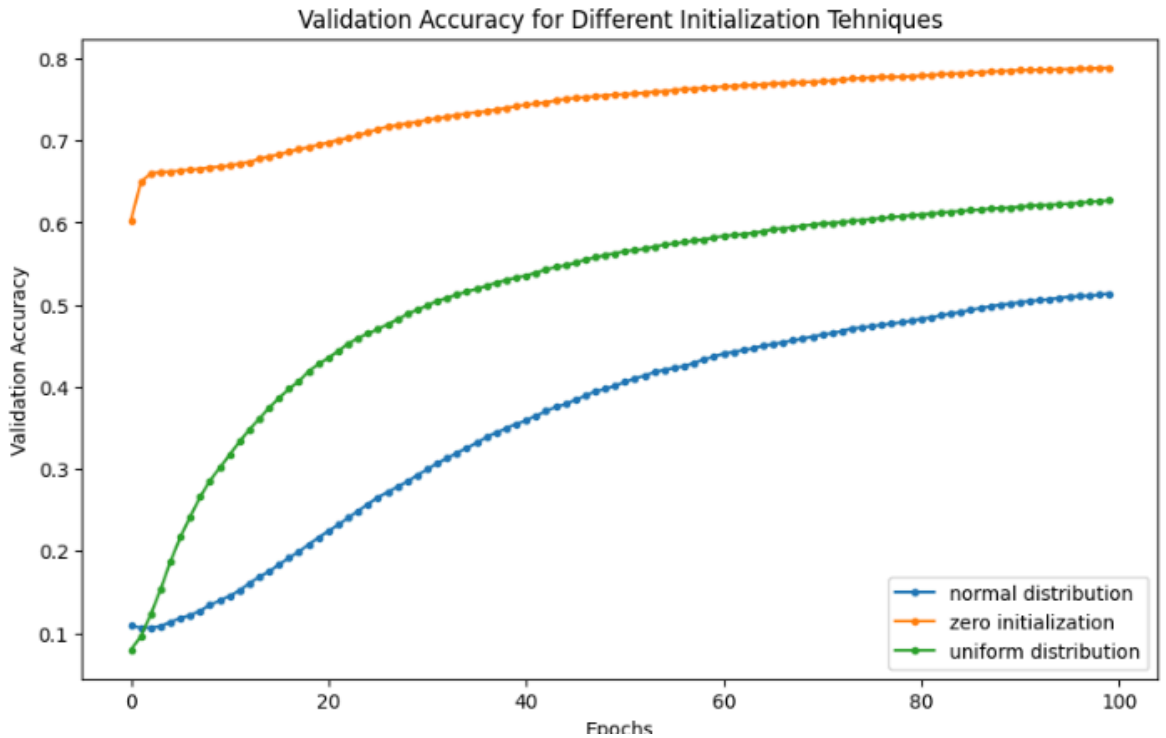
Smallest Regularization Coefficient gave the best result through the epochs. I choose 10^{-9} Regularization Coefficient for my best model.

Batch Size



Batch size 64 is more efficient than batch size 1 and it starts good and keep accuracy slightly better.

Initialization Tecniques

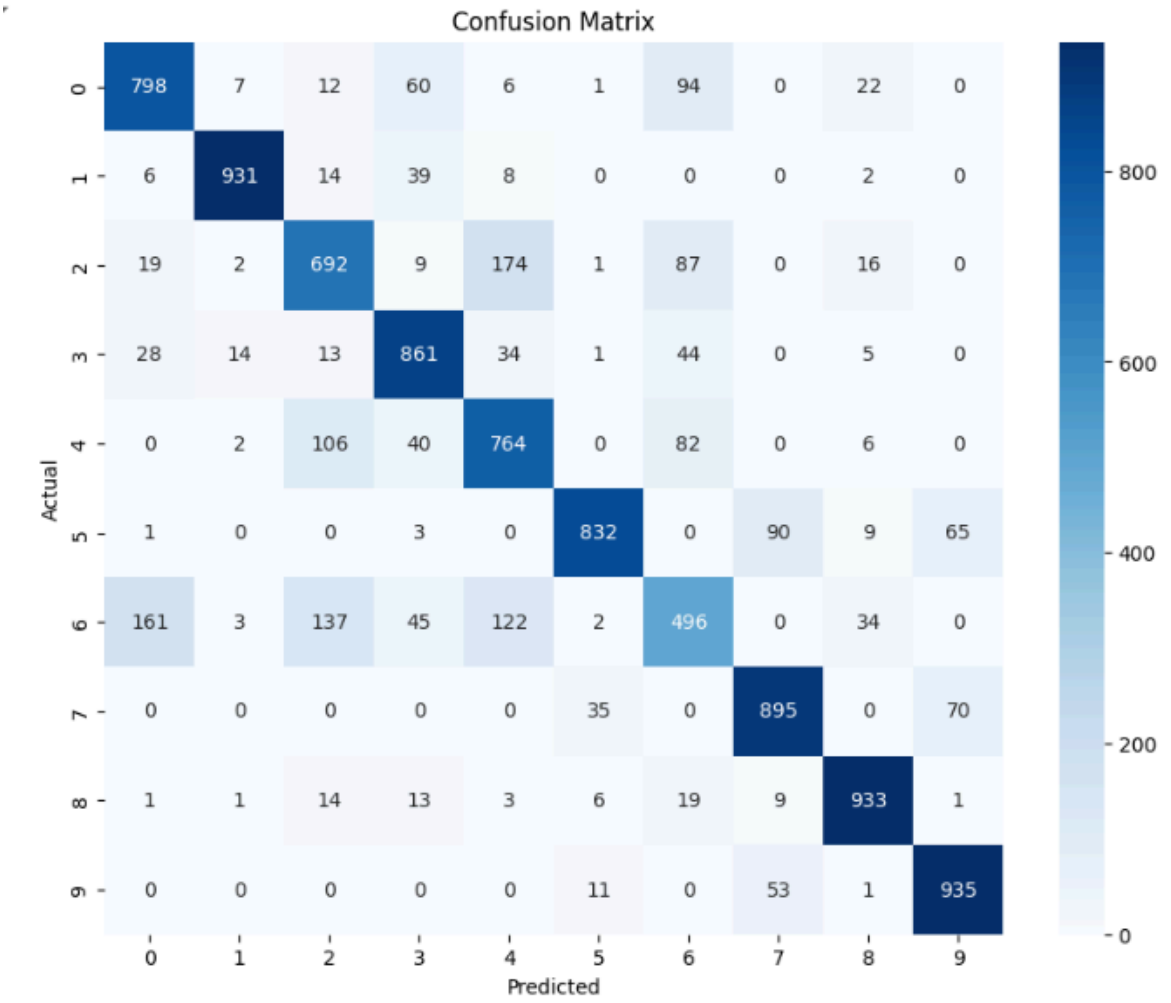


Zero initialization gives the best accuracy.

Question 2.3

Best Parameters: Regularization Coefficient = 10^{-9}, Learning Rate = 0.0001, Zero Weight Initialization, Batch Size = 64

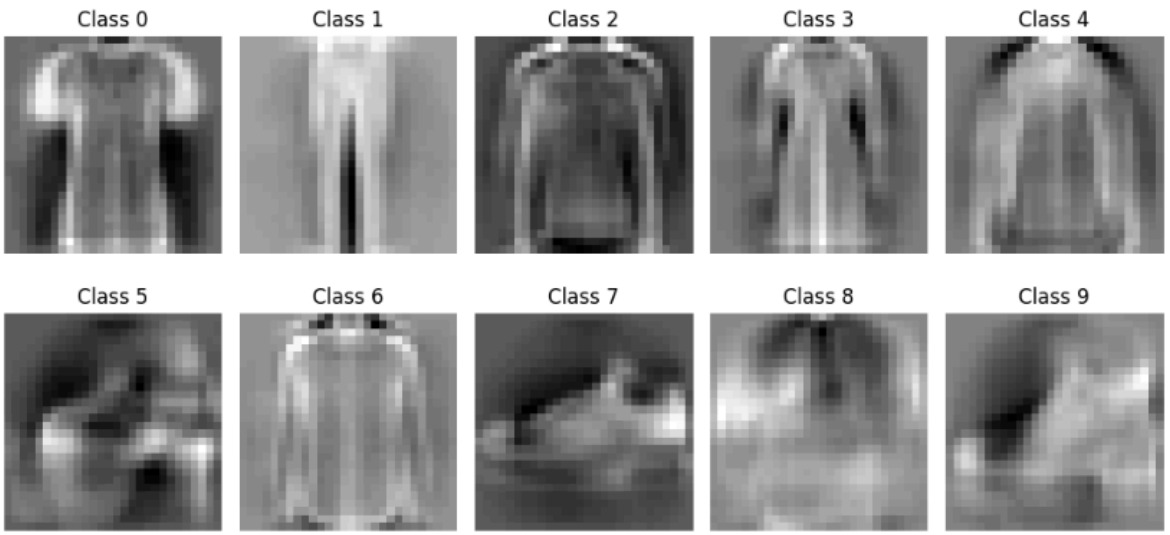
Accuracy is **81.37%**



Confusion Matrix

Question 2.4

I visualized my final weights.



Each weights corresponds to a specific type of clothing. For example, first image similar to tshirt, second image similar to pant, etc.. . The weights are trained to highlight pixel patterns that associated with each clothing type. They represent the significant patterns in each class.

Question 2.5

	Precision	Recall	F1 Score	F2 Score
Class 0	0.7870	0.7980	0.7925	0.7958
Class 1	0.9698	0.9310	0.9500	0.9385
Class 2	0.7004	0.6920	0.6962	0.6937
Class 3	0.8047	0.8610	0.8319	0.8491
Class 4	0.6877	0.7640	0.7238	0.7474
Class 5	0.9359	0.8320	0.8809	0.8509
Class 6	0.6034	0.4960	0.5445	0.5143
Class 7	0.8548	0.8950	0.8745	0.8867
Class 8	0.9076	0.9330	0.9201	0.9278
Class 9	0.8730	0.9350	0.9029	0.9219

Based on the confusion matrix and scores, model has problem to predict Class 6. It has the worst scores on Precision, Recall and F scores and can be seen model predict it as Class0, Class 2 and Class 4 mostly. Class 2 is also one of the problematic but not bad as Class 6.