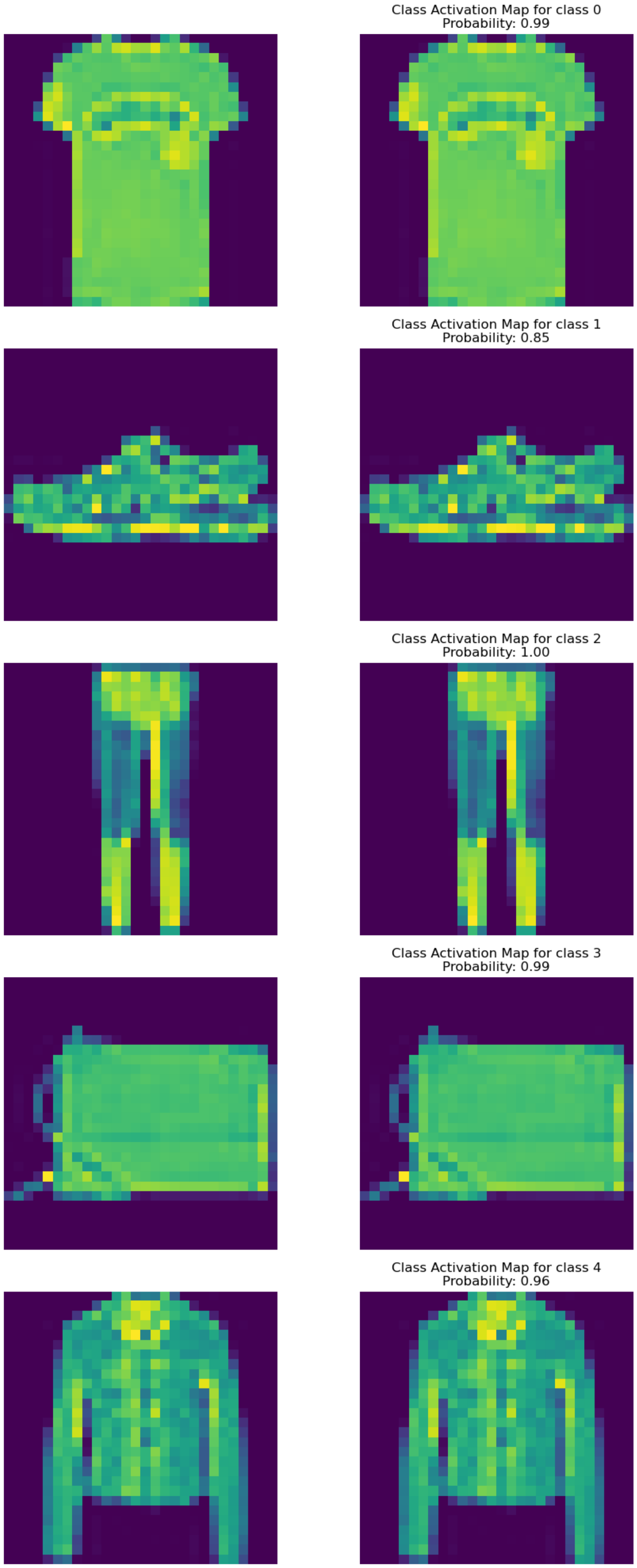
Q3: Analysis on Question2 model1.

Plotting the Class Activation Map for all the classes along with their SoftMax probabilities:



Class Activation Map of CNN

Class activation maps (CAMs) are used in convolutional neural networks (CNNs) to visualize the regions of an input image that are important for a particular classification decision.

To generate a CAM, we first obtain the feature maps from the last convolutional layer of the CNN. These feature maps represent the learned features of the input image. We then compute the class activation map by taking a weighted sum of these feature maps, where the weights are the learned coefficients of the last fully connected layer of the CNN. The resulting activation map highlights the regions of the input image that are most important for the predicted class.

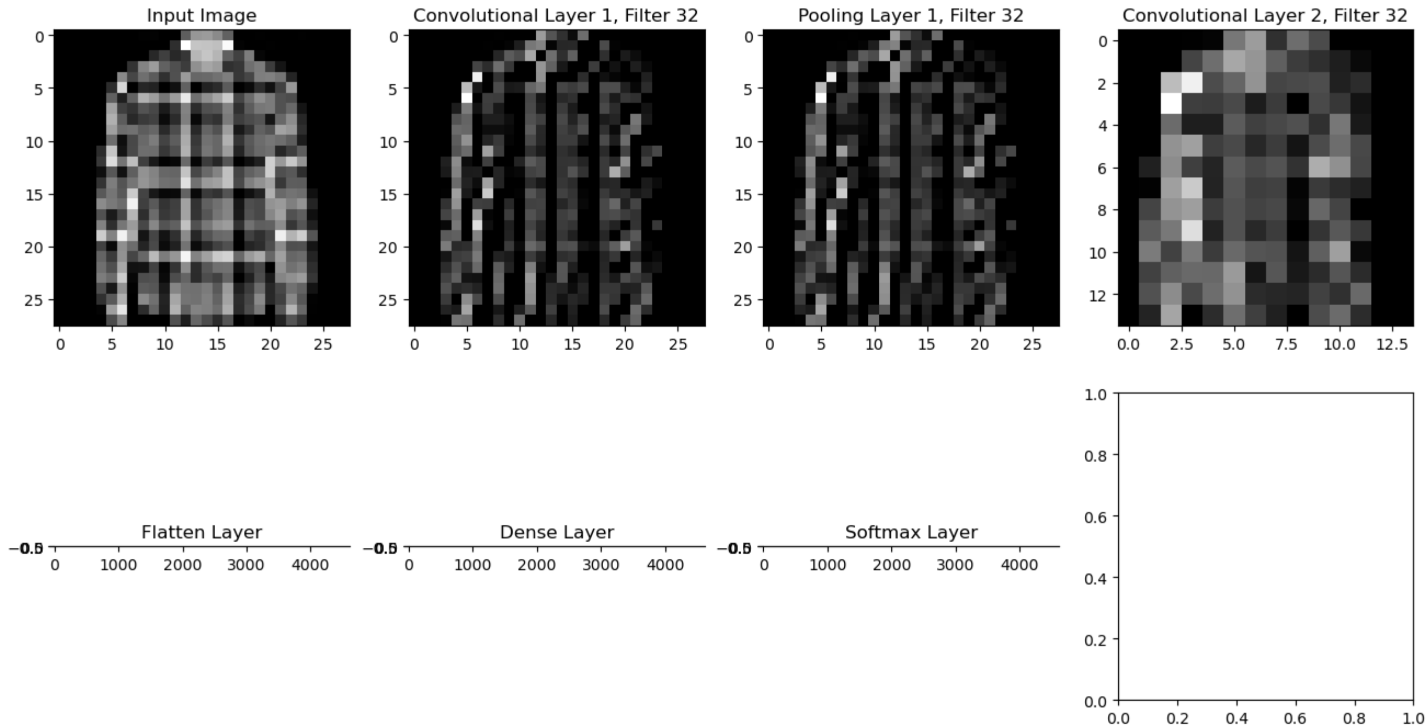
From the plot, we have a CNN that has been trained to classify images into 5 classes: t-shirt, sandal, trouser, bag, and coat. Let's assume that the input image has been classified as t-shirts with a probability of 0.99. We can generate a CAM for this class by taking the weighted sum of the feature maps, where the weights are the coefficients of the last fully connected layer that correspond to the t-shirts class.

The resulting CAM will highlight the regions of the input image that were most important for predicting the t-shirts class. This can give us insight into what features the CNN is looking for when it makes its predictions. Similarly, we can generate CAMs for the other classes to see what regions of the input image are important for each class.

For example, if we generate a CAM for the trouser class, which was predicted with a probability of 1.00, the resulting activation map will highlight the regions of the input image that were most important for predicting the trouser class. We can repeat this process for each of the other classes to get a better understanding of how the CNN is making its predictions.

Conclusion: the region which has higher intensity has contributed the most in classifying one class from another class

Analysing the CNN architecture:



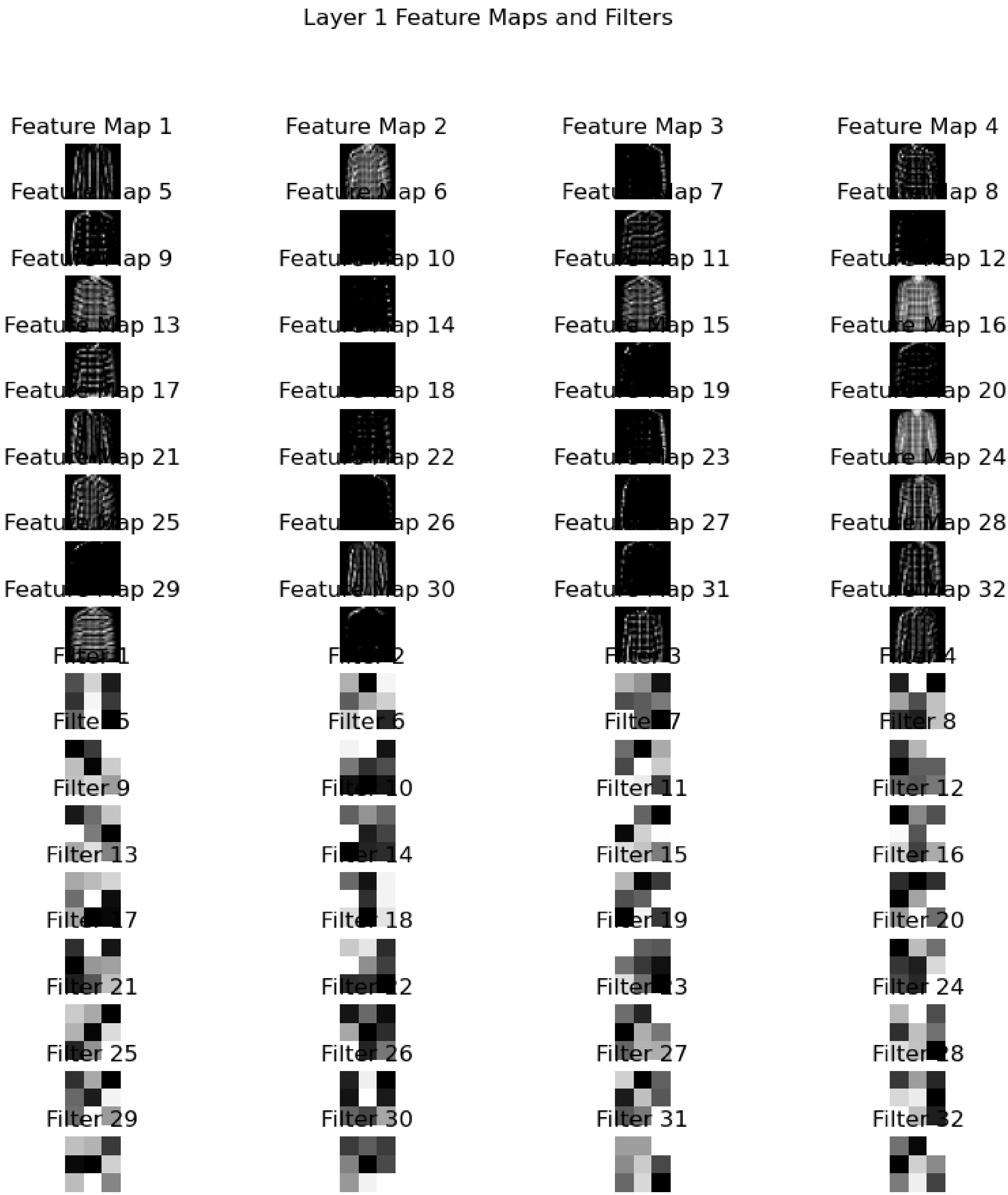
From the above figure we can analyse the CNN architecture in detail. Given the input image, we first apply the convolution with 32 3x3 filter, resulting image can be analysed form the above figure. Then we apply max pooling to further reduce the dimension capturing only the pixel values that contains the most information. After max pooling we again apply a layer of convolution to further reduce the dimension and to extract the features. Once the features has been extracted, we then flatten the pixel values. This flattened layer is known as encodings.

"encoding" refers to the process of transforming input data (usually images) into a more compact and abstract representation that captures important features and patterns in the input. This is typically done using a series of convolutional and pooling layers that progressively reduce the spatial dimensions of the input while increasing the number of feature maps or channels.

The output of the encoding layers is often flattened and fed into one or more fully connected layers that performs the classification task. In our case, we have used the dense layer.

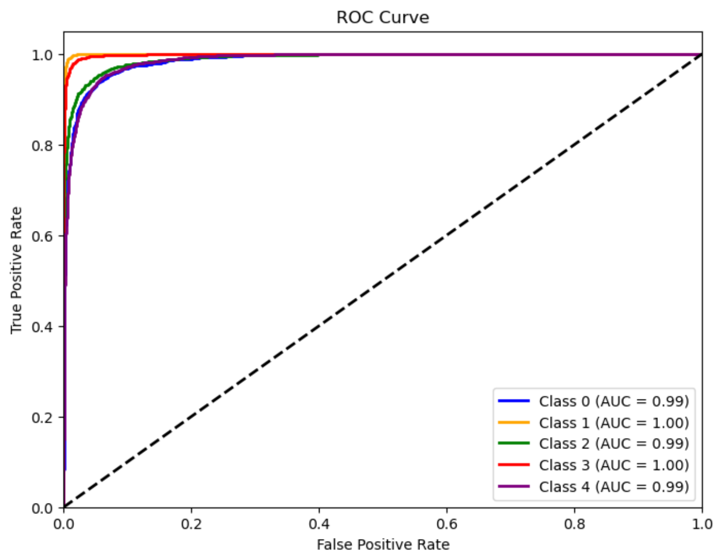
The fully connected layer is then connected to the SoftMax layer. the SoftMax activation function is typically used in the final layer to convert the output of the network into a probability distribution over the classes. This allows us to interpret the output of the network as a probability that the input belongs to each class, which can be used to make predictions or perform classification.

Analysing the CNN feature map for layer 1



Analysing the feature map for layer 2

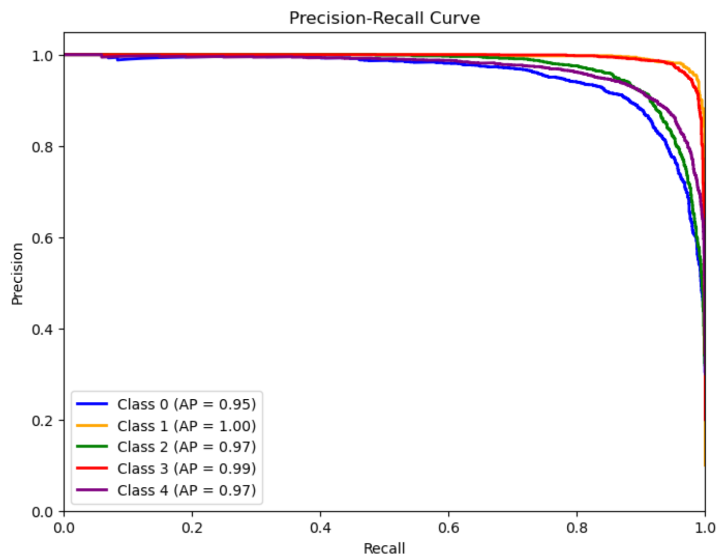
Plot the ROC and Precision Recall Curve for Q2 model1.



ROC Curve interpretation

In this ROC curve, there are five classes (Class 0, Class 1, Class 2, Class 3, and Class 4), and each class has an associated AUC value. The AUC (Area Under the Curve) is a metric that represents the overall performance of the model. An AUC value of 1.0 indicates a perfect classifier, while an AUC value of 0.5 indicates a classifier that performs no better than random chance.

Looking at the AUC values for each class in this ROC curve, we can see that they are all quite high, ranging from 0.99 to 1.0. This suggests that the model is performing very well for each of the five classes. Specifically, Class 1 and Class 3 have an AUC value of 1.0, which indicates that the model can perfectly distinguish between positive and negative examples for these classes.

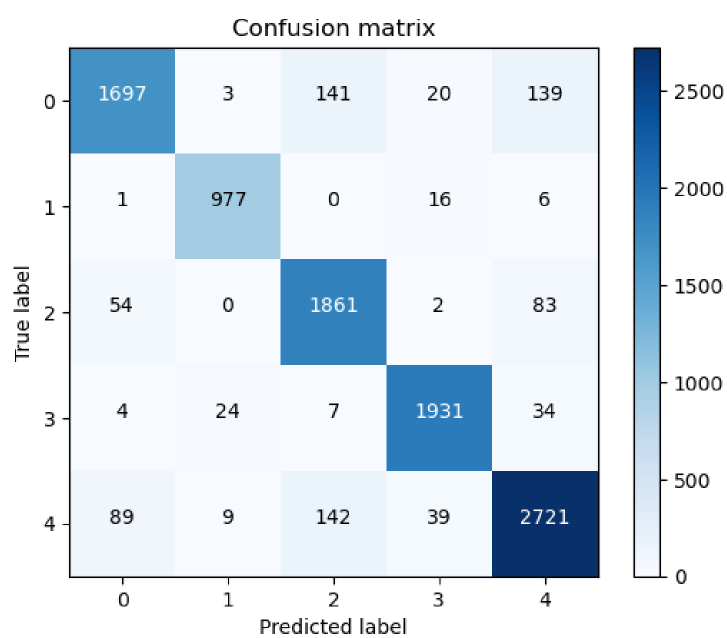


PR Curve interpretation

In this PR curve, there are five classes, and each class has an associated average precision (AP) value. An AP value of 1.0 indicates a perfect classifier, while an AP value of 0.0 indicates a classifier that performs no better than random chance.

Looking at the AP values for each class in this PR curve, we can see that they are all relatively high, ranging from 0.95 to 1.0. This suggests that the model is performing very well for each of the five classes. Specifically, Class 1 has an AP value of 1.0, which indicates that the model can perfectly distinguish between positive and negative examples for this class.

It also highlights the areas where the model's performance could potentially be improved, such as for Class 0 and Class 2 where the AP values are slightly lower than the others.



Confusion Matrix interpretation

Looking at the values in the matrix, we can see that the majority of the diagonal values are high, indicating that the model is making correct predictions for those classes. For example, the value of 1697 in the (1,1) position indicates that there were 1697 instances where the true class was class 0, and the model correctly predicted it as class 0. Similarly, the value of 977 in the (2,2) position indicates that there were 977 instances where the true class was class 1, and the model correctly predicted it as class 1.

However, there are also some off-diagonal values, indicating misclassifications. For example, the value of 54 in the (3,1) position indicates that there were 54 instances where the true class was class 2, but the model incorrectly predicted it as class 0. Similarly, the value of 34 in the (4,5) position indicates that there were 34 instances where the true class was class 4, but the model incorrectly predicted it as class 5.