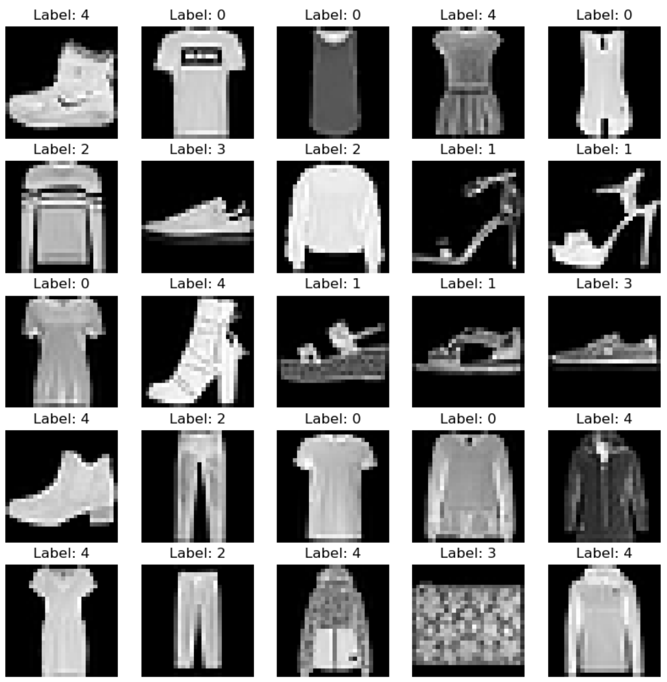
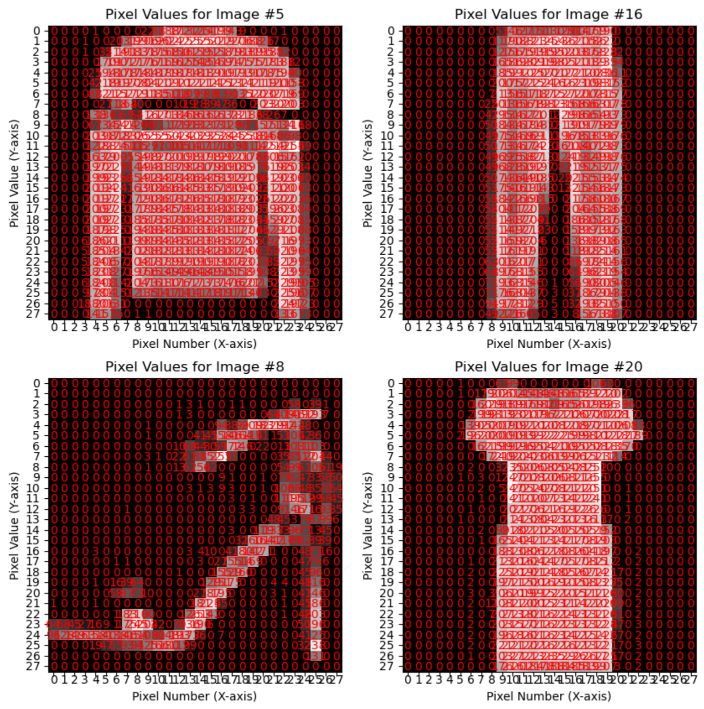
Question1: Using default network.

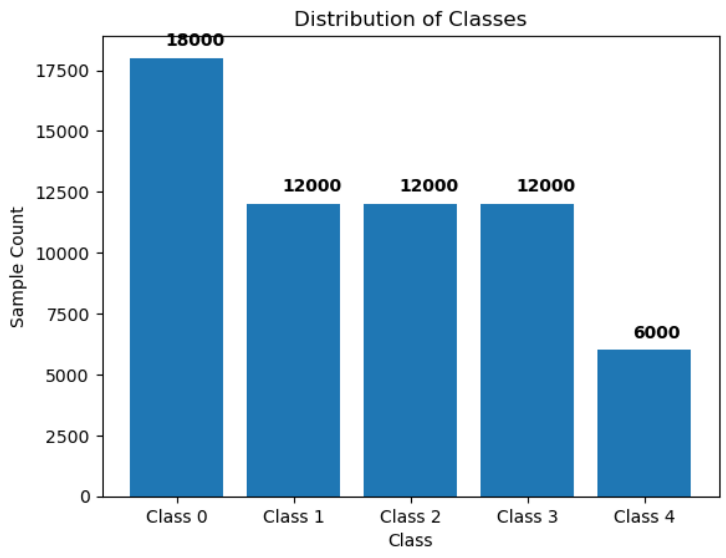
Plot the train images and its corresponding labels. From the below labels we are not able to figure out the classes to which a particular group of images belongs.

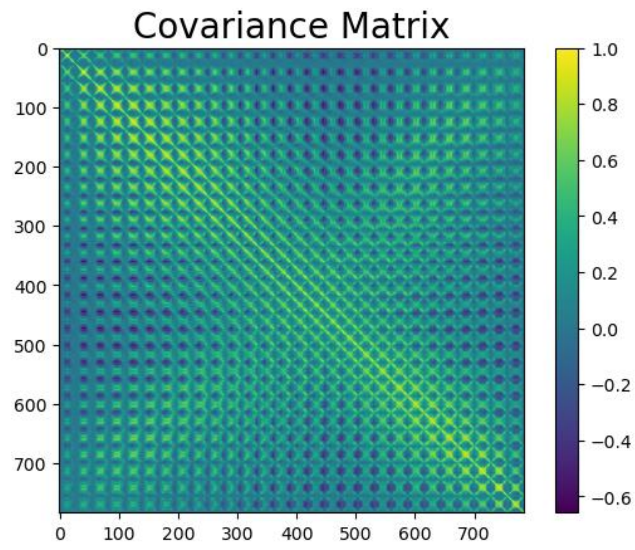
The dataset consists of 70000 grayscale images of 28x28 pixels each, representing 5 mystery classes. The images are split into 60000 training samples and 10000 test samples. Visualizing the input samples and their corresponding pixels.





Plotting the data distribution





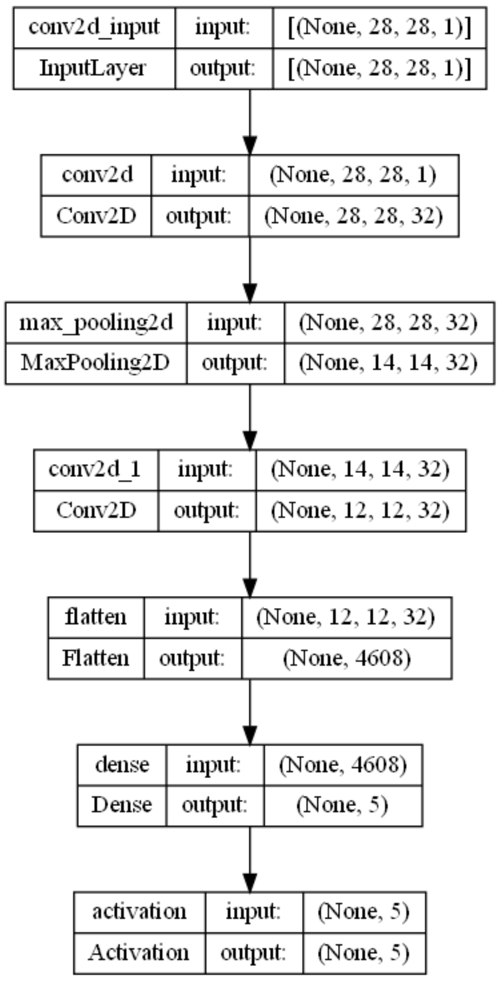
Distribution of the class shows that for 3 classes we have equal amount of data. But for class 1 and class 4, the dataset is not balanced properly.

We also tested for outliers' detection following which we got that there are few outliers. These outliers may be because of the mystery classes. Following are the results after removing outliers. This is just for analysis purpose, for training the CNN we will be using the entire dataset provided.

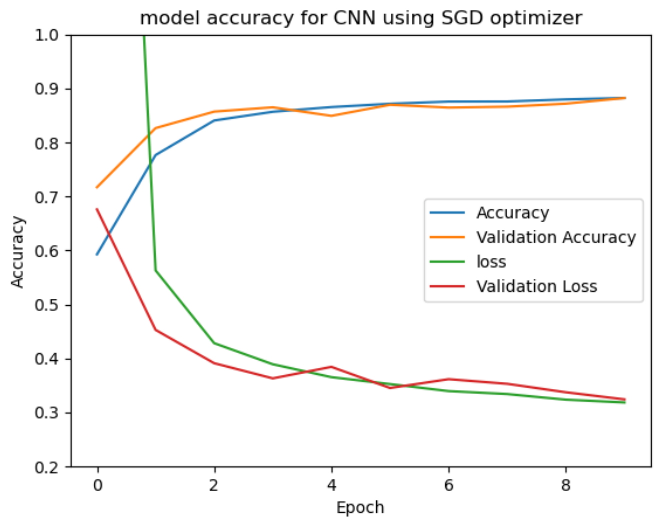
```
Shape of x_train data after outlier removal: (24553, 784)
Shape of y_train data after outlier removal: (24553,)
```

Plotting the default CNN Model: `Test accuracy: 86.15000247955322`

Plotting the training loss, validation loss, training accuracy and validation accuracy for default CNN. Epoch: 10, LR: 0.01, Loss: Categorical Cross entropy



conv2d_input | input: | [(None, 28, 28, 1)]
InputLayer | output: | [(None, 28, 28, 1)]
conv2d | input: | (None, 28, 28, 1)
Conv2D | output: | (None, 28, 28, 32)
max_pooling2d | input: | (None, 28, 28, 32)
MaxPooling2D | output: | (None, 14, 14, 32)
conv2d_1 | input: | (None, 14, 14, 32)
Conv2D | output: | (None, 12, 12, 32)
flatten | input: | (None, 12, 12, 32)
Flatten | output: | (None, 4608)
dense | input: | (None, 4608)
Dense | output: | (None, 5)
activation | input: | (None, 5)
Activation | output: | (None, 5)



The output shows the training and validation accuracy and loss for each epoch during the training of a neural network model. The model was trained for 10 epochs and achieved an accuracy of 86.15% on the test set.

It seems that the model is learning, as the training and validation accuracies increase with each epoch, while the loss decreases. However, it appears that the model may be overfitting to the training data, as the validation accuracy starts to plateau after the 6th epoch, while the training accuracy continues to increase. This suggests that the model may not generalize well to new data.

To address this issue, techniques such as regularization, dropout, or early stopping could be used to prevent overfitting and improve the model's performance on new data.

Runtime Performance

```
Epoch 1/10
2700/2700 [==============================] - 12s 3ms/step - loss: 4.8767 - accuracy: 0.7084 - val_loss: 0.5392 - val_accuracy:
0.7898
Epoch 2/10
2700/2700 [==============================] - 6s 2ms/step - loss: 0.4785 - accuracy: 0.8205 - val_loss: 0.4739 - val_accuracy:
0.8160
Epoch 3/10
2700/2700 [==============================] - 7s 2ms/step - loss: 0.4164 - accuracy: 0.8421 - val_loss: 0.3870 - val_accuracy:
0.8525
Epoch 4/10
2700/2700 [==============================] - 7s 3ms/step - loss: 0.3883 - accuracy: 0.8530 - val_loss: 0.3648 - val_accuracy:
0.8583
Epoch 5/10
2700/2700 [==============================] - 6s 2ms/step - loss: 0.3767 - accuracy: 0.8588 - val_loss: 0.3637 - val_accuracy:
0.8618
Epoch 6/10
2700/2700 [==============================] - 6s 2ms/step - loss: 0.3655 - accuracy: 0.8621 - val_loss: 0.3525 - val_accuracy:
0.8678
Epoch 7/10
2700/2700 [==============================] - 6s 2ms/step - loss: 0.3565 - accuracy: 0.8663 - val_loss: 0.3831 - val_accuracy:
0.8517
Epoch 8/10
2700/2700 [==============================] - 6s 2ms/step - loss: 0.3497 - accuracy: 0.8694 - val_loss: 0.3384 - val_accuracy:
0.8718
Epoch 9/10
2700/2700 [==============================] - 6s 2ms/step - loss: 0.3443 - accuracy: 0.8706 - val_loss: 0.3333 - val_accuracy:
0.8727
Epoch 10/10
2700/2700 [==============================] - 6s 2ms/step - loss: 0.3377 - accuracy: 0.8739 - val_loss: 0.3411 - val_accuracy:
0.8698
313/313 [==============================] - 1s 2ms/step - loss: 0.3641 - accuracy: 0.8615
Test accuracy: 86.15000247955322
```