MACHINE LEARNING
ORANGE LEVEL PROBLEM
Output Screenshots

PES2UG22CS093
Areeb Ahmed
Section B


```
[7]
...  Epoch 1/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.1649 - loss: 2.1252 - val_accuracy: 0.2951 - val_loss: 1.7667
     Epoch 2/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.2869 - loss: 1.8422 - val_accuracy: 0.3993 - val_loss: 1.6421
     Epoch 3/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.3508 - loss: 1.7286 - val_accuracy: 0.4167 - val_loss: 1.5772
     Epoch 4/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.4066 - loss: 1.5919 - val_accuracy: 0.4271 - val_loss: 1.5170
     Epoch 5/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4594 - loss: 1.4600 - val_accuracy: 0.4583 - val_loss: 1.4515
     Epoch 6/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.4821 - loss: 1.4309 - val_accuracy: 0.4653 - val_loss: 1.4314
     Epoch 7/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.4809 - loss: 1.3910 - val_accuracy: 0.4896 - val_loss: 1.3945
     Epoch 8/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.5307 - loss: 1.2788 - val_accuracy: 0.4931 - val_loss: 1.3614
     Epoch 9/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.5462 - loss: 1.1997 - val_accuracy: 0.4931 - val_loss: 1.3292
     Epoch 10/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.5723 - loss: 1.2134 - val_accuracy: 0.5035 - val_loss: 1.3099
     Epoch 11/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.5648 - loss: 1.2152 - val_accuracy: 0.5278 - val_loss: 1.2716
     Epoch 12/50
     36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.5804 - loss: 1.1394 - val_accuracy: 0.5278 - val_loss: 1.2636
     Epoch 13/50
     ...
         accuracy                           0.58       288
        macro avg       0.58      0.58      0.57       288
     weighted avg       0.59      0.58      0.58       288

     Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Layer (type)	Output Shape	Param #
dense_14 (Dense)	(None, 512)	85,504
batch_normalization_3 (BatchNormalization)	(None, 512)	2,048
dropout_10 (Dropout)	(None, 512)	0
dense_15 (Dense)	(None, 256)	131,328
batch_normalization_4 (BatchNormalization)	(None, 256)	1,024
dropout_11 (Dropout)	(None, 256)	0
dense_16 (Dense)	(None, 128)	32,896
batch_normalization_5 (BatchNormalization)	(None, 128)	512
dropout_12 (Dropout)	(None, 128)	0
dense_17 (Dense)	(None, 8)	1,032

Total params: 254,344 (993.53 KB)

Trainable params: 252,552 (986.53 KB)

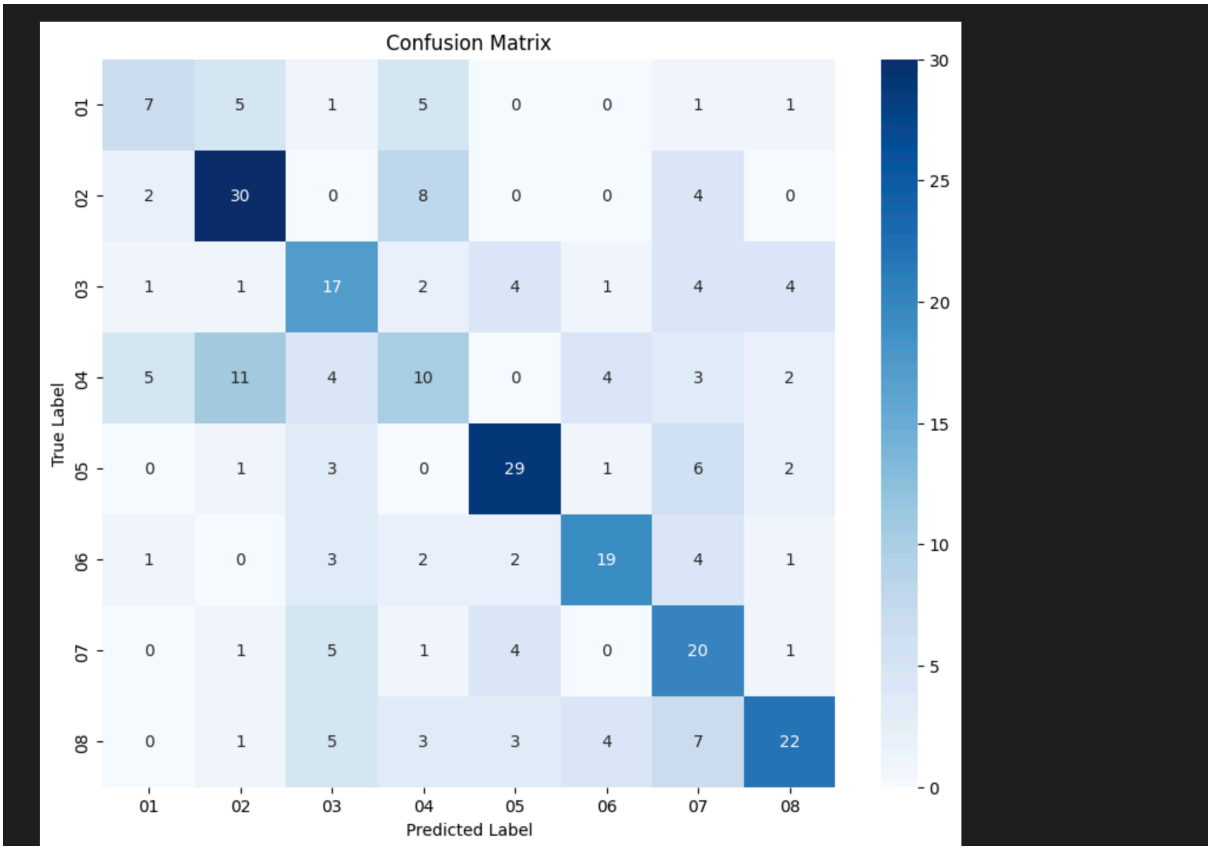
Non-trainable params: 1,792 (7.00 KB)

```
Epoch 1/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 2s 7ms/step - accuracy: 0.1746 - loss: 10.4321 - val_accuracy: 0.3333 - val_loss: 8.9752 - learning_rate
Epoch 2/100
```

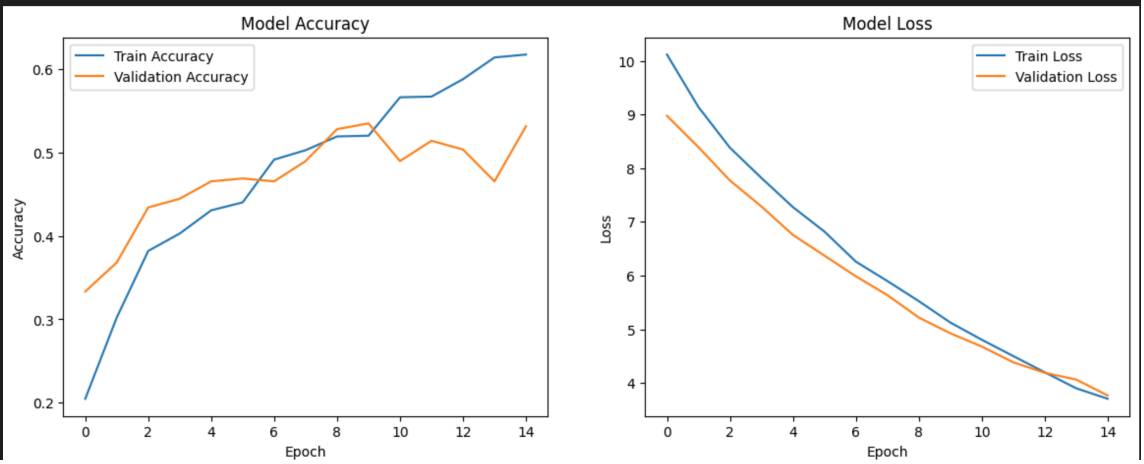
```
Epoch 1/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 2s 7ms/step - accuracy: 0.1746 - loss: 10.4321 - val_accuracy: 0.3333 - val_loss: 8.9752 - learning_rate: 0.0010
Epoch 2/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.2906 - loss: 9.3329 - val_accuracy: 0.3681 - val_loss: 8.3912 - learning_rate: 0.0010
Epoch 3/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.3682 - loss: 8.5195 - val_accuracy: 0.4340 - val_loss: 7.7708 - learning_rate: 0.0010
Epoch 4/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4204 - loss: 7.9189 - val_accuracy: 0.4444 - val_loss: 7.2871 - learning_rate: 0.0010
Epoch 5/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4381 - loss: 7.3367 - val_accuracy: 0.4653 - val_loss: 6.7576 - learning_rate: 0.0010
Epoch 6/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4286 - loss: 6.9568 - val_accuracy: 0.4688 - val_loss: 6.3720 - learning_rate: 0.0010
Epoch 7/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.4995 - loss: 6.3641 - val_accuracy: 0.4653 - val_loss: 5.9867 - learning_rate: 0.0010
Epoch 8/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5185 - loss: 5.9295 - val_accuracy: 0.4896 - val_loss: 5.6362 - learning_rate: 0.0010
Epoch 9/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5451 - loss: 5.5607 - val_accuracy: 0.5278 - val_loss: 5.2184 - learning_rate: 0.0010
Epoch 10/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5408 - loss: 5.1786 - val_accuracy: 0.5347 - val_loss: 4.9252 - learning_rate: 0.0010
Epoch 11/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5584 - loss: 4.8692 - val_accuracy: 0.4896 - val_loss: 4.6779 - learning_rate: 0.0010
Epoch 12/100
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5721 - loss: 4.5217 - val_accuracy: 0.5139 - val_loss: 4.3849 - learning_rate: 0.0010
Epoch 13/100
...
Training Accuracy: 0.71
Testing Accuracy: 0.53
9/9 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step
F1 Score: 0.5319
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Confusion Matrix



	precision	recall	f1-score	support
01	0.44	0.35	0.39	20
02	0.60	0.68	0.64	44
03	0.45	0.50	0.47	34
04	0.32	0.26	0.29	39
05	0.69	0.69	0.69	42
06	0.66	0.59	0.62	32
07	0.41	0.62	0.49	32
08	0.67	0.49	0.56	45
accuracy			0.53	288
macro avg	0.53	0.52	0.52	288
weighted avg	0.54	0.53	0.53	288



Best KNN Accuracy: 0.5868055555555556

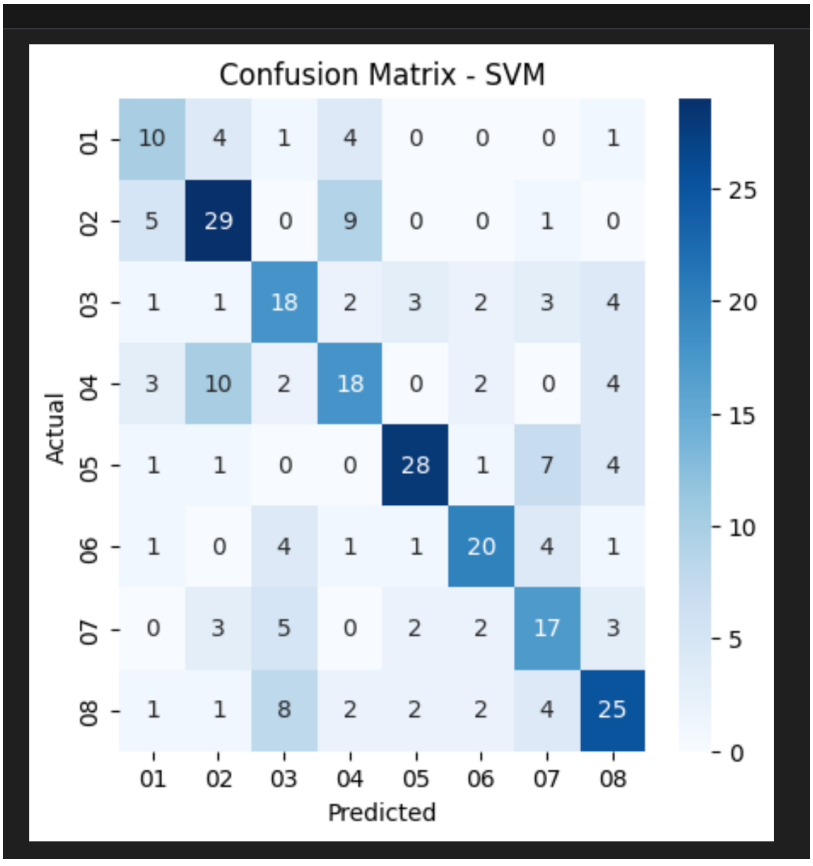
	precision	recall	f1-score	support
01	0.47	0.75	0.58	20
02	0.73	0.68	0.71	44
03	0.47	0.44	0.45	34
04	0.47	0.46	0.47	39
05	0.76	0.62	0.68	42
06	0.50	0.44	0.47	32
07	0.56	0.69	0.62	32
08	0.66	0.64	0.65	45
accuracy			0.59	288
macro avg	0.58	0.59	0.58	288
weighted avg	0.60	0.59	0.59	288

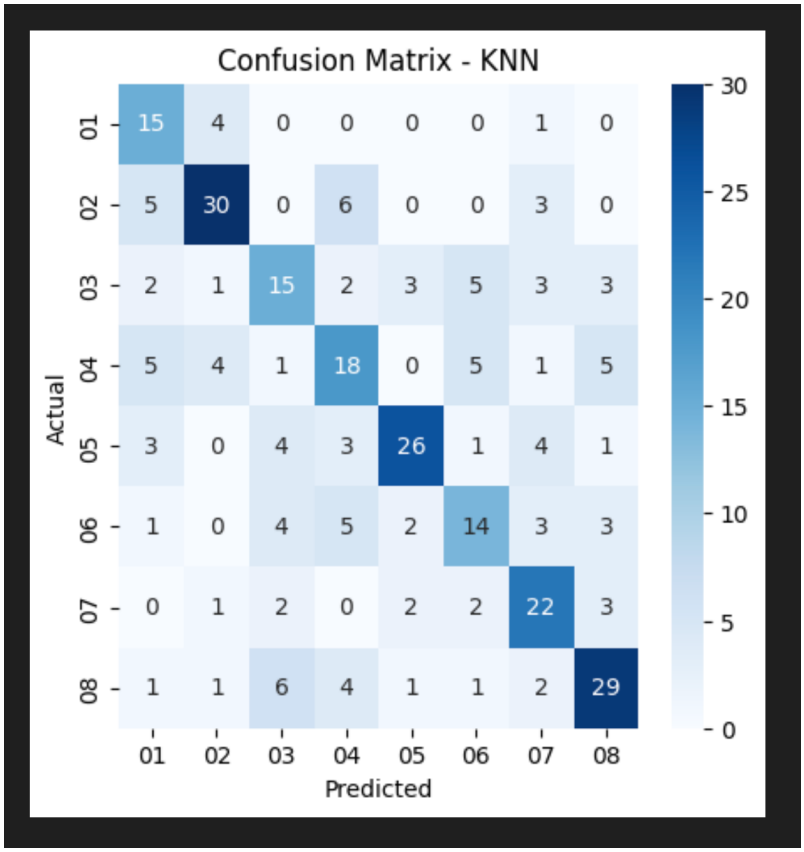
```
Best SVM Accuracy: 0.5729166666666666
              precision    recall  f1-score   support

          01       0.45      0.50      0.48        20
          02       0.59      0.66      0.62        44
          03       0.47      0.53      0.50        34
          04       0.50      0.46      0.48        39
          05       0.78      0.67      0.72        42
          06       0.69      0.62      0.66        32
          07       0.47      0.53      0.50        32
          08       0.60      0.56      0.57        45

    accuracy                           0.57       288
   macro avg       0.57      0.57      0.57       288
weighted avg       0.58      0.57      0.58       288
```

```
import matplotlib.pyplot as plt
```




Confusion Matrix - KNN
Actual
Predicted
01
02
03
04
05
06
07
08
15 4 0 0 0 0 1 0
5 30 0 6 0 0 3 0
2 1 15 2 3 5 3 3
5 4 1 18 0 5 1 5
3 0 4 3 26 1 4 1
1 0 4 5 2 14 3 3
0 1 2 0 2 2 22 3
1 1 6 4 1 1 2 29
30
25
20
15
10
5
0