Classification using Support Vector Machine (SVM)

This code performs image classification on the provided image data and labels using SVM

This model was trained on personal computer by Archit Jaiswal

In [12]:

```
import numpy as np
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import cv2

import matplotlib.pyplot as plt
%matplotlib inline
```

The data was already preprocessed by rotating the images and correcting the labels

In [22]:

```
# Loading Data
X = np.load('data_train_corrected.npy')
labels = np.load('labels_train_corrected.npy')

print(X.shape, labels.shape)
```

```
(90000, 9032) (9032,)
```

In [23]:

```
# Scaling the data

X_scaled = StandardScaler().fit_transform(X)
```

In [24]:

```
# Partitioning the data into training set and test set

X_train, X_test, labels_train, labels_test = train_test_split(X_scaled.T, labels,
```

In [25]:

```
X_train.shape
```

Out[25]:

```
(6322, 90000)
```

In [26]:

```
X_test.shape
```

Out[26]:

```
(2710, 90000)
```

In [27]:

```
labels_train.shape
```

Out[27]:

```
(6322,)
```

In [31]:

```
svc = SVC(kernel = 'linear')
svc.fit(X_train, labels_train)
```

Out[31]:

```
SVC(kernel='linear')
```

In [33]:

```
predict_labels = svc.predict(X_test)
```

In [34]:

```
from sklearn.metrics import accuracy_score
```

```
accuracy = accuracy_score(labels_test, predict_labels)
```

In [36]:

```
print('Prediction accuracy on test set: ', accuracy)
```

```
Prediction accuracy on test set:  0.46494464944649444
```

In [37]:

```
predict_training_labels = svc.predict(X_train)
```

In [38]:

```
# accuracy on training data

training_accuracy = accuracy_score(labels_train, predict_training_labels)
print('Prediction accuracy on training set: ', training_accuracy)
```

```
Prediction accuracy on training set:  0.999841822208162
```

In [39]:

```
# Confusion matrix for training set

from sklearn.metrics import confusion_matrix

confusion_matrix(labels_train, predict_training_labels)
```

Out[39]:

```
array([[614,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0],
       [  0, 620,   0,   0,   0,   0,   0,   0,   0,   0,   0],
       [  0,   0, 605,   0,   0,   0,   0,   0,   0,   0,   0],
       [  0,   0,   0, 623,   0,   0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0, 642,   0,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0, 614,   0,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0, 620,   0,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0,   0, 636,   0,   0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0, 624,   0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0, 636,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   1,  87]],
      dtype=int64)
```

In [40]:

```
# Confusion matrix for test set

confusion_matrix(labels_test, predict_labels)
```

Out[40]:

```
array([[130,  13,  12,  12,  14,  27,  10,  10,  16,  23,   4],
       [ 12, 178,   7,   7,  13,   9,  12,   9,   5,   9,  15],
       [ 14,   9, 196,  19,   6,   3,  14,   9,  10,   4,   4],
       [  4,  10,  32, 174,  18,   2,   4,   7,   7,   2,   9],
       [ 24,   6,  15,  41,  95,  10,  11,   8,  12,  13,   3],
       [ 50,  15,  19,  17,  34,  84,  19,  12,  14,  17,   2],
       [ 22,  24,  10,  27,  19,  22, 116,   8,   6,  12,   9],
       [ 12,  12,   4,  19,  18,  13,   5, 117,  34,   7,   8],
       [ 32,  12,  17,  22,  21,  17,  15,  47,  71,   8,   7],
       [ 20,  15,   8,   9,  39,  27,  11,  19,  13,  97,   4],
       [  6,   1,   3,   1,   5,   2,   5,   2,   1,   2,   2]],
      dtype=int64)
```

Due to a significant difference in accuracy of test set and training set, it can be concluded that the SVM classifier model is overfitting the training data. The model can be improved by using soft margin SVM. CNN model can provide far better classification accuracy compared to SVM.

In [ ]: