Computer Vision Coursework Submission (IN3060/INM460)

Student name, ID and cohort: Rajani Mohan Janipalli (210049506) - PG

Training and testing script of SIFT-SVM, SIFT-MLP, HOG-SVM and HOG-MLP models.

Google Colab Setup

In [ ]:
```
from google.colab import drive
drive.mount('/content/drive')

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

```
Mounted at /content/drive
```

Updating Open CV

In [ ]:
```
!pip install opencv-python==4.5.5.64

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

```
Requirement already satisfied: opencv-python==4.5.5.64 in /usr/local/lib/python3.7/dist-packages (4.5.5.64)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from opencv-python==4.5.5.64) (1.21.6)
```

Check the version open CV

In [ ]:
```
!pip show opencv-python

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

```
Name: opencv-python
Version: 4.5.5.64
Summary: Wrapper package for OpenCV python bindings.
Home-page: https://github.com/skvark/opencv-python
Author: None
Author-email: None
License: MIT
Location: /usr/local/lib/python3.7/dist-packages
Requires: numpy
Required-by: imgaug, dopamine-rl, albumentations
```

Assign path to link the folder containing the colab notebook.

In [ ]:
```
import os


GOOGLE_DRIVE_PATH_AFTER_MYDRIVE = 'Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/Code'
GOOGLE_DRIVE_PATH = os.path.join('drive', 'My Drive', GOOGLE_DRIVE_PATH_AFTER_MYDRIVE)
print(os.listdir(GOOGLE_DRIVE_PATH))


## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

```
['CV_CW_SVM_MLP_1.ipynb', 'CV_CW_CNN1.ipynb', 'CV_CW_draft1.ipynb', 'dummy_CNN3.ipynb', 'CV_CW_CNN3.ipynb', 'Copy of CV_CW_CNN3.ip
ynb']
```

Assign path to link the folder containing the train dataset.

In [ ]:
```
GOOGLE_DRIVE_PATH_AFTER_MYDRIVE_DS = 'Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/CW_Dataset'
GOOGLE_DRIVE_PATH_DS = os.path.join('drive', 'My Drive', GOOGLE_DRIVE_PATH_AFTER_MYDRIVE_DS)
print(os.listdir(GOOGLE_DRIVE_PATH_DS))


# Code from above cell was modified in the this cell, as the dataset resides in a different folder.
```

```
['CW_Dataset.zip']
```

Copy and unzip the dataset directly in colab server.

In [ ]:
```
# Identify path to zipped dataset
CW_zip_path = os.path.join(GOOGLE_DRIVE_PATH_DS, 'CW_Dataset.zip')

# Copy it to Colab
!cp '{CW_zip_path}' .

# Unzip it
!yes|unzip -q CW_Dataset.zip

# Delete zipped version from Colab (not from Drive)
!rm CW_Dataset.zip


## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

Import necessary libraries.

In [ ]:
```
import cv2
from sklearn.model_selection import train_test_split
from skimage import img_as_ubyte, io, color
from sklearn.cluster import MiniBatchKMeans
from sklearn import svm, metrics
import matplotlib.pyplot as plt
import numpy as np
from collections import Counter

%matplotlib inline

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

In [ ]:
```
import pickle
import time
```

Create a function to import images & labels from dataset.

In [ ]:
```
def import_imagelabels_lists(path):
    """Load images from selected directories"""
    images = []

    file_names = [file for file in sorted(os.listdir(os.path.join(path))) if file.endswith('.jpg')]
    for file in file_names:
        images.append(io.imread(os.path.join(path, file)))

    label_set = np.loadtxt(os.path.join('labels', 'list_label_{}.txt'.format(path)), dtype='str')
    label_nums = [] # create an empty list to append label numbers.
    for i in range(len(label_set)): # execute a for loop to extract the exact labels from data and append them to a list.
      label_nums.append(label_set[i][1])

    label_names = ['Suprise' if p == '1' else 'Fear' if p == '2' else 'Disgust' if p == '3' else 'Happiness' if p == '4' else 'Sad

    return images,label_names


## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module
## and modified as per the requirement in this task.
## https://tutorial.eyehunts.com/python/python-elif-in-list-comprehension-conditionals-example-code/
```

Import training set images & labels.

In [ ]:
```
X_1, y_1 = import_imagelabels_lists('train')
```

In [ ]:
```
type(X_1) # Check the data type.
```

Out[ ]:
```
list
```

In [ ]:
```
len(X_1) # Check the total number of images in train data.
```

Out[ ]:
```
12271
```

Above output showing the total number of images in train data matches with that menstioned in the coursework specification document.

In [ ]:
```
type(X_1[0])
```

Out[ ]:
```
numpy.ndarray
```

In [ ]:
```
X_1[0] # Check an element of train features to verify that it is an image array.
```

Out[ ]:
```
array([[[ 65,  74,  73],
        [ 97, 103, 103],
        [ 70,  76,  76],
        ...,
        [ 24,  22,  27],
        [ 73,  72,  77],
        [ 59,  60,  64]],

       [[117, 126, 125],
        [ 86,  92,  92],
        [ 66,  72,  72],
        ...,
        [ 51,  50,  55],
        [ 87,  88,  92],
        [ 44,  47,  52]],

       [[ 50,  59,  58],
        [ 78,  87,  86],
        [ 76,  82,  82],
        ...,
        [ 55,  60,  64],
        [ 67,  75,  78],
```

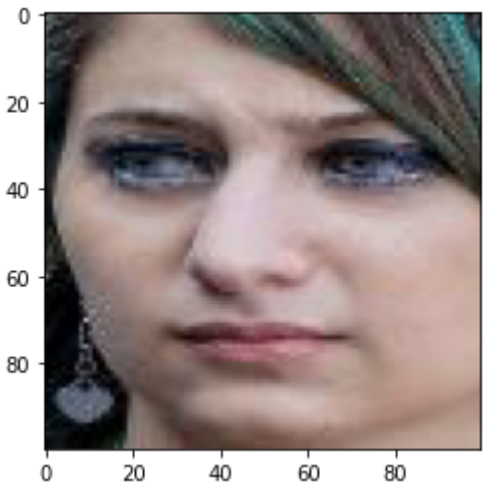
```
        [ 29,  36,  42]],

       ...,

       [[ 14,  19,  22],
        [ 11,  16,  19],
        [ 12,  16,  17],
        ...,
        [191, 169, 155],
        [193, 171, 157],
        [194, 172, 158]],

       [[  2,   7,  10],
        [  7,  12,  15],
        [ 15,  19,  20],
        ...,
        [191, 169, 155],
        [194, 172, 158],
        [194, 172, 158]],

       [[  4,   9,  12],
        [ 10,  15,  18],
        [ 19,  23,  24],
        ...,
        [192, 170, 156],
        [194, 172, 158],
        [194, 172, 158]]], dtype=uint8)
```

In [ ]:

```
plt.imshow(X_1[0]) # plot an element of train features data to see the image.
```

Out[ ]:

```
<matplotlib.image.AxesImage at 0x7f50abb97d10>
```



Label explanation: (taken from Readme file of dataset)

1: Surprise

2: Fear

3: Disgust

4: Happiness

5: Sadness

6: Anger

7: Neutral

In [ ]:

```
type(y_1)
```

Out[ ]:

```
list
```

In [ ]:

```
len(y_1)
```

Out[ ]:

```
12271
```

The above output shows that the total number of labels in train target matches with the total number of images in the train features.

In [ ]:

```
y_1[:10]
```

Out[ ]:

```
['Sadness',
 'Sadness',
 'Happiness',
 'Happiness',
 'Sadness',
 'Suprise',
 'Sadness',
 'Happiness',
 'Happiness',
 'Suprise']
```

Value counts of labels.

In [ ]:
```
print(Counter(y_1))
```

```
Counter({'Happiness': 4772, 'Neutral': 2524, 'Sadness': 1982, 'Suprise': 1290, 'Disgust': 717, 'Anger': 705, 'Fear': 281})
```

Clearly, the dataset is imbalanced.

Plot first 10 images with their labels.

In [ ]:
```
fig, axes = plt.subplots(2, 5, figsize=(14, 7), sharex=True, sharey=True)
ax = axes.ravel()

for i in range(10):
    ax[i].imshow(X_1[i])
    ax[i].set_title(f'Label: {y_1[i]}')
    ax[i].set_axis_off()
fig.tight_layout()
plt.show()

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

In [ ]:

In [ ]:
```
# X_train, X_val, y_train, y_val = train_test_split(X, y_12, test_size=0.1, shuffle=True, stratify=y_12)

# ## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

In [ ]:
```
X_train = X_1
y_train = y_1
```

In [ ]:
```
len(X_train)
```

Out[ ]:
```
12271
```

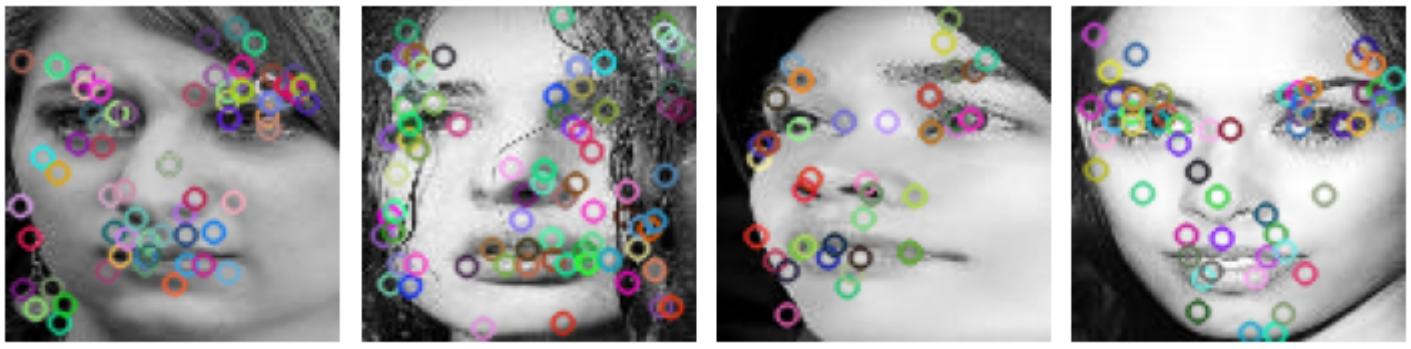
In [ ]:
```
len(y_train)
```

Out[ ]:
```
12271
```

Extract SIFT descriptors for training data.

In [ ]:
```
# Initiate SIFT detector
sift = cv2.SIFT_create()

# Create empty lists for feature descriptors and labels
des_list = []
y_train_list = []

fig, ax = plt.subplots(1, 4, figsize=(10, 8), sharey=True)

for i in range(len(X_train)):
    # Identify keypoints and extract descriptors with SIFT
    img = img_as_ubyte(color.rgb2gray(X_train[i]))
    kp, des = sift.detectAndCompute(img, None)

    # Show results for first 4 images
```

```
    if i<4:
        img_with_SIFT = cv2.drawKeypoints(img, kp, img)
        ax[i].imshow(img_with_SIFT)
        ax[i].set_axis_off()

    # Append list of descriptors and label to respective lists
    if des is not None:
        des_list.append(des)
        y_train_list.append(y_train[i])

fig.tight_layout()
plt.show()

# Convert to array for easier handling
des_array = np.vstack(des_list)


## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```



Perform KMeans clustering of SIFT descriptors to obtain codewords or Bag of Visual Words.

In [ ]:
```
# Number of centroids/codewords: good rule of thumb is 10*num_classes
k = len(np.unique(y_train)) * 10

# Use MiniBatchKMeans for faster computation and lower memory usage
batch_size = des_array.shape[0] // 4
kmeans = MiniBatchKMeans(n_clusters=k, batch_size=batch_size).fit(des_array)

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

Save the KMeans model to use it for extracting SIFT descriptors for test data.

In [ ]:
```
import pickle

siftKMeans = 'drive/My Drive/Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/Models/siftKMeans.pickle'

with open (siftKMeans, 'wb') as f:
    pickle.dump(kmeans, f)
```

In [ ]:
```
print(k)
```

```
70
```

Find the optimal number of clusters from elbow diagram.

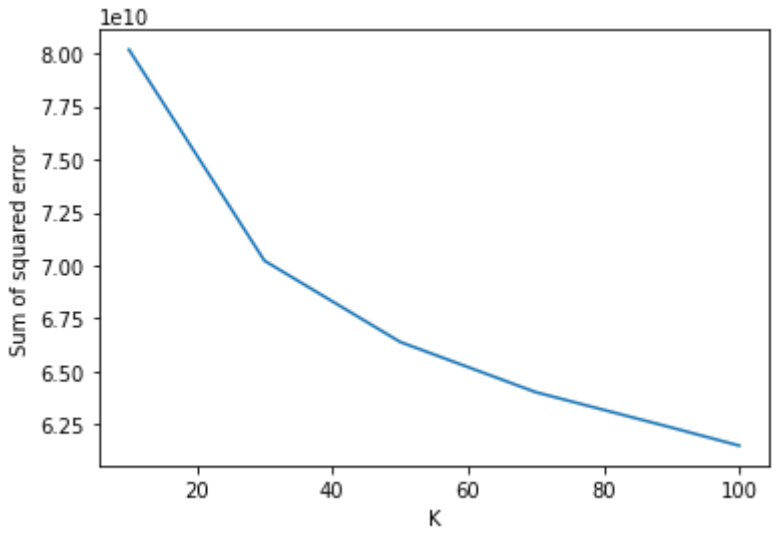
In [ ]:
```
k_rng = [10, 30, 50, 70, 90, 100]
sse = []
for knum in k_rng:
  kmelb = MiniBatchKMeans(n_clusters=knum, batch_size=batch_size).fit(des_array)
  sse.append(kmelb.inertia_)

  ## CODING REFERENCE:
  ## https://www.youtube.com/watch?v=EItlUEPCIzM
```

In [ ]:
```
len(des_array)
```

Out[ ]:
```
715991
```

In [ ]:
```
plt.plot(k_rng, sse)
plt.xlabel('K')
plt.ylabel('Sum of squared error')
```

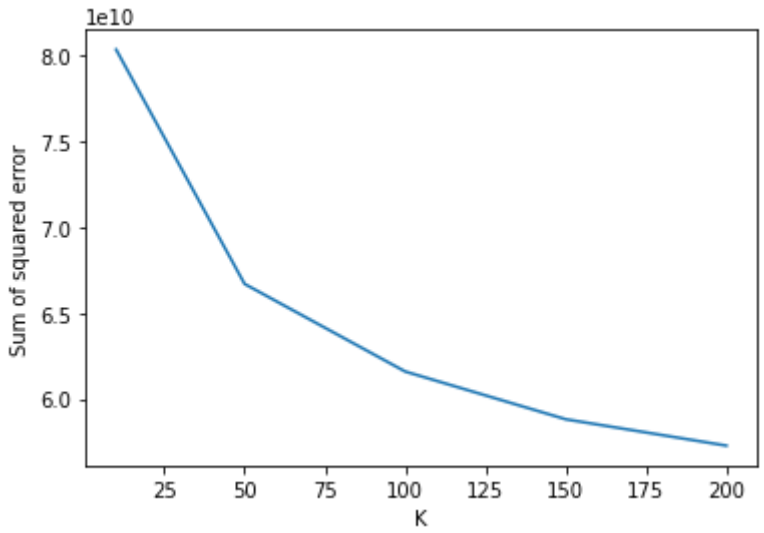
Out[ ]:
```
Text(0, 0.5, 'Sum of squared error')
```



In [ ]:
```
k_rng1 = [10, 50, 100, 150, 200]
sse1 = []
for knum1 in k_rng1:
  kmelb1 = MiniBatchKMeans(n_clusters=knum1, batch_size=batch_size).fit(des_array)
  sse1.append(kmelb1.inertia_)
```

In [ ]:
```
plt.plot(k_rng1, sse1)
plt.xlabel('K')
plt.ylabel('Sum of squared error')
```

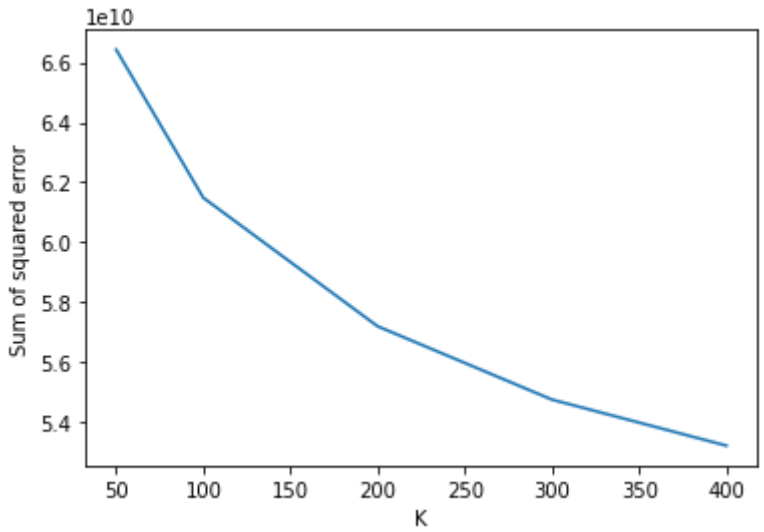
Out[ ]:
```
Text(0, 0.5, 'Sum of squared error')
```



In [ ]:
```
k_rng2 = [50, 100, 200, 300, 400]
sse2 = []
for knum2 in k_rng2:
  kmelb2 = MiniBatchKMeans(n_clusters=knum2, batch_size=batch_size).fit(des_array)
  sse2.append(kmelb2.inertia_)
```

In [ ]:
```
plt.plot(k_rng2, sse2)
plt.xlabel('K')
plt.ylabel('Sum of squared error')
```

Out[ ]:
```
Text(0, 0.5, 'Sum of squared error')
```



All the above search iterations show that 70 is nearly the optimal number of clusters.

In [ ]:
```
# Convert descriptors into histograms of codewords for each image
hist_list = []
idx_list = []

for des in des_list:
    hist = np.zeros(k)

    idx = kmeans.predict(des)
    idx_list.append(idx)
```

```
    for j in idx:
        hist[j] = hist[j] + (1 / len(des))
    hist_list.append(hist)

hist_array = np.vstack(hist_list)

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

In [ ]:

```
# fig, ax = plt.subplots(figsize=(8, 3))
# ax.hist(np.array(idx_list, dtype=object), bins=k)
# ax.set_title('Codewords occurrence in training set')
# plt.show()

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

NOTE: The above cell produces a histograms of codewords generated from feature descriptors. But it has been commented as it always crashes the RAM of colab.

Create baseline SVM model using SIFT descriptors.

In [ ]:

```
# Create a classifier: a support vector classifier
baseline_siftsvm = svm.SVC(kernel='rbf', class_weight='balanced')

# We learn the digits on the first half of the digits
# baseline_siftsvm.fit(hist_array, y_train_list)

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

In [ ]:

```
from sklearn.model_selection import cross_val_score
```

Perform K-Fold Crossvalidation of the model befor fitting the complete traning data to the model.

In [ ]:

```
baseline_siftsvm_cvscores = cross_val_score(baseline_siftsvm, hist_array, y_train_list, cv=5)

## CODING REFERENCE:

## https://scikit-learn.org/stable/modules/cross_validation.html
```

In [ ]:

```
baseline_siftsvm_cvscores
```

Out[ ]:

```
array([0.32001631, 0.32613127, 0.30819405, 0.30615573, 0.30954323])
```

In [ ]:

```
baseline_siftsvm_cvscores.mean()
```

Out[ ]:

```
0.31400811603995243
```

In [ ]:

```
baseline_siftsvm.fit(hist_array, y_train_list)
```

Out[ ]:

```
SVC(class_weight='balanced')
```

In [ ]:

```
baseline_siftsvm.score(hist_array, y_train_list)
```

Out[ ]:

```
0.5539791258969341
```

In [ ]:

```
baseline_siftsvm.get_params()
```

Out[ ]:

```
{'C': 1.0,
 'break_ties': False,
 'cache_size': 200,
 'class_weight': 'balanced',
 'coef0': 0.0,
 'decision_function_shape': 'ovr',
 'degree': 3,
 'gamma': 'scale',
 'kernel': 'rbf',
 'max_iter': -1,
 'probability': False,
 'random_state': None,
 'shrinking': True,
 'tol': 0.001,
 'verbose': False}
```

Perform exhaustive grid search for tuning hyperparameters.

In [ ]:

```
from sklearn.model_selection import GridSearchCV
```

In [ ]:

```
siftsvm_improve1 = svm.SVC(class_weight='balanced')
```

In [ ]:
```
siftsvm_gs1 = GridSearchCV(estimator=siftsvm_improve1,
             param_grid={'C': [0.5, 1, 10, 100], 'kernel': ('linear', 'poly','rbf')})


## CODING REFERENCE:
## https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchC
```

In [ ]:
```
siftsvm_gs1.fit(hist_array, y_train_list)
```

Out[ ]:
```
GridSearchCV(estimator=SVC(class_weight='balanced'),
             param_grid={'C': [0.5, 1, 10, 100],
                         'kernel': ('linear', 'poly', 'rbf')})
```

In [ ]:
```
siftsvm_gs1.best_params_
```

Out[ ]:
```
{'C': 100, 'kernel': 'rbf'}
```

In [ ]:
```
siftsvm_gs1.best_score_
```

Out[ ]:
```
0.3500480152478338
```

In [ ]:
```
siftsvm_1 = svm.SVC(C=10, kernel='rbf', class_weight='balanced')
```

In [ ]:
```
siftsvm_1_cvscores = cross_val_score(siftsvm_1, hist_array, y_train_list, cv=5)
```

In [ ]:
```
siftsvm_1_cvscores
```

Out[ ]:
```
array([0.3473298 , 0.36608235, 0.34896046, 0.33306156, 0.34502447])
```

In [ ]:
```
siftsvm_1_cvscores.mean()
```

Out[ ]:
```
0.3480917264141721
```

In [ ]:
```
siftsvm_1.fit(hist_array, y_train_list)
```

Out[ ]:
```
SVC(C=10, class_weight='balanced')
```

In [ ]:
```
siftsvm_1.score(hist_array, y_train_list)
```

Out[ ]:
```
0.9570287018917156
```

After hyperparameter tuning done above, create best SVM model that uses SIFT descriptors.

In [ ]:
```
best_siftSVM_model = svm.SVC(kernel='rbf', class_weight='balanced')
```

In [ ]:
```
best_siftSVM_model_cvscores = cross_val_score(best_siftSVM_model, hist_array, y_train_list, cv=5)
```

In [ ]:
```
best_siftSVM_model_cvscores
```

Out[ ]:
```
array([0.32001631, 0.32613127, 0.30819405, 0.30615573, 0.30954323])
```

In [ ]:
```
best_siftSVM_model_cvscores.mean()
```

Out[ ]:
```
0.31400811603995243
```

In [ ]:
```
import time
```

In [ ]:
```
t0 = time.time()
best_siftSVM_model.fit(hist_array, y_train_list)
tt = time.time()
print(tt-t0)
```

```
36.35411357879639
```

In [ ]:
```
best_siftSVM_model.score(hist_array, y_train_list)
```

Out[ ]:
```
0.565068493150685
```

Save best SVM model that uses SIFT descriptor.

In [ ]:
```
import pickle
```

```
bestsiftsvmpath = 'drive/My Drive/Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/Models/best_siftSVM_model.pickle'

with open (bestsiftsvmpath, 'wb') as f:
    pickle.dump(best_siftSVM_model, f)

## CODING REFERENCE
## https://www.youtube.com/watch?v=KfnhNlD8WZI
```

In [ ]:

In [ ]:

In [ ]:

```
from sklearn.neural_network import MLPClassifier
```

Create baseline MLP model that uses SIFT descriptors.

In [ ]:

```
baseline_siftMLP = MLPClassifier(hidden_layer_sizes=(50,), max_iter=100, alpha=1e-4,
                    solver='sgd', verbose=True, random_state=1,
                    learning_rate_init=.1)

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

In [ ]:

```
baseline_siftMLP_cvscores = cross_val_score(baseline_siftMLP, hist_array, y_train_list, cv=5)
```

```
Iteration 1, loss = 1.66581995
Iteration 2, loss = 1.64099238
Iteration 3, loss = 1.63722333
Iteration 4, loss = 1.63449359
Iteration 5, loss = 1.63236846
Iteration 6, loss = 1.62714451
Iteration 7, loss = 1.62249152
Iteration 8, loss = 1.61696324
Iteration 9, loss = 1.61130682
Iteration 10, loss = 1.60535298
Iteration 11, loss = 1.60112884
Iteration 12, loss = 1.59494581
Iteration 13, loss = 1.59156350
Iteration 14, loss = 1.58552805
Iteration 15, loss = 1.58256438
Iteration 16, loss = 1.57747700
Iteration 17, loss = 1.57499554
Iteration 18, loss = 1.57102678
Iteration 19, loss = 1.56783029
Iteration 20, loss = 1.56444155
Iteration 21, loss = 1.56323953
Iteration 22, loss = 1.55974905
Iteration 23, loss = 1.55704921
Iteration 24, loss = 1.55761086
Iteration 25, loss = 1.55375689
Iteration 26, loss = 1.55018691
Iteration 27, loss = 1.55427064
Iteration 28, loss = 1.55213047
Iteration 29, loss = 1.55153200
Iteration 30, loss = 1.54583155
Iteration 31, loss = 1.54532107
Iteration 32, loss = 1.55005704
Iteration 33, loss = 1.55029119
Iteration 34, loss = 1.54295271
Iteration 35, loss = 1.53976201
Iteration 36, loss = 1.53929462
Iteration 37, loss = 1.54097366
Iteration 38, loss = 1.53830231
Iteration 39, loss = 1.53556783
Iteration 40, loss = 1.53853604
Iteration 41, loss = 1.53945272
Iteration 42, loss = 1.53796322
Iteration 43, loss = 1.53578377
Iteration 44, loss = 1.53582084
Iteration 45, loss = 1.53575143
Iteration 46, loss = 1.53409052
Iteration 47, loss = 1.53176140
Iteration 48, loss = 1.53073224
Iteration 49, loss = 1.53077407
Iteration 50, loss = 1.52895957
Iteration 51, loss = 1.53586335
Iteration 52, loss = 1.52851806
Iteration 53, loss = 1.53217255
Iteration 54, loss = 1.52963171
Iteration 55, loss = 1.53469442
Iteration 56, loss = 1.52864275
Iteration 57, loss = 1.52764007
Iteration 58, loss = 1.52740497
Iteration 59, loss = 1.52971957
Iteration 60, loss = 1.53922572
Iteration 61, loss = 1.52821647
```

```
Iteration 62, loss = 1.52791100
Iteration 63, loss = 1.52628803
Iteration 64, loss = 1.52675132
Iteration 65, loss = 1.52729300
Iteration 66, loss = 1.52673237
Iteration 67, loss = 1.52478671
Iteration 68, loss = 1.53088841
Iteration 69, loss = 1.52994777
Iteration 70, loss = 1.52923827
Iteration 71, loss = 1.52575320
Iteration 72, loss = 1.52290549
Iteration 73, loss = 1.53161872
Iteration 74, loss = 1.52315391
Iteration 75, loss = 1.52059567
Iteration 76, loss = 1.52007734
Iteration 77, loss = 1.51939606
Iteration 78, loss = 1.51925087
Iteration 79, loss = 1.51974135
Iteration 80, loss = 1.51978901
Iteration 81, loss = 1.51921309
Iteration 82, loss = 1.52726706
Iteration 83, loss = 1.51914188
Iteration 84, loss = 1.51699570
Iteration 85, loss = 1.52004478
Iteration 86, loss = 1.51826734
Iteration 87, loss = 1.51657908
Iteration 88, loss = 1.51981359
Iteration 89, loss = 1.51790277
Iteration 90, loss = 1.51954142
Iteration 91, loss = 1.51528193
Iteration 92, loss = 1.51620881
Iteration 93, loss = 1.51980362
Iteration 94, loss = 1.51917794
Iteration 95, loss = 1.51835428
Iteration 96, loss = 1.52036102
Iteration 97, loss = 1.52127369
Iteration 98, loss = 1.51554527
Iteration 99, loss = 1.51563418
Iteration 100, loss = 1.51123073
Iteration 1, loss = 1.66821503
Iteration 2, loss = 1.63968031
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (100) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
Iteration 3, loss = 1.63830677
Iteration 4, loss = 1.63556625
Iteration 5, loss = 1.63180902
Iteration 6, loss = 1.62754182
Iteration 7, loss = 1.62146787
Iteration 8, loss = 1.61618007
Iteration 9, loss = 1.61293524
Iteration 10, loss = 1.60580385
Iteration 11, loss = 1.59887412
Iteration 12, loss = 1.59505684
Iteration 13, loss = 1.59598440
Iteration 14, loss = 1.58365138
Iteration 15, loss = 1.58561434
Iteration 16, loss = 1.57834344
Iteration 17, loss = 1.57614825
Iteration 18, loss = 1.56962201
Iteration 19, loss = 1.56649294
Iteration 20, loss = 1.56713913
Iteration 21, loss = 1.56026069
Iteration 22, loss = 1.56676212
Iteration 23, loss = 1.56015474
Iteration 24, loss = 1.55686504
Iteration 25, loss = 1.55548326
Iteration 26, loss = 1.54993599
Iteration 27, loss = 1.56120321
Iteration 28, loss = 1.54856846
Iteration 29, loss = 1.55126973
Iteration 30, loss = 1.54728787
Iteration 31, loss = 1.54839977
Iteration 32, loss = 1.54603045
Iteration 33, loss = 1.54176568
Iteration 34, loss = 1.54221485
Iteration 35, loss = 1.53690520
Iteration 36, loss = 1.53825340
Iteration 37, loss = 1.53855308
Iteration 38, loss = 1.53817349
Iteration 39, loss = 1.53604251
Iteration 40, loss = 1.53236332
Iteration 41, loss = 1.54028179
Iteration 42, loss = 1.53429601
Iteration 43, loss = 1.53244262
Iteration 44, loss = 1.53255131
Iteration 45, loss = 1.52952737
Iteration 46, loss = 1.52811395
Iteration 47, loss = 1.52978265
Iteration 48, loss = 1.52800135
Iteration 49, loss = 1.53016447
Iteration 50, loss = 1.52970607
```

```
Iteration 51, loss = 1.52957006
Iteration 52, loss = 1.53826594
Iteration 53, loss = 1.53376798
Iteration 54, loss = 1.53185045
Iteration 55, loss = 1.53062556
Iteration 56, loss = 1.52833026
Iteration 57, loss = 1.52693627
Iteration 58, loss = 1.52980594
Iteration 59, loss = 1.53130105
Iteration 60, loss = 1.52476173
Iteration 61, loss = 1.52552778
Iteration 62, loss = 1.52702591
Iteration 63, loss = 1.52320667
Iteration 64, loss = 1.54241660
Iteration 65, loss = 1.53002824
Iteration 66, loss = 1.53077319
Iteration 67, loss = 1.52903575
Iteration 68, loss = 1.52500373
Iteration 69, loss = 1.53066960
Iteration 70, loss = 1.53580998
Iteration 71, loss = 1.52511021
Iteration 72, loss = 1.52335176
Iteration 73, loss = 1.52191095
Iteration 74, loss = 1.52159290
Iteration 75, loss = 1.52116112
Iteration 76, loss = 1.51888088
Iteration 77, loss = 1.52310973
Iteration 78, loss = 1.52709417
Iteration 79, loss = 1.52163404
Iteration 80, loss = 1.52166581
Iteration 81, loss = 1.51936446
Iteration 82, loss = 1.52106129
Iteration 83, loss = 1.52095653
Iteration 84, loss = 1.52163604
Iteration 85, loss = 1.51692663
Iteration 86, loss = 1.52236882
Iteration 87, loss = 1.51782908
Iteration 88, loss = 1.52003960
Iteration 89, loss = 1.51822888
Iteration 90, loss = 1.51503258
Iteration 91, loss = 1.52319219
Iteration 92, loss = 1.52038415
Iteration 93, loss = 1.51758192
Iteration 94, loss = 1.51582153
Iteration 95, loss = 1.51646430
Iteration 96, loss = 1.51477008
Iteration 97, loss = 1.51987641
Iteration 98, loss = 1.51322298
Iteration 99, loss = 1.51650000
Iteration 100, loss = 1.51385499
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (100) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
Iteration 1, loss = 1.66832187
Iteration 2, loss = 1.64015818
Iteration 3, loss = 1.63746112
Iteration 4, loss = 1.63614499
Iteration 5, loss = 1.63213556
Iteration 6, loss = 1.62837316
Iteration 7, loss = 1.62390071
Iteration 8, loss = 1.61843629
Iteration 9, loss = 1.61393511
Iteration 10, loss = 1.60753370
Iteration 11, loss = 1.60175901
Iteration 12, loss = 1.59575451
Iteration 13, loss = 1.59554606
Iteration 14, loss = 1.58862333
Iteration 15, loss = 1.58265177
Iteration 16, loss = 1.57875942
Iteration 17, loss = 1.57549096
Iteration 18, loss = 1.57176393
Iteration 19, loss = 1.56972217
Iteration 20, loss = 1.56585082
Iteration 21, loss = 1.56087214
Iteration 22, loss = 1.55947311
Iteration 23, loss = 1.55813345
Iteration 24, loss = 1.55599472
Iteration 25, loss = 1.55792962
Iteration 26, loss = 1.54889476
Iteration 27, loss = 1.55148795
Iteration 28, loss = 1.54645036
Iteration 29, loss = 1.54890051
Iteration 30, loss = 1.54366533
Iteration 31, loss = 1.54366954
Iteration 32, loss = 1.54713400
Iteration 33, loss = 1.54284538
Iteration 34, loss = 1.54128137
Iteration 35, loss = 1.53888913
Iteration 36, loss = 1.53569450
Iteration 37, loss = 1.54098952
Iteration 38, loss = 1.53904456
Iteration 39, loss = 1.53444672
```

```
Iteration 40, loss = 1.53088135
Iteration 41, loss = 1.53522533
Iteration 42, loss = 1.53004891
Iteration 43, loss = 1.53055492
Iteration 44, loss = 1.52878146
Iteration 45, loss = 1.53083484
Iteration 46, loss = 1.53020240
Iteration 47, loss = 1.52814007
Iteration 48, loss = 1.52749574
Iteration 49, loss = 1.52585182
Iteration 50, loss = 1.52643774
Iteration 51, loss = 1.53217201
Iteration 52, loss = 1.52817850
Iteration 53, loss = 1.52546198
Iteration 54, loss = 1.52623729
Iteration 55, loss = 1.52170561
Iteration 56, loss = 1.52482385
Iteration 57, loss = 1.52447495
Iteration 58, loss = 1.52309030
Iteration 59, loss = 1.52298588
Iteration 60, loss = 1.52338302
Iteration 61, loss = 1.52025751
Iteration 62, loss = 1.52527080
Iteration 63, loss = 1.52780254
Iteration 64, loss = 1.52501006
Iteration 65, loss = 1.52180483
Iteration 66, loss = 1.52619197
Iteration 67, loss = 1.52209759
Iteration 68, loss = 1.51960308
Iteration 69, loss = 1.51700709
Iteration 70, loss = 1.53015257
Iteration 71, loss = 1.52033030
Iteration 72, loss = 1.51824245
Iteration 73, loss = 1.51973329
Iteration 74, loss = 1.52037035
Iteration 75, loss = 1.52000861
Iteration 76, loss = 1.52099205
Iteration 77, loss = 1.52313737
Iteration 78, loss = 1.51653233
Iteration 79, loss = 1.51837004
Iteration 80, loss = 1.51684159
Iteration 81, loss = 1.51825752
Iteration 82, loss = 1.52255306
Iteration 83, loss = 1.51837136
Iteration 84, loss = 1.51491736
Iteration 85, loss = 1.51334612
Iteration 86, loss = 1.51501903
Iteration 87, loss = 1.51405608
Iteration 88, loss = 1.51884652
Iteration 89, loss = 1.51495298
Iteration 90, loss = 1.51325957
Iteration 91, loss = 1.51600616
Iteration 92, loss = 1.51401736
Iteration 93, loss = 1.51526621
Iteration 94, loss = 1.51254740
Iteration 95, loss = 1.51406770
Iteration 96, loss = 1.51271817
Iteration 97, loss = 1.51704073
Iteration 98, loss = 1.50794000
Iteration 99, loss = 1.51175848
Iteration 100, loss = 1.51077894
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (100) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
Iteration 1, loss = 1.66802019
Iteration 2, loss = 1.63916447
Iteration 3, loss = 1.63694161
Iteration 4, loss = 1.63465836
Iteration 5, loss = 1.63129758
Iteration 6, loss = 1.62612137
Iteration 7, loss = 1.62124848
Iteration 8, loss = 1.61538112
Iteration 9, loss = 1.60951618
Iteration 10, loss = 1.60129912
Iteration 11, loss = 1.59790631
Iteration 12, loss = 1.59183835
Iteration 13, loss = 1.58535218
Iteration 14, loss = 1.58084834
Iteration 15, loss = 1.57436387
Iteration 16, loss = 1.57037805
Iteration 17, loss = 1.56757820
Iteration 18, loss = 1.56746127
Iteration 19, loss = 1.56263934
Iteration 20, loss = 1.55932091
Iteration 21, loss = 1.55483110
Iteration 22, loss = 1.55229517
Iteration 23, loss = 1.55551062
Iteration 24, loss = 1.55022736
Iteration 25, loss = 1.54721023
Iteration 26, loss = 1.54282228
Iteration 27, loss = 1.54170835
Iteration 28, loss = 1.54140351
```

```
Iteration 29, loss = 1.54204996
Iteration 30, loss = 1.53670491
Iteration 31, loss = 1.53724437
Iteration 32, loss = 1.53925917
Iteration 33, loss = 1.53521189
Iteration 34, loss = 1.53506515
Iteration 35, loss = 1.53383131
Iteration 36, loss = 1.53418848
Iteration 37, loss = 1.53219883
Iteration 38, loss = 1.52969399
Iteration 39, loss = 1.52775442
Iteration 40, loss = 1.52778504
Iteration 41, loss = 1.52920192
Iteration 42, loss = 1.52632755
Iteration 43, loss = 1.52720330
Iteration 44, loss = 1.52427923
Iteration 45, loss = 1.52597653
Iteration 46, loss = 1.52375216
Iteration 47, loss = 1.52306481
Iteration 48, loss = 1.52872687
Iteration 49, loss = 1.53854553
Iteration 50, loss = 1.52812669
Iteration 51, loss = 1.54200506
Iteration 52, loss = 1.53946547
Iteration 53, loss = 1.53328095
Iteration 54, loss = 1.53080216
Iteration 55, loss = 1.52952416
Iteration 56, loss = 1.52711441
Iteration 57, loss = 1.52727859
Iteration 58, loss = 1.52402856
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 1.66725404
Iteration 2, loss = 1.63949153
Iteration 3, loss = 1.63841261
Iteration 4, loss = 1.63465047
Iteration 5, loss = 1.63113681
Iteration 6, loss = 1.62914772
Iteration 7, loss = 1.62134545
Iteration 8, loss = 1.61513460
Iteration 9, loss = 1.60968399
Iteration 10, loss = 1.60199933
Iteration 11, loss = 1.59680989
Iteration 12, loss = 1.59061968
Iteration 13, loss = 1.58561967
Iteration 14, loss = 1.58183053
Iteration 15, loss = 1.57641543
Iteration 16, loss = 1.57416777
Iteration 17, loss = 1.57200392
Iteration 18, loss = 1.56851399
Iteration 19, loss = 1.56648235
Iteration 20, loss = 1.56369744
Iteration 21, loss = 1.55835380
Iteration 22, loss = 1.55913290
Iteration 23, loss = 1.55602459
Iteration 24, loss = 1.55514821
Iteration 25, loss = 1.55135962
Iteration 26, loss = 1.54852353
Iteration 27, loss = 1.54821656
Iteration 28, loss = 1.54487741
Iteration 29, loss = 1.54467570
Iteration 30, loss = 1.54232150
Iteration 31, loss = 1.54046734
Iteration 32, loss = 1.53650861
Iteration 33, loss = 1.54037883
Iteration 34, loss = 1.53453082
Iteration 35, loss = 1.53962022
Iteration 36, loss = 1.53450351
Iteration 37, loss = 1.53166005
Iteration 38, loss = 1.52949786
Iteration 39, loss = 1.52962744
Iteration 40, loss = 1.52929077
Iteration 41, loss = 1.52794724
Iteration 42, loss = 1.53049012
Iteration 43, loss = 1.52796455
Iteration 44, loss = 1.52679317
Iteration 45, loss = 1.52498736
Iteration 46, loss = 1.52868714
Iteration 47, loss = 1.52881519
Iteration 48, loss = 1.52245072
Iteration 49, loss = 1.52501335
Iteration 50, loss = 1.52412953
Iteration 51, loss = 1.52555129
Iteration 52, loss = 1.52102286
Iteration 53, loss = 1.52446367
Iteration 54, loss = 1.52096862
Iteration 55, loss = 1.52534793
Iteration 56, loss = 1.52129418
Iteration 57, loss = 1.52088167
Iteration 58, loss = 1.51997080
Iteration 59, loss = 1.51661418
Iteration 60, loss = 1.51600601
Iteration 61, loss = 1.51647231
```

```
Iteration 62, loss = 1.52032660
Iteration 63, loss = 1.51904475
Iteration 64, loss = 1.51674676
Iteration 65, loss = 1.51780719
Iteration 66, loss = 1.51532715
Iteration 67, loss = 1.51649802
Iteration 68, loss = 1.51524241
Iteration 69, loss = 1.51241143
Iteration 70, loss = 1.51645697
Iteration 71, loss = 1.51591380
Iteration 72, loss = 1.51408486
Iteration 73, loss = 1.51250036
Iteration 74, loss = 1.51846931
Iteration 75, loss = 1.50903620
Iteration 76, loss = 1.51361983
Iteration 77, loss = 1.51574065
Iteration 78, loss = 1.51460473
Iteration 79, loss = 1.51228106
Iteration 80, loss = 1.51200988
Iteration 81, loss = 1.50879842
Iteration 82, loss = 1.51174619
Iteration 83, loss = 1.50672348
Iteration 84, loss = 1.50683406
Iteration 85, loss = 1.51301352
Iteration 86, loss = 1.50917337
Iteration 87, loss = 1.51389699
Iteration 88, loss = 1.50810495
Iteration 89, loss = 1.50502592
Iteration 90, loss = 1.50910860
Iteration 91, loss = 1.51388060
Iteration 92, loss = 1.50915414
Iteration 93, loss = 1.50663632
Iteration 94, loss = 1.50576608
Iteration 95, loss = 1.50764510
Iteration 96, loss = 1.50662050
Iteration 97, loss = 1.50603493
Iteration 98, loss = 1.50935025
Iteration 99, loss = 1.50464956
Iteration 100, loss = 1.50502172
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (100) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

In [ ]:

```
baseline_siftMLP_cvscores
```

Out[ ]:

```
array([0.39298818, 0.42600897, 0.41255605, 0.38768854, 0.40171289])
```

In [ ]:

```
baseline_siftMLP_cvscores.mean()
```

Out[ ]:

```
0.4041909264482217
```

In [ ]:

```
baseline_siftMLP.fit(hist_array, y_train_list)
```

```
Iteration 1, loss = 1.65855951
Iteration 2, loss = 1.63918617
Iteration 3, loss = 1.63576490
Iteration 4, loss = 1.63199051
Iteration 5, loss = 1.62650293
Iteration 6, loss = 1.62039781
Iteration 7, loss = 1.61398949
Iteration 8, loss = 1.60633908
Iteration 9, loss = 1.59794023
Iteration 10, loss = 1.59315630
Iteration 11, loss = 1.58686482
Iteration 12, loss = 1.58130255
Iteration 13, loss = 1.57576712
Iteration 14, loss = 1.57294560
Iteration 15, loss = 1.56863409
Iteration 16, loss = 1.56690035
Iteration 17, loss = 1.56048481
Iteration 18, loss = 1.55963466
Iteration 19, loss = 1.55639623
Iteration 20, loss = 1.55266713
Iteration 21, loss = 1.55154372
Iteration 22, loss = 1.54769332
Iteration 23, loss = 1.54530911
Iteration 24, loss = 1.54586111
Iteration 25, loss = 1.54302070
Iteration 26, loss = 1.54109644
Iteration 27, loss = 1.53845330
Iteration 28, loss = 1.53859766
Iteration 29, loss = 1.53632876
Iteration 30, loss = 1.53615368
Iteration 31, loss = 1.53557420
Iteration 32, loss = 1.53329967
Iteration 33, loss = 1.53553144
Iteration 34, loss = 1.53245726
Iteration 35, loss = 1.53199170
Iteration 36, loss = 1.53244659
```

```
Iteration 37, loss = 1.52981344
Iteration 38, loss = 1.52753501
Iteration 39, loss = 1.52639756
Iteration 40, loss = 1.52618800
Iteration 41, loss = 1.52622858
Iteration 42, loss = 1.52482421
Iteration 43, loss = 1.52486911
Iteration 44, loss = 1.52393731
Iteration 45, loss = 1.52530493
Iteration 46, loss = 1.52195013
Iteration 47, loss = 1.52306309
Iteration 48, loss = 1.52112516
Iteration 49, loss = 1.52143083
Iteration 50, loss = 1.52101199
Iteration 51, loss = 1.52100826
Iteration 52, loss = 1.51999992
Iteration 53, loss = 1.51929952
Iteration 54, loss = 1.51939555
Iteration 55, loss = 1.51825222
Iteration 56, loss = 1.51698819
Iteration 57, loss = 1.51875224
Iteration 58, loss = 1.51779162
Iteration 59, loss = 1.51691849
Iteration 60, loss = 1.51758769
Iteration 61, loss = 1.51631112
Iteration 62, loss = 1.51537303
Iteration 63, loss = 1.51419377
Iteration 64, loss = 1.51496487
Iteration 65, loss = 1.51424089
Iteration 66, loss = 1.51373188
Iteration 67, loss = 1.51445834
Iteration 68, loss = 1.51576078
Iteration 69, loss = 1.51347244
Iteration 70, loss = 1.51111079
Iteration 71, loss = 1.51346975
Iteration 72, loss = 1.51477103
Iteration 73, loss = 1.51301872
Iteration 74, loss = 1.51168525
Iteration 75, loss = 1.51184228
Iteration 76, loss = 1.50875221
Iteration 77, loss = 1.50978946
Iteration 78, loss = 1.51102509
Iteration 79, loss = 1.51004705
Iteration 80, loss = 1.50862968
Iteration 81, loss = 1.50726754
Iteration 82, loss = 1.51080362
Iteration 83, loss = 1.50840418
Iteration 84, loss = 1.51024549
Iteration 85, loss = 1.50593353
Iteration 86, loss = 1.50620661
Iteration 87, loss = 1.50859221
Iteration 88, loss = 1.50671545
Iteration 89, loss = 1.51037171
Iteration 90, loss = 1.50648259
Iteration 91, loss = 1.50481956
Iteration 92, loss = 1.50471999
Iteration 93, loss = 1.50453248
Iteration 94, loss = 1.50688289
Iteration 95, loss = 1.50303694
Iteration 96, loss = 1.50592996
Iteration 97, loss = 1.50464945
Iteration 98, loss = 1.50501200
Iteration 99, loss = 1.50362791
Iteration 100, loss = 1.50485222
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (100) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:

```
MLPClassifier(hidden_layer_sizes=(50,), learning_rate_init=0.1, max_iter=100,
              random_state=1, solver='sgd', verbose=True)
```

In [ ]:

```
baseline_siftMLP.score(hist_array, y_train_list)
```

Out[ ]:

```
0.4260437051532942
```

In [ ]:

```
baseline_siftMLP.get_params()
```

Out[ ]:

```
{'activation': 'relu',
 'alpha': 0.0001,
 'batch_size': 'auto',
 'beta_1': 0.9,
 'beta_2': 0.999,
 'early_stopping': False,
 'epsilon': 1e-08,
 'hidden_layer_sizes': (50,),
 'learning_rate': 'constant',
 'learning_rate_init': 0.1,
 'max_fun': 15000,
 'max_iter': 100,
 'momentum': 0.9,
 'n_iter_no_change': 10,
```

```
 'nesterovs_momentum': True,
 'power_t': 0.5,
 'random_state': 1,
 'shuffle': True,
 'solver': 'sgd',
 'tol': 0.0001,
 'validation_fraction': 0.1,
 'verbose': True,
 'warm_start': False}
```

Perform hyperparameter tuning.

In [ ]:
```
siftMLP_improve1 = MLPClassifier(max_iter=200, alpha=1e-4,
                    solver='sgd', random_state=1,
                    learning_rate_init=.1)
```

In [ ]:
```
siftMLP_gs1 = GridSearchCV(estimator=siftMLP_improve1,
             param_grid={'hidden_layer_sizes': [(50,), (100,), (50,50), (50,100)]})
```

In [ ]:
```
siftMLP_gs1.fit(hist_array, y_train_list)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:
```
GridSearchCV(estimator=MLPClassifier(learning_rate_init=0.1, random_state=1,
                                     solver='sgd'),
             param_grid={'hidden_layer_sizes': [(50,), (100,), (50, 50),
                                                (50, 100)]})
```

In [ ]:
```
siftMLP_gs1.best_params_
```

Out[ ]:
```
{'hidden_layer_sizes': (100,)}
```

In [ ]:
```
siftMLP_gs1.best_score_
```

Out[ ]:
```
0.40402520068977027
```

In [ ]:
```
siftMLP_improve2 = MLPClassifier(hidden_layer_sizes=(100,), alpha=1e-4,
                    solver='sgd', random_state=1,
                    learning_rate_init=.1)
```

In [ ]:
```
siftMLP_gs2 = GridSearchCV(estimator=siftMLP_improve2,
             param_grid={'max_iter': [200, 300, 400, 500]})
```

In [ ]:
```
siftMLP_gs2.fit(hist_array, y_train_list)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:
```
GridSearchCV(estimator=MLPClassifier(learning_rate_init=0.1, random_state=1,
                                     solver='sgd'),
             param_grid={'max_iter': [200, 300, 400, 500]})
```

In [ ]:
```
siftMLP_gs2.best_params_
```

Out[ ]:
```
{'max_iter': 200}
```

In [ ]:
```
siftMLP_gs2.best_score_
```

Out[ ]:
```
0.40402520068977027
```

In [ ]:
```
siftMLP_improve3 = MLPClassifier(hidden_layer_sizes=(100,), max_iter=200,alpha=1e-4,
                                 random_state=1, learning_rate_init=.1)
```

In [ ]:
```
siftMLP_gs3 = GridSearchCV(estimator=siftMLP_improve3,
             param_grid={'activation': ['identity', 'tanh', 'relu'], 'solver': ['sgd', 'adam']})
```

In [ ]:
```
siftMLP_gs3.fit(hist_array, y_train_list)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:
```
GridSearchCV(estimator=MLPClassifier(learning_rate_init=0.1, random_state=1),
             param_grid={'activation': ['identity', 'tanh', 'relu'],
                         'solver': ['sgd', 'adam']})
```

In [ ]:
```
siftMLP_gs3.best_params_
```

Out[ ]:
```
{'activation': 'relu', 'solver': 'sgd'}
```

In [ ]:
```
siftMLP_gs3.best_score_
```

Out[ ]:
```
0.40402520068977027
```

In [ ]:
```
siftMLP_improve4 = MLPClassifier(hidden_layer_sizes=(100,), activation='relu', max_iter=200,alpha=1e-4,
                    solver='sgd', random_state=1)
```

In [ ]:
```
siftMLP_gs4 = GridSearchCV(estimator=siftMLP_improve4,
             param_grid={'learning_rate_init': [0.001, 0.01, 0.1]})
```

In [ ]:
```
siftMLP_gs4.fit(hist_array, y_train_list)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:
```
GridSearchCV(estimator=MLPClassifier(random_state=1, solver='sgd'),
             param_grid={'learning_rate_init': [0.001, 0.01, 0.1]})
```

In [ ]:
```
siftMLP_gs4.best_params_
```

Out[ ]:
```
{'learning_rate_init': 0.01}
```

In [ ]:
```
siftMLP_gs4.best_score_
```

Out[ ]:
```
0.4127522712475784
```

In [ ]:
```
siftMLP_improve5 = MLPClassifier(hidden_layer_sizes=(100,), activation='relu', max_iter=200,alpha=1e-4,
                    solver='sgd', learning_rate_init=0.01, random_state=1)
```

In [ ]:
```
siftMLP_improve5_cvscores = cross_val_score(siftMLP_improve5, hist_array, y_train_list, cv=5)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

In [ ]:
```
siftMLP_improve5_cvscores.mean()
```

Out[ ]:
```
0.4127522712475784
```

In [ ]:
```
siftMLP_improve5.fit(hist_array, y_train_list)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:
```
MLPClassifier(learning_rate_init=0.01, random_state=1, solver='sgd')
```

In [ ]:
```
siftMLP_improve5.score(hist_array, y_train_list)
```

Out[ ]:
```
0.4212328767123288
```

Create best MLP model that uses SIFT descriptors.

In [ ]:
```
best_siftMLP_model = MLPClassifier(hidden_layer_sizes=(100,), activation='relu', max_iter=200,alpha=1e-4,
                    solver='sgd', verbose=True, learning_rate_init=0.01, random_state=1)
```

In [ ]:
```
best_siftMLP_model_cvscores = cross_val_score(best_siftMLP_model, hist_array, y_train_list, cv=5)
```

```
Iteration 1, loss = 1.73982916
Iteration 2, loss = 1.64522139
Iteration 3, loss = 1.64047280
Iteration 4, loss = 1.63932516
Iteration 5, loss = 1.63898974
Iteration 6, loss = 1.63846608
Iteration 7, loss = 1.63818854
Iteration 8, loss = 1.63753789
Iteration 9, loss = 1.63742672
Iteration 10, loss = 1.63709098
Iteration 11, loss = 1.63678084
Iteration 12, loss = 1.63660070
Iteration 13, loss = 1.63623876
Iteration 14, loss = 1.63571689
Iteration 15, loss = 1.63593976
Iteration 16, loss = 1.63512411
Iteration 17, loss = 1.63467543
Iteration 18, loss = 1.63452690
Iteration 19, loss = 1.63412965
Iteration 20, loss = 1.63381928
Iteration 21, loss = 1.63342630
Iteration 22, loss = 1.63328478
Iteration 23, loss = 1.63278067
Iteration 24, loss = 1.63219960
Iteration 25, loss = 1.63188497
Iteration 26, loss = 1.63142418
Iteration 27, loss = 1.63079598
Iteration 28, loss = 1.63083192
Iteration 29, loss = 1.63013237
Iteration 30, loss = 1.62979923
Iteration 31, loss = 1.62918190
Iteration 32, loss = 1.62918090
Iteration 33, loss = 1.62858841
Iteration 34, loss = 1.62778196
Iteration 35, loss = 1.62747871
Iteration 36, loss = 1.62692535
Iteration 37, loss = 1.62671099
Iteration 38, loss = 1.62634741
Iteration 39, loss = 1.62540515
Iteration 40, loss = 1.62503227
Iteration 41, loss = 1.62418819
Iteration 42, loss = 1.62358755
Iteration 43, loss = 1.62301432
Iteration 44, loss = 1.62256063
Iteration 45, loss = 1.62202398
Iteration 46, loss = 1.62158127
Iteration 47, loss = 1.62055990
Iteration 48, loss = 1.61998407
Iteration 49, loss = 1.61948530
Iteration 50, loss = 1.61895182
Iteration 51, loss = 1.61826366
Iteration 52, loss = 1.61741365
Iteration 53, loss = 1.61682477
Iteration 54, loss = 1.61663447
Iteration 55, loss = 1.61508554
Iteration 56, loss = 1.61480606
Iteration 57, loss = 1.61388766
Iteration 58, loss = 1.61396274
Iteration 59, loss = 1.61339238
Iteration 60, loss = 1.61243501
Iteration 61, loss = 1.61136453
Iteration 62, loss = 1.61052147
Iteration 63, loss = 1.60983982
Iteration 64, loss = 1.60912867
Iteration 65, loss = 1.60889703
Iteration 66, loss = 1.60776094
Iteration 67, loss = 1.60792971
Iteration 68, loss = 1.60638514
Iteration 69, loss = 1.60542499
Iteration 70, loss = 1.60470432
Iteration 71, loss = 1.60437702
Iteration 72, loss = 1.60344203
Iteration 73, loss = 1.60330919
Iteration 74, loss = 1.60174666
Iteration 75, loss = 1.60241322
Iteration 76, loss = 1.60018695
Iteration 77, loss = 1.59963203
Iteration 78, loss = 1.59867832
Iteration 79, loss = 1.59827838
Iteration 80, loss = 1.59792223
```

```
Iteration 81, loss = 1.59686231
Iteration 82, loss = 1.59662044
Iteration 83, loss = 1.59575708
Iteration 84, loss = 1.59481524
Iteration 85, loss = 1.59411239
Iteration 86, loss = 1.59384305
Iteration 87, loss = 1.59278327
Iteration 88, loss = 1.59233466
Iteration 89, loss = 1.59139189
Iteration 90, loss = 1.59090119
Iteration 91, loss = 1.59092023
Iteration 92, loss = 1.58970574
Iteration 93, loss = 1.58889610
Iteration 94, loss = 1.58869343
Iteration 95, loss = 1.58779623
Iteration 96, loss = 1.58757781
Iteration 97, loss = 1.58681958
Iteration 98, loss = 1.58612046
Iteration 99, loss = 1.58589863
Iteration 100, loss = 1.58536193
Iteration 101, loss = 1.58475153
Iteration 102, loss = 1.58393966
Iteration 103, loss = 1.58408132
Iteration 104, loss = 1.58265258
Iteration 105, loss = 1.58223626
Iteration 106, loss = 1.58207838
Iteration 107, loss = 1.58119795
Iteration 108, loss = 1.58040788
Iteration 109, loss = 1.58052863
Iteration 110, loss = 1.58011448
Iteration 111, loss = 1.57990466
Iteration 112, loss = 1.57880559
Iteration 113, loss = 1.57844008
Iteration 114, loss = 1.57742845
Iteration 115, loss = 1.57803970
Iteration 116, loss = 1.57742487
Iteration 117, loss = 1.57642072
Iteration 118, loss = 1.57621041
Iteration 119, loss = 1.57568491
Iteration 120, loss = 1.57466715
Iteration 121, loss = 1.57470256
Iteration 122, loss = 1.57351278
Iteration 123, loss = 1.57336807
Iteration 124, loss = 1.57330684
Iteration 125, loss = 1.57313688
Iteration 126, loss = 1.57219929
Iteration 127, loss = 1.57246518
Iteration 128, loss = 1.57174384
Iteration 129, loss = 1.57098805
Iteration 130, loss = 1.57100605
Iteration 131, loss = 1.57034418
Iteration 132, loss = 1.57325320
Iteration 133, loss = 1.56939845
Iteration 134, loss = 1.57004177
Iteration 135, loss = 1.56853774
Iteration 136, loss = 1.56830689
Iteration 137, loss = 1.56743325
Iteration 138, loss = 1.56718059
Iteration 139, loss = 1.56815128
Iteration 140, loss = 1.56629578
Iteration 141, loss = 1.56600640
Iteration 142, loss = 1.56599033
Iteration 143, loss = 1.56526316
Iteration 144, loss = 1.56572856
Iteration 145, loss = 1.56448992
Iteration 146, loss = 1.56457419
Iteration 147, loss = 1.56421781
Iteration 148, loss = 1.56414832
Iteration 149, loss = 1.56284332
Iteration 150, loss = 1.56251651
Iteration 151, loss = 1.56295147
Iteration 152, loss = 1.56232100
Iteration 153, loss = 1.56171107
Iteration 154, loss = 1.56248222
Iteration 155, loss = 1.56107104
Iteration 156, loss = 1.56125856
Iteration 157, loss = 1.56032721
Iteration 158, loss = 1.56019675
Iteration 159, loss = 1.55992723
Iteration 160, loss = 1.56069302
Iteration 161, loss = 1.55924213
Iteration 162, loss = 1.55870566
Iteration 163, loss = 1.55861816
Iteration 164, loss = 1.55840817
Iteration 165, loss = 1.55783615
Iteration 166, loss = 1.55750147
Iteration 167, loss = 1.55706906
Iteration 168, loss = 1.55989454
Iteration 169, loss = 1.55650047
Iteration 170, loss = 1.55604816
Iteration 171, loss = 1.55649985
Iteration 172, loss = 1.55585979
```

```
Iteration 173, loss = 1.55550014
Iteration 174, loss = 1.55514103
Iteration 175, loss = 1.55504217
Iteration 176, loss = 1.55482804
Iteration 177, loss = 1.55481664
Iteration 178, loss = 1.55482592
Iteration 179, loss = 1.55388223
Iteration 180, loss = 1.55478969
Iteration 181, loss = 1.55399109
Iteration 182, loss = 1.55279833
Iteration 183, loss = 1.55301054
Iteration 184, loss = 1.55296524
Iteration 185, loss = 1.55278536
Iteration 186, loss = 1.55207825
Iteration 187, loss = 1.55165938
Iteration 188, loss = 1.55229395
Iteration 189, loss = 1.55113958
Iteration 190, loss = 1.55116700
Iteration 191, loss = 1.55081365
Iteration 192, loss = 1.55039809
Iteration 193, loss = 1.55092128
Iteration 194, loss = 1.55032287
Iteration 195, loss = 1.54985993
Iteration 196, loss = 1.54922683
Iteration 197, loss = 1.54963175
Iteration 198, loss = 1.54898539
Iteration 199, loss = 1.54934480
Iteration 200, loss = 1.54845358
Iteration 1, loss = 1.74005196
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
Iteration 2, loss = 1.64432467
Iteration 3, loss = 1.64026517
Iteration 4, loss = 1.63916845
Iteration 5, loss = 1.63875403
Iteration 6, loss = 1.63835069
Iteration 7, loss = 1.63792887
Iteration 8, loss = 1.63784081
Iteration 9, loss = 1.63732089
Iteration 10, loss = 1.63701391
Iteration 11, loss = 1.63673155
Iteration 12, loss = 1.63638213
Iteration 13, loss = 1.63595514
Iteration 14, loss = 1.63553705
Iteration 15, loss = 1.63625204
Iteration 16, loss = 1.63494724
Iteration 17, loss = 1.63447103
Iteration 18, loss = 1.63442455
Iteration 19, loss = 1.63403123
Iteration 20, loss = 1.63356699
Iteration 21, loss = 1.63312032
Iteration 22, loss = 1.63308799
Iteration 23, loss = 1.63234583
Iteration 24, loss = 1.63222664
Iteration 25, loss = 1.63151459
Iteration 26, loss = 1.63139452
Iteration 27, loss = 1.63066130
Iteration 28, loss = 1.63063029
Iteration 29, loss = 1.62990664
Iteration 30, loss = 1.62950362
Iteration 31, loss = 1.62874398
Iteration 32, loss = 1.62849183
Iteration 33, loss = 1.62797620
Iteration 34, loss = 1.62753884
Iteration 35, loss = 1.62750771
Iteration 36, loss = 1.62658927
Iteration 37, loss = 1.62600318
Iteration 38, loss = 1.62575621
Iteration 39, loss = 1.62506138
Iteration 40, loss = 1.62440546
Iteration 41, loss = 1.62369574
Iteration 42, loss = 1.62322597
Iteration 43, loss = 1.62251796
Iteration 44, loss = 1.62183282
Iteration 45, loss = 1.62146599
Iteration 46, loss = 1.62099152
Iteration 47, loss = 1.62014558
Iteration 48, loss = 1.61939109
Iteration 49, loss = 1.61891526
Iteration 50, loss = 1.61803632
Iteration 51, loss = 1.61764411
Iteration 52, loss = 1.61668950
Iteration 53, loss = 1.61621997
Iteration 54, loss = 1.61564116
Iteration 55, loss = 1.61461547
Iteration 56, loss = 1.61384165
Iteration 57, loss = 1.61321926
Iteration 58, loss = 1.61270551
Iteration 59, loss = 1.61179263
Iteration 60, loss = 1.61072409
Iteration 61, loss = 1.61036384
```

```
Iteration 62, loss = 1.60935434
Iteration 63, loss = 1.60863505
Iteration 64, loss = 1.60835129
Iteration 65, loss = 1.60790033
Iteration 66, loss = 1.60695864
Iteration 67, loss = 1.60627954
Iteration 68, loss = 1.60524674
Iteration 69, loss = 1.60460901
Iteration 70, loss = 1.60384032
Iteration 71, loss = 1.60299762
Iteration 72, loss = 1.60204242
Iteration 73, loss = 1.60185629
Iteration 74, loss = 1.60133948
Iteration 75, loss = 1.59996538
Iteration 76, loss = 1.59936210
Iteration 77, loss = 1.59829213
Iteration 78, loss = 1.59798070
Iteration 79, loss = 1.59735073
Iteration 80, loss = 1.59671512
Iteration 81, loss = 1.59569901
Iteration 82, loss = 1.59559811
Iteration 83, loss = 1.59468528
Iteration 84, loss = 1.59406444
Iteration 85, loss = 1.59288225
Iteration 86, loss = 1.59241304
Iteration 87, loss = 1.59175028
Iteration 88, loss = 1.59099760
Iteration 89, loss = 1.59125308
Iteration 90, loss = 1.58978599
Iteration 91, loss = 1.58932046
Iteration 92, loss = 1.58863716
Iteration 93, loss = 1.58801696
Iteration 94, loss = 1.58721869
Iteration 95, loss = 1.58704181
Iteration 96, loss = 1.58598640
Iteration 97, loss = 1.58545956
Iteration 98, loss = 1.58521613
Iteration 99, loss = 1.58437091
Iteration 100, loss = 1.58383096
Iteration 101, loss = 1.58350999
Iteration 102, loss = 1.58291189
Iteration 103, loss = 1.58270581
Iteration 104, loss = 1.58189441
Iteration 105, loss = 1.58205544
Iteration 106, loss = 1.58242224
Iteration 107, loss = 1.58008088
Iteration 108, loss = 1.57953920
Iteration 109, loss = 1.57981746
Iteration 110, loss = 1.57862671
Iteration 111, loss = 1.57794878
Iteration 112, loss = 1.57708374
Iteration 113, loss = 1.57695267
Iteration 114, loss = 1.57675538
Iteration 115, loss = 1.57633280
Iteration 116, loss = 1.57528804
Iteration 117, loss = 1.57492261
Iteration 118, loss = 1.57515486
Iteration 119, loss = 1.57450682
Iteration 120, loss = 1.57349983
Iteration 121, loss = 1.57306453
Iteration 122, loss = 1.57233418
Iteration 123, loss = 1.57201747
Iteration 124, loss = 1.57199801
Iteration 125, loss = 1.57137188
Iteration 126, loss = 1.57078649
Iteration 127, loss = 1.57064113
Iteration 128, loss = 1.57068042
Iteration 129, loss = 1.56968155
Iteration 130, loss = 1.56944386
Iteration 131, loss = 1.56927461
Iteration 132, loss = 1.56849935
Iteration 133, loss = 1.56793057
Iteration 134, loss = 1.56722698
Iteration 135, loss = 1.56678057
Iteration 136, loss = 1.56584894
Iteration 137, loss = 1.56604563
Iteration 138, loss = 1.56550389
Iteration 139, loss = 1.56484521
Iteration 140, loss = 1.56453010
Iteration 141, loss = 1.56444177
Iteration 142, loss = 1.56410299
Iteration 143, loss = 1.56370419
Iteration 144, loss = 1.56417739
Iteration 145, loss = 1.56262918
Iteration 146, loss = 1.56242013
Iteration 147, loss = 1.56215598
Iteration 148, loss = 1.56180782
Iteration 149, loss = 1.56132105
Iteration 150, loss = 1.56149404
Iteration 151, loss = 1.56085391
Iteration 152, loss = 1.56019505
Iteration 153, loss = 1.55979044
```

```
Iteration 154, loss = 1.55998776
Iteration 155, loss = 1.55851100
Iteration 156, loss = 1.55828665
Iteration 157, loss = 1.55815123
Iteration 158, loss = 1.55834203
Iteration 159, loss = 1.55753738
Iteration 160, loss = 1.55733324
Iteration 161, loss = 1.55695952
Iteration 162, loss = 1.55604232
Iteration 163, loss = 1.55690345
Iteration 164, loss = 1.55616476
Iteration 165, loss = 1.55533575
Iteration 166, loss = 1.55567993
Iteration 167, loss = 1.55432369
Iteration 168, loss = 1.55532488
Iteration 169, loss = 1.55422574
Iteration 170, loss = 1.55417539
Iteration 171, loss = 1.55356438
Iteration 172, loss = 1.55306799
Iteration 173, loss = 1.55348607
Iteration 174, loss = 1.55236781
Iteration 175, loss = 1.55336151
Iteration 176, loss = 1.55253754
Iteration 177, loss = 1.55171090
Iteration 178, loss = 1.55166321
Iteration 179, loss = 1.55127875
Iteration 180, loss = 1.55093849
Iteration 181, loss = 1.55023790
Iteration 182, loss = 1.55081529
Iteration 183, loss = 1.54997213
Iteration 184, loss = 1.55090790
Iteration 185, loss = 1.54947892
Iteration 186, loss = 1.54923198
Iteration 187, loss = 1.54919556
Iteration 188, loss = 1.54927305
Iteration 189, loss = 1.54974247
Iteration 190, loss = 1.54875204
Iteration 191, loss = 1.54768885
Iteration 192, loss = 1.54745380
Iteration 193, loss = 1.54708165
Iteration 194, loss = 1.54661381
Iteration 195, loss = 1.54734687
Iteration 196, loss = 1.54710689
Iteration 197, loss = 1.54758898
Iteration 198, loss = 1.54594320
Iteration 199, loss = 1.54609607
Iteration 200, loss = 1.54573778
Iteration 1, loss = 1.73885129
Iteration 2, loss = 1.64478595
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
Iteration 3, loss = 1.64041842
Iteration 4, loss = 1.63935656
Iteration 5, loss = 1.63896434
Iteration 6, loss = 1.63879053
Iteration 7, loss = 1.63827404
Iteration 8, loss = 1.63778687
Iteration 9, loss = 1.63764235
Iteration 10, loss = 1.63709153
Iteration 11, loss = 1.63666945
Iteration 12, loss = 1.63655138
Iteration 13, loss = 1.63622297
Iteration 14, loss = 1.63601467
Iteration 15, loss = 1.63611028
Iteration 16, loss = 1.63529322
Iteration 17, loss = 1.63496937
Iteration 18, loss = 1.63479449
Iteration 19, loss = 1.63422420
Iteration 20, loss = 1.63389858
Iteration 21, loss = 1.63338713
Iteration 22, loss = 1.63328885
Iteration 23, loss = 1.63267528
Iteration 24, loss = 1.63241217
Iteration 25, loss = 1.63206784
Iteration 26, loss = 1.63158577
Iteration 27, loss = 1.63107503
Iteration 28, loss = 1.63065694
Iteration 29, loss = 1.63031649
Iteration 30, loss = 1.63041372
Iteration 31, loss = 1.62921371
Iteration 32, loss = 1.62925372
Iteration 33, loss = 1.62910960
Iteration 34, loss = 1.62847932
Iteration 35, loss = 1.62790101
Iteration 36, loss = 1.62730791
Iteration 37, loss = 1.62676869
Iteration 38, loss = 1.62635029
Iteration 39, loss = 1.62582809
Iteration 40, loss = 1.62518890
Iteration 41, loss = 1.62492557
Iteration 42, loss = 1.62409948
```

```
Iteration 43, loss = 1.62371008
Iteration 44, loss = 1.62299536
Iteration 45, loss = 1.62267037
Iteration 46, loss = 1.62214796
Iteration 47, loss = 1.62128824
Iteration 48, loss = 1.62075626
Iteration 49, loss = 1.61993829
Iteration 50, loss = 1.61952453
Iteration 51, loss = 1.61899929
Iteration 52, loss = 1.61809670
Iteration 53, loss = 1.61773394
Iteration 54, loss = 1.61726328
Iteration 55, loss = 1.61613039
Iteration 56, loss = 1.61546096
Iteration 57, loss = 1.61476163
Iteration 58, loss = 1.61411598
Iteration 59, loss = 1.61351245
Iteration 60, loss = 1.61261260
Iteration 61, loss = 1.61230681
Iteration 62, loss = 1.61120520
Iteration 63, loss = 1.61044811
Iteration 64, loss = 1.61018869
Iteration 65, loss = 1.60937529
Iteration 66, loss = 1.60920685
Iteration 67, loss = 1.60774060
Iteration 68, loss = 1.60734279
Iteration 69, loss = 1.60651487
Iteration 70, loss = 1.60598584
Iteration 71, loss = 1.60500305
Iteration 72, loss = 1.60473320
Iteration 73, loss = 1.60348689
Iteration 74, loss = 1.60293733
Iteration 75, loss = 1.60238517
Iteration 76, loss = 1.60137404
Iteration 77, loss = 1.60074690
Iteration 78, loss = 1.59994695
Iteration 79, loss = 1.59962270
Iteration 80, loss = 1.59881318
Iteration 81, loss = 1.59812831
Iteration 82, loss = 1.59806904
Iteration 83, loss = 1.59665930
Iteration 84, loss = 1.59670960
Iteration 85, loss = 1.59552966
Iteration 86, loss = 1.59483149
Iteration 87, loss = 1.59414743
Iteration 88, loss = 1.59356379
Iteration 89, loss = 1.59332027
Iteration 90, loss = 1.59245974
Iteration 91, loss = 1.59202726
Iteration 92, loss = 1.59099328
Iteration 93, loss = 1.59081365
Iteration 94, loss = 1.58963208
Iteration 95, loss = 1.58893078
Iteration 96, loss = 1.58858330
Iteration 97, loss = 1.58795184
Iteration 98, loss = 1.58901869
Iteration 99, loss = 1.58674539
Iteration 100, loss = 1.58627768
Iteration 101, loss = 1.58597220
Iteration 102, loss = 1.58641154
Iteration 103, loss = 1.58435540
Iteration 104, loss = 1.58442488
Iteration 105, loss = 1.58352752
Iteration 106, loss = 1.58348460
Iteration 107, loss = 1.58242815
Iteration 108, loss = 1.58209580
Iteration 109, loss = 1.58153312
Iteration 110, loss = 1.58080274
Iteration 111, loss = 1.58092747
Iteration 112, loss = 1.57973625
Iteration 113, loss = 1.57958413
Iteration 114, loss = 1.57930652
Iteration 115, loss = 1.57853145
Iteration 116, loss = 1.57814495
Iteration 117, loss = 1.57722732
Iteration 118, loss = 1.57739894
Iteration 119, loss = 1.57703831
Iteration 120, loss = 1.57675714
Iteration 121, loss = 1.57649871
Iteration 122, loss = 1.57557039
Iteration 123, loss = 1.57486739
Iteration 124, loss = 1.57463958
Iteration 125, loss = 1.57389041
Iteration 126, loss = 1.57404902
Iteration 127, loss = 1.57340944
Iteration 128, loss = 1.57242793
Iteration 129, loss = 1.57203656
Iteration 130, loss = 1.57144592
Iteration 131, loss = 1.57117423
Iteration 132, loss = 1.57077697
Iteration 133, loss = 1.57025027
Iteration 134, loss = 1.56944996
```

```
Iteration 135, loss = 1.56936780
Iteration 136, loss = 1.56878285
Iteration 137, loss = 1.56846993
Iteration 138, loss = 1.56895743
Iteration 139, loss = 1.56776337
Iteration 140, loss = 1.56687537
Iteration 141, loss = 1.56666082
Iteration 142, loss = 1.56680574
Iteration 143, loss = 1.56673972
Iteration 144, loss = 1.56621734
Iteration 145, loss = 1.56528046
Iteration 146, loss = 1.56503029
Iteration 147, loss = 1.56441446
Iteration 148, loss = 1.56475176
Iteration 149, loss = 1.56407388
Iteration 150, loss = 1.56458251
Iteration 151, loss = 1.56363218
Iteration 152, loss = 1.56295594
Iteration 153, loss = 1.56212002
Iteration 154, loss = 1.56224992
Iteration 155, loss = 1.56160021
Iteration 156, loss = 1.56092907
Iteration 157, loss = 1.56075670
Iteration 158, loss = 1.56017003
Iteration 159, loss = 1.56030919
Iteration 160, loss = 1.55958620
Iteration 161, loss = 1.55907521
Iteration 162, loss = 1.55910059
Iteration 163, loss = 1.55901770
Iteration 164, loss = 1.55855246
Iteration 165, loss = 1.55842569
Iteration 166, loss = 1.55745261
Iteration 167, loss = 1.55696397
Iteration 168, loss = 1.55768805
Iteration 169, loss = 1.55639683
Iteration 170, loss = 1.55650560
Iteration 171, loss = 1.55622752
Iteration 172, loss = 1.55608994
Iteration 173, loss = 1.55497138
Iteration 174, loss = 1.55443170
Iteration 175, loss = 1.55494521
Iteration 176, loss = 1.55438604
Iteration 177, loss = 1.55532676
Iteration 178, loss = 1.55390738
Iteration 179, loss = 1.55358876
Iteration 180, loss = 1.55282494
Iteration 181, loss = 1.55297649
Iteration 182, loss = 1.55316186
Iteration 183, loss = 1.55290106
Iteration 184, loss = 1.55221039
Iteration 185, loss = 1.55182204
Iteration 186, loss = 1.55101101
Iteration 187, loss = 1.55119102
Iteration 188, loss = 1.55148857
Iteration 189, loss = 1.55066962
Iteration 190, loss = 1.55054563
Iteration 191, loss = 1.54963565
Iteration 192, loss = 1.55010169
Iteration 193, loss = 1.54904442
Iteration 194, loss = 1.54909253
Iteration 195, loss = 1.55082181
Iteration 196, loss = 1.54860872
Iteration 197, loss = 1.54865668
Iteration 198, loss = 1.54913713
Iteration 199, loss = 1.54800341
Iteration 200, loss = 1.54769329
Iteration 1, loss = 1.73991947
Iteration 2, loss = 1.64465536
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
Iteration 3, loss = 1.64046362
Iteration 4, loss = 1.63955672
Iteration 5, loss = 1.63886268
Iteration 6, loss = 1.63841251
Iteration 7, loss = 1.63786952
Iteration 8, loss = 1.63734025
Iteration 9, loss = 1.63726564
Iteration 10, loss = 1.63689211
Iteration 11, loss = 1.63634850
Iteration 12, loss = 1.63629702
Iteration 13, loss = 1.63574729
Iteration 14, loss = 1.63519446
Iteration 15, loss = 1.63569949
Iteration 16, loss = 1.63467934
Iteration 17, loss = 1.63445755
Iteration 18, loss = 1.63427322
Iteration 19, loss = 1.63374484
Iteration 20, loss = 1.63340616
Iteration 21, loss = 1.63266158
Iteration 22, loss = 1.63254872
Iteration 23, loss = 1.63184328
```

```
Iteration 24, loss = 1.63178276
Iteration 25, loss = 1.63101770
Iteration 26, loss = 1.63056336
Iteration 27, loss = 1.62995804
Iteration 28, loss = 1.62975051
Iteration 29, loss = 1.62919677
Iteration 30, loss = 1.62878581
Iteration 31, loss = 1.62800210
Iteration 32, loss = 1.62768759
Iteration 33, loss = 1.62734374
Iteration 34, loss = 1.62655649
Iteration 35, loss = 1.62620275
Iteration 36, loss = 1.62535868
Iteration 37, loss = 1.62485349
Iteration 38, loss = 1.62456207
Iteration 39, loss = 1.62400066
Iteration 40, loss = 1.62329108
Iteration 41, loss = 1.62255017
Iteration 42, loss = 1.62188625
Iteration 43, loss = 1.62095907
Iteration 44, loss = 1.62065782
Iteration 45, loss = 1.61986294
Iteration 46, loss = 1.61923067
Iteration 47, loss = 1.61837179
Iteration 48, loss = 1.61762284
Iteration 49, loss = 1.61676342
Iteration 50, loss = 1.61615943
Iteration 51, loss = 1.61536221
Iteration 52, loss = 1.61458362
Iteration 53, loss = 1.61411009
Iteration 54, loss = 1.61333186
Iteration 55, loss = 1.61230286
Iteration 56, loss = 1.61145662
Iteration 57, loss = 1.61082854
Iteration 58, loss = 1.61017051
Iteration 59, loss = 1.60959888
Iteration 60, loss = 1.60833245
Iteration 61, loss = 1.60765003
Iteration 62, loss = 1.60677773
Iteration 63, loss = 1.60584922
Iteration 64, loss = 1.60517154
Iteration 65, loss = 1.60491127
Iteration 66, loss = 1.60391226
Iteration 67, loss = 1.60261760
Iteration 68, loss = 1.60234175
Iteration 69, loss = 1.60153270
Iteration 70, loss = 1.60014416
Iteration 71, loss = 1.59941469
Iteration 72, loss = 1.59863441
Iteration 73, loss = 1.59763893
Iteration 74, loss = 1.59795935
Iteration 75, loss = 1.59651837
Iteration 76, loss = 1.59551672
Iteration 77, loss = 1.59543559
Iteration 78, loss = 1.59406598
Iteration 79, loss = 1.59338644
Iteration 80, loss = 1.59277539
Iteration 81, loss = 1.59213072
Iteration 82, loss = 1.59196966
Iteration 83, loss = 1.59022186
Iteration 84, loss = 1.58999901
Iteration 85, loss = 1.58890093
Iteration 86, loss = 1.58801246
Iteration 87, loss = 1.58758245
Iteration 88, loss = 1.58692601
Iteration 89, loss = 1.58617196
Iteration 90, loss = 1.58557171
Iteration 91, loss = 1.58498451
Iteration 92, loss = 1.58423610
Iteration 93, loss = 1.58376480
Iteration 94, loss = 1.58327971
Iteration 95, loss = 1.58240093
Iteration 96, loss = 1.58173204
Iteration 97, loss = 1.58149902
Iteration 98, loss = 1.58064717
Iteration 99, loss = 1.58032640
Iteration 100, loss = 1.57931907
Iteration 101, loss = 1.57897638
Iteration 102, loss = 1.57869755
Iteration 103, loss = 1.57782782
Iteration 104, loss = 1.57722075
Iteration 105, loss = 1.57681203
Iteration 106, loss = 1.57612233
Iteration 107, loss = 1.57582174
Iteration 108, loss = 1.57504695
Iteration 109, loss = 1.57392551
Iteration 110, loss = 1.57381273
Iteration 111, loss = 1.57298153
Iteration 112, loss = 1.57264342
Iteration 113, loss = 1.57202409
Iteration 114, loss = 1.57236824
Iteration 115, loss = 1.57127632
```

```
Iteration 116, loss = 1.57023546
Iteration 117, loss = 1.56977782
Iteration 118, loss = 1.56999231
Iteration 119, loss = 1.56940072
Iteration 120, loss = 1.56850700
Iteration 121, loss = 1.56815197
Iteration 122, loss = 1.56816241
Iteration 123, loss = 1.56703923
Iteration 124, loss = 1.56670930
Iteration 125, loss = 1.56587249
Iteration 126, loss = 1.56574815
Iteration 127, loss = 1.56551822
Iteration 128, loss = 1.56483731
Iteration 129, loss = 1.56397037
Iteration 130, loss = 1.56416518
Iteration 131, loss = 1.56337036
Iteration 132, loss = 1.56329888
Iteration 133, loss = 1.56285118
Iteration 134, loss = 1.56174409
Iteration 135, loss = 1.56209008
Iteration 136, loss = 1.56073983
Iteration 137, loss = 1.56069039
Iteration 138, loss = 1.56074371
Iteration 139, loss = 1.55944122
Iteration 140, loss = 1.55908326
Iteration 141, loss = 1.55879482
Iteration 142, loss = 1.55847406
Iteration 143, loss = 1.55784860
Iteration 144, loss = 1.55809418
Iteration 145, loss = 1.55734600
Iteration 146, loss = 1.55673785
Iteration 147, loss = 1.55609240
Iteration 148, loss = 1.55641617
Iteration 149, loss = 1.55555317
Iteration 150, loss = 1.55620904
Iteration 151, loss = 1.55518755
Iteration 152, loss = 1.55434004
Iteration 153, loss = 1.55423545
Iteration 154, loss = 1.55440332
Iteration 155, loss = 1.55357458
Iteration 156, loss = 1.55302934
Iteration 157, loss = 1.55286031
Iteration 158, loss = 1.55231612
Iteration 159, loss = 1.55217903
Iteration 160, loss = 1.55186137
Iteration 161, loss = 1.55095041
Iteration 162, loss = 1.55108411
Iteration 163, loss = 1.54984152
Iteration 164, loss = 1.55212899
Iteration 165, loss = 1.55010979
Iteration 166, loss = 1.55008025
Iteration 167, loss = 1.54888277
Iteration 168, loss = 1.55000062
Iteration 169, loss = 1.54879805
Iteration 170, loss = 1.54861169
Iteration 171, loss = 1.54771468
Iteration 172, loss = 1.54790614
Iteration 173, loss = 1.54736006
Iteration 174, loss = 1.54653686
Iteration 175, loss = 1.54717302
Iteration 176, loss = 1.54637744
Iteration 177, loss = 1.54662476
Iteration 178, loss = 1.54600159
Iteration 179, loss = 1.54538368
Iteration 180, loss = 1.54560615
Iteration 181, loss = 1.54548099
Iteration 182, loss = 1.54393598
Iteration 183, loss = 1.54506643
Iteration 184, loss = 1.54456650
Iteration 185, loss = 1.54386254
Iteration 186, loss = 1.54344337
Iteration 187, loss = 1.54298565
Iteration 188, loss = 1.54407984
Iteration 189, loss = 1.54336746
Iteration 190, loss = 1.54306835
Iteration 191, loss = 1.54191034
Iteration 192, loss = 1.54200056
Iteration 193, loss = 1.54149549
Iteration 194, loss = 1.54082788
Iteration 195, loss = 1.54220960
Iteration 196, loss = 1.54143103
Iteration 197, loss = 1.54248370
Iteration 198, loss = 1.54087449
Iteration 199, loss = 1.54045493
Iteration 200, loss = 1.54045574
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
Iteration 1, loss = 1.74126714
Iteration 2, loss = 1.64500120
Iteration 3, loss = 1.64054328
Iteration 4, loss = 1.63933691
```

```
Iteration 5, loss = 1.63879272
Iteration 6, loss = 1.63847307
Iteration 7, loss = 1.63800442
Iteration 8, loss = 1.63770690
Iteration 9, loss = 1.63721845
Iteration 10, loss = 1.63699028
Iteration 11, loss = 1.63654175
Iteration 12, loss = 1.63632504
Iteration 13, loss = 1.63581104
Iteration 14, loss = 1.63555610
Iteration 15, loss = 1.63535377
Iteration 16, loss = 1.63466584
Iteration 17, loss = 1.63435879
Iteration 18, loss = 1.63408811
Iteration 19, loss = 1.63355092
Iteration 20, loss = 1.63342250
Iteration 21, loss = 1.63281662
Iteration 22, loss = 1.63242755
Iteration 23, loss = 1.63209281
Iteration 24, loss = 1.63148422
Iteration 25, loss = 1.63103077
Iteration 26, loss = 1.63135071
Iteration 27, loss = 1.63007914
Iteration 28, loss = 1.62979199
Iteration 29, loss = 1.62942380
Iteration 30, loss = 1.62865561
Iteration 31, loss = 1.62832250
Iteration 32, loss = 1.62758937
Iteration 33, loss = 1.62705762
Iteration 34, loss = 1.62628293
Iteration 35, loss = 1.62624477
Iteration 36, loss = 1.62529988
Iteration 37, loss = 1.62510009
Iteration 38, loss = 1.62421285
Iteration 39, loss = 1.62377059
Iteration 40, loss = 1.62388159
Iteration 41, loss = 1.62261232
Iteration 42, loss = 1.62166670
Iteration 43, loss = 1.62117583
Iteration 44, loss = 1.62093025
Iteration 45, loss = 1.61974621
Iteration 46, loss = 1.61929187
Iteration 47, loss = 1.61807123
Iteration 48, loss = 1.61736631
Iteration 49, loss = 1.61670124
Iteration 50, loss = 1.61601996
Iteration 51, loss = 1.61521405
Iteration 52, loss = 1.61464079
Iteration 53, loss = 1.61379662
Iteration 54, loss = 1.61289007
Iteration 55, loss = 1.61231946
Iteration 56, loss = 1.61174179
Iteration 57, loss = 1.61081158
Iteration 58, loss = 1.60964204
Iteration 59, loss = 1.60908242
Iteration 60, loss = 1.60833328
Iteration 61, loss = 1.60774294
Iteration 62, loss = 1.60686835
Iteration 63, loss = 1.60618286
Iteration 64, loss = 1.60519697
Iteration 65, loss = 1.60444644
Iteration 66, loss = 1.60349528
Iteration 67, loss = 1.60334852
Iteration 68, loss = 1.60233537
Iteration 69, loss = 1.60108384
Iteration 70, loss = 1.60027231
Iteration 71, loss = 1.59975150
Iteration 72, loss = 1.59967210
Iteration 73, loss = 1.59818534
Iteration 74, loss = 1.59733278
Iteration 75, loss = 1.59737421
Iteration 76, loss = 1.59746708
Iteration 77, loss = 1.59545394
Iteration 78, loss = 1.59488771
Iteration 79, loss = 1.59416273
Iteration 80, loss = 1.59294831
Iteration 81, loss = 1.59219080
Iteration 82, loss = 1.59183760
Iteration 83, loss = 1.59125111
Iteration 84, loss = 1.59087663
Iteration 85, loss = 1.58952085
Iteration 86, loss = 1.58894662
Iteration 87, loss = 1.58835503
Iteration 88, loss = 1.58822453
Iteration 89, loss = 1.58696463
Iteration 90, loss = 1.58683968
Iteration 91, loss = 1.58601814
Iteration 92, loss = 1.58525974
Iteration 93, loss = 1.58491890
Iteration 94, loss = 1.58432606
Iteration 95, loss = 1.58344564
Iteration 96, loss = 1.58297389
```

```
Iteration 97, loss = 1.58248815
Iteration 98, loss = 1.58207094
Iteration 99, loss = 1.58152510
Iteration 100, loss = 1.58078769
Iteration 101, loss = 1.58054584
Iteration 102, loss = 1.58015319
Iteration 103, loss = 1.58083935
Iteration 104, loss = 1.57908412
Iteration 105, loss = 1.57818038
Iteration 106, loss = 1.57743797
Iteration 107, loss = 1.57781469
Iteration 108, loss = 1.57715035
Iteration 109, loss = 1.57708383
Iteration 110, loss = 1.57651860
Iteration 111, loss = 1.57533867
Iteration 112, loss = 1.57499523
Iteration 113, loss = 1.57519675
Iteration 114, loss = 1.57388321
Iteration 115, loss = 1.57378595
Iteration 116, loss = 1.57377135
Iteration 117, loss = 1.57282657
Iteration 118, loss = 1.57242856
Iteration 119, loss = 1.57207312
Iteration 120, loss = 1.57213813
Iteration 121, loss = 1.57196435
Iteration 122, loss = 1.57064757
Iteration 123, loss = 1.57011312
Iteration 124, loss = 1.56970999
Iteration 125, loss = 1.56955460
Iteration 126, loss = 1.57005601
Iteration 127, loss = 1.56989637
Iteration 128, loss = 1.56833466
Iteration 129, loss = 1.56778548
Iteration 130, loss = 1.56735010
Iteration 131, loss = 1.56673823
Iteration 132, loss = 1.56653985
Iteration 133, loss = 1.56600324
Iteration 134, loss = 1.56636599
Iteration 135, loss = 1.56511401
Iteration 136, loss = 1.56495220
Iteration 137, loss = 1.56465630
Iteration 138, loss = 1.56403825
Iteration 139, loss = 1.56484359
Iteration 140, loss = 1.56340801
Iteration 141, loss = 1.56246871
Iteration 142, loss = 1.56306574
Iteration 143, loss = 1.56196546
Iteration 144, loss = 1.56212303
Iteration 145, loss = 1.56162990
Iteration 146, loss = 1.56178915
Iteration 147, loss = 1.56060532
Iteration 148, loss = 1.56079765
Iteration 149, loss = 1.55953388
Iteration 150, loss = 1.55992428
Iteration 151, loss = 1.55901814
Iteration 152, loss = 1.55918655
Iteration 153, loss = 1.55932122
Iteration 154, loss = 1.55907299
Iteration 155, loss = 1.55892716
Iteration 156, loss = 1.55795088
Iteration 157, loss = 1.55734885
Iteration 158, loss = 1.55640420
Iteration 159, loss = 1.55620305
Iteration 160, loss = 1.55683899
Iteration 161, loss = 1.55613771
Iteration 162, loss = 1.55558817
Iteration 163, loss = 1.55487129
Iteration 164, loss = 1.55485938
Iteration 165, loss = 1.55454591
Iteration 166, loss = 1.55388466
Iteration 167, loss = 1.55383350
Iteration 168, loss = 1.55334840
Iteration 169, loss = 1.55312631
Iteration 170, loss = 1.55230970
Iteration 171, loss = 1.55268039
Iteration 172, loss = 1.55233537
Iteration 173, loss = 1.55243552
Iteration 174, loss = 1.55170866
Iteration 175, loss = 1.55197248
Iteration 176, loss = 1.55126703
Iteration 177, loss = 1.55098996
Iteration 178, loss = 1.55061014
Iteration 179, loss = 1.55041030
Iteration 180, loss = 1.55062715
Iteration 181, loss = 1.54948174
Iteration 182, loss = 1.54998453
Iteration 183, loss = 1.54881921
Iteration 184, loss = 1.54858291
Iteration 185, loss = 1.54839848
Iteration 186, loss = 1.54925849
Iteration 187, loss = 1.54923016
Iteration 188, loss = 1.54821410
```

```
Iteration 189, loss = 1.54826222
Iteration 190, loss = 1.54699106
Iteration 191, loss = 1.54722985
Iteration 192, loss = 1.54717907
Iteration 193, loss = 1.54574452
Iteration 194, loss = 1.54728659
Iteration 195, loss = 1.54567695
Iteration 196, loss = 1.54583152
Iteration 197, loss = 1.54482982
Iteration 198, loss = 1.54509863
Iteration 199, loss = 1.54497320
Iteration 200, loss = 1.54422470
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

In [ ]:

```
best_siftMLP_model_cvscores
```

Out[ ]:

```
array([0.41867102, 0.41867102, 0.41704036, 0.40277212, 0.40660685])
```

In [ ]:

```
best_siftMLP_model_cvscores.mean()
```

Out[ ]:

```
0.4127522712475784
```

In [ ]:

```
t0 = time.time()
best_siftMLP_model.fit(hist_array, y_train_list)
tt = time.time()
print(tt - t0)
```

```
Iteration 1, loss = 1.72211328
Iteration 2, loss = 1.64199257
Iteration 3, loss = 1.63961353
Iteration 4, loss = 1.63875336
Iteration 5, loss = 1.63813757
Iteration 6, loss = 1.63758412
Iteration 7, loss = 1.63716188
Iteration 8, loss = 1.63677199
Iteration 9, loss = 1.63639797
Iteration 10, loss = 1.63612659
Iteration 11, loss = 1.63564339
Iteration 12, loss = 1.63509039
Iteration 13, loss = 1.63468251
Iteration 14, loss = 1.63433831
Iteration 15, loss = 1.63401839
Iteration 16, loss = 1.63354417
Iteration 17, loss = 1.63293706
Iteration 18, loss = 1.63238046
Iteration 19, loss = 1.63194910
Iteration 20, loss = 1.63140521
Iteration 21, loss = 1.63088356
Iteration 22, loss = 1.63006052
Iteration 23, loss = 1.62981165
Iteration 24, loss = 1.62942280
Iteration 25, loss = 1.62869155
Iteration 26, loss = 1.62820943
Iteration 27, loss = 1.62732822
Iteration 28, loss = 1.62673698
Iteration 29, loss = 1.62586410
Iteration 30, loss = 1.62544657
Iteration 31, loss = 1.62483201
Iteration 32, loss = 1.62381884
Iteration 33, loss = 1.62323955
Iteration 34, loss = 1.62262486
Iteration 35, loss = 1.62161258
Iteration 36, loss = 1.62060345
Iteration 37, loss = 1.61991913
Iteration 38, loss = 1.61908263
Iteration 39, loss = 1.61832665
Iteration 40, loss = 1.61735915
Iteration 41, loss = 1.61648811
Iteration 42, loss = 1.61565572
Iteration 43, loss = 1.61484352
Iteration 44, loss = 1.61419101
Iteration 45, loss = 1.61291193
Iteration 46, loss = 1.61237263
Iteration 47, loss = 1.61108983
Iteration 48, loss = 1.61017611
Iteration 49, loss = 1.60918202
Iteration 50, loss = 1.60841502
Iteration 51, loss = 1.60728467
Iteration 52, loss = 1.60666684
Iteration 53, loss = 1.60569955
Iteration 54, loss = 1.60477468
Iteration 55, loss = 1.60356068
Iteration 56, loss = 1.60281633
Iteration 57, loss = 1.60212259
Iteration 58, loss = 1.60124787
Iteration 59, loss = 1.60035110
Iteration 60, loss = 1.59956257
```

```
Iteration 61, loss = 1.59852488
Iteration 62, loss = 1.59780832
Iteration 63, loss = 1.59703450
Iteration 64, loss = 1.59631666
Iteration 65, loss = 1.59531227
Iteration 66, loss = 1.59443125
Iteration 67, loss = 1.59355545
Iteration 68, loss = 1.59299933
Iteration 69, loss = 1.59245463
Iteration 70, loss = 1.59168629
Iteration 71, loss = 1.59078141
Iteration 72, loss = 1.59001044
Iteration 73, loss = 1.58938015
Iteration 74, loss = 1.58915165
Iteration 75, loss = 1.58817441
Iteration 76, loss = 1.58764923
Iteration 77, loss = 1.58728833
Iteration 78, loss = 1.58629603
Iteration 79, loss = 1.58557933
Iteration 80, loss = 1.58526720
Iteration 81, loss = 1.58437354
Iteration 82, loss = 1.58401975
Iteration 83, loss = 1.58372310
Iteration 84, loss = 1.58296792
Iteration 85, loss = 1.58239595
Iteration 86, loss = 1.58195269
Iteration 87, loss = 1.58103186
Iteration 88, loss = 1.58055206
Iteration 89, loss = 1.58010979
Iteration 90, loss = 1.57993454
Iteration 91, loss = 1.57900279
Iteration 92, loss = 1.57842023
Iteration 93, loss = 1.57811363
Iteration 94, loss = 1.57790824
Iteration 95, loss = 1.57776197
Iteration 96, loss = 1.57681208
Iteration 97, loss = 1.57649856
Iteration 98, loss = 1.57545538
Iteration 99, loss = 1.57503895
Iteration 100, loss = 1.57494181
Iteration 101, loss = 1.57431767
Iteration 102, loss = 1.57365553
Iteration 103, loss = 1.57309185
Iteration 104, loss = 1.57287864
Iteration 105, loss = 1.57176393
Iteration 106, loss = 1.57172226
Iteration 107, loss = 1.57107272
Iteration 108, loss = 1.57173934
Iteration 109, loss = 1.57072484
Iteration 110, loss = 1.57014567
Iteration 111, loss = 1.57003486
Iteration 112, loss = 1.56920264
Iteration 113, loss = 1.56820488
Iteration 114, loss = 1.56807380
Iteration 115, loss = 1.56770539
Iteration 116, loss = 1.56753257
Iteration 117, loss = 1.56690195
Iteration 118, loss = 1.56613028
Iteration 119, loss = 1.56596720
Iteration 120, loss = 1.56562387
Iteration 121, loss = 1.56568796
Iteration 122, loss = 1.56465039
Iteration 123, loss = 1.56471624
Iteration 124, loss = 1.56406597
Iteration 125, loss = 1.56340126
Iteration 126, loss = 1.56304385
Iteration 127, loss = 1.56238483
Iteration 128, loss = 1.56231445
Iteration 129, loss = 1.56177616
Iteration 130, loss = 1.56163400
Iteration 131, loss = 1.56103457
Iteration 132, loss = 1.56096078
Iteration 133, loss = 1.56045019
Iteration 134, loss = 1.55960052
Iteration 135, loss = 1.55901007
Iteration 136, loss = 1.55894765
Iteration 137, loss = 1.55872203
Iteration 138, loss = 1.55773726
Iteration 139, loss = 1.55811239
Iteration 140, loss = 1.55829871
Iteration 141, loss = 1.55700320
Iteration 142, loss = 1.55659395
Iteration 143, loss = 1.55645187
Iteration 144, loss = 1.55599557
Iteration 145, loss = 1.55547587
Iteration 146, loss = 1.55513017
Iteration 147, loss = 1.55486828
Iteration 148, loss = 1.55434999
Iteration 149, loss = 1.55392828
Iteration 150, loss = 1.55432467
Iteration 151, loss = 1.55312201
Iteration 152, loss = 1.55285349
```

```
Iteration 153, loss = 1.55261408
Iteration 154, loss = 1.55258808
Iteration 155, loss = 1.55170370
Iteration 156, loss = 1.55134466
Iteration 157, loss = 1.55139409
Iteration 158, loss = 1.55131739
Iteration 159, loss = 1.55064543
Iteration 160, loss = 1.55055179
Iteration 161, loss = 1.55005221
Iteration 162, loss = 1.54978555
Iteration 163, loss = 1.54903914
Iteration 164, loss = 1.54828479
Iteration 165, loss = 1.54831823
Iteration 166, loss = 1.54851002
Iteration 167, loss = 1.54784194
Iteration 168, loss = 1.54751506
Iteration 169, loss = 1.54762270
Iteration 170, loss = 1.54688131
Iteration 171, loss = 1.54679664
Iteration 172, loss = 1.54657290
Iteration 173, loss = 1.54541134
Iteration 174, loss = 1.54587909
Iteration 175, loss = 1.54559634
Iteration 176, loss = 1.54554236
Iteration 177, loss = 1.54492736
Iteration 178, loss = 1.54472279
Iteration 179, loss = 1.54450519
Iteration 180, loss = 1.54413296
Iteration 181, loss = 1.54406524
Iteration 182, loss = 1.54374379
Iteration 183, loss = 1.54291746
Iteration 184, loss = 1.54310915
Iteration 185, loss = 1.54277881
Iteration 186, loss = 1.54239182
Iteration 187, loss = 1.54267827
Iteration 188, loss = 1.54206211
Iteration 189, loss = 1.54139798
Iteration 190, loss = 1.54131357
Iteration 191, loss = 1.54140499
Iteration 192, loss = 1.54055110
Iteration 193, loss = 1.54082864
Iteration 194, loss = 1.54091741
Iteration 195, loss = 1.54059833
Iteration 196, loss = 1.53977767
Iteration 197, loss = 1.53954206
Iteration 198, loss = 1.53982064
Iteration 199, loss = 1.54003968
Iteration 200, loss = 1.53921498
22.775389432907104
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

In [ ]:

```
best_siftMLP_model.score(hist_array, y_train_list)
```

Out[ ]:

```
0.4195205479452055
```

Save best MLP model that uses SIFT descriptors.

In [ ]:

```
bestsiftmlppath = 'drive/My Drive/Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/Models/best_siftMLP_model.pickle'

with open (bestsiftmlppath, 'wb') as f:
    pickle.dump(best_siftMLP_model, f)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

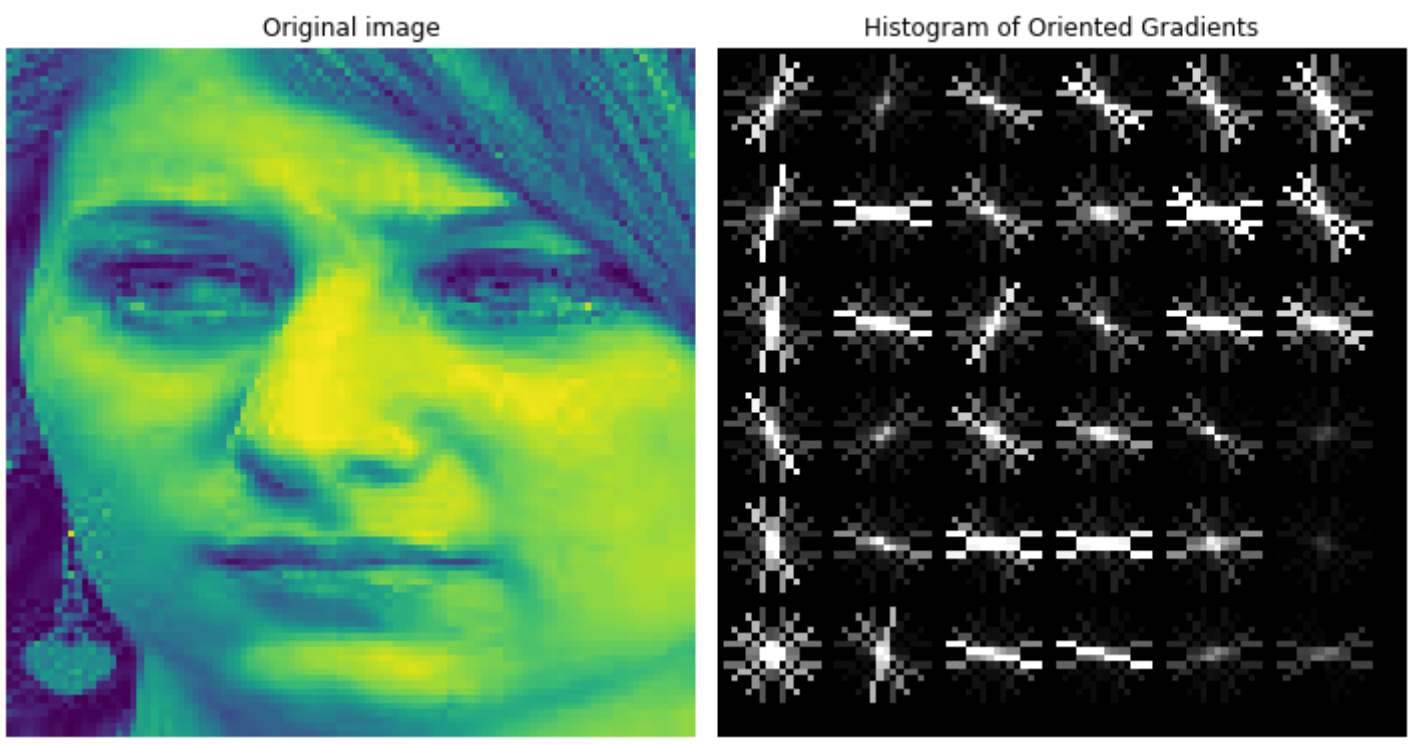
```
from skimage.feature import hog
from skimage import data, exposure
import matplotlib.pyplot as plt
```

Extract HOG descriptors for training data.

In [ ]:

```
hog_images = []
hog_descriptors = []

X_train_gray = [ color.rgb2gray(i) for i in X_train]

for i in range(len(X_train)):
    # Identify keypoints and extract descriptors with SIFT
    img_hog = img_as_ubyte(np.array(X_train_gray[i]))
```

```
    fet_des, hog_img = hog(img_hog, orientations=8, pixels_per_cell=(16, 16), cells_per_block=(4, 4), visualize=True, multichannel
    hog_descriptors.append(fet_des)
    hog_images.append(hog_img)


fig, ax = plt.subplots(1, 2, figsize=(10, 8), sharex=True, sharey=True)

ax[0].axis('off')
ax[0].imshow(X_train_gray[0])
ax[0].set_title('Original image')

# Rescale histogram for better display
hog_img_rescale = exposure.rescale_intensity(hog_images[0], in_range=(0, 10))

ax[1].axis('off')
ax[1].imshow(hog_img_rescale, cmap='gray')
ax[1].set_title('Histogram of Oriented Gradients')
fig.tight_layout()
plt.show()


## CODING REFERENCE: Code was taken from Lab Tutorial 06 of Computer Vision - IN3060/INM460 module
## and modified as per the requirement in this task.
## https://www.kaggle.com/code/manikg/training-svm-classifier-with-hog-features/notebook
```



In [ ]:
```
len(hog_descriptors)
```

Out[ ]:
```
12271
```

Create baseline SVM model that uses HOG descriptors.

In [ ]:
```
baseline_hogSVM = svm.SVC(kernel='rbf', class_weight='balanced')
```

In [ ]:
```
baseline_hogSVM_cvscores = cross_val_score(baseline_hogSVM, hog_descriptors, y_train, cv=5)
```

In [ ]:
```
baseline_hogSVM_cvscores
```

Out[ ]:
```
array([0.6395112 , 0.69274654, 0.65729421, 0.6202119 , 0.60676447])
```

In [ ]:
```
baseline_hogSVM_cvscores.mean()
```

Out[ ]:
```
0.6433056633087506
```

In [ ]:
```
baseline_hogSVM.fit(hog_descriptors, y_train)
```

Out[ ]:
```
SVC(class_weight='balanced')
```

In [ ]:
```
baseline_hogSVM.score(hog_descriptors, y_train)
```

Out[ ]:
```
0.8024610871159644
```

In [ ]:
```
baseline_hogSVM.get_params()
```

Out[ ]:
```
{'C': 1.0,
 'break_ties': False,
 'cache_size': 200,
 'class_weight': 'balanced',
 'coef0': 0.0,
```

```
 'decision_function_shape': 'ovr',
 'degree': 3,
 'gamma': 'scale',
 'kernel': 'rbf',
 'max_iter': -1,
 'probability': False,
 'random_state': None,
 'shrinking': True,
 'tol': 0.001,
 'verbose': False}
```

Perform hyperparameter tuning.

In [ ]:
```
hogSVM_improve1 = svm.SVC(class_weight='balanced')
```

In [ ]:
```
hogSVM_gs1 = GridSearchCV(estimator=hogSVM_improve1,
             param_grid={'C': [0.5, 1, 10, 100], 'kernel': ('linear', 'poly','rbf')})
```

In [ ]:
```
hogSVM_gs1.fit(hog_descriptors, y_train)
```

Out[ ]:
```
GridSearchCV(estimator=SVC(class_weight='balanced'),
             param_grid={'C': [0.5, 1, 10, 100],
                         'kernel': ('linear', 'poly', 'rbf')})
```

In [ ]:
```
hogSVM_gs1.best_params_
```

Out[ ]:
```
{'C': 10, 'kernel': 'rbf'}
```

In [ ]:
```
hogSVM_gs1.best_score_
```

Out[ ]:
```
0.6797336241424701
```

In [ ]:
```
hogSVM_imporve1 = svm.SVC(C=10, kernel='rbf', class_weight='balanced')
```

In [ ]:
```
hogSVM_imporve1_cvscores = cross_val_score(hogSVM_imporve1, hog_descriptors, y_train, cv=5)
```

In [ ]:
```
hogSVM_imporve1_cvscores
```

Out[ ]:
```
array([0.66843177, 0.71271394, 0.70252649, 0.66177669, 0.64710676])
```

In [ ]:
```
hogSVM_imporve1_cvscores.mean()
```

Out[ ]:
```
0.6785111302549394
```

In [ ]:
```
hogSVM_imporve1.fit(hog_descriptors, y_train)
```

Out[ ]:
```
SVC(C=10, class_weight='balanced')
```

In [ ]:
```
hogSVM_imporve1.score(hog_descriptors, y_train)
```

Out[ ]:
```
0.9914432401597262
```

Create best SVM model that uses HOG descriptors.

In [ ]:
```
best_hogSVM_model = svm.SVC(kernel='rbf', class_weight='balanced')
```

In [ ]:
```
best_hogSVM_model_cvscores = cross_val_score(best_hogSVM_model, hog_descriptors, y_train, cv=5)
```

In [ ]:
```
best_hogSVM_model_cvscores
```

Out[ ]:
```
array([0.6395112 , 0.69437653, 0.65892421, 0.6206194 , 0.60717196])
```

In [ ]:
```
best_hogSVM_model_cvscores.mean()
```

Out[ ]:
```
0.644120659233771
```

In [ ]:
```
t0 = time.time()
best_hogSVM_model.fit(hog_descriptors, y_train)
tt = time.time()
print(tt - t0)
```

```
153.30672788619995
```

In [ ]:
```
best_hogSVM_model.score(hog_descriptors, y_train)
```

Out[ ]:
```
0.8024610871159644
```

Save best SVM model that uses HOG descriptors.

In [ ]:
```
besthogsvmpath = 'drive/My Drive/Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/Models/best_hogSVM_model.pickle'

with open (besthogsvmpath, 'wb') as f:
    pickle.dump(best_hogSVM_model, f)
```

Create baseline MLP model that uses HOG descriptors.

In [ ]:
```
baseline_hogMLP = MLPClassifier(hidden_layer_sizes=(50,), max_iter=100, alpha=1e-4,
                    solver='sgd', verbose=True, random_state=1,
                    learning_rate_init=.1)
```

In [ ]:
```
baseline_hogMLP_cvscores = cross_val_score(baseline_hogMLP, hog_descriptors, y_train, cv=5)
```

```
Iteration 1, loss = 1.52813486
Iteration 2, loss = 1.19921585
Iteration 3, loss = 1.06760607
Iteration 4, loss = 1.01513078
Iteration 5, loss = 0.98387255
Iteration 6, loss = 0.96905944
Iteration 7, loss = 0.94943476
Iteration 8, loss = 0.94239081
Iteration 9, loss = 0.92788645
Iteration 10, loss = 0.91090613
Iteration 11, loss = 0.91117174
Iteration 12, loss = 0.89972986
Iteration 13, loss = 0.89652192
Iteration 14, loss = 0.87803317
Iteration 15, loss = 0.88181002
Iteration 16, loss = 0.85384648
Iteration 17, loss = 0.86421964
Iteration 18, loss = 0.84367620
Iteration 19, loss = 0.83199950
Iteration 20, loss = 0.82628562
Iteration 21, loss = 0.82457949
Iteration 22, loss = 0.81713299
Iteration 23, loss = 0.80733955
Iteration 24, loss = 0.80041216
Iteration 25, loss = 0.79548580
Iteration 26, loss = 0.78510070
Iteration 27, loss = 0.78007304
Iteration 28, loss = 0.77546138
Iteration 29, loss = 0.76870387
Iteration 30, loss = 0.75503830
Iteration 31, loss = 0.75036982
Iteration 32, loss = 0.74311459
Iteration 33, loss = 0.74022194
Iteration 34, loss = 0.72826642
Iteration 35, loss = 0.73492715
Iteration 36, loss = 0.72325782
Iteration 37, loss = 0.71571870
Iteration 38, loss = 0.72791836
Iteration 39, loss = 0.70747258
Iteration 40, loss = 0.70195450
Iteration 41, loss = 0.69170705
Iteration 42, loss = 0.68550047
Iteration 43, loss = 0.69519725
Iteration 44, loss = 0.67730870
Iteration 45, loss = 0.69996390
Iteration 46, loss = 0.66605691
Iteration 47, loss = 0.66422134
Iteration 48, loss = 0.66139287
Iteration 49, loss = 0.64489835
Iteration 50, loss = 0.66134859
Iteration 51, loss = 0.64434455
Iteration 52, loss = 0.64265614
Iteration 53, loss = 0.62934790
Iteration 54, loss = 0.63014069
Iteration 55, loss = 0.61620793
Iteration 56, loss = 0.62658875
Iteration 57, loss = 0.63828638
Iteration 58, loss = 0.62102276
Iteration 59, loss = 0.60481591
Iteration 60, loss = 0.59331461
Iteration 61, loss = 0.60475634
Iteration 62, loss = 0.58160269
Iteration 63, loss = 0.59130744
Iteration 64, loss = 0.59245092
Iteration 65, loss = 0.58480456
Iteration 66, loss = 0.56570773
Iteration 67, loss = 0.62437263
Iteration 68, loss = 0.56371885
Iteration 69, loss = 0.58517118
```

```
Iteration 70, loss = 0.58471697
Iteration 71, loss = 0.56589867
Iteration 72, loss = 0.55196837
Iteration 73, loss = 0.54066077
Iteration 74, loss = 0.54850166
Iteration 75, loss = 0.58721336
Iteration 76, loss = 0.68130356
Iteration 77, loss = 0.60347423
Iteration 78, loss = 0.60998520
Iteration 79, loss = 0.58850841
Iteration 80, loss = 0.58898875
Iteration 81, loss = 0.74449649
Iteration 82, loss = 0.61842520
Iteration 83, loss = 0.59479101
Iteration 84, loss = 0.58422457
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 1.54425899
Iteration 2, loss = 1.20224290
Iteration 3, loss = 1.09496376
Iteration 4, loss = 1.03137267
Iteration 5, loss = 1.00845208
Iteration 6, loss = 0.98406170
Iteration 7, loss = 0.96910296
Iteration 8, loss = 0.95322053
Iteration 9, loss = 0.94165629
Iteration 10, loss = 0.92511840
Iteration 11, loss = 0.92145143
Iteration 12, loss = 0.90895041
Iteration 13, loss = 0.90263774
Iteration 14, loss = 0.89188237
Iteration 15, loss = 0.87559103
Iteration 16, loss = 0.86686459
Iteration 17, loss = 0.86097540
Iteration 18, loss = 0.85048386
Iteration 19, loss = 0.85406738
Iteration 20, loss = 0.89659015
Iteration 21, loss = 0.83879354
Iteration 22, loss = 0.82935765
Iteration 23, loss = 0.81707138
Iteration 24, loss = 0.80649657
Iteration 25, loss = 0.80105439
Iteration 26, loss = 0.79227364
Iteration 27, loss = 0.78894288
Iteration 28, loss = 0.77628848
Iteration 29, loss = 0.77193242
Iteration 30, loss = 0.76749656
Iteration 31, loss = 0.75346865
Iteration 32, loss = 0.75472364
Iteration 33, loss = 0.74813293
Iteration 34, loss = 0.73315378
Iteration 35, loss = 0.72621192
Iteration 36, loss = 0.72673343
Iteration 37, loss = 0.72002050
Iteration 38, loss = 0.74105244
Iteration 39, loss = 0.71344208
Iteration 40, loss = 0.69795350
Iteration 41, loss = 0.70265150
Iteration 42, loss = 0.71800567
Iteration 43, loss = 0.69091918
Iteration 44, loss = 0.68929134
Iteration 45, loss = 0.69059027
Iteration 46, loss = 0.68441135
Iteration 47, loss = 0.68675373
Iteration 48, loss = 0.66506575
Iteration 49, loss = 0.65568787
Iteration 50, loss = 0.65540100
Iteration 51, loss = 0.65331087
Iteration 52, loss = 0.63487645
Iteration 53, loss = 0.62891665
Iteration 54, loss = 0.62886560
Iteration 55, loss = 0.63588083
Iteration 56, loss = 0.62101163
Iteration 57, loss = 0.61735242
Iteration 58, loss = 0.61953606
Iteration 59, loss = 0.59617409
Iteration 60, loss = 0.86327586
Iteration 61, loss = 0.68579397
Iteration 62, loss = 0.68183957
Iteration 63, loss = 0.65702014
Iteration 64, loss = 0.71118803
Iteration 65, loss = 0.66567976
Iteration 66, loss = 0.66524851
Iteration 67, loss = 0.66453539
Iteration 68, loss = 0.62945883
Iteration 69, loss = 0.62059106
Iteration 70, loss = 0.63726950
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 1.53040815
Iteration 2, loss = 1.19236902
Iteration 3, loss = 1.06676381
Iteration 4, loss = 1.02449129
Iteration 5, loss = 0.99375766
```

```
Iteration 6, loss = 0.97278465
Iteration 7, loss = 0.96569614
Iteration 8, loss = 0.94138573
Iteration 9, loss = 0.92970475
Iteration 10, loss = 0.91995449
Iteration 11, loss = 0.91050242
Iteration 12, loss = 0.89526120
Iteration 13, loss = 0.89354098
Iteration 14, loss = 0.87725917
Iteration 15, loss = 0.87444056
Iteration 16, loss = 0.86733589
Iteration 17, loss = 0.85727998
Iteration 18, loss = 0.84620326
Iteration 19, loss = 0.83459028
Iteration 20, loss = 0.83633297
Iteration 21, loss = 0.81942733
Iteration 22, loss = 0.80284878
Iteration 23, loss = 0.80094388
Iteration 24, loss = 0.78795547
Iteration 25, loss = 0.78356066
Iteration 26, loss = 0.78057209
Iteration 27, loss = 0.77174710
Iteration 28, loss = 0.76182547
Iteration 29, loss = 0.75381958
Iteration 30, loss = 0.75116352
Iteration 31, loss = 0.75913557
Iteration 32, loss = 0.74047108
Iteration 33, loss = 0.73721341
Iteration 34, loss = 0.72614687
Iteration 35, loss = 0.71524406
Iteration 36, loss = 0.70978984
Iteration 37, loss = 0.70887347
Iteration 38, loss = 0.69756817
Iteration 39, loss = 0.70444457
Iteration 40, loss = 0.67838139
Iteration 41, loss = 0.67614764
Iteration 42, loss = 0.68247001
Iteration 43, loss = 0.65581666
Iteration 44, loss = 0.65866028
Iteration 45, loss = 0.64430123
Iteration 46, loss = 0.65897114
Iteration 47, loss = 0.64795410
Iteration 48, loss = 0.62707835
Iteration 49, loss = 0.62405366
Iteration 50, loss = 0.63738344
Iteration 51, loss = 0.61072758
Iteration 52, loss = 0.61621941
Iteration 53, loss = 0.61261164
Iteration 54, loss = 0.60805160
Iteration 55, loss = 0.59307089
Iteration 56, loss = 0.59273804
Iteration 57, loss = 0.57855482
Iteration 58, loss = 0.57381708
Iteration 59, loss = 0.59551322
Iteration 60, loss = 0.58990937
Iteration 61, loss = 0.55091207
Iteration 62, loss = 0.56876018
Iteration 63, loss = 0.56001756
Iteration 64, loss = 0.55714479
Iteration 65, loss = 0.54602012
Iteration 66, loss = 0.53762664
Iteration 67, loss = 0.54391194
Iteration 68, loss = 0.52294577
Iteration 69, loss = 0.52389053
Iteration 70, loss = 0.51170126
Iteration 71, loss = 0.51258617
Iteration 72, loss = 0.50797519
Iteration 73, loss = 0.49317942
Iteration 74, loss = 0.48567151
Iteration 75, loss = 0.49351746
Iteration 76, loss = 0.48697587
Iteration 77, loss = 0.48912561
Iteration 78, loss = 0.47883711
Iteration 79, loss = 0.46775831
Iteration 80, loss = 0.47386395
Iteration 81, loss = 0.45865904
Iteration 82, loss = 0.45819929
Iteration 83, loss = 0.46801211
Iteration 84, loss = 0.46033260
Iteration 85, loss = 0.45409444
Iteration 86, loss = 0.45221846
Iteration 87, loss = 0.43984513
Iteration 88, loss = 0.42215282
Iteration 89, loss = 0.47008992
Iteration 90, loss = 0.43682394
Iteration 91, loss = 0.46674776
Iteration 92, loss = 0.45126976
Iteration 93, loss = 0.40907424
Iteration 94, loss = 0.44008320
Iteration 95, loss = 0.41860150
Iteration 96, loss = 0.41945066
Iteration 97, loss = 0.38526692
```

```
Iteration 98, loss = 0.39119673
Iteration 99, loss = 0.38271531
Iteration 100, loss = 0.39396660
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (100) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
Iteration 1, loss = 1.53388857
Iteration 2, loss = 1.17987449
Iteration 3, loss = 1.06857904
Iteration 4, loss = 1.02156172
Iteration 5, loss = 0.98040945
Iteration 6, loss = 0.95924706
Iteration 7, loss = 0.94940098
Iteration 8, loss = 0.93053525
Iteration 9, loss = 0.92302183
Iteration 10, loss = 0.90681429
Iteration 11, loss = 0.90369717
Iteration 12, loss = 0.88638464
Iteration 13, loss = 0.90413208
Iteration 14, loss = 0.86718933
Iteration 15, loss = 0.86064897
Iteration 16, loss = 0.85286563
Iteration 17, loss = 0.84283107
Iteration 18, loss = 0.83572505
Iteration 19, loss = 0.84471148
Iteration 20, loss = 0.82560652
Iteration 21, loss = 0.81792950
Iteration 22, loss = 0.79613914
Iteration 23, loss = 0.79291867
Iteration 24, loss = 0.79478623
Iteration 25, loss = 0.77555092
Iteration 26, loss = 0.77122197
Iteration 27, loss = 0.76502434
Iteration 28, loss = 0.75804167
Iteration 29, loss = 0.75089428
Iteration 30, loss = 0.76050947
Iteration 31, loss = 0.73923543
Iteration 32, loss = 0.72807632
Iteration 33, loss = 0.72798946
Iteration 34, loss = 0.73471540
Iteration 35, loss = 0.70857912
Iteration 36, loss = 0.69733986
Iteration 37, loss = 0.71656419
Iteration 38, loss = 0.68327825
Iteration 39, loss = 0.68095680
Iteration 40, loss = 0.66633491
Iteration 41, loss = 0.66866657
Iteration 42, loss = 0.67139071
Iteration 43, loss = 0.64773453
Iteration 44, loss = 0.64295702
Iteration 45, loss = 0.64008628
Iteration 46, loss = 0.64979850
Iteration 47, loss = 0.63350617
Iteration 48, loss = 0.64180530
Iteration 49, loss = 0.61617191
Iteration 50, loss = 0.61262232
Iteration 51, loss = 0.61494554
Iteration 52, loss = 0.61353537
Iteration 53, loss = 0.59050599
Iteration 54, loss = 0.59592254
Iteration 55, loss = 0.57921739
Iteration 56, loss = 0.58515295
Iteration 57, loss = 0.57214337
Iteration 58, loss = 0.57970726
Iteration 59, loss = 0.56093559
Iteration 60, loss = 0.56144114
Iteration 61, loss = 0.56480678
Iteration 62, loss = 0.57655415
Iteration 63, loss = 0.54250494
Iteration 64, loss = 0.53981872
Iteration 65, loss = 0.56744657
Iteration 66, loss = 0.52554225
Iteration 67, loss = 0.53651441
Iteration 68, loss = 0.53769013
Iteration 69, loss = 0.53946224
Iteration 70, loss = 0.54262322
Iteration 71, loss = 0.54723337
Iteration 72, loss = 0.50326041
Iteration 73, loss = 0.49722788
Iteration 74, loss = 0.50286915
Iteration 75, loss = 0.49651315
Iteration 76, loss = 0.49484286
Iteration 77, loss = 0.48530508
Iteration 78, loss = 0.48244849
Iteration 79, loss = 0.47904356
Iteration 80, loss = 0.49564705
Iteration 81, loss = 0.50517108
Iteration 82, loss = 0.47499723
Iteration 83, loss = 0.45775617
Iteration 84, loss = 0.45079594
Iteration 85, loss = 0.45692328
Iteration 86, loss = 0.45152970
```

```
Iteration 87, loss = 0.45484583
Iteration 88, loss = 0.43406931
Iteration 89, loss = 0.44418976
Iteration 90, loss = 0.43501292
Iteration 91, loss = 0.48557705
Iteration 92, loss = 0.68808441
Iteration 93, loss = 0.53133628
Iteration 94, loss = 0.51238773
Iteration 95, loss = 0.53265039
Iteration 96, loss = 0.49376378
Iteration 97, loss = 0.49404880
Iteration 98, loss = 0.50169542
Iteration 99, loss = 0.50405633
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 1.52351259
Iteration 2, loss = 1.17454869
Iteration 3, loss = 1.05284497
Iteration 4, loss = 0.99809964
Iteration 5, loss = 0.96972416
Iteration 6, loss = 0.94461345
Iteration 7, loss = 0.93889599
Iteration 8, loss = 0.91918564
Iteration 9, loss = 0.90984383
Iteration 10, loss = 0.89026401
Iteration 11, loss = 0.88476180
Iteration 12, loss = 0.87138205
Iteration 13, loss = 0.87068450
Iteration 14, loss = 0.84916590
Iteration 15, loss = 0.85279828
Iteration 16, loss = 0.83807520
Iteration 17, loss = 0.82113751
Iteration 18, loss = 0.81892259
Iteration 19, loss = 0.81625247
Iteration 20, loss = 0.80255343
Iteration 21, loss = 0.79191966
Iteration 22, loss = 0.77444120
Iteration 23, loss = 0.76669557
Iteration 24, loss = 0.76115242
Iteration 25, loss = 0.75168824
Iteration 26, loss = 0.74205778
Iteration 27, loss = 0.74408126
Iteration 28, loss = 0.73742004
Iteration 29, loss = 0.72542851
Iteration 30, loss = 0.71823057
Iteration 31, loss = 0.71788784
Iteration 32, loss = 0.70404989
Iteration 33, loss = 0.69943686
Iteration 34, loss = 0.68694293
Iteration 35, loss = 0.68601271
Iteration 36, loss = 0.67205957
Iteration 37, loss = 0.68802866
Iteration 38, loss = 0.66256964
Iteration 39, loss = 0.65575473
Iteration 40, loss = 0.64616415
Iteration 41, loss = 0.64357530
Iteration 42, loss = 0.63640628
Iteration 43, loss = 0.62785088
Iteration 44, loss = 0.62107076
Iteration 45, loss = 0.61777707
Iteration 46, loss = 0.62131398
Iteration 47, loss = 0.60637958
Iteration 48, loss = 0.59605443
Iteration 49, loss = 0.61119511
Iteration 50, loss = 0.59512930
Iteration 51, loss = 0.57717755
Iteration 52, loss = 0.59035824
Iteration 53, loss = 0.58336193
Iteration 54, loss = 0.56422818
Iteration 55, loss = 0.56586687
Iteration 56, loss = 0.55143566
Iteration 57, loss = 0.54648693
Iteration 58, loss = 0.56128045
Iteration 59, loss = 0.53568240
Iteration 60, loss = 0.56099938
Iteration 61, loss = 0.52957298
Iteration 62, loss = 0.53427102
Iteration 63, loss = 0.52892270
Iteration 64, loss = 0.50752872
Iteration 65, loss = 0.50444903
Iteration 66, loss = 0.50579628
Iteration 67, loss = 0.51794857
Iteration 68, loss = 0.50841368
Iteration 69, loss = 0.49415141
Iteration 70, loss = 0.48619578
Iteration 71, loss = 0.48946290
Iteration 72, loss = 0.52066690
Iteration 73, loss = 0.48621707
Iteration 74, loss = 0.47201295
Iteration 75, loss = 0.47733040
Iteration 76, loss = 0.47157743
Iteration 77, loss = 0.46323434
Iteration 78, loss = 0.45593855
```

```
Iteration 79, loss = 0.45205237
Iteration 80, loss = 0.44969880
Iteration 81, loss = 0.43814456
Iteration 82, loss = 0.43436477
Iteration 83, loss = 0.43386001
Iteration 84, loss = 0.43164178
Iteration 85, loss = 0.44975837
Iteration 86, loss = 0.47202636
Iteration 87, loss = 0.42850795
Iteration 88, loss = 0.40615078
Iteration 89, loss = 0.40232204
Iteration 90, loss = 0.39930873
Iteration 91, loss = 0.51424341
Iteration 92, loss = 0.39844414
Iteration 93, loss = 0.39595477
Iteration 94, loss = 0.39378207
Iteration 95, loss = 0.39170021
Iteration 96, loss = 0.38836342
Iteration 97, loss = 0.38480584
Iteration 98, loss = 0.36246191
Iteration 99, loss = 0.38304808
Iteration 100, loss = 0.35528494
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (100) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

In [ ]:

```
baseline_hogMLP_cvscores
```

Out[ ]:

```
array([0.62729124, 0.61409943, 0.58231459, 0.599022  , 0.58272209])
```

In [ ]:

```
baseline_hogMLP_cvscores.mean()
```

Out[ ]:

```
0.6010898703143959
```

In [ ]:

```
baseline_hogMLP.fit(hog_descriptors, y_train)
```

```
Iteration 1, loss = 1.48096560
Iteration 2, loss = 1.13623880
Iteration 3, loss = 1.03452115
Iteration 4, loss = 0.99260436
Iteration 5, loss = 0.97104326
Iteration 6, loss = 0.95667011
Iteration 7, loss = 0.93945609
Iteration 8, loss = 0.92579354
Iteration 9, loss = 0.91839740
Iteration 10, loss = 0.90428701
Iteration 11, loss = 0.89280959
Iteration 12, loss = 0.88221779
Iteration 13, loss = 0.87551000
Iteration 14, loss = 0.86050821
Iteration 15, loss = 0.85822813
Iteration 16, loss = 0.84103007
Iteration 17, loss = 0.83381380
Iteration 18, loss = 0.82591770
Iteration 19, loss = 0.81904386
Iteration 20, loss = 0.81242549
Iteration 21, loss = 0.80359569
Iteration 22, loss = 0.79752688
Iteration 23, loss = 0.78727094
Iteration 24, loss = 0.77988874
Iteration 25, loss = 0.77130303
Iteration 26, loss = 0.75955425
Iteration 27, loss = 0.75607860
Iteration 28, loss = 0.75144072
Iteration 29, loss = 0.73945648
Iteration 30, loss = 0.73662818
Iteration 31, loss = 0.72350615
Iteration 32, loss = 0.71430677
Iteration 33, loss = 0.71143150
Iteration 34, loss = 0.70887128
Iteration 35, loss = 0.69432689
Iteration 36, loss = 0.69321145
Iteration 37, loss = 0.68289245
Iteration 38, loss = 0.68216374
Iteration 39, loss = 0.67110188
Iteration 40, loss = 0.67199179
Iteration 41, loss = 0.66471071
Iteration 42, loss = 0.65354043
Iteration 43, loss = 0.64827774
Iteration 44, loss = 0.63653195
Iteration 45, loss = 0.63364598
Iteration 46, loss = 0.63419051
Iteration 47, loss = 0.63200190
Iteration 48, loss = 0.62194291
Iteration 49, loss = 0.61708878
Iteration 50, loss = 0.60535701
Iteration 51, loss = 0.60136622
Iteration 52, loss = 0.59682135
Iteration 53, loss = 0.58859340
```

```
Iteration 54, loss = 0.58452232
Iteration 55, loss = 0.58585069
Iteration 56, loss = 0.58237856
Iteration 57, loss = 0.57448622
Iteration 58, loss = 0.57372838
Iteration 59, loss = 0.57684142
Iteration 60, loss = 0.55902605
Iteration 61, loss = 0.55730049
Iteration 62, loss = 0.54756986
Iteration 63, loss = 0.54880330
Iteration 64, loss = 0.54819119
Iteration 65, loss = 0.53439186
Iteration 66, loss = 0.53875351
Iteration 67, loss = 0.53881537
Iteration 68, loss = 0.53386196
Iteration 69, loss = 0.52668930
Iteration 70, loss = 0.52216860
Iteration 71, loss = 0.52918845
Iteration 72, loss = 0.51913129
Iteration 73, loss = 0.50068871
Iteration 74, loss = 0.51098233
Iteration 75, loss = 0.49836829
Iteration 76, loss = 0.49573949
Iteration 77, loss = 0.50530416
Iteration 78, loss = 0.49051982
Iteration 79, loss = 0.48501053
Iteration 80, loss = 0.48354273
Iteration 81, loss = 0.48983535
Iteration 82, loss = 0.47073860
Iteration 83, loss = 0.46020875
Iteration 84, loss = 0.47769969
Iteration 85, loss = 0.47097389
Iteration 86, loss = 0.45193951
Iteration 87, loss = 0.45191547
Iteration 88, loss = 0.46010444
Iteration 89, loss = 0.45153452
Iteration 90, loss = 0.45304607
Iteration 91, loss = 0.44371767
Iteration 92, loss = 0.43627252
Iteration 93, loss = 0.44426194
Iteration 94, loss = 0.43843055
Iteration 95, loss = 0.43308345
Iteration 96, loss = 0.42353281
Iteration 97, loss = 0.42191831
Iteration 98, loss = 0.42822741
Iteration 99, loss = 0.41101690
Iteration 100, loss = 0.42002154
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (100) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:

```
MLPClassifier(hidden_layer_sizes=(50,), learning_rate_init=0.1, max_iter=100,
              random_state=1, solver='sgd', verbose=True)
```

In [ ]:

```
baseline_hogMLP.score(hog_descriptors, y_train)
```

Out[ ]:

```
0.8502159563197783
```

Perform hyperparameter tuning.

In [ ]:

```
hogMLP_improve1 = MLPClassifier(max_iter=200, alpha=1e-4,
                    solver='sgd', random_state=1,
                    learning_rate_init=.1)
```

In [ ]:

```
hotMLP_gs1 = GridSearchCV(estimator=hogMLP_improve1,
             param_grid={'hidden_layer_sizes': [(50,), (100,), (50,50), (50,100)]})
```

In [ ]:

```
hotMLP_gs1.fit(hog_descriptors, y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:

```
GridSearchCV(estimator=MLPClassifier(learning_rate_init=0.1, random_state=1,
```

```
                                solver='sgd'),
             param_grid={'hidden_layer_sizes': [(50,), (100,), (50, 50),
                                                (50, 100)]})
```

In [ ]:
```
hotMLP_gs1.best_params_
```

Out[ ]:
```
{'hidden_layer_sizes': (100,)}
```

In [ ]:
```
hotMLP_gs1.best_score_
```

Out[ ]:
```
0.6287177010143463
```

In [ ]:
```
hogMLP_improve2 = MLPClassifier(hidden_layer_sizes=(100,), alpha=1e-4,
                    solver='sgd', random_state=1,
                    learning_rate_init=.1)
```

In [ ]:
```
hogMLP_gs2 = GridSearchCV(estimator=hogMLP_improve2,
             param_grid={'max_iter': [200, 300, 400, 500]})
```

In [ ]:
```
hogMLP_gs2.fit(hog_descriptors, y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:
```
GridSearchCV(estimator=MLPClassifier(learning_rate_init=0.1, random_state=1,
                                     solver='sgd'),
             param_grid={'max_iter': [200, 300, 400, 500]})
```

In [ ]:
```
hogMLP_gs2.best_params_
```

Out[ ]:
```
{'max_iter': 200}
```

In [ ]:
```
hogMLP_gs2.best_score_
```

Out[ ]:
```
0.6287177010143463
```

In [ ]:
```
hogMLP_improve3 = MLPClassifier(hidden_layer_sizes=(100,), max_iter=200,
                                alpha=1e-4,random_state=1,
                    learning_rate_init=.1)
```

In [ ]:
```
hogMLP_gs3 = GridSearchCV(estimator=hogMLP_improve3,
             param_grid={'activation': ['identity', 'tanh', 'relu'], 'solver': ['sgd', 'adam']})
```

In [ ]:
```
hogMLP_gs3.fit(hog_descriptors, y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
```

```
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:

```
GridSearchCV(estimator=MLPClassifier(learning_rate_init=0.1, random_state=1,
                                     solver='sgd'),
             param_grid={'activation': ['identity', 'tanh', 'relu'],
                         'solver': ['sgd', 'adam']})
```

In [ ]:

```
hogMLP_gs3.best_params_
```

Out[ ]:

```
{'activation': 'relu', 'solver': 'sgd'}
```

In [ ]:

```
hogMLP_gs3.best_score_
```

Out[ ]:

```
0.6287177010143463
```

In [ ]:

```
hogMLP_improve4 = MLPClassifier(hidden_layer_sizes=(100,), activation='relu', max_iter=200,alpha=1e-4,
                    solver='sgd', random_state=1)
```

In [ ]:

```
hogMLP_gs4 = GridSearchCV(estimator=hogMLP_improve4,
             param_grid={'learning_rate_init': [0.001, 0.01, 0.1]})
```

In [ ]:

```
hogMLP_gs4.fit(hog_descriptors, y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:

```
GridSearchCV(estimator=MLPClassifier(random_state=1, solver='sgd'),
             param_grid={'learning_rate_init': [0.001, 0.01, 0.1]})
```

In [ ]:

```
hogMLP_gs4.best_params_
```

Out[ ]:
```
{'learning_rate_init': 0.01}
```

In [ ]:
```
hogMLP_gs4.best_score_
```

Out[ ]:
```
0.6556105746966174
```

In [ ]:
```
hogMLP_improve5 = MLPClassifier(hidden_layer_sizes=(100,), activation='relu', max_iter=600,alpha=1e-4,
                    solver='sgd', learning_rate_init=0.01, random_state=1)
```

In [ ]:
```
hogMLP_improve5_cvscores = cross_val_score(hogMLP_improve5, hog_descriptors, y_train, cv=5)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (600) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (600) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (600) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (600) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (600) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

In [ ]:
```
hogMLP_improve5_cvscores
```

Out[ ]:
```
array([0.65824847, 0.68296659, 0.66462918, 0.64466178, 0.62754686])
```

In [ ]:
```
hogMLP_improve5_cvscores.mean()
```

Out[ ]:
```
0.6556105746966174
```

In [ ]:
```
hogMLP_improve5.fit(hog_descriptors, y_train,)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

Out[ ]:
```
MLPClassifier(learning_rate_init=0.01, random_state=1, solver='sgd')
```

In [ ]:
```
hogMLP_improve5.score(hog_descriptors, y_train,)
```

Out[ ]:
```
0.769945399722924
```

Create best MLP model that uses HOG descriptors.

In [ ]:
```
best_hogMLP_model = MLPClassifier(hidden_layer_sizes=(100,), activation='relu', max_iter=200,alpha=1e-4,
                    solver='sgd', learning_rate_init=0.01, verbose=True, random_state=1)
```

In [ ]:
```
best_hogMLP_model_cvscores = cross_val_score(best_hogMLP_model, hog_descriptors, y_train, cv=5)
```

```
Iteration 1, loss = 1.67385090
Iteration 2, loss = 1.56700147
Iteration 3, loss = 1.49292645
Iteration 4, loss = 1.41381672
Iteration 5, loss = 1.34431089
Iteration 6, loss = 1.28943973
Iteration 7, loss = 1.24768260
Iteration 8, loss = 1.21151399
Iteration 9, loss = 1.18162149
Iteration 10, loss = 1.15529326
Iteration 11, loss = 1.13104897
Iteration 12, loss = 1.11044091
Iteration 13, loss = 1.09344674
Iteration 14, loss = 1.07702241
Iteration 15, loss = 1.06291205
Iteration 16, loss = 1.04975792
Iteration 17, loss = 1.03958215
Iteration 18, loss = 1.03094066
Iteration 19, loss = 1.02131085
Iteration 20, loss = 1.01335991
Iteration 21, loss = 1.00475339
Iteration 22, loss = 0.99928651
Iteration 23, loss = 0.99260071
Iteration 24, loss = 0.98731188
Iteration 25, loss = 0.98213504
Iteration 26, loss = 0.97679565
```

```
Iteration 27, loss = 0.97372935
Iteration 28, loss = 0.96930031
Iteration 29, loss = 0.96594457
Iteration 30, loss = 0.95988457
Iteration 31, loss = 0.95745682
Iteration 32, loss = 0.95541850
Iteration 33, loss = 0.95233685
Iteration 34, loss = 0.94909487
Iteration 35, loss = 0.94624848
Iteration 36, loss = 0.94359159
Iteration 37, loss = 0.93956059
Iteration 38, loss = 0.93821405
Iteration 39, loss = 0.93463311
Iteration 40, loss = 0.93200041
Iteration 41, loss = 0.92873325
Iteration 42, loss = 0.92688423
Iteration 43, loss = 0.92558859
Iteration 44, loss = 0.92173856
Iteration 45, loss = 0.92251927
Iteration 46, loss = 0.91802386
Iteration 47, loss = 0.91880622
Iteration 48, loss = 0.91344139
Iteration 49, loss = 0.91281672
Iteration 50, loss = 0.91107107
Iteration 51, loss = 0.90898768
Iteration 52, loss = 0.90986116
Iteration 53, loss = 0.90653035
Iteration 54, loss = 0.90418640
Iteration 55, loss = 0.90320411
Iteration 56, loss = 0.90027371
Iteration 57, loss = 0.89779114
Iteration 58, loss = 0.89649023
Iteration 59, loss = 0.89610805
Iteration 60, loss = 0.89440750
Iteration 61, loss = 0.89117200
Iteration 62, loss = 0.89089226
Iteration 63, loss = 0.88825069
Iteration 64, loss = 0.88731903
Iteration 65, loss = 0.88499387
Iteration 66, loss = 0.88372972
Iteration 67, loss = 0.88182036
Iteration 68, loss = 0.88127947
Iteration 69, loss = 0.87987319
Iteration 70, loss = 0.87916236
Iteration 71, loss = 0.87762139
Iteration 72, loss = 0.87460354
Iteration 73, loss = 0.87686582
Iteration 74, loss = 0.87300677
Iteration 75, loss = 0.87139781
Iteration 76, loss = 0.86891314
Iteration 77, loss = 0.86905564
Iteration 78, loss = 0.86769486
Iteration 79, loss = 0.86712892
Iteration 80, loss = 0.86484463
Iteration 81, loss = 0.86263077
Iteration 82, loss = 0.86156867
Iteration 83, loss = 0.86081983
Iteration 84, loss = 0.85766119
Iteration 85, loss = 0.85734751
Iteration 86, loss = 0.85536950
Iteration 87, loss = 0.85533509
Iteration 88, loss = 0.85187859
Iteration 89, loss = 0.85232897
Iteration 90, loss = 0.85438851
Iteration 91, loss = 0.84887913
Iteration 92, loss = 0.84839709
Iteration 93, loss = 0.84760972
Iteration 94, loss = 0.84563199
Iteration 95, loss = 0.84333395
Iteration 96, loss = 0.84214983
Iteration 97, loss = 0.84214892
Iteration 98, loss = 0.84074620
Iteration 99, loss = 0.83856479
Iteration 100, loss = 0.83828619
Iteration 101, loss = 0.83768124
Iteration 102, loss = 0.83391468
Iteration 103, loss = 0.83296452
Iteration 104, loss = 0.83048023
Iteration 105, loss = 0.83137878
Iteration 106, loss = 0.83062498
Iteration 107, loss = 0.82799289
Iteration 108, loss = 0.82683002
Iteration 109, loss = 0.82649740
Iteration 110, loss = 0.82332253
Iteration 111, loss = 0.82381538
Iteration 112, loss = 0.82142814
Iteration 113, loss = 0.82224984
Iteration 114, loss = 0.81747138
Iteration 115, loss = 0.81961298
Iteration 116, loss = 0.81805710
Iteration 117, loss = 0.81469145
Iteration 118, loss = 0.81243134
```

```
Iteration 119, loss = 0.81177815
Iteration 120, loss = 0.81144826
Iteration 121, loss = 0.81063813
Iteration 122, loss = 0.80869717
Iteration 123, loss = 0.80515873
Iteration 124, loss = 0.80434828
Iteration 125, loss = 0.80410539
Iteration 126, loss = 0.80384103
Iteration 127, loss = 0.80232407
Iteration 128, loss = 0.80028203
Iteration 129, loss = 0.79721020
Iteration 130, loss = 0.79636719
Iteration 131, loss = 0.79392381
Iteration 132, loss = 0.79349951
Iteration 133, loss = 0.79320498
Iteration 134, loss = 0.79095639
Iteration 135, loss = 0.79130389
Iteration 136, loss = 0.78979856
Iteration 137, loss = 0.78603939
Iteration 138, loss = 0.78599827
Iteration 139, loss = 0.78519513
Iteration 140, loss = 0.78272771
Iteration 141, loss = 0.77988300
Iteration 142, loss = 0.78007678
Iteration 143, loss = 0.77911143
Iteration 144, loss = 0.77799977
Iteration 145, loss = 0.77769669
Iteration 146, loss = 0.77688613
Iteration 147, loss = 0.77231953
Iteration 148, loss = 0.77115087
Iteration 149, loss = 0.77127640
Iteration 150, loss = 0.76969953
Iteration 151, loss = 0.76677869
Iteration 152, loss = 0.76498071
Iteration 153, loss = 0.76408955
Iteration 154, loss = 0.76592874
Iteration 155, loss = 0.76186859
Iteration 156, loss = 0.76021707
Iteration 157, loss = 0.75965360
Iteration 158, loss = 0.75743335
Iteration 159, loss = 0.75464661
Iteration 160, loss = 0.75500803
Iteration 161, loss = 0.75075493
Iteration 162, loss = 0.75100900
Iteration 163, loss = 0.74937300
Iteration 164, loss = 0.74768576
Iteration 165, loss = 0.74870673
Iteration 166, loss = 0.74570147
Iteration 167, loss = 0.74520113
Iteration 168, loss = 0.74459798
Iteration 169, loss = 0.74376815
Iteration 170, loss = 0.74046202
Iteration 171, loss = 0.74175836
Iteration 172, loss = 0.73776084
Iteration 173, loss = 0.73734688
Iteration 174, loss = 0.73370975
Iteration 175, loss = 0.73342648
Iteration 176, loss = 0.73327306
Iteration 177, loss = 0.73080144
Iteration 178, loss = 0.73209472
Iteration 179, loss = 0.72663415
Iteration 180, loss = 0.72648495
Iteration 181, loss = 0.72595082
Iteration 182, loss = 0.72329848
Iteration 183, loss = 0.72169169
Iteration 184, loss = 0.72063624
Iteration 185, loss = 0.72007692
Iteration 186, loss = 0.71798368
Iteration 187, loss = 0.71996968
Iteration 188, loss = 0.71595749
Iteration 189, loss = 0.71339254
Iteration 190, loss = 0.71203161
Iteration 191, loss = 0.71044610
Iteration 192, loss = 0.71103188
Iteration 193, loss = 0.70791145
Iteration 194, loss = 0.70615364
Iteration 195, loss = 0.70392818
Iteration 196, loss = 0.70357188
Iteration 197, loss = 0.70073238
Iteration 198, loss = 0.70160485
Iteration 199, loss = 0.69758246
Iteration 200, loss = 0.69674153
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
Iteration 1, loss = 1.67447401
Iteration 2, loss = 1.57069877
Iteration 3, loss = 1.50045755
Iteration 4, loss = 1.42322533
Iteration 5, loss = 1.35752834
Iteration 6, loss = 1.30220776
Iteration 7, loss = 1.26005216
```

```
Iteration 8, loss = 1.22357811
Iteration 9, loss = 1.19367912
Iteration 10, loss = 1.16851442
Iteration 11, loss = 1.14438475
Iteration 12, loss = 1.12419037
Iteration 13, loss = 1.10608556
Iteration 14, loss = 1.09031720
Iteration 15, loss = 1.07650201
Iteration 16, loss = 1.06447801
Iteration 17, loss = 1.05501930
Iteration 18, loss = 1.04335875
Iteration 19, loss = 1.03486966
Iteration 20, loss = 1.02785827
Iteration 21, loss = 1.02064899
Iteration 22, loss = 1.01407271
Iteration 23, loss = 1.00802868
Iteration 24, loss = 1.00220986
Iteration 25, loss = 0.99811756
Iteration 26, loss = 0.99310232
Iteration 27, loss = 0.98777671
Iteration 28, loss = 0.98498058
Iteration 29, loss = 0.97933209
Iteration 30, loss = 0.97768658
Iteration 31, loss = 0.97171459
Iteration 32, loss = 0.96903546
Iteration 33, loss = 0.96552344
Iteration 34, loss = 0.96356840
Iteration 35, loss = 0.95976303
Iteration 36, loss = 0.95740187
Iteration 37, loss = 0.95398739
Iteration 38, loss = 0.95169341
Iteration 39, loss = 0.94900875
Iteration 40, loss = 0.94783442
Iteration 41, loss = 0.94381009
Iteration 42, loss = 0.94370614
Iteration 43, loss = 0.94041581
Iteration 44, loss = 0.93956803
Iteration 45, loss = 0.93588326
Iteration 46, loss = 0.93452041
Iteration 47, loss = 0.93316627
Iteration 48, loss = 0.92895178
Iteration 49, loss = 0.92933905
Iteration 50, loss = 0.92511895
Iteration 51, loss = 0.92388807
Iteration 52, loss = 0.92233603
Iteration 53, loss = 0.92142974
Iteration 54, loss = 0.91844791
Iteration 55, loss = 0.91730704
Iteration 56, loss = 0.91644291
Iteration 57, loss = 0.91676695
Iteration 58, loss = 0.91212642
Iteration 59, loss = 0.91170172
Iteration 60, loss = 0.90848947
Iteration 61, loss = 0.90712638
Iteration 62, loss = 0.90574459
Iteration 63, loss = 0.90553018
Iteration 64, loss = 0.90265494
Iteration 65, loss = 0.90204595
Iteration 66, loss = 0.90128630
Iteration 67, loss = 0.89832830
Iteration 68, loss = 0.89639401
Iteration 69, loss = 0.89543665
Iteration 70, loss = 0.89467470
Iteration 71, loss = 0.89363493
Iteration 72, loss = 0.89202928
Iteration 73, loss = 0.89032891
Iteration 74, loss = 0.89048859
Iteration 75, loss = 0.88741926
Iteration 76, loss = 0.88863692
Iteration 77, loss = 0.88547965
Iteration 78, loss = 0.88394370
Iteration 79, loss = 0.88259845
Iteration 80, loss = 0.88104063
Iteration 81, loss = 0.87983361
Iteration 82, loss = 0.87859043
Iteration 83, loss = 0.87747086
Iteration 84, loss = 0.87567671
Iteration 85, loss = 0.87392934
Iteration 86, loss = 0.87419918
Iteration 87, loss = 0.87105890
Iteration 88, loss = 0.86971801
Iteration 89, loss = 0.86969431
Iteration 90, loss = 0.86910678
Iteration 91, loss = 0.86567311
Iteration 92, loss = 0.86509825
Iteration 93, loss = 0.86447902
Iteration 94, loss = 0.86395475
Iteration 95, loss = 0.86177944
Iteration 96, loss = 0.86077005
Iteration 97, loss = 0.86058697
Iteration 98, loss = 0.85770980
Iteration 99, loss = 0.85715178
```

```
Iteration 100, loss = 0.85459163
Iteration 101, loss = 0.85405402
Iteration 102, loss = 0.85221634
Iteration 103, loss = 0.85206422
Iteration 104, loss = 0.85018473
Iteration 105, loss = 0.84974830
Iteration 106, loss = 0.84681938
Iteration 107, loss = 0.84531433
Iteration 108, loss = 0.84392786
Iteration 109, loss = 0.84453125
Iteration 110, loss = 0.84266700
Iteration 111, loss = 0.84112753
Iteration 112, loss = 0.84061625
Iteration 113, loss = 0.83771368
Iteration 114, loss = 0.83781655
Iteration 115, loss = 0.83454191
Iteration 116, loss = 0.83445947
Iteration 117, loss = 0.83428293
Iteration 118, loss = 0.83209680
Iteration 119, loss = 0.83161120
Iteration 120, loss = 0.82950150
Iteration 121, loss = 0.82816918
Iteration 122, loss = 0.82571957
Iteration 123, loss = 0.82528706
Iteration 124, loss = 0.82521880
Iteration 125, loss = 0.82098006
Iteration 126, loss = 0.82264744
Iteration 127, loss = 0.82057122
Iteration 128, loss = 0.81972945
Iteration 129, loss = 0.81794096
Iteration 130, loss = 0.81698015
Iteration 131, loss = 0.81392097
Iteration 132, loss = 0.81311342
Iteration 133, loss = 0.81261042
Iteration 134, loss = 0.81082857
Iteration 135, loss = 0.81059350
Iteration 136, loss = 0.80865985
Iteration 137, loss = 0.80638086
Iteration 138, loss = 0.80524424
Iteration 139, loss = 0.80387332
Iteration 140, loss = 0.80104203
Iteration 141, loss = 0.80072861
Iteration 142, loss = 0.79982334
Iteration 143, loss = 0.79801018
Iteration 144, loss = 0.79613507
Iteration 145, loss = 0.79778843
Iteration 146, loss = 0.79368467
Iteration 147, loss = 0.79476152
Iteration 148, loss = 0.79268230
Iteration 149, loss = 0.79114816
Iteration 150, loss = 0.78808048
Iteration 151, loss = 0.78675166
Iteration 152, loss = 0.78585843
Iteration 153, loss = 0.78450756
Iteration 154, loss = 0.78366222
Iteration 155, loss = 0.77951481
Iteration 156, loss = 0.77914377
Iteration 157, loss = 0.77737721
Iteration 158, loss = 0.77755512
Iteration 159, loss = 0.77618153
Iteration 160, loss = 0.77329782
Iteration 161, loss = 0.77517440
Iteration 162, loss = 0.77110444
Iteration 163, loss = 0.76952054
Iteration 164, loss = 0.76818725
Iteration 165, loss = 0.76766355
Iteration 166, loss = 0.76483223
Iteration 167, loss = 0.76292675
Iteration 168, loss = 0.76094059
Iteration 169, loss = 0.76152903
Iteration 170, loss = 0.76066005
Iteration 171, loss = 0.75779942
Iteration 172, loss = 0.75704351
Iteration 173, loss = 0.75520212
Iteration 174, loss = 0.75569495
Iteration 175, loss = 0.75306535
Iteration 176, loss = 0.75218323
Iteration 177, loss = 0.74973374
Iteration 178, loss = 0.74836026
Iteration 179, loss = 0.74800990
Iteration 180, loss = 0.74509579
Iteration 181, loss = 0.74545993
Iteration 182, loss = 0.74320766
Iteration 183, loss = 0.74048360
Iteration 184, loss = 0.74136001
Iteration 185, loss = 0.73614960
Iteration 186, loss = 0.73681684
Iteration 187, loss = 0.73564819
Iteration 188, loss = 0.73389324
Iteration 189, loss = 0.73186270
Iteration 190, loss = 0.73066608
Iteration 191, loss = 0.72797491
```

```
Iteration 192, loss = 0.72682043
Iteration 193, loss = 0.72877336
Iteration 194, loss = 0.72363776
Iteration 195, loss = 0.72134120
Iteration 196, loss = 0.72286306
Iteration 197, loss = 0.71916330
Iteration 198, loss = 0.72007071
Iteration 199, loss = 0.71502176
Iteration 200, loss = 0.71702866
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
Iteration 1, loss = 1.67509555
Iteration 2, loss = 1.56829368
Iteration 3, loss = 1.49639704
Iteration 4, loss = 1.41790089
Iteration 5, loss = 1.34839274
Iteration 6, loss = 1.29459817
Iteration 7, loss = 1.25055938
Iteration 8, loss = 1.21392153
Iteration 9, loss = 1.18487481
Iteration 10, loss = 1.15677643
Iteration 11, loss = 1.13349353
Iteration 12, loss = 1.11456184
Iteration 13, loss = 1.09556660
Iteration 14, loss = 1.07987852
Iteration 15, loss = 1.06601523
Iteration 16, loss = 1.05403721
Iteration 17, loss = 1.04497431
Iteration 18, loss = 1.03582203
Iteration 19, loss = 1.02399481
Iteration 20, loss = 1.01861250
Iteration 21, loss = 1.01174742
Iteration 22, loss = 1.00335535
Iteration 23, loss = 0.99717686
Iteration 24, loss = 0.99555620
Iteration 25, loss = 0.98854509
Iteration 26, loss = 0.98354305
Iteration 27, loss = 0.97910246
Iteration 28, loss = 0.97643754
Iteration 29, loss = 0.97117682
Iteration 30, loss = 0.96800639
Iteration 31, loss = 0.96428667
Iteration 32, loss = 0.96042253
Iteration 33, loss = 0.95747817
Iteration 34, loss = 0.95577916
Iteration 35, loss = 0.95081463
Iteration 36, loss = 0.94843113
Iteration 37, loss = 0.94555259
Iteration 38, loss = 0.94293048
Iteration 39, loss = 0.94222485
Iteration 40, loss = 0.93827211
Iteration 41, loss = 0.93462222
Iteration 42, loss = 0.93332020
Iteration 43, loss = 0.93260409
Iteration 44, loss = 0.92924190
Iteration 45, loss = 0.92746368
Iteration 46, loss = 0.92575395
Iteration 47, loss = 0.92300589
Iteration 48, loss = 0.92034660
Iteration 49, loss = 0.91870056
Iteration 50, loss = 0.91596187
Iteration 51, loss = 0.91701957
Iteration 52, loss = 0.91341207
Iteration 53, loss = 0.91232075
Iteration 54, loss = 0.91028722
Iteration 55, loss = 0.90938324
Iteration 56, loss = 0.90585707
Iteration 57, loss = 0.90495865
Iteration 58, loss = 0.90250675
Iteration 59, loss = 0.90231612
Iteration 60, loss = 0.89965630
Iteration 61, loss = 0.89832420
Iteration 62, loss = 0.89569001
Iteration 63, loss = 0.89646847
Iteration 64, loss = 0.89473899
Iteration 65, loss = 0.89298154
Iteration 66, loss = 0.89152845
Iteration 67, loss = 0.88731298
Iteration 68, loss = 0.88714059
Iteration 69, loss = 0.88490521
Iteration 70, loss = 0.88374971
Iteration 71, loss = 0.88135867
Iteration 72, loss = 0.88097778
Iteration 73, loss = 0.87875270
Iteration 74, loss = 0.88017778
Iteration 75, loss = 0.87785084
Iteration 76, loss = 0.87687654
Iteration 77, loss = 0.87399600
Iteration 78, loss = 0.87178965
Iteration 79, loss = 0.87028276
Iteration 80, loss = 0.86984741
```

```
Iteration 81, loss = 0.86872597
Iteration 82, loss = 0.86629688
Iteration 83, loss = 0.86593730
Iteration 84, loss = 0.86341338
Iteration 85, loss = 0.86251916
Iteration 86, loss = 0.85983900
Iteration 87, loss = 0.86012220
Iteration 88, loss = 0.85849649
Iteration 89, loss = 0.85749315
Iteration 90, loss = 0.85622865
Iteration 91, loss = 0.85418908
Iteration 92, loss = 0.85253912
Iteration 93, loss = 0.85248909
Iteration 94, loss = 0.85055614
Iteration 95, loss = 0.84973505
Iteration 96, loss = 0.84831413
Iteration 97, loss = 0.84671586
Iteration 98, loss = 0.84374481
Iteration 99, loss = 0.84292325
Iteration 100, loss = 0.84094912
Iteration 101, loss = 0.83888229
Iteration 102, loss = 0.83987632
Iteration 103, loss = 0.83824647
Iteration 104, loss = 0.83777916
Iteration 105, loss = 0.83530488
Iteration 106, loss = 0.83404190
Iteration 107, loss = 0.83284502
Iteration 108, loss = 0.83073880
Iteration 109, loss = 0.82879043
Iteration 110, loss = 0.82761196
Iteration 111, loss = 0.82582388
Iteration 112, loss = 0.82680388
Iteration 113, loss = 0.82446026
Iteration 114, loss = 0.82187237
Iteration 115, loss = 0.81983568
Iteration 116, loss = 0.81851582
Iteration 117, loss = 0.81886171
Iteration 118, loss = 0.81785585
Iteration 119, loss = 0.81493429
Iteration 120, loss = 0.81267985
Iteration 121, loss = 0.81146866
Iteration 122, loss = 0.81005085
Iteration 123, loss = 0.80838787
Iteration 124, loss = 0.80980986
Iteration 125, loss = 0.80818163
Iteration 126, loss = 0.80666884
Iteration 127, loss = 0.80630068
Iteration 128, loss = 0.80248294
Iteration 129, loss = 0.80139753
Iteration 130, loss = 0.79969970
Iteration 131, loss = 0.79811086
Iteration 132, loss = 0.79738884
Iteration 133, loss = 0.79534187
Iteration 134, loss = 0.79482863
Iteration 135, loss = 0.79318778
Iteration 136, loss = 0.79401119
Iteration 137, loss = 0.78912053
Iteration 138, loss = 0.78933522
Iteration 139, loss = 0.78758889
Iteration 140, loss = 0.78751990
Iteration 141, loss = 0.78300081
Iteration 142, loss = 0.78331822
Iteration 143, loss = 0.78175334
Iteration 144, loss = 0.77945628
Iteration 145, loss = 0.78018515
Iteration 146, loss = 0.77776979
Iteration 147, loss = 0.77570797
Iteration 148, loss = 0.77398752
Iteration 149, loss = 0.77210963
Iteration 150, loss = 0.77101060
Iteration 151, loss = 0.76958738
Iteration 152, loss = 0.76700837
Iteration 153, loss = 0.76732365
Iteration 154, loss = 0.76410302
Iteration 155, loss = 0.76711796
Iteration 156, loss = 0.76329403
Iteration 157, loss = 0.76139900
Iteration 158, loss = 0.76008940
Iteration 159, loss = 0.75809602
Iteration 160, loss = 0.75627687
Iteration 161, loss = 0.75423844
Iteration 162, loss = 0.75195083
Iteration 163, loss = 0.75126119
Iteration 164, loss = 0.75022181
Iteration 165, loss = 0.74700575
Iteration 166, loss = 0.74694241
Iteration 167, loss = 0.74455652
Iteration 168, loss = 0.74231548
Iteration 169, loss = 0.74282271
Iteration 170, loss = 0.74211857
Iteration 171, loss = 0.73872374
Iteration 172, loss = 0.73700110
```

```
Iteration 173, loss = 0.73609593
Iteration 174, loss = 0.73481954
Iteration 175, loss = 0.73367092
Iteration 176, loss = 0.73214727
Iteration 177, loss = 0.73188682
Iteration 178, loss = 0.72725224
Iteration 179, loss = 0.72775059
Iteration 180, loss = 0.72582294
Iteration 181, loss = 0.72493586
Iteration 182, loss = 0.72216995
Iteration 183, loss = 0.72155428
Iteration 184, loss = 0.71937495
Iteration 185, loss = 0.71702542
Iteration 186, loss = 0.71680356
Iteration 187, loss = 0.71497466
Iteration 188, loss = 0.71422825
Iteration 189, loss = 0.71178947
Iteration 190, loss = 0.71092229
Iteration 191, loss = 0.70836341
Iteration 192, loss = 0.70605112
Iteration 193, loss = 0.70997936
Iteration 194, loss = 0.70427665
Iteration 195, loss = 0.70468981
Iteration 196, loss = 0.70293589
Iteration 197, loss = 0.70054791
Iteration 198, loss = 0.69911233
Iteration 199, loss = 0.69747447
Iteration 200, loss = 0.69484860
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
Iteration 1, loss = 1.67314339
Iteration 2, loss = 1.56559002
Iteration 3, loss = 1.48771021
Iteration 4, loss = 1.40506929
Iteration 5, loss = 1.33417944
Iteration 6, loss = 1.27936672
Iteration 7, loss = 1.23640745
Iteration 8, loss = 1.20043532
Iteration 9, loss = 1.17131616
Iteration 10, loss = 1.14515648
Iteration 11, loss = 1.12128179
Iteration 12, loss = 1.10176978
Iteration 13, loss = 1.08450539
Iteration 14, loss = 1.06916958
Iteration 15, loss = 1.05485670
Iteration 16, loss = 1.04351063
Iteration 17, loss = 1.03152550
Iteration 18, loss = 1.02174962
Iteration 19, loss = 1.01345327
Iteration 20, loss = 1.00747298
Iteration 21, loss = 0.99872959
Iteration 22, loss = 0.99294582
Iteration 23, loss = 0.98665575
Iteration 24, loss = 0.98362952
Iteration 25, loss = 0.97677781
Iteration 26, loss = 0.97297113
Iteration 27, loss = 0.96770427
Iteration 28, loss = 0.96376955
Iteration 29, loss = 0.95979105
Iteration 30, loss = 0.95657621
Iteration 31, loss = 0.95278983
Iteration 32, loss = 0.94980070
Iteration 33, loss = 0.94817992
Iteration 34, loss = 0.94525910
Iteration 35, loss = 0.94088731
Iteration 36, loss = 0.93891728
Iteration 37, loss = 0.93524930
Iteration 38, loss = 0.93377039
Iteration 39, loss = 0.93167479
Iteration 40, loss = 0.92852161
Iteration 41, loss = 0.92584820
Iteration 42, loss = 0.92307627
Iteration 43, loss = 0.92250973
Iteration 44, loss = 0.91848809
Iteration 45, loss = 0.91710395
Iteration 46, loss = 0.91563262
Iteration 47, loss = 0.91427406
Iteration 48, loss = 0.91180993
Iteration 49, loss = 0.90926126
Iteration 50, loss = 0.90678166
Iteration 51, loss = 0.90674850
Iteration 52, loss = 0.90383985
Iteration 53, loss = 0.90235702
Iteration 54, loss = 0.89975997
Iteration 55, loss = 0.89890704
Iteration 56, loss = 0.89542805
Iteration 57, loss = 0.89417757
Iteration 58, loss = 0.89319789
Iteration 59, loss = 0.89124010
Iteration 60, loss = 0.88892177
Iteration 61, loss = 0.88924799
```

```
Iteration 62, loss = 0.88611136
Iteration 63, loss = 0.88438809
Iteration 64, loss = 0.88408850
Iteration 65, loss = 0.88144581
Iteration 66, loss = 0.87995990
Iteration 67, loss = 0.87683497
Iteration 68, loss = 0.87627146
Iteration 69, loss = 0.87473835
Iteration 70, loss = 0.87472954
Iteration 71, loss = 0.87192310
Iteration 72, loss = 0.87038541
Iteration 73, loss = 0.86817219
Iteration 74, loss = 0.86938812
Iteration 75, loss = 0.86555981
Iteration 76, loss = 0.86505638
Iteration 77, loss = 0.86249319
Iteration 78, loss = 0.86075039
Iteration 79, loss = 0.86014926
Iteration 80, loss = 0.85842809
Iteration 81, loss = 0.85627173
Iteration 82, loss = 0.85519465
Iteration 83, loss = 0.85475547
Iteration 84, loss = 0.85260987
Iteration 85, loss = 0.85051451
Iteration 86, loss = 0.85163503
Iteration 87, loss = 0.84897001
Iteration 88, loss = 0.84691028
Iteration 89, loss = 0.84632261
Iteration 90, loss = 0.84567713
Iteration 91, loss = 0.84357687
Iteration 92, loss = 0.84093181
Iteration 93, loss = 0.84135359
Iteration 94, loss = 0.83861126
Iteration 95, loss = 0.83668697
Iteration 96, loss = 0.83572124
Iteration 97, loss = 0.83440294
Iteration 98, loss = 0.83216211
Iteration 99, loss = 0.83315453
Iteration 100, loss = 0.83106760
Iteration 101, loss = 0.82760893
Iteration 102, loss = 0.82676608
Iteration 103, loss = 0.82634158
Iteration 104, loss = 0.82549514
Iteration 105, loss = 0.82299507
Iteration 106, loss = 0.82031810
Iteration 107, loss = 0.81944979
Iteration 108, loss = 0.81637126
Iteration 109, loss = 0.81708061
Iteration 110, loss = 0.81434545
Iteration 111, loss = 0.81366607
Iteration 112, loss = 0.81178159
Iteration 113, loss = 0.81089939
Iteration 114, loss = 0.80963451
Iteration 115, loss = 0.80880534
Iteration 116, loss = 0.80536055
Iteration 117, loss = 0.80422827
Iteration 118, loss = 0.80314924
Iteration 119, loss = 0.80320059
Iteration 120, loss = 0.80037856
Iteration 121, loss = 0.79902394
Iteration 122, loss = 0.79731658
Iteration 123, loss = 0.79666646
Iteration 124, loss = 0.79418159
Iteration 125, loss = 0.79354171
Iteration 126, loss = 0.79253642
Iteration 127, loss = 0.79011588
Iteration 128, loss = 0.78751103
Iteration 129, loss = 0.78730774
Iteration 130, loss = 0.78761545
Iteration 131, loss = 0.78497687
Iteration 132, loss = 0.78352158
Iteration 133, loss = 0.78262127
Iteration 134, loss = 0.77976274
Iteration 135, loss = 0.77822790
Iteration 136, loss = 0.77915207
Iteration 137, loss = 0.77604400
Iteration 138, loss = 0.77432571
Iteration 139, loss = 0.77387559
Iteration 140, loss = 0.77146121
Iteration 141, loss = 0.76890143
Iteration 142, loss = 0.76810680
Iteration 143, loss = 0.76709668
Iteration 144, loss = 0.76389110
Iteration 145, loss = 0.76351277
Iteration 146, loss = 0.76272038
Iteration 147, loss = 0.76180215
Iteration 148, loss = 0.75946259
Iteration 149, loss = 0.75792892
Iteration 150, loss = 0.75786265
Iteration 151, loss = 0.75395063
Iteration 152, loss = 0.75256695
Iteration 153, loss = 0.75439325
```

```
Iteration 154, loss = 0.75222779
Iteration 155, loss = 0.74813792
Iteration 156, loss = 0.74736315
Iteration 157, loss = 0.74571177
Iteration 158, loss = 0.74407620
Iteration 159, loss = 0.74450031
Iteration 160, loss = 0.73984923
Iteration 161, loss = 0.74060179
Iteration 162, loss = 0.73979247
Iteration 163, loss = 0.73653412
Iteration 164, loss = 0.73600845
Iteration 165, loss = 0.73222396
Iteration 166, loss = 0.73322893
Iteration 167, loss = 0.72843065
Iteration 168, loss = 0.72733865
Iteration 169, loss = 0.72881604
Iteration 170, loss = 0.72672460
Iteration 171, loss = 0.72513536
Iteration 172, loss = 0.72179706
Iteration 173, loss = 0.72343002
Iteration 174, loss = 0.71931556
Iteration 175, loss = 0.71592261
Iteration 176, loss = 0.71518479
Iteration 177, loss = 0.71409673
Iteration 178, loss = 0.71225435
Iteration 179, loss = 0.71102383
Iteration 180, loss = 0.70964938
Iteration 181, loss = 0.70874986
Iteration 182, loss = 0.70634652
Iteration 183, loss = 0.70500585
Iteration 184, loss = 0.70430478
Iteration 185, loss = 0.70066254
Iteration 186, loss = 0.70121417
Iteration 187, loss = 0.69746486
Iteration 188, loss = 0.69812784
Iteration 189, loss = 0.69499408
Iteration 190, loss = 0.69568965
Iteration 191, loss = 0.69183704
Iteration 192, loss = 0.69152521
Iteration 193, loss = 0.68993056
Iteration 194, loss = 0.68806764
Iteration 195, loss = 0.68665132
Iteration 196, loss = 0.68552418
Iteration 197, loss = 0.68178593
Iteration 198, loss = 0.68206940
Iteration 199, loss = 0.67844615
Iteration 200, loss = 0.67791363
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

```
Iteration 1, loss = 1.66919454
Iteration 2, loss = 1.56072795
Iteration 3, loss = 1.48047527
Iteration 4, loss = 1.39573322
Iteration 5, loss = 1.32516931
Iteration 6, loss = 1.26983717
Iteration 7, loss = 1.22695135
Iteration 8, loss = 1.19142153
Iteration 9, loss = 1.16124583
Iteration 10, loss = 1.13465982
Iteration 11, loss = 1.11146086
Iteration 12, loss = 1.09073673
Iteration 13, loss = 1.07345110
Iteration 14, loss = 1.05718527
Iteration 15, loss = 1.04309499
Iteration 16, loss = 1.03121229
Iteration 17, loss = 1.01951235
Iteration 18, loss = 1.01038774
Iteration 19, loss = 1.00131917
Iteration 20, loss = 0.99577284
Iteration 21, loss = 0.98737561
Iteration 22, loss = 0.98042966
Iteration 23, loss = 0.97396130
Iteration 24, loss = 0.97222007
Iteration 25, loss = 0.96520638
Iteration 26, loss = 0.96040211
Iteration 27, loss = 0.95672423
Iteration 28, loss = 0.95200801
Iteration 29, loss = 0.94809003
Iteration 30, loss = 0.94508684
Iteration 31, loss = 0.94137777
Iteration 32, loss = 0.93729173
Iteration 33, loss = 0.93821351
Iteration 34, loss = 0.93276809
Iteration 35, loss = 0.92903198
Iteration 36, loss = 0.92775245
Iteration 37, loss = 0.92248161
Iteration 38, loss = 0.92150011
Iteration 39, loss = 0.91916683
Iteration 40, loss = 0.91615200
Iteration 41, loss = 0.91370441
Iteration 42, loss = 0.91093523
```

```
Iteration 43, loss = 0.90911766
Iteration 44, loss = 0.90634397
Iteration 45, loss = 0.90476114
Iteration 46, loss = 0.90283252
Iteration 47, loss = 0.90189247
Iteration 48, loss = 0.89820381
Iteration 49, loss = 0.89747409
Iteration 50, loss = 0.89431481
Iteration 51, loss = 0.89566138
Iteration 52, loss = 0.89188573
Iteration 53, loss = 0.89009052
Iteration 54, loss = 0.88790658
Iteration 55, loss = 0.88535009
Iteration 56, loss = 0.88399699
Iteration 57, loss = 0.88323804
Iteration 58, loss = 0.88334936
Iteration 59, loss = 0.87944327
Iteration 60, loss = 0.87685449
Iteration 61, loss = 0.87611136
Iteration 62, loss = 0.87419505
Iteration 63, loss = 0.87249591
Iteration 64, loss = 0.87309562
Iteration 65, loss = 0.86989657
Iteration 66, loss = 0.86738061
Iteration 67, loss = 0.86598991
Iteration 68, loss = 0.86504945
Iteration 69, loss = 0.86334120
Iteration 70, loss = 0.86424018
Iteration 71, loss = 0.86106270
Iteration 72, loss = 0.85827610
Iteration 73, loss = 0.85708806
Iteration 74, loss = 0.85628035
Iteration 75, loss = 0.85573580
Iteration 76, loss = 0.85293919
Iteration 77, loss = 0.85156247
Iteration 78, loss = 0.84963173
Iteration 79, loss = 0.84903135
Iteration 80, loss = 0.84700868
Iteration 81, loss = 0.84466392
Iteration 82, loss = 0.84367115
Iteration 83, loss = 0.84254694
Iteration 84, loss = 0.84187639
Iteration 85, loss = 0.83945451
Iteration 86, loss = 0.83972687
Iteration 87, loss = 0.83777331
Iteration 88, loss = 0.83477857
Iteration 89, loss = 0.83581171
Iteration 90, loss = 0.83394307
Iteration 91, loss = 0.83238700
Iteration 92, loss = 0.83003493
Iteration 93, loss = 0.82921124
Iteration 94, loss = 0.82806804
Iteration 95, loss = 0.82553869
Iteration 96, loss = 0.82466516
Iteration 97, loss = 0.82239014
Iteration 98, loss = 0.82138774
Iteration 99, loss = 0.82013123
Iteration 100, loss = 0.82074365
Iteration 101, loss = 0.81637311
Iteration 102, loss = 0.81673964
Iteration 103, loss = 0.81512620
Iteration 104, loss = 0.81532085
Iteration 105, loss = 0.81232041
Iteration 106, loss = 0.81014170
Iteration 107, loss = 0.80928217
Iteration 108, loss = 0.80853229
Iteration 109, loss = 0.80723594
Iteration 110, loss = 0.80714872
Iteration 111, loss = 0.80396147
Iteration 112, loss = 0.80153935
Iteration 113, loss = 0.80182149
Iteration 114, loss = 0.80030828
Iteration 115, loss = 0.79787880
Iteration 116, loss = 0.79647186
Iteration 117, loss = 0.79413016
Iteration 118, loss = 0.79416565
Iteration 119, loss = 0.79299415
Iteration 120, loss = 0.79172335
Iteration 121, loss = 0.78908092
Iteration 122, loss = 0.78847833
Iteration 123, loss = 0.78588118
Iteration 124, loss = 0.78621547
Iteration 125, loss = 0.78344973
Iteration 126, loss = 0.78333864
Iteration 127, loss = 0.78129419
Iteration 128, loss = 0.78163841
Iteration 129, loss = 0.77892250
Iteration 130, loss = 0.77883732
Iteration 131, loss = 0.77586191
Iteration 132, loss = 0.77521813
Iteration 133, loss = 0.77281733
Iteration 134, loss = 0.77161181
```

```
Iteration 135, loss = 0.77159709
Iteration 136, loss = 0.76921242
Iteration 137, loss = 0.76888663
Iteration 138, loss = 0.76561099
Iteration 139, loss = 0.76463032
Iteration 140, loss = 0.76270012
Iteration 141, loss = 0.76159820
Iteration 142, loss = 0.76047748
Iteration 143, loss = 0.75926362
Iteration 144, loss = 0.75658203
Iteration 145, loss = 0.75613822
Iteration 146, loss = 0.75498969
Iteration 147, loss = 0.75301887
Iteration 148, loss = 0.75149767
Iteration 149, loss = 0.74992572
Iteration 150, loss = 0.74910370
Iteration 151, loss = 0.74790306
Iteration 152, loss = 0.74578654
Iteration 153, loss = 0.74498413
Iteration 154, loss = 0.74383635
Iteration 155, loss = 0.74107248
Iteration 156, loss = 0.73941445
Iteration 157, loss = 0.73698481
Iteration 158, loss = 0.73667704
Iteration 159, loss = 0.73552379
Iteration 160, loss = 0.73342010
Iteration 161, loss = 0.73165700
Iteration 162, loss = 0.73089048
Iteration 163, loss = 0.72927218
Iteration 164, loss = 0.72913804
Iteration 165, loss = 0.72540250
Iteration 166, loss = 0.72287765
Iteration 167, loss = 0.72293030
Iteration 168, loss = 0.72078749
Iteration 169, loss = 0.71922545
Iteration 170, loss = 0.71979830
Iteration 171, loss = 0.71696778
Iteration 172, loss = 0.71517475
Iteration 173, loss = 0.71665681
Iteration 174, loss = 0.71297380
Iteration 175, loss = 0.71287534
Iteration 176, loss = 0.70892479
Iteration 177, loss = 0.70849261
Iteration 178, loss = 0.70809115
Iteration 179, loss = 0.70536761
Iteration 180, loss = 0.70353848
Iteration 181, loss = 0.70330641
Iteration 182, loss = 0.69955822
Iteration 183, loss = 0.70007339
Iteration 184, loss = 0.69901473
Iteration 185, loss = 0.69541130
Iteration 186, loss = 0.69554989
Iteration 187, loss = 0.69232317
Iteration 188, loss = 0.69245574
Iteration 189, loss = 0.68884405
Iteration 190, loss = 0.68724353
Iteration 191, loss = 0.68685774
Iteration 192, loss = 0.68557707
Iteration 193, loss = 0.68287654
Iteration 194, loss = 0.68281181
Iteration 195, loss = 0.68106624
Iteration 196, loss = 0.67920517
Iteration 197, loss = 0.67646497
Iteration 198, loss = 0.67732125
Iteration 199, loss = 0.67322880
Iteration 200, loss = 0.67286786
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

In [ ]:

```
best_hogMLP_model_cvscores
```

Out[ ]:

```
array([0.65824847, 0.68296659, 0.66462918, 0.64466178, 0.62754686])
```

In [ ]:

```
best_hogMLP_model_cvscores.mean()
```

Out[ ]:

```
0.6556105746966174
```

In [ ]:

```
t0 = time.time()
best_hogMLP_model.fit(hog_descriptors, y_train)
tt = time.time()
print(tt - t0)
```

```
Iteration 1, loss = 1.66022210
Iteration 2, loss = 1.53852764
Iteration 3, loss = 1.44102658
Iteration 4, loss = 1.34943261
Iteration 5, loss = 1.28173618
Iteration 6, loss = 1.23125369
```

```
Iteration 7, loss = 1.19142438
Iteration 8, loss = 1.15880580
Iteration 9, loss = 1.13107478
Iteration 10, loss = 1.10659160
Iteration 11, loss = 1.08707569
Iteration 12, loss = 1.06923561
Iteration 13, loss = 1.05410797
Iteration 14, loss = 1.04120961
Iteration 15, loss = 1.02993540
Iteration 16, loss = 1.02154976
Iteration 17, loss = 1.01139144
Iteration 18, loss = 1.00427718
Iteration 19, loss = 0.99629329
Iteration 20, loss = 0.98971069
Iteration 21, loss = 0.98382644
Iteration 22, loss = 0.97875124
Iteration 23, loss = 0.97449902
Iteration 24, loss = 0.96998680
Iteration 25, loss = 0.96579261
Iteration 26, loss = 0.96253225
Iteration 27, loss = 0.95782631
Iteration 28, loss = 0.95610398
Iteration 29, loss = 0.95189695
Iteration 30, loss = 0.94785844
Iteration 31, loss = 0.94646855
Iteration 32, loss = 0.94260068
Iteration 33, loss = 0.93995173
Iteration 34, loss = 0.93709441
Iteration 35, loss = 0.93408861
Iteration 36, loss = 0.93070399
Iteration 37, loss = 0.92885894
Iteration 38, loss = 0.92751233
Iteration 39, loss = 0.92493887
Iteration 40, loss = 0.92272640
Iteration 41, loss = 0.92046118
Iteration 42, loss = 0.91748665
Iteration 43, loss = 0.91675164
Iteration 44, loss = 0.91355706
Iteration 45, loss = 0.91143035
Iteration 46, loss = 0.90993811
Iteration 47, loss = 0.90869999
Iteration 48, loss = 0.90590969
Iteration 49, loss = 0.90380210
Iteration 50, loss = 0.90302541
Iteration 51, loss = 0.90065005
Iteration 52, loss = 0.89886322
Iteration 53, loss = 0.89688260
Iteration 54, loss = 0.89557192
Iteration 55, loss = 0.89419216
Iteration 56, loss = 0.89268483
Iteration 57, loss = 0.89105729
Iteration 58, loss = 0.88846806
Iteration 59, loss = 0.88736609
Iteration 60, loss = 0.88418880
Iteration 61, loss = 0.88449661
Iteration 62, loss = 0.88287422
Iteration 63, loss = 0.88032854
Iteration 64, loss = 0.87877064
Iteration 65, loss = 0.87758045
Iteration 66, loss = 0.87653087
Iteration 67, loss = 0.87391123
Iteration 68, loss = 0.87215561
Iteration 69, loss = 0.86973011
Iteration 70, loss = 0.86944530
Iteration 71, loss = 0.86701477
Iteration 72, loss = 0.86631009
Iteration 73, loss = 0.86355813
Iteration 74, loss = 0.86306163
Iteration 75, loss = 0.86049695
Iteration 76, loss = 0.85896641
Iteration 77, loss = 0.85668399
Iteration 78, loss = 0.85723759
Iteration 79, loss = 0.85537190
Iteration 80, loss = 0.85366653
Iteration 81, loss = 0.85073082
Iteration 82, loss = 0.85121579
Iteration 83, loss = 0.84874966
Iteration 84, loss = 0.84650791
Iteration 85, loss = 0.84687226
Iteration 86, loss = 0.84381323
Iteration 87, loss = 0.84296388
Iteration 88, loss = 0.84123999
Iteration 89, loss = 0.83867536
Iteration 90, loss = 0.83701410
Iteration 91, loss = 0.83599607
Iteration 92, loss = 0.83358834
Iteration 93, loss = 0.83292911
Iteration 94, loss = 0.83125898
Iteration 95, loss = 0.82979178
Iteration 96, loss = 0.82796463
Iteration 97, loss = 0.82551782
Iteration 98, loss = 0.82451583
```

```
Iteration 99, loss = 0.82293629
Iteration 100, loss = 0.82144583
Iteration 101, loss = 0.82074828
Iteration 102, loss = 0.81796965
Iteration 103, loss = 0.81620041
Iteration 104, loss = 0.81503674
Iteration 105, loss = 0.81313890
Iteration 106, loss = 0.81136658
Iteration 107, loss = 0.81105265
Iteration 108, loss = 0.80916059
Iteration 109, loss = 0.80663192
Iteration 110, loss = 0.80519359
Iteration 111, loss = 0.80364595
Iteration 112, loss = 0.80274954
Iteration 113, loss = 0.80073347
Iteration 114, loss = 0.79858058
Iteration 115, loss = 0.79769417
Iteration 116, loss = 0.79626513
Iteration 117, loss = 0.79342762
Iteration 118, loss = 0.79306528
Iteration 119, loss = 0.79139481
Iteration 120, loss = 0.78902043
Iteration 121, loss = 0.78807154
Iteration 122, loss = 0.78594917
Iteration 123, loss = 0.78495535
Iteration 124, loss = 0.78319043
Iteration 125, loss = 0.78081608
Iteration 126, loss = 0.78026359
Iteration 127, loss = 0.77832699
Iteration 128, loss = 0.77622881
Iteration 129, loss = 0.77510066
Iteration 130, loss = 0.77289506
Iteration 131, loss = 0.77242879
Iteration 132, loss = 0.76949574
Iteration 133, loss = 0.77115772
Iteration 134, loss = 0.76786512
Iteration 135, loss = 0.76585383
Iteration 136, loss = 0.76301431
Iteration 137, loss = 0.76181565
Iteration 138, loss = 0.76014658
Iteration 139, loss = 0.75813924
Iteration 140, loss = 0.75749566
Iteration 141, loss = 0.75592921
Iteration 142, loss = 0.75407061
Iteration 143, loss = 0.75229985
Iteration 144, loss = 0.75086530
Iteration 145, loss = 0.74846297
Iteration 146, loss = 0.74725761
Iteration 147, loss = 0.74652781
Iteration 148, loss = 0.74392943
Iteration 149, loss = 0.74345151
Iteration 150, loss = 0.74207289
Iteration 151, loss = 0.73971018
Iteration 152, loss = 0.73854251
Iteration 153, loss = 0.73618882
Iteration 154, loss = 0.73357934
Iteration 155, loss = 0.73264356
Iteration 156, loss = 0.73033305
Iteration 157, loss = 0.72922589
Iteration 158, loss = 0.72786866
Iteration 159, loss = 0.72639832
Iteration 160, loss = 0.72494633
Iteration 161, loss = 0.72306203
Iteration 162, loss = 0.72225219
Iteration 163, loss = 0.71925520
Iteration 164, loss = 0.71915558
Iteration 165, loss = 0.71780752
Iteration 166, loss = 0.71499527
Iteration 167, loss = 0.71360715
Iteration 168, loss = 0.71253605
Iteration 169, loss = 0.70960759
Iteration 170, loss = 0.70843239
Iteration 171, loss = 0.70760414
Iteration 172, loss = 0.70495821
Iteration 173, loss = 0.70424051
Iteration 174, loss = 0.70203887
Iteration 175, loss = 0.70054074
Iteration 176, loss = 0.69894078
Iteration 177, loss = 0.69848813
Iteration 178, loss = 0.69489244
Iteration 179, loss = 0.69370047
Iteration 180, loss = 0.69206124
Iteration 181, loss = 0.69052210
Iteration 182, loss = 0.69060336
Iteration 183, loss = 0.68760214
Iteration 184, loss = 0.68429177
Iteration 185, loss = 0.68483646
Iteration 186, loss = 0.68297543
Iteration 187, loss = 0.68054476
Iteration 188, loss = 0.67847748
Iteration 189, loss = 0.67723157
Iteration 190, loss = 0.67622889
```

```
Iteration 191, loss = 0.67361934
Iteration 192, loss = 0.67157186
Iteration 193, loss = 0.67157598
Iteration 194, loss = 0.66946183
Iteration 195, loss = 0.66747929
Iteration 196, loss = 0.66533363
Iteration 197, loss = 0.66553185
Iteration 198, loss = 0.66257045
Iteration 199, loss = 0.65928290
Iteration 200, loss = 0.65946556
117.7620804309845
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: ConvergenceWarning: Stochastic Optimi
zer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  ConvergenceWarning,
```

In [ ]:
```
best_hogMLP_model.score(hog_descriptors, y_train)
```

Out[ ]:
```
0.7789096243174966
```

Save best MLP model that uses HOG descriptors.

In [ ]:
```
besthogmlppath = 'drive/My Drive/Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/Models/best_hogMLP_model.pickle'

with open (besthogmlppath, 'wb') as f:
    pickle.dump(best_hogMLP_model, f)
```

Import images and labels from test data.

In [ ]:
```
X_2, y_2 = import_imagelabels_lists('test')
```

In [ ]:
```
type(X_2)
```

Out[ ]:
```
list
```

In [ ]:
```
len(X_2)
```

Out[ ]:
```
3068
```

In [ ]:
```
type(y_2)
```

Out[ ]:
```
list
```

In [ ]:
```
len(y_2)
```

Out[ ]:
```
3068
```

The above shown number of images and labels matches with that mentioned in the coursework specification document.

Plot first 10 images of test data along with their labels.

In [ ]:
```
fig, axes = plt.subplots(2, 5, figsize=(14, 7), sharex=True, sharey=True)
ax = axes.ravel()

for i in range(10):
    ax[i].imshow(X_2[i])
    ax[i].set_title(f'Label: {y_2[i]}')
    ax[i].set_axis_off()
fig.tight_layout()
plt.show()
```

In [ ]:
```
X_test = X_2
y_test = y_2
```

In [ ]:
```
testk = len(np.unique(y_test)) * 10

# Use MiniBatchKMeans for faster computation and lower memory usage
batch_size = des_array.shape[0] // 4
kmeans = MiniBatchKMeans(n_clusters=k, batch_size=batch_size).fit(des_array)
```

Extract SIFT descriptors for test data.

In [ ]:
```
testhist_list = []

for i in range(len(X_test)):
    testimg = img_as_ubyte(color.rgb2gray(X_test[i]))
    testkp, testdes = sift.detectAndCompute(testimg, None)

    if testdes is not None:
        testhist = np.zeros(k)

        testidx = kmeans.predict(testdes)

        for j in testidx:
            testhist[j] = testhist[j] + (1 / len(testdes))

        # hist = scale.transform(hist.reshape(1, -1))
        testhist_list.append(testhist)

    else:
        testhist_list.append(None)

# Remove potential cases of images with no descriptors
testidx_not_empty = [i for i, x in enumerate(testhist_list) if x is not None]
testhist_list = [testhist_list[i] for i in testidx_not_empty]
y_test1 = [y_test[i] for i in testidx_not_empty]
testhist_array = np.vstack(testhist_list)

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

In [ ]:
```
len(y_test1)
```

Out[ ]:
```
3066
```

Load best SVM model that uses SIFT descriptors.

In [ ]:
```
bestsiftsvmpath = 'drive/My Drive/Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/Models/best_siftSVM_model.pickle'

with open (bestsiftsvmpath, 'rb') as f:
    best_siftSVM_model_loaded = pickle.load(f)
```

Make predictions from test features.

In [ ]:
```
y_pred = best_siftSVM_model_loaded.predict(testhist_array).tolist()

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

Plot first 10 images of test data along with their actual and predicted labels.

In [ ]:
```
fig, axes = plt.subplots(2, 5, figsize=(14, 7), sharex=True, sharey=True)
ax = axes.ravel()

for i in range(10):
    ax[i].imshow(X_test[i])
    ax[i].set_title(f'Label: {y_test1[i]} \n Prediction: {y_pred[i]}')
    ax[i].set_axis_off()
fig.tight_layout()
plt.show()

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

Calculate test score of best SVM model that uses SIFT descriptors.

In [ ]:
```
siftSVM_test_score = best_siftSVM_model_loaded.score(testhist_array, y_test1)
siftSVM_test_score
```

Out[ ]:
```
0.22504892367906065
```

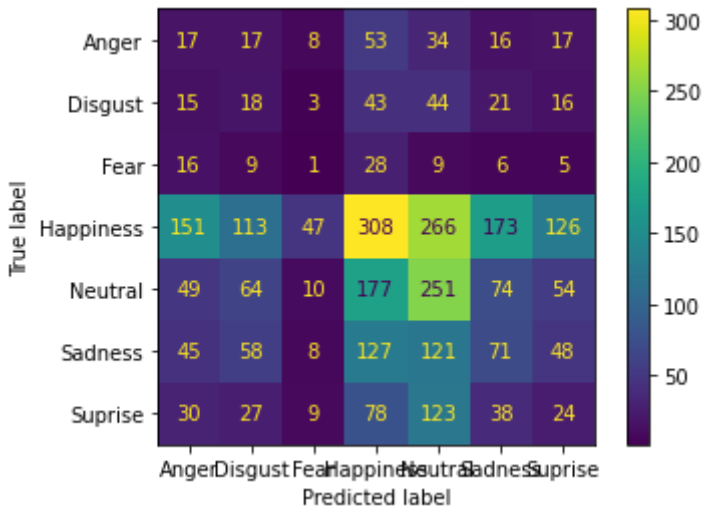
Generate a classification report of the best SVM model that uses SIFT descriptors.

In [ ]:
```
print(f"""Classification report for classifier best siftSVM- {best_siftSVM_model_loaded}:
      {metrics.classification_report(y_test1, y_pred)}\n""")

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```

```
Classification report for classifier best siftSVM- SVC(class_weight='balanced'):
                   precision    recall  f1-score   support

       Anger       0.05      0.10      0.07       162
     Disgust       0.06      0.11      0.08       160
        Fear       0.01      0.01      0.01        74
   Happiness       0.38      0.26      0.31      1184
     Neutral       0.30      0.37      0.33       679
     Sadness       0.18      0.15      0.16       478
     Suprise       0.08      0.07      0.08       329

    accuracy                           0.23      3066
   macro avg       0.15      0.15      0.15      3066
weighted avg       0.25      0.23      0.23      3066
```

Plot Confusion matrix for best SVM model that uses SIFT descriptors.

In [ ]:
```
metrics.ConfusionMatrixDisplay.from_predictions(y_test1, y_pred)
plt.show()

## CODING REFERENCE: Code was taken from Lab Tutorial 07 of Computer Vision - IN3060/INM460 module.
```



Load best MLP model that uses SIFT descriptors and then evaluate its performance over test data.

In [ ]:
```
bestsiftmlppath = 'drive/My Drive/Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/Models/best_siftMLP_model.pickle'

with open (bestsiftmlppath, 'rb') as f:
    best_siftMLP_model_loaded = pickle.load(f)
```

In [ ]:
```
y_pred_siftMLP = best_siftMLP_model_loaded.predict(testhist_array).tolist()
```

In [ ]:
```
fig, axes = plt.subplots(2, 5, figsize=(14, 7), sharex=True, sharey=True)
ax = axes.ravel()

for i in range(10):
    ax[i].imshow(X_test[i])
    ax[i].set_title(f'Label: {y_test1[i]} \n Prediction: {y_pred_siftMLP[i]}')
    ax[i].set_axis_off()
fig.tight_layout()
plt.show()
```

In [ ]:
```
siftMLP_test_score = best_siftMLP_model_loaded.score(testhist_array, y_test1)
siftMLP_test_score
```

Out[ ]:
```
0.3679060665362035
```

In [ ]:
```
print(f"""Classification report for classifier best siftMPL- {best_siftMLP_model_loaded}:
      {metrics.classification_report(y_test1, y_pred_siftMLP)}\n""")
```

```
Classification report for classifier best siftMPL- MLPClassifier(learning_rate_init=0.01, random_state=1, solver='sgd',
              verbose=True):
                    precision    recall  f1-score   support

       Anger       0.00      0.00      0.00       162
     Disgust       0.00      0.00      0.00       160
        Fear       0.00      0.00      0.00        74
   Happiness       0.38      0.88      0.54      1184
     Neutral       0.25      0.12      0.16       679
     Sadness       0.10      0.00      0.01       478
     Suprise       0.00      0.00      0.00       329

    accuracy                           0.37      3066
   macro avg       0.10      0.14      0.10      3066
```

```
weighted avg       0.22      0.37      0.24      3066
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

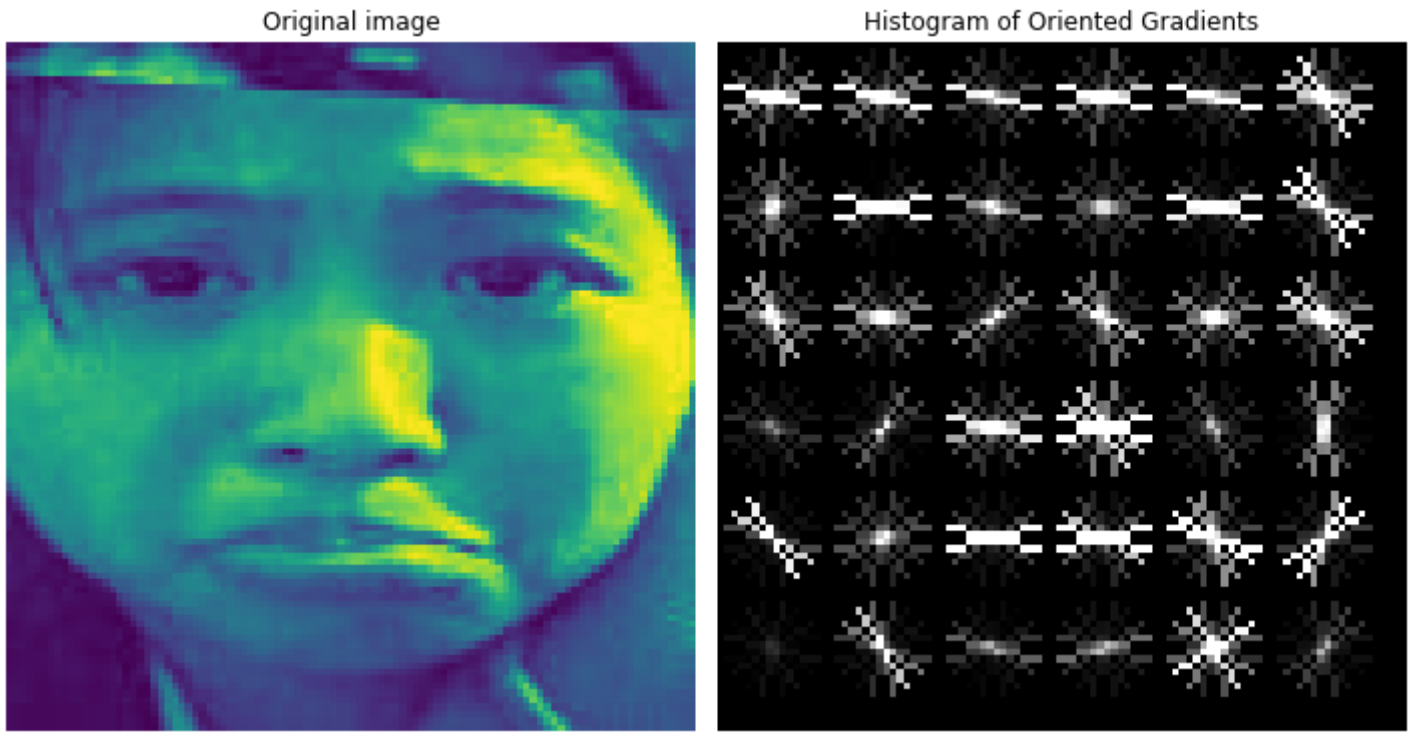
In [ ]:

```
metrics.ConfusionMatrixDisplay.from_predictions(y_test1, y_pred_siftMLP)
plt.show()
```

Extract HOG descriptors for test data.

In [ ]:

```
testhog_images = []
testhog_descriptors = []

X_test_gray = [ color.rgb2gray(i) for i in X_test]

for i in range(len(X_test)):
    # Identify keypoints and extract descriptors with SIFT
    testimg_hog = img_as_ubyte(np.array(X_test_gray[i]))
    testfet_des, testhog_img = hog(testimg_hog, orientations=8, pixels_per_cell=(16, 16), cells_per_block=(4, 4), visualize=True,
    testhog_descriptors.append(testfet_des)
    testhog_images.append(testhog_img)


fig, ax = plt.subplots(1, 2, figsize=(10, 8), sharex=True, sharey=True)

ax[0].axis('off')
ax[0].imshow(X_test_gray[0])
ax[0].set_title('Original image')

# Rescale histogram for better display
testhog_img_rescale = exposure.rescale_intensity(testhog_images[0], in_range=(0, 10))

ax[1].axis('off')
ax[1].imshow(testhog_img_rescale, cmap='gray')
ax[1].set_title('Histogram of Oriented Gradients')
fig.tight_layout()
plt.show()


## CODING REFERENCE: Code was taken from Lab Tutorial 06 of Computer Vision - IN3060/INM460 module
## and modified as per the requirement in this task.
## https://www.kaggle.com/code/manikg/training-svm-classifier-with-hog-features/notebook
```



Load best SVM model that uses HOG descriptors and then evaluate its performance over test data.

In [ ]:
```
besthogsvmpath = 'drive/My Drive/Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/Models/best_hogSVM_model.pickle'

with open (besthogsvmpath, 'rb') as f:
    best_hogSVM_model_loaded = pickle.load(f)
```

In [ ]:
```
y_pred_hogSVM = best_hogSVM_model_loaded.predict(testhog_descriptors)
```

In [ ]:
```
fig, axes = plt.subplots(2, 5, figsize=(14, 7), sharex=True, sharey=True)
ax = axes.ravel()

for i in range(10):
    ax[i].imshow(X_test[i])
    ax[i].set_title(f'Label: {y_test[i]} \n Prediction: {y_pred_hogSVM[i]}')
    ax[i].set_axis_off()
fig.tight_layout()
plt.show()
```

In [ ]:
```
hogSVM_test_score = best_hogSVM_model_loaded.score(testhog_descriptors, y_test)
hogSVM_test_score
```

Out[ ]:
```
0.6734028683181226
```

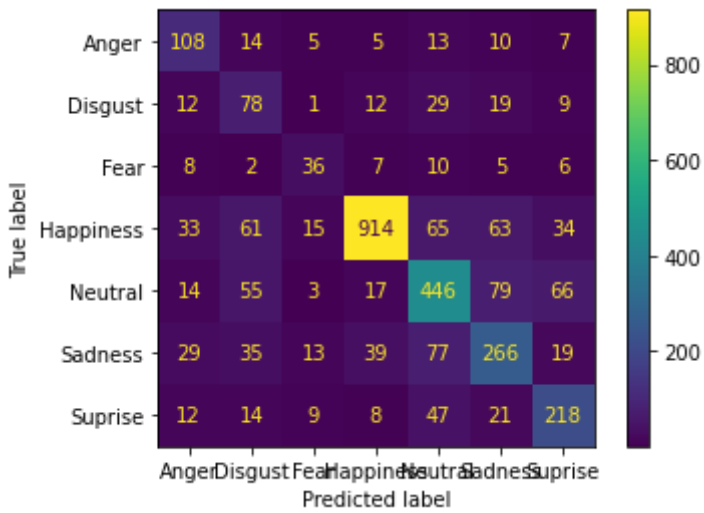
In [ ]:
```
print(f"""Classification report for classifier best hogSVM- {best_hogSVM_model_loaded}:
      {metrics.classification_report(y_test, y_pred_hogSVM)}\n""")
```

```
Classification report for classifier best hogSVM- SVC(class_weight='balanced'):
                    precision    recall  f1-score   support

       Anger       0.50      0.67      0.57       162
     Disgust       0.30      0.49      0.37       160
        Fear       0.44      0.49      0.46        74
   Happiness       0.91      0.77      0.84      1185
     Neutral       0.65      0.66      0.65       680
```

```
     Sadness       0.57      0.56      0.57       478
     Suprise       0.61      0.66      0.63       329

    accuracy                           0.67      3068
   macro avg       0.57      0.61      0.58      3068
weighted avg       0.70      0.67      0.68      3068
```

In [ ]:

```
metrics.ConfusionMatrixDisplay.from_predictions(y_test, y_pred_hogSVM)
plt.show()
```



Load best MLP model that uses HOG descriptors and then evaluate its performance over test data.

In [ ]:

```
besthogmlppath = 'drive/My Drive/Colab Notebooks/Computer Vision Coursework/CW_Folder_PG/Models/best_hogMLP_model.pickle'

with open (besthogmlppath, 'rb') as f:
    best_hogMLP_model_loaded = pickle.load(f)
```

In [ ]:

```
y_pred_hogMLP = best_hogMLP_model_loaded.predict(testhog_descriptors)
```

In [ ]:

```
fig, axes = plt.subplots(2, 5, figsize=(14, 7), sharex=True, sharey=True)
ax = axes.ravel()

for i in range(10):
    ax[i].imshow(X_test[i])
    ax[i].set_title(f'Label: {y_test[i]} \n Prediction: {y_pred_hogMLP[i]}')
    ax[i].set_axis_off()
fig.tight_layout()
plt.show()
```

In [ ]:

```
hogMLP_test_score = best_hogMLP_model_loaded.score(testhog_descriptors, y_test)
hogMLP_test_score
```

Out[ ]:

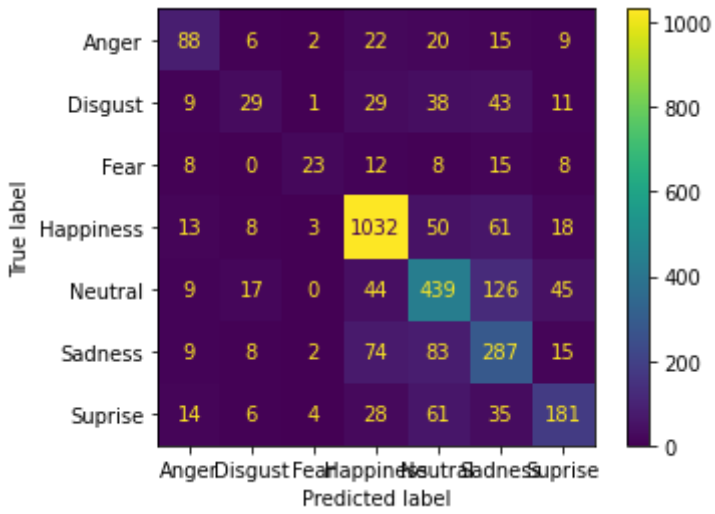
```
0.6776401564537158
```

In [ ]:

```
print(f"""Classification report for classifier best hogMLP- {best_hogMLP_model_loaded}:
      {metrics.classification_report(y_test, y_pred_hogMLP)}\n""")
```

```
Classification report for classifier best hogMLP- MLPClassifier(learning_rate_init=0.01, random_state=1, solver='sgd',
              verbose=True):
```

```
              precision    recall  f1-score   support

       Anger       0.59      0.54      0.56       162
     Disgust       0.39      0.18      0.25       160
        Fear       0.66      0.31      0.42        74
   Happiness       0.83      0.87      0.85      1185
     Neutral       0.63      0.65      0.64       680
     Sadness       0.49      0.60      0.54       478
     Suprise       0.63      0.55      0.59       329

    accuracy                           0.68      3068
   macro avg       0.60      0.53      0.55      3068
weighted avg       0.67      0.68      0.67      3068
```

In [ ]:

```
metrics.ConfusionMatrixDisplay.from_predictions(y_test, y_pred_hogMLP)
plt.show()
```