REPORT ON TASK 4 OF ASSIGNMENT 1

Muntashir Bin Solaiman 22021468

Computer vision

Task 4

a) (b) The code for KNN classification and its accuracy score was:-

```
def classification_accuracy( predictions, labels ):
    return ( predictions == labels ).sum()/len(
predictions )

knnClassifier = sklearn.neighbors.KNeighborsClassifier()
knnClassifier.fit( X_train, y_train )
predictionsFromKNN = knnClassifier.predict( X_test )

accuracy = classification_accuracy( predictionsFromKNN,
y_test )
print(f'{ accuracy*100:5.2f}%')
```

The output was:-

```
28.57%
```

The accuracy score was low. KNN accurately predicted 28.67% of the classes. KNN couldn't predict the class of 71.33% of the testing data.

The code for SVM classification and its accuracy score was:-

```
svm_classifier = sklearn.svm.SVC()
svm_classifier.fit(X_train, y_train)

predictionsFromSVM = svm_classifier.predict(X_test)

accuracy = classification_accuracy( predictionsFromSVM,
y_test )
print(f'{accuracy*100:5.2f}%')
```

The output was:-

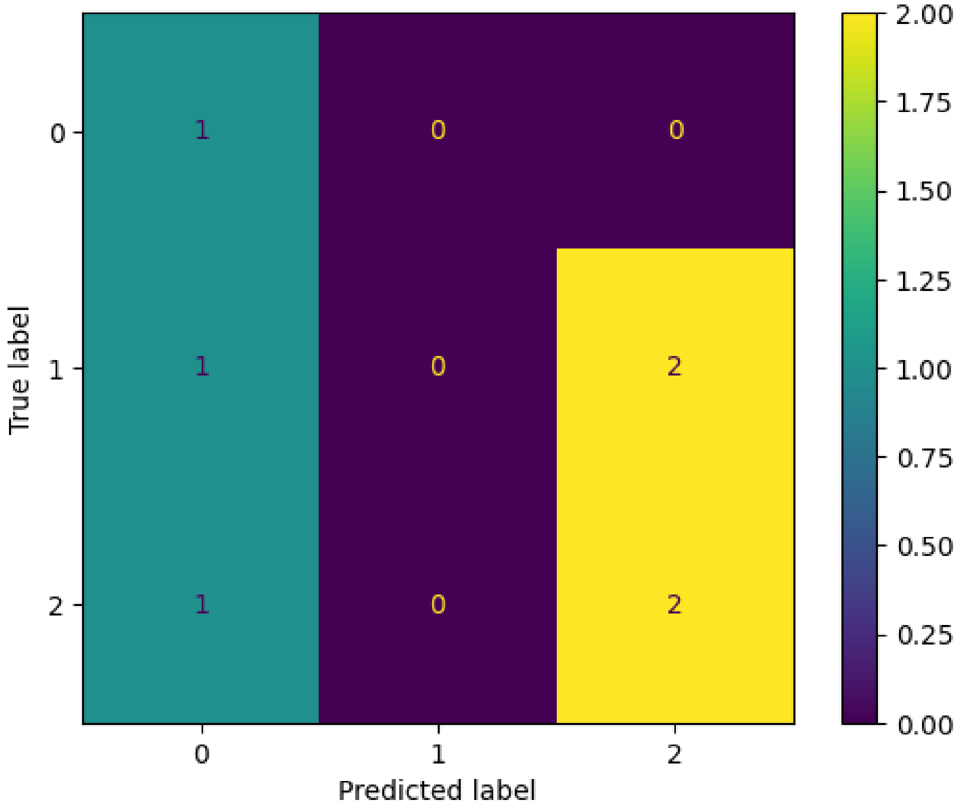
```
14.29%
```

The accuracy score was low. It was lower than KNN. SVM accurately predicted 14.29% of the classes. SVM couldn't predict the class of 85.71% of the testing data.

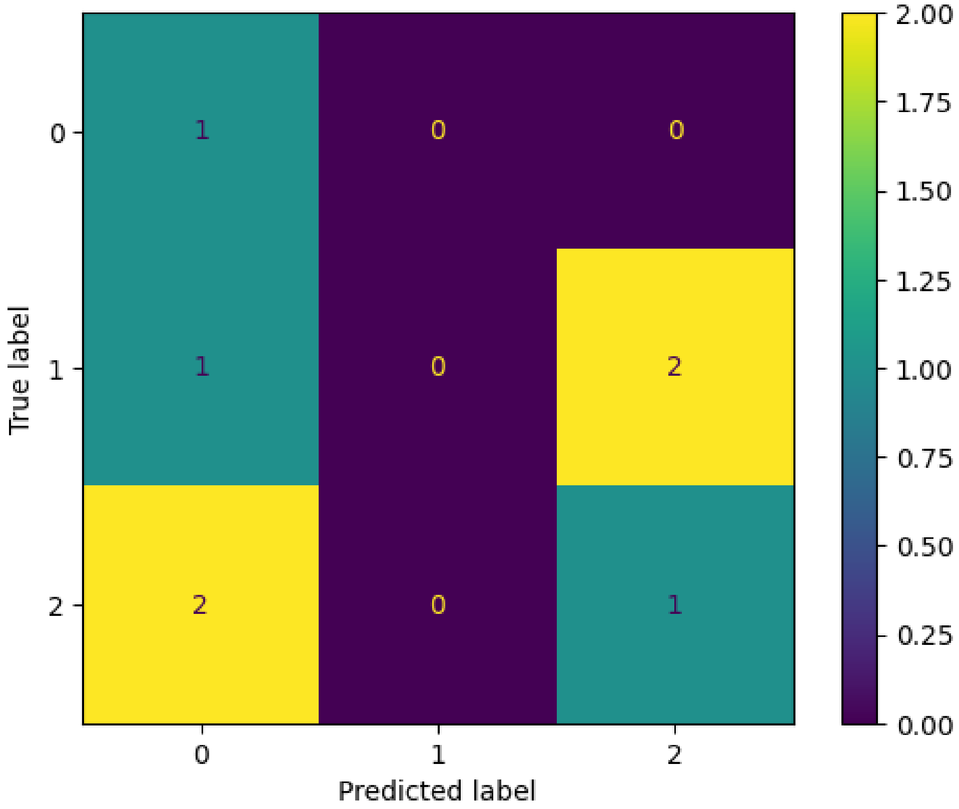
The code for generating and displaying the confusion matrix was as follows:-

```
sklearn.metrics.ConfusionMatrixDisplay.from_predictions(
y_test, predictionsFromKNN )
plt.show()
sklearn.metrics.ConfusionMatrixDisplay.from_predictions(
y_test, predictionsFromSVM )
plt.show()
```

As a result the confusion matrix for KNN was:-



Alternatively, the confusion matrix for SVM was:-



According to the confusion matrix for KNN, horses and birds were predicted accurately. However, ships couldn’t be predicted. To identify which classes created the most errors the false negative and false positives were added . The result were:-

Horse = 2

Ship = 3

Bird = 3

Therefore, ships and bird caused the most errors.

For SVM, Horses and birds were identified correctly. However, ships could’nt. The prediction errors were mostly caused by horses = 3 and birds = 2.

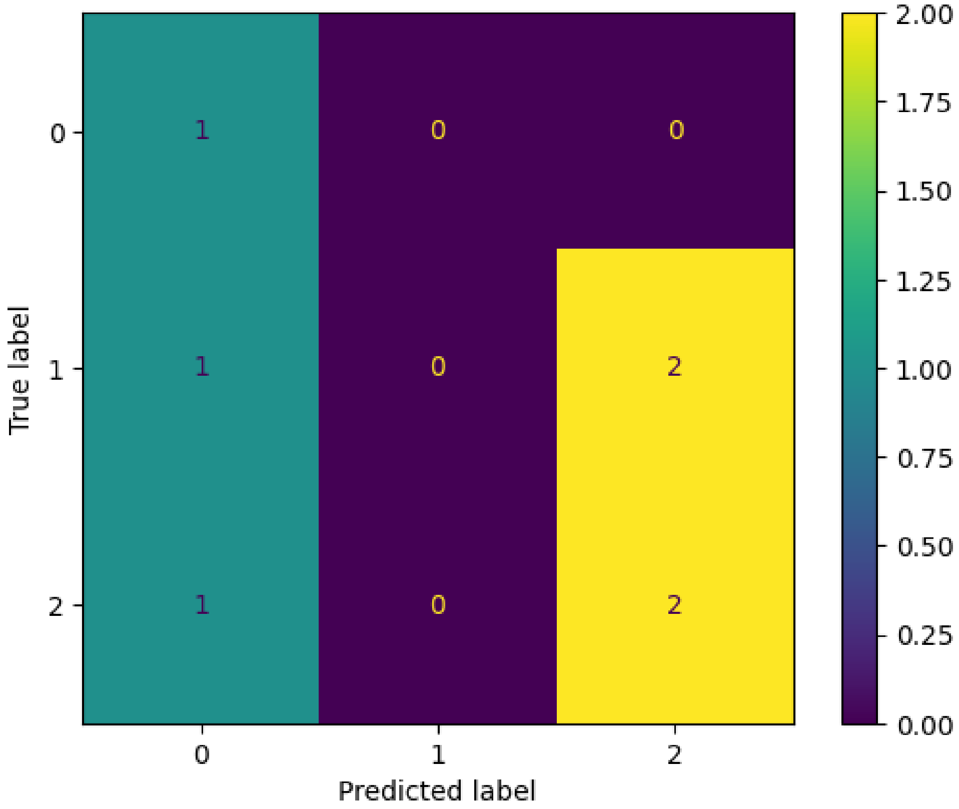
C) KNN:-

With features colour standard deviations and k neighbors = 4 an accuracy score of 57.4% was achieved. With GLCM statistics and neighbor values of 1 and 4 the same score was achieved. The code below used glcm features and k neighbor of 4.

```
def trainKnnClassifierWithNeighbours( features, labels,
nNeighbours, testSize=0.2, randomState=42 ):


    xTrain, xTest, yTrain, yTest = train_test_split(
features, labels, test_size=testSize,
random_state=randomState )

    knnClassifier = sklearn.neighbors.KNeighborsClassifier(
nNeighbours )

    knnClassifier.fit( xTrain, yTrain )

    return knnClassifier, xTest, yTest


def evaluateKnnClassifier( knnClassifier, xTest, yTest ):


    predictionsFromKnn = knnClassifier.predict( xTest )

    accuracy = classificationAccuracy( predictionsFromKnn,
yTest )

    print( f'Accuracy with k={knnClassifier.n_neighbors}:
{accuracy * 100:.2f}%' )

    sklearn.metrics.ConfusionMatrixDisplay.from_predictions(
y_test, predictionsFromKNN )

    plt.show()



features = np.array( glcmFeatures )

labels = np.array( labels )


knnClassifier, xTest, yTest =
trainKnnClassifierWithNeighbours( features, labels, 4 )

evaluateKnnClassifier( knnClassifier, xTest, yTest )
```

The output was:-

```
Accuracy with k=4: 57.14%
```



However, Ships seemed still difficult to identify for KNN. Similarly, ships and bird still caused the most errors.

SVM:-

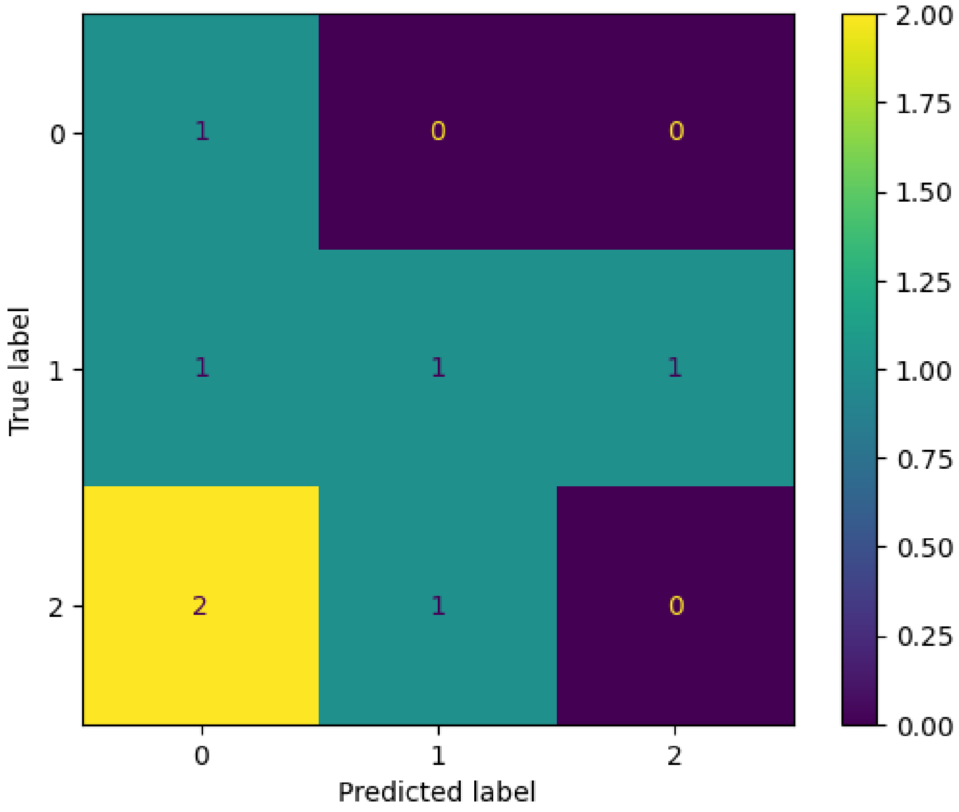
Sigmoid kernel and GLCM features seemed to provide the highest score of 57.14%. The Code used were:-

kernels = ['linear', 'poly', 'rbf', 'sigmoid']

```
def trainSvmClassifierWithKernel( features, labels,
kernel, testSize=0.2, randomState=42):

    xTrain, xTest, yTrain, yTest = train_test_split(
features, labels, test_size=testSize,
random_state=randomState )
    svmClassifier = sklearn.svm.SVC( kernel = kernel )
    svmClassifier.fit(xTrain, yTrain)
    return svmClassifier, xTest, yTest

def evaluateSvmClassifier( svmClassifier, xTest, yTest ):

    predictionsFromSvm = svmClassifier.predict(xTest)
    accuracy = classificationAccuracy(predictionsFromSvm,
yTest)
    print( f'Accuracy with kernel={svmClassifier.kernel}:
{accuracy * 100:.2f}%' )

sklearn.metrics.ConfusionMatrixDisplay.from_predictions(
y_test, predictionsFromSVM )
    plt.show()

features = np.array( glcmFeatures )
labels = np.array( labels )

svmClassifier, xTest, yTest =
trainSvmClassifierWithKernel( features, labels,
kernels[3] )

evaluateSvmClassifier( svmClassifier, xTest, yTest )
```

The outputs were:-

```
Accuracy with kernel=sigmoid: 57.14%
```



The sum of all values were Horse=3, Ships= 3 and bird =4. Birds seemed to cause the most error.