Faculty of Engineering and Technology
Electrical and Computer Engineering Department
MACHINE LEARNING AND DATA SCIENCE
ENCS5341

Fashion-MNIST

Instructor: Dr.Yazan Abu Farha
Section : 1

Prepared by:
Baraa Fatony-1180566
Qussay ZeinEddin-1180235

Date : 11/02/2023

ABSTRACT

In this project,we’re given a set of images(Fashion-MNIST dataset) to classify the Image , using machine learning based techniques. The given images might be T-shirt/top ,Trouser , Pullover , Dress , Coat , Sandal , Shirt , Sneaker , Bag or Ankle boot. We tested machine learning models for classifying images , We’ve achieved 89.32% testing set accuracy using SVM with LBP .

Contents:

1 INTRODUCTION

1.1 Problem Statement

In this task, we're given a set of images of clothes , and we need to classify them to one of ten classes. In this task we limit our work with four supervised models. Along with the image, we just have matrix of pixels .

1.2 Features

As required by the project, we have the raw pixels of the images as features-vector. Each image size is (28 × 28) pixel. In other words, each image will be represented by a 784- dimensional features vector, where the first 28 elements in the vector are the 28 pixels of the first row in the image, and the elements in column 29 to column 56 are the 28 pixels of the second raw in the image, and so on. Also we used MinMaxScaler() to normalize the features.

Some features we extracted from the data to help the model learn better:

- **Edge** : Edges are significant local changes of intensity in a digital image. An edge can be defined as a set of connected pixels that forms a boundary between two disjoint regions , It reduces the amount of data in an image and preserves the structural properties of an image.

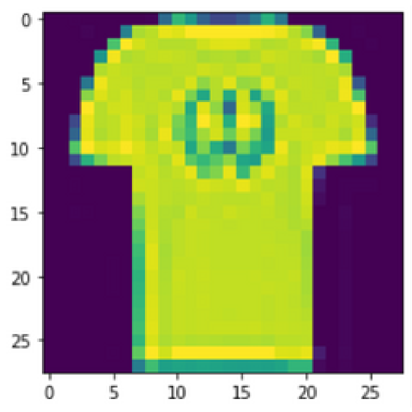


Figure 1.1: Original image

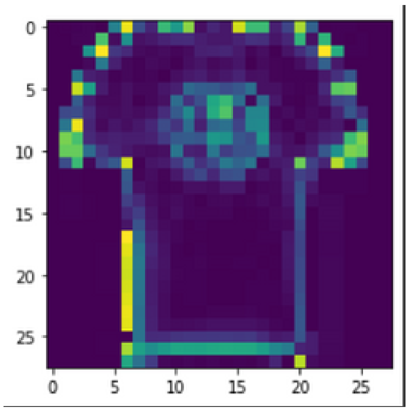


Figure 1.2: Edged image

- **Fast Fourier Transform** : fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image , these frequencies can then be used as new features in machine learning algorithms, providing a more informative representation of the image data.

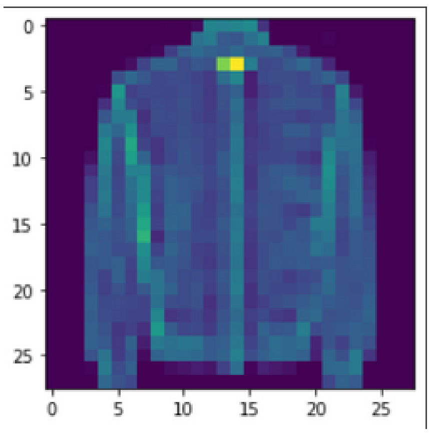


Figure 13: Original image

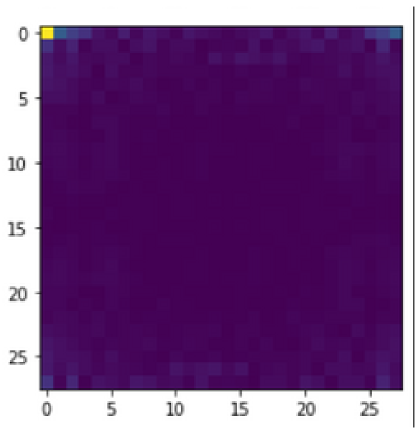


Figure 1.4: FFT of image

- **Local Binary Pattern (LBP)** : is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number , LBP features are easy to calculate and have strong representation ability, and have good effect on describe feature description.

1.3 Training , Testing and Validation Sets

In order to reduce the training time, we are required to split the training data for training and validation sets with randomness and shuffling to tune hyperparameters . For training set, we need to take only 40000 examples , for validation set, we need to take 20000 examples and for testing set , we need to take 10000 examples . Similarity is defined as distance between the feature vectors ,we used manhattan distance equation .

1.4 Models

1.4.1 Baseline model

We will compare our models and findings using a baseline model as a point of comparison. Our starting point would be one of the most straightforward models. We are using K-Nearest Neighbors with K = 3 for this challenge. The pixel is classed in this model based on the training set pixel with the highest similarity. Similarity is defined as distance between the feature vectors ,we used manhattan distance equation .

1.4.2 Support Vector Machines (SVM)

We employed a Support Vector Machine model with a Radial Basis Function(rbf) kernel as an improvement . SVM classifiers are constructed and each one trains data from two classes. To provide a consistent interface with other classifiers, the decision_function_shape option allows to monotonically transform the results of the “one-versus-one” classifiers to a “one-vs-rest” decision function of shape (n_samples, n_classes).

1.4.3 Logistic Regression

Logistic Regression is a classification technique is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme.

1.4.4 Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) is a feedforward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between the input and output layers. MLP uses backpropogation for training the network. This model optimizes the log-loss function using LBFGS or stochastic gradient descent.

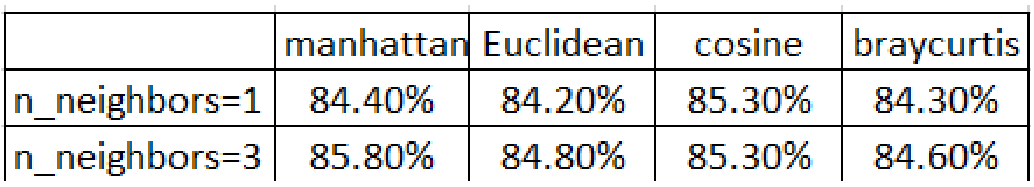
1.5 Evaluation

In order to evaluate our models we used accuracy to know the percentage of predictions it got right , precision to measure the ability of a classifier to identify only the correct instances for each class , Recall to measure the ability of a classifier to find all correct instances per class and F1 score which balances Precision and Recall.

2 Technical Details

For the improvement models, we've used a validation set to tune some of the hyper– parameters:

• **KNN** : for the KNN model we've relied on the default hyper–parameters, Except the number of neighbors and the type of distance used to measure similarity.
We got the table below, showing the accuracy of each training operation on validation set , so depend on maximum accuracy we choose n_neighbors=3 with manhattan distance .



	manhattan	Euclidean	cosine	braycurtis
n_neighbors=1	84.40%	84.20%	85.30%	84.30%
n_neighbors=3	85.80%	84.80%	85.30%	84.60%

Table 2.1: Accuracy in KNN hyper-parameter tuning

• **SVM** : for the SVM model we have tuned the Regularization parameter , it denotesInverse of regularization strength C and the Maximum number of iterations taken for the solvers to converge. We got the table below, showing the accuracy of each training operation on validation set , so depend on maximum accuracy we choose C=1000 and max_iter=100 .

	Linear	Poly	rbf
C=5.0	84.30%	87.80%	90.00%
C=6.0	84.10%	87.90%	89.90%
C=7.0	84.10%	87.90%	89.90%
C=8.0	84.00%	88.20%	89.90%
C=9.0	84.10%	88.10%	90.03%
C=10.0	84.10%	87.70%	90.20%
C=11.0	83.90%	88.50%	90.20%
C=12.0	84.00%	88.10%	90.30%
C=13.0	83.80%	88.10%	90.40%
C=14.0	83.80%	87.60%	89.80%
C=15.0	83.90%	87.70%	89.80%

Table 2.2: Accuracy in SVM hyper-parameter tuning

• **Logistic Regression** : for the Logistic Regression model we have tuned the Regularization parameter , it denotes the strength of the regularization is inversely proportional to C and max_iter which represents the maximum number of iterations.
We got the table below, showing the accuracy of each training operation on validation set , so depend on maximum accuracy we choose C=1000 with maximum iteration =100 .

	C=2	C=5	C=100	C=1000	C=5000
max_iter=100	84.80%	85.20%	85.20%	85.32%	85.25%
max_iter=1000	84.38%	83.90%	83.40%	83.50%	83.48

Table 2.3: Accuracy in Logistic Regression hyper-parameter tuning

• **MLP** : for the MLP model we have tuned the hidden_layer_sizes , it represents the number of neurons in the ith hidden layer , alpha which represents the strength of the L2 regularization term and max_iter which represents the maximum number of iterations .
We got the table below, showing the accuracy of each training operation on validation set , so depend on maximum accuracy we choose hidden_layer_sizes=300 , maximum iteration =300 and alpha = 0.001 .

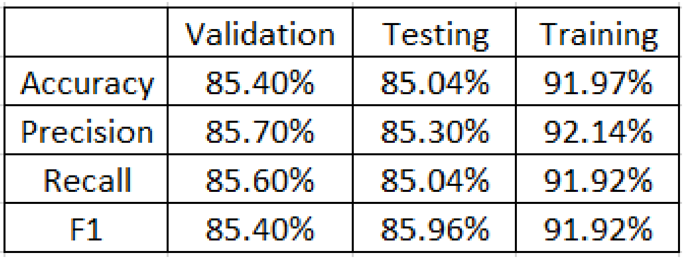
			Accuracy
hidden_layer_sizes=100	alpha=0.005	max_iter=500	83.95%
hidden_layer_sizes=200	alpha=0.0002	max_iter=100	86.81%
hidden_layer_sizes=300	alpha=0.001	max_iter=300	87.45%
hidden_layer_sizes=500	alpha=0.01	max_iter=500	86.16%

Table 2.4: Accuracy in MLP hyper-parameter tuning

3 Experiments and Result

3.1 Baseline Model

Baseline model has achieved 85.04% testing set accuracy and 91.98% training set accuracy. These results are expected from this model (KNN with K = 3 and manhattan distance) , as it does not take into account any features in the image , it only finds the distance and similarity in the pixel values, and in the image histogram (28x28 = 784) some information and the correct arrangement of pixels is lost .

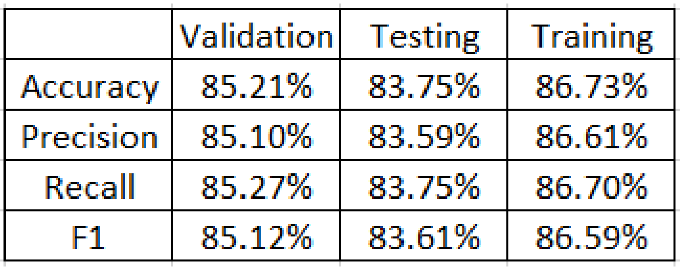


	Validation	Testing	Training
Accuracy	85.40%	85.04%	91.97%
Precision	85.70%	85.30%	92.14%
Recall	85.60%	85.04%	91.92%
F1	85.40%	85.96%	91.92%

Table 3.1: KNN evaluation matrix

3.2 Logistic Regression

Logistic Regression has achieved 83.75% testing set accuracy and 86.73% training set accuracy. It appears that the results and accuracy have become worse in this model, because the logistic regression assumes that the data is divided exactly in a linear way, and it is difficult for it to find complex relationships between the variables, and this is difficult to obtain in a scenario such as a classification of clothes out of 10 classes .



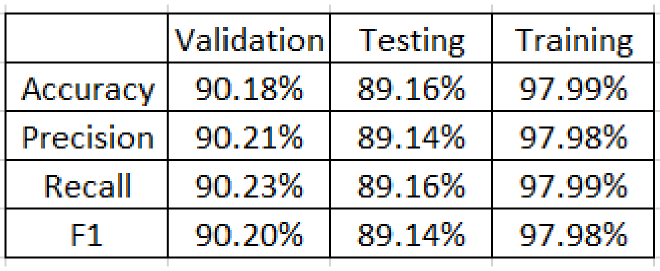
	Validation	Testing	Training
Accuracy	85.21%	83.75%	86.73%
Precision	85.10%	83.59%	86.61%
Recall	85.27%	83.75%	86.70%
F1	85.12%	83.61%	86.59%

Table 3.2: Logistic Regression evaluation matrix

3.3 SVM

Support Vector Machine has achieved 89.16% testing set accuracy and 97.99% training set accuracy.

It appears that the results and accuracy have become better in this model, as SVM model performs well in image classification, and uses the kernel trick to overcome the problem of non-linear separation of data like Logistic Regression model , which maps the data into a higher-dimensional space where the data becomes linearly separable.



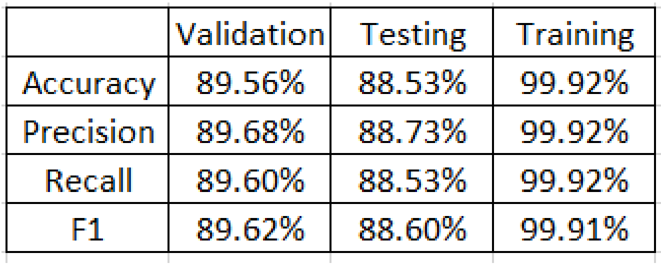
	Validation	Testing	Training
Accuracy	90.18%	89.16%	97.99%
Precision	90.21%	89.14%	97.98%
Recall	90.23%	89.16%	97.99%
F1	90.20%	89.14%	97.98%

Table 3.3: SVM evaluation matrix

3.4 MLP

Multilayer Perceptrons has achieved 88.53% testing set accuracy and 99.92% training set accuracy.

It appears that the results and accuracy have become better in this model, as SVM model performs well when the pixels of an image reduced down to one long row of data and it good for complex problem with for nonlinearly separated data.



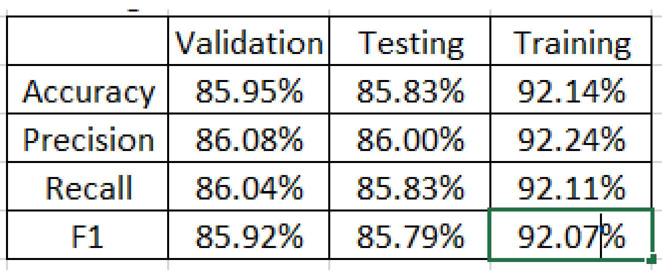
	Validation	Testing	Training
Accuracy	89.56%	88.53%	99.92%
Precision	89.68%	88.73%	99.92%
Recall	89.60%	88.53%	99.92%
F1	89.62%	88.60%	99.91%

Table 3.4: MLP evaluation matrix

3.5 Images with Edge

- **KNN**

Baseline model with edge feature has achieved 85.83% set accuracy(More than the original model by 0.79) and 92.14% training set accuracy (More than the original model by 0.17) .

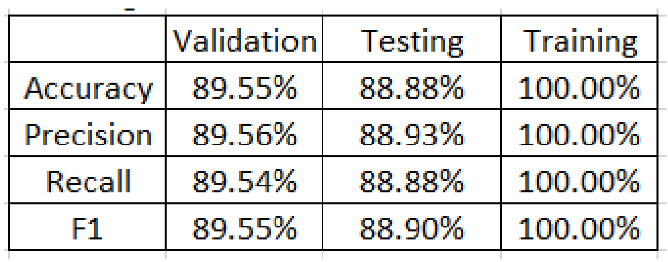


	Validation	Testing	Training
Accuracy	85.95%	85.83%	92.14%
Precision	86.08%	86.00%	92.24%
Recall	86.04%	85.83%	92.11%
F1	85.92%	85.79%	92.07%

Table 3.5: KNN(images with edge) evaluation matrix

- **MLP**

MLP model with edge feature has achieved 88.88% set accuracy(More than the original model by 0.35) and 100% training set accuracy (More than the original model by 0.08) .



	Validation	Testing	Training
Accuracy	89.55%	88.88%	100.00%
Precision	89.56%	88.93%	100.00%
Recall	89.54%	88.88%	100.00%
F1	89.55%	88.90%	100.00%

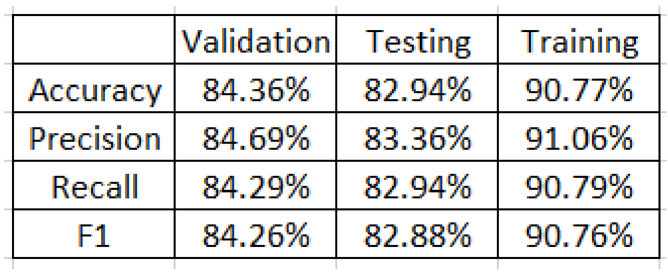
Table 3.6: MLP(images with edge) evaluation matrix

It seems that this feature helps models understand shapes because some classes are similar in pixels, but edges may help classify shapes between similar classes .

3.6 Fourier Transform

- **KNN**

Baseline model with FFT of image has achieved 82.94% set accuracy(less than the original model by 2.1) and 90.77% training set accuracy (less than the original model by 1.2) .

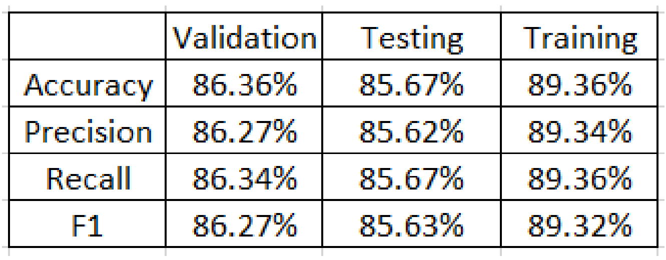


	Validation	Testing	Training
Accuracy	84.36%	82.94%	90.77%
Precision	84.69%	83.36%	91.06%
Recall	84.29%	82.94%	90.79%
F1	84.26%	82.88%	90.76%

Table 3.7: KNN(FFT) evaluation matrix

- **SVM**

SVM with FFT of image has achieved 85.67% set accuracy(less than the original model by 3.49) and 89.36% training set accuracy (less than the original model by 8.63) .



	Validation	Testing	Training
Accuracy	86.36%	85.67%	89.36%
Precision	86.27%	85.62%	89.34%
Recall	86.34%	85.67%	89.36%
F1	86.27%	85.63%	89.32%

Table 3.8: SVM(FFT) evaluation matrix

It seems that this feature does not give models as much information as pixels, and it is possible that the blurring of the image (28x28) does not make this feature compose well.
The Fourier transform has led to a very specific and limited view of frequency in the image so we lose many information that varies.

3.7 Local Binary Pattern (LBP)

- **KNN**

Baseline model with LBP has achieved 85.16% set accuracy(More than the original model by 0.12) and 92.03% training set accuracy (More than the original model by 0.06) .

	Validation	Testing	Training
Accuracy	85.19%	85.16%	92.03%
Precision	85.34%	85.43%	92.04%
Recall	85.14%	85.16%	91.89%
F1	85.03%	85.13%	91.85%

Table 3.9: KNN(LBP) evaluation matrix

- **SVM**

Baseline model with LBP of image has achieved 89.32% set accuracy(less than the original model by 3.49) and 98.02% training set accuracy (less than the original model by 8.63) .

	Validation	Testing	Training
Accuracy	89.96%	89.32%	98.02%
Precision	90.01%	89.30%	98.03%
Recall	90.05%	89.32%	98.02%
F1	90.05%	89.30%	98.00%

Table 3.10: SVM(LBP) evaluation matrix

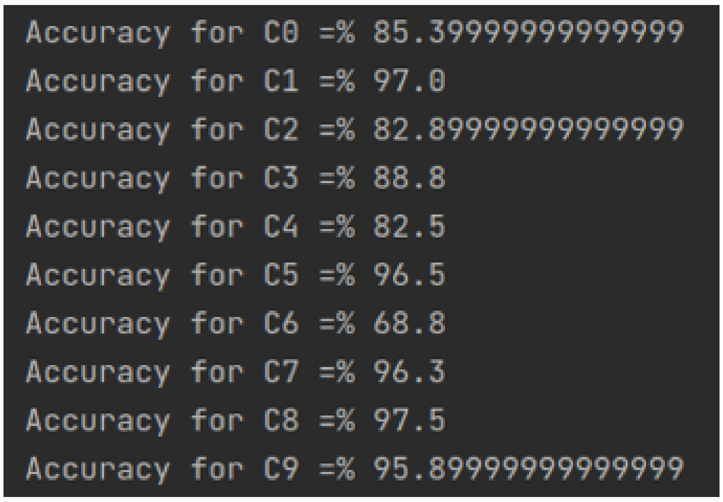
We noted that this feature has increased accuracy slightly, as LBP works on making descriptors efficiently capture the local spatial patterns and the gray scale contrast in an image.

3.8 Error and Evaluation Metrics Analysis

The problem with taking accuracy for model evaluation of multi-class data is that it looks for predictions on the data as a whole, and in a problem like the nature of our project when the data contains ten classifications, It may make us less confident about our results even though the model may have failed to classify just one or two classes out of ten.

The solution is to try to understand the accuracy of the model by classifying each model separately because multiclass problem is broken down into a series of binary problems, this is done using “One-vs-Rest (OVR)” strategy . In OVR, the actual and predicted classes and the corresponding decision scores are re-calculated for each class, combining all other classes as a single class. Basically this generates a binary compilation output, and these results are viewed for each class separately to see which class failed the model.

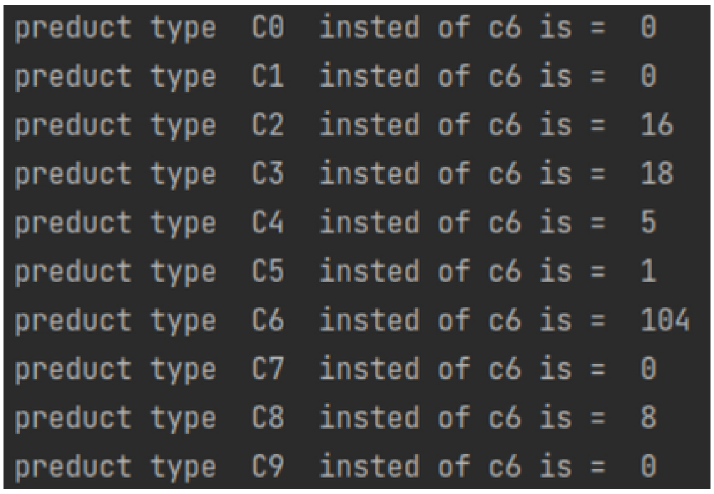
In our project we found the accuracy for each class in SVM and it was found that the lowest accuracy was for class 6 (shirt).



```
Accuracy for C0 =% 85.39999999999999
Accuracy for C1 =% 97.0
Accuracy for C2 =% 82.89999999999999
Accuracy for C3 =% 88.8
Accuracy for C4 =% 82.5
Accuracy for C5 =% 96.5
Accuracy for C6 =% 68.8
Accuracy for C7 =% 96.3
Accuracy for C8 =% 97.5
Accuracy for C9 =% 95.89999999999999
```

Figure 3.1: Accuracy for each class in SNM

Then we find one row of confusion matrix just for class 6 to compare all possible two-class combinations of the dataset to find the largest mixing between any class and class 6.



```
preduct type  C0  insted of c6 is =  0
preduct type  C1  insted of c6 is =  0
preduct type  C2  insted of c6 is =  16
preduct type  C3  insted of c6 is =  18
preduct type  C4  insted of c6 is =  5
preduct type  C5  insted of c6 is =  1
preduct type  C6  insted of c6 is =  104
preduct type  C7  insted of c6 is =  0
preduct type  C8  insted of c6 is =  8
preduct type  C9  insted of c6 is =  0
```

Figure 3.2: total predictions for class 6 from other classes

We noted that class 6(Shirt)looks a lot like classes (2,3) , The model cannot accurately distinguish between classes 3(Dress) ,2(Pullover) , and 6 because of the high similarity between the images, which results in the model's failure . Therefore, in the previous section, we tried to find new features of images that help distinguish them, such as: LBP to make descriptors efficiently capture the local spatial patterns and the gray scale contrast in an image and Edge feature to find the edges of image to try to distinguish , this increased the accuracies slightly, but there was still a weakness in its classification due to the poor quality of the images.

4 Conclusion and Discussions

We have learned during this project how to deal with images and insert pixels for different machine learning models. We also tried during this project to learn some deep learning techniques, which are MLB, and deal with it.

During our work on this project, we noticed that the machine learning models are sensitive to the features added to the images during training, as we forced many of the features not mentioned in the previous section like equivalent diameter to try to increase the accuracy, but we were surprised that these added features reduce the accuracy, not increase it , also we've noticed that the Edge and LBR feature are useful in images, since they define the nature of the image.

In terms of models, we've seen that models have their advantages in some tasks and not the others. For example, Logistic Regression assumes that the classed are linearly separable, that's why it performs even worse than the baseline model, since we're using it outside its domain. While for the RBF SVM we have noticed two main advantages: first, it can generalize and perform well even in the presence of noise, and second, it can handle the case where the classes are not linearly separable by embedding them into higher dimensional space (the kernel trick).

In terms of the results, we have noticed that taking accuracy as an evaluation method for multiclass problems may make us less confident about our results even though the model may have failed to classify one or two classes out of ten, so we tried to use the same equation for accuracy but in different ways (OVR ,OVO and confusion matrix) to understand the accuracy of each class separately and find out the distinguish that occurs in any classes and produce new features accordingly.