

CMPE471/ Machine Learning

Project 2 Report

The “Fruits” Problem

Semester: Spring 2021

Instructor: Dr. Abdulaziz Al-Ali

Students:

|  |  |
| --- | --- |
| Mohamad Alnass | 201704075 |
| Omar Omar | 201704146 |
| Nayef Radwi | 201707337 |

# Abstract

The purpose of this project is to create a multi-class fruit classifier using the perceptron algorithm and one V.S all technique. The project tackles multiple challenges such as feature extraction, implementation, and testing the performance of the classifier during training and testing. Features were extracted to reduce the number of features from 30,000 to 1500 useful features for every fruit image. Implementation of the project was to train a perceptron for each class and then use the confidence of the predictions to classify the image. Finally for performance, the accuracy was calculated for training and for testing, accuracy, recall, precision, and F1 were calculated for each class as well as overall performance were measured.

# Feature Extraction

In order to enhance the performance of the classifier as well as reduce the time taken to train each perceptron, images were preprocessed to extract useful features and reduce the effect of the curse of dimensionality that 30,000 features possesses. The first step of feature extraction was to split the image into sub-areas, for this project the image was split into 10 by 10 areas. After that, the standard deviation, variance, max value, min value, and average were calculated for each area. This process of splitting and calculating was repeated for each channel of the image (which are red, green, and blue). The whole process extracted 1500 features (300 for each calculation method) that are used in the training algorithm.

After looping over every image, the features are combine in N by 1501 array, where N is the number of images, 1500 are the features, and the 1 is the label appended to the end of the features list. This N by 1501 array is then stored in a csv file called.

To show how some of the features extracted affected the image, these are 2 images of 2 different colored fruits with average, standard deviation, and minimum value applied separately on the images:

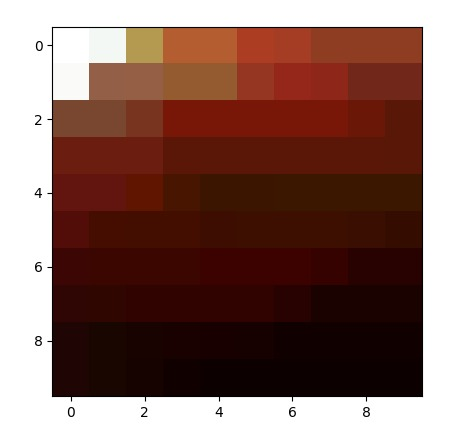
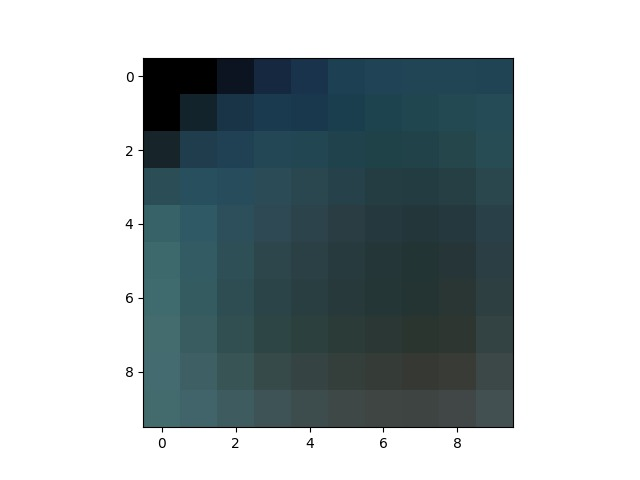
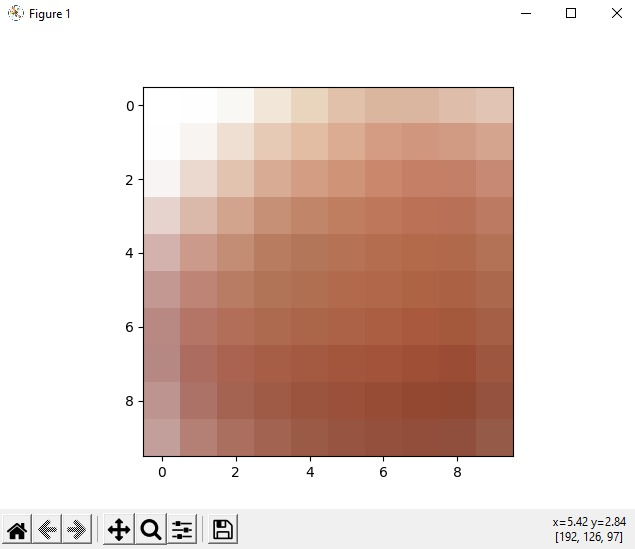


Figure 1 – apple

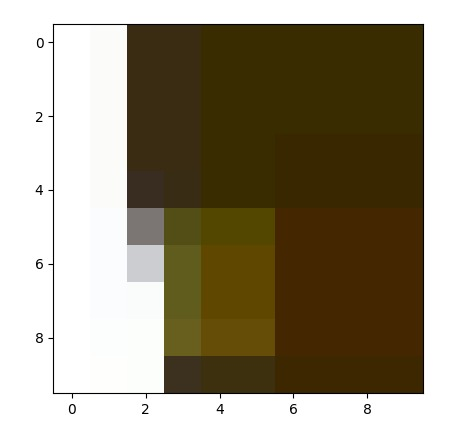
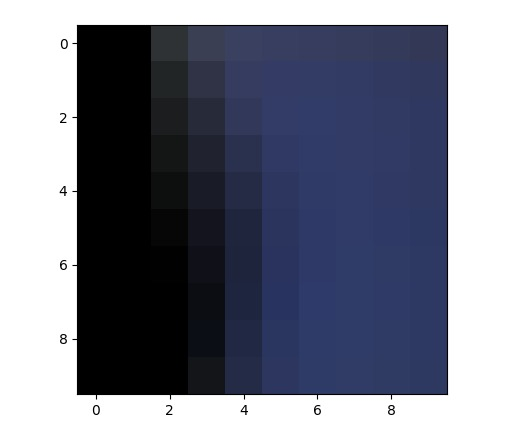
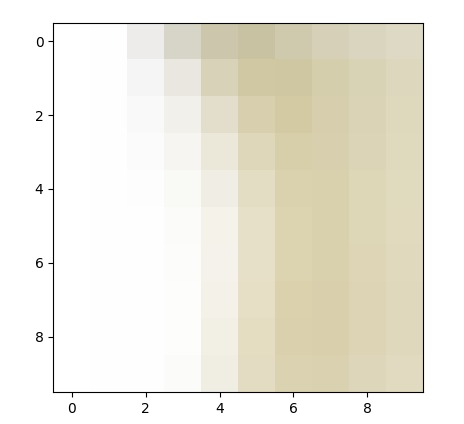


Figure 2 - banana

# Implementation

The classifier was implemented by using the one V.S all technique, this means that each perceptron was trained to classify a specific class (whether it belongs to the class or not). when it comes to prediction the classifier will give the same image to every perceptron and then choose the classification that had the highest positive dot product.

The training of one perceptron was done by feeding the entire training set to the learning algorithm along with the number of epochs (training set is 75% of the entire dataset). The perceptron predicts each sample and then uses the update weights function to modify its weights if necessary. Initially the weights of each perceptron are random with 0.005 as the lower bound and 0.01 as the upper bound. The update weights function is:

Where wi is the weight, is the learning rate, C(x) is the real output, h(x) is the perceptron’s output, and xi is the feature value of the corresponding weight (for the bias it is 1).

This whole process is repeated based on the number of epochs. After each epoch, the training set is shuffled and used again. After each perceptron has been trained on the training set, the weights of all the perceptrons are then stored in csv file to be used later.

# Performance and Testing

After storing the weights, the classifier is then tested on the testing set and the performance of training and testing are measured.

When it came to training the accuracy of the perceptron is plotted against the number of epochs in the graph below

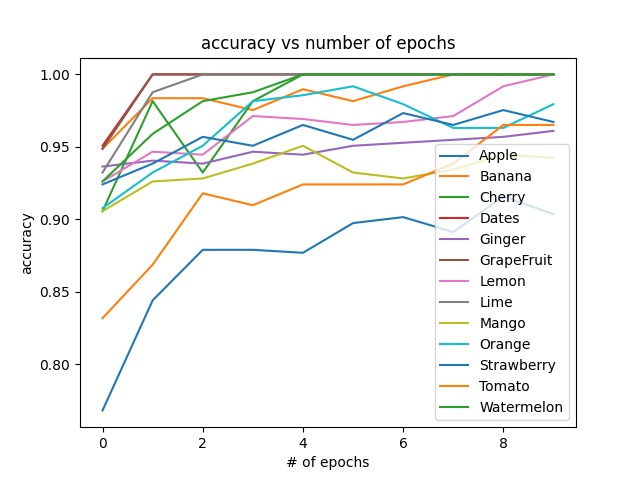


Figure 3 - accuracy vs 10 epochs

As the graph shows, the accuracy increases slightly as the number of epochs increase. After a large number of epochs, the perceptron starts to over fit.

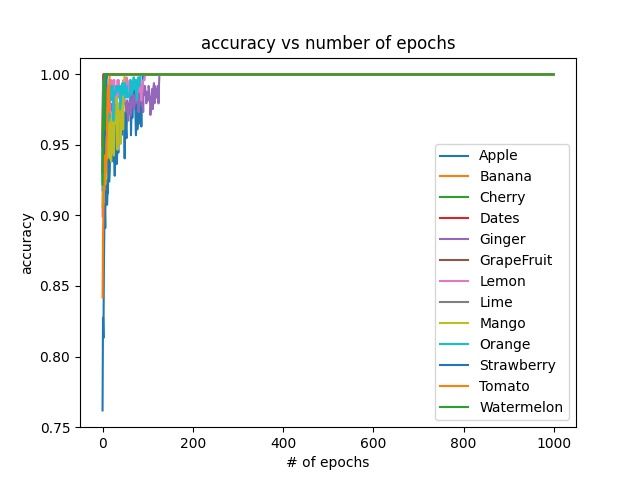


Figure 4 - accuracy vs 1000 epochs

Testing the performance of the entire classifier was done calculating the precision, F1, and recall of each class as well as the macro-recall, macro-precision, and macro F1. These calculations are repeated 3 times for 3 different classifiers.

1. Classifier trained with 10 epochs
2. Classifier trained with 100 epochs
3. Classifier trained with 1000 epochs

## Classifier trained with 10 epochs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| class | F1 | Precision | Accuracy | Recall |
| 0 | 0.918 | 1.000 | 0.969 | 0.848 |
| 1 | 1.000 | 1.000 | 1.000 | 1.000 |
| 2 | 0.810 | 0.739 | 0.951 | 0.895 |
| 3 | 1.000 | 1.000 | 1.000 | 1.000 |
| 4 | 0.800 | 1.000 | 0.994 | 0.667 |
| 5 | 0.400 | 1.000 | 0.982 | 0.250 |
| 6 | 0.625 | 0.500 | 0.963 | 0.833 |
| 7 | 0.957 | 0.917 | 0.994 | 1.000 |
| 8 | 0.571 | 1.000 | 0.963 | 0.400 |
| 9 | 0.963 | 0.929 | 0.994 | 1.000 |
| 10 | 0.857 | 0.750 | 0.988 | 1.000 |
| 11 | 0.909 | 0.882 | 0.982 | 0.938 |
| 12 | 0.923 | 0.857 | 0.988 | 1.000 |
| **Macro** | **0.826** | **0.890** | **0.982** | **0.833** |

## Classifier trained with 100 epochs