DWIT COLLEGE

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Institute of Science and Technology

AUTOMATIC FRUIT CLASSIFICATION USING ADABOOST ALGORITHM

A PROJECT REPORT

Submitted to

Department of Computer Science and Information Technology

DWIT College

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted by

Sagar Giri

August, 2016

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

SUPERVISOR'S RECOMENDATION

I hereby recommend that this project prepared under my supervision by SAGAR GIRI entitled **"AUTOMATIC FRUIT CLASSIFICATION USING ADABOOST ALGORITHM"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

…………………………………………

Rituraj Lamsal

Lecturer

Deerwalk Institute of Technology

DWIT College

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

LETTER OF APPROVAL

This is to certify that this project prepared by SAGAR GIRI entitled **"AUTOMATIC FRUIT CLASSIFICATION USING ADABOOST ALGORITHM"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

…………………………………… Rituraj Lamsal [Supervisor] Lecturer DWIT College	………………………………………… Hitesh Karki Chief Academic Officer DWIT College
………………………………………….. Jagdish Bhatta [External Examiner] IOST, Tribhuvan University	………………………………………….. Sarbin Sayami [Internal Examiner] Assistant Professor IOST, Tribhuvan University

ACKNOWLEDGEMENT

I cannot express enough thanks to DWIT college committee for their continued support and encouragement. I would like to thank my supervisor Mr. Ritu Raj Lamsal for his continuous support and expert advice and encouragement throughout this difficult project, as well as Mr. Sarbin Sayami for his advice in report organization and story writing. Also, I would like to thank Mr. Hitesh Karki for his continuous encouragement to do this project.

My completion of this project could not have been accomplished without the support of my associate, Mr. Subigya Kumar Nepal. I offer my sincere appreciation for the learning opportunities and resources provided by the DWIT College.

My heartfelt thanks to my friends for your encouragement when the times got rough are much appreciated and duly noted.

Sagar Giri
TU Exam Roll no: 1809/069

STUDENT’S DECLARATION

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

... ...

Sagar Giri

Date: ... ...

ABSTRACT

Object Recognition is an important study in Computer Science. Object recognition is emerging technology to detect and classify objects based on their characteristics. Fruit recognition and automatic classification of fruits is also a domain of object recognition and it is still a complicated task due to the various properties of numerous types of fruits. Different fruits have different shapes, sizes, color, textures and other properties. Similarly, some of the fruits like Tangerines and Mandarin Oranges share the same characteristics like color, texture, size, etc. This project aims to find a better way of a fruit classification method using supervised machine learning algorithms and image processing mechanisms based on multi-feature extraction methods. Firstly, we pre-process the training sample of fruits' images. Preprocessing includes separating foreground and background, scaling and cropping the image to reduce the dimension so that the processing is fast. Then, we extract features from the fruit's image, which includes color, texture and shape of the fruit image. Extracted features are then fitted into the AdaBoost classifier machine learning algorithm. Finally, the results obtained from the machine learning network are cross validated with the test sample. The output obtained will give us the prediction accuracy and class of the fruit that it has acknowledged. Experimental results have been collected using a fruit image database consisting of 5 different classes of fruits and 120 fruits images overall. Therefore, average prediction accuracy of more than 55% is obtained with a learning rate of 0.7.

Keywords: Classification, Feature Extraction, AdaBoost Classifier, Object Recognition, Fruit Classification

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

PCA	Principal Component Analysis
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
SIFT	Scale Invariant Feature Transform
RF	Random Forest
LIN	Linear
HPOL	Homogeneous Polynomial
GRB	Gaussian Radial Basis
WTA	Winner-Takes-All
MWV	Max-Wins-Voting
DAG	Directed Acyclic Graph
CPM	Critical Path Method
WBS	Work Breakdown Structure
HTML	Hypertext Markup Language
CSS	Cascading Styles Sheet
IBM	International Business Machines

CHAPTER 1: INTRODUCTION

Object Recognition implements pattern recognition of different objects. Pattern recognition builds up from different areas such as statistics and machine learning. To achieve good object detection, classification and recognition, different machine learning algorithms and object's feature extraction algorithms are used. While using machine learning algorithms, it is not a guarantee that every algorithm gives accurate result. The achievement of accuracy can be different for different algorithms. Hence, we need to select the (best) algorithm with the highest classification and prediction accuracy. Also, while training the system, proper learning rate also plays a vital role.

For fruit classification and detection this project implements a portion of computer vision and object recognition with machine learning model. The rapid development of computer vision, image processing and recognition, advancement in computer technology provides the possibility of fruit classification through computer vision. In recent years, fruit recognition using computer vision is being gradually applied in agriculture sector, education sector and supermarkets [1]. Computer vision has been widely used in industries to aid in automatic checking processes [2]. The important problem in computer vision and pattern recognition is shape matching. Shape comparison and shape matching can be carried out by using computer vision and image processing algorithms. Shape matching applications contain image registration, object detection and recognition, and image content based retrieval [3]. Many agricultural applications use image processing to automate their duty. Detecting crop diseases are one of these applications in which the crop images are analyzed in order to discover the affected diseases [4].

1.1 Problem Statement

Despite of advancement in computer vision, image processing, recognition and advancement in computer technology, automatic fruit classification is a challenging task. The primary parameters that play vital role while classifying a fruit include the machine learning algorithm that is being used, quality of images in the fruit database, fruit's images' shape and size and fruit's color. Secondary parameters that affect the classification are similar characters of fruits like color, shape, size, etc. If both primary and secondary parameters are not analyzed properly in the beginning then it may cause problem during classification and may lead to less accuracy and unexpected results.

Many related works have been conducted in fruit classification using different classification algorithms but those approaches still lack in some aspects. A research in fruit classification has been carried out by just taking only three fruits into consideration with 100% accuracy [5]. However, considering only three fruit in the sample is not enough because the trained model may not recognize the fruit's images' that are out of the training sample.

Similarly, proper implementation of machine learning algorithm should also be taken into consideration while performing classification. Using Multiclass SVM algorithm, gives success rate or accuracy in range of 70% to 75% [6]. However, this model may provide wrong interpretation or recognition for fruits with similar features like shape, size, color, texture, etc.

Some approaches are only focused on one feature while others combine two features, resulting in distinctive methods. However, different fruit images can have same color and shape, which may pose a problem. So, it is required to have more features to make the recognition process more robust and effective.

1.2 Objective of the Project

The main objectives of this project are:

a) To extract at least three features from the fruit's image. The features extracted are Haar-Like Features [7], Hue Histogram Feature or Color Histogram [8] and Edge Histogram Feature [9].

b) To implement the AdaBoost classifier algorithm [10] for automatic fruit classification.

c) To develop a web interface platform for testing the prediction of fruit image.

1.3 Scope and Limitation of the Project

1.3.1 Scope

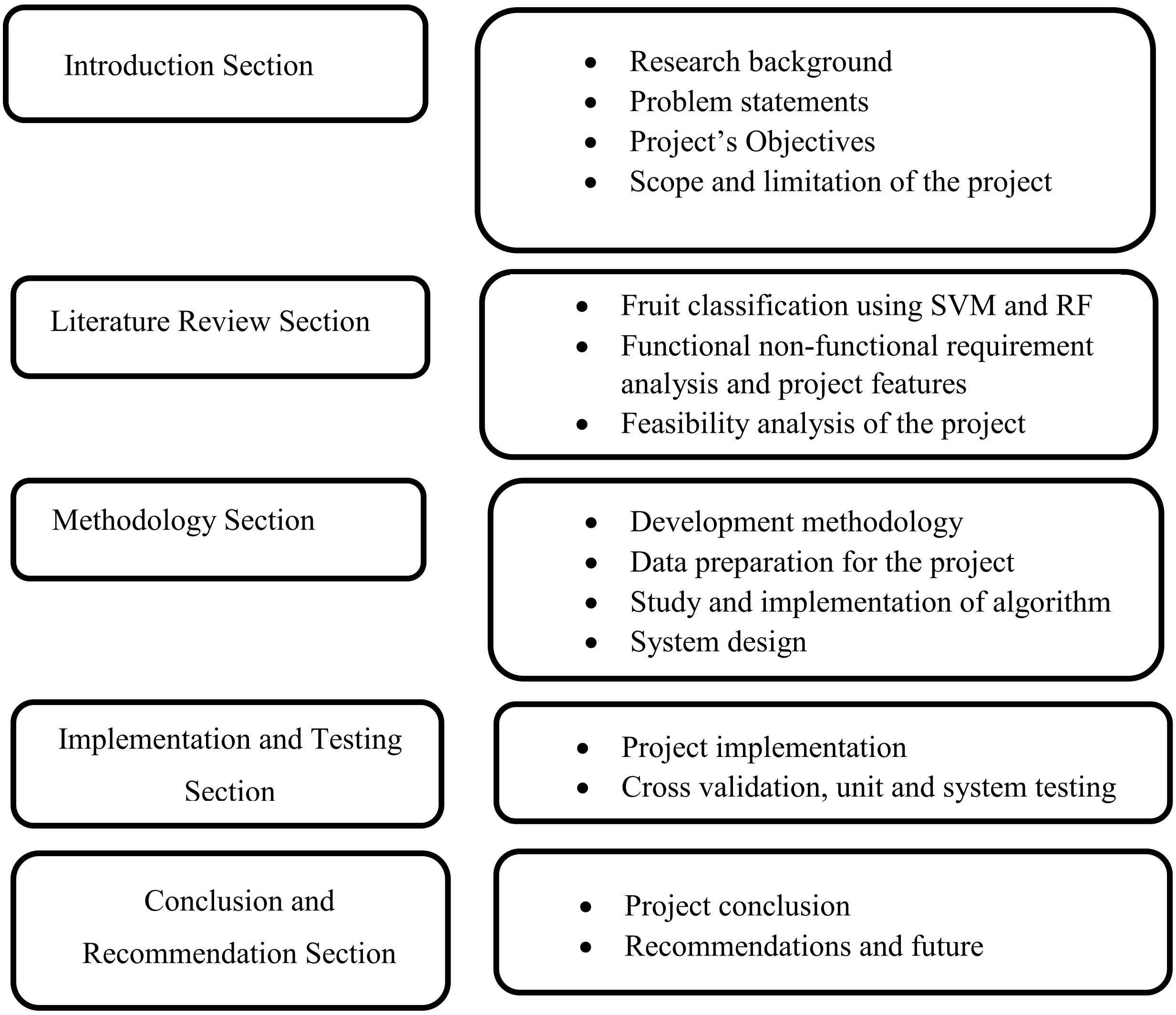
The scope of this project is only limited to edible fruits that are available in the fruit data sets which is used to train the system. Leafy and other vegetables like lettuce, cabbage, spinach, etc. may not be in the scope of this project. So, when images of these are provided as input to the system, it may not recognize and may not produce the desired result.

1.3.2 Limitation

For this project only 5 categories of fruits are used to train the system, hence, there might be false prediction and misclassification of fruits that are out of the training classes. Similarly, only three features from the fruit's image are extracted to reduce the processing complexities which possibly will limits the prediction accuracy score.

1.4 Report Organization

This project report is organized as shown in block diagram in Figure 1:



Introduction Section

- Research background
- Problem statements
- Project's Objectives
- Scope and limitation of the project

Literature Review Section

- Fruit classification using SVM and RF
- Functional non-functional requirement analysis and project features
- Feasibility analysis of the project

Methodology Section

- Development methodology
- Data preparation for the project
- Study and implementation of algorithm
- System design

Implementation and Testing Section

- Project implementation
- Cross validation, unit and system testing

Conclusion and Recommendation Section

- Project conclusion
- Recommendations and future

Figure 1 - Project Block Diagram

CHAPTER 2: LITERATURE REVIEW

Background

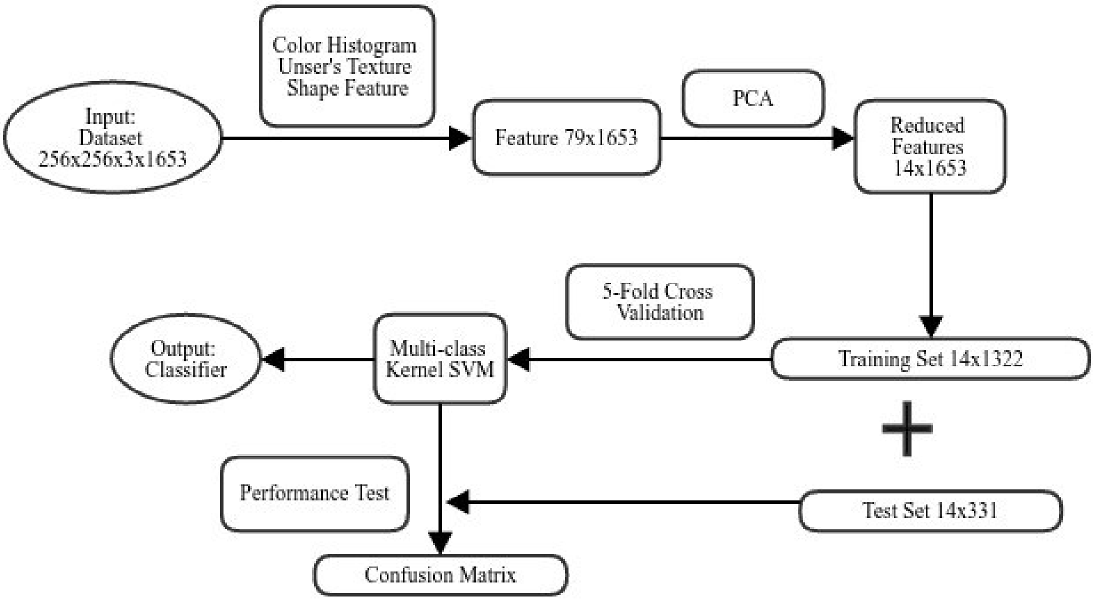
Object recognition has been studied for more than four decades [11]. Significant efforts have been paid to develop representation schemes and algorithms aiming at recognizing generic objects in images taken under different imaging conditions (e.g., viewpoint, illumination, and occlusion). Within a limited scope of distinct objects, such as handwritten digits, fingerprints, faces, and road signs, substantial success has been progress towards object categorization from images has been made in the recent years [13]. Object recognition along with machine learning algorithm and image processing have been implemented in fruit classification as well. Some of the works related to fruit classification are mentioned below:

Fruit Classification using SVM

Yudong Zhang and Lenan Wu have implemented fruit classification in their "Classification of Fruits Using Computer Vision and a Multiclass Support Vector Machine" [6]. In their research, first the fruit images were acquired by the digital camera. Second pre-processing of the image was carried out by removing the background of each image by split-and-merge algorithm. The input was a database of 1,653 images with 18 categories of fruits, and each image size was 256x256. Third, the color histogram, texture and shape features of each pre-processed image were extracted to compose a feature space. Altogether 79 features were extracted from each 256x256 image. These 79 feature included 64 color features, seven texture features, and eight shape features. Fourthly, principal component analysis (PCA) was used to reduce the dimensions of feature space. achieved. Object recognition is also related to content-based image retrieval and multimedia indexing as a number of generic objects can be recognized [12]. In addition, significant

Principal component analysis (PCA) is an efficient tool to reduce the dimensionality of a data set consisting of a large number of interrelated variables while retaining the most significant variations [14]. By applying PCA, 79 features were reduced to 14. These 1,653 images were divided in the proportion of 4:1. Out of 1,653 images, 1,322 image sets were considered as training set and 331 image sets were considered as test sets. Test sets was constructed by randomly sampling in each group. Finally, the SVMs were trained using 5-fold stratified cross validation with the reduced feature vectors as input. The training sets (1,322 fruit images) was used to train the multi-class SVM. The weights of SVM were adjusted to make minimal the average error of 5-fold cross validation. The test data sets (331 fruit images) was used to analyze the performance of the classifier and to calculate the Confusion Matrix. If test was acceptable, then output the classifier, otherwise re-train the weights of SVM. The experimental results demonstrated that the Max-Wins-Voting SVM with Gaussian Radial Basis kernel achieves the best classification accuracy of 88.2%. While considering the computation time, the performance of Directed Acyclic Graph SVMs was founded the swiftest.

According to Yudong Zhang and Lenan Wu, the flowchart created is shown in Figure 2.



(Source: Yudong Zhang and Lenan Wu)

Figure 2 - Flowchart for fruit recognition system

Their experiment were carried out on a Pentium 4 IBM platform with 3 GHz main frequency and 2GB memory under Microsoft Windows XP operating system. The proposed algorithm was developed and implemented on Matlab 2011b (The Mathworks$^{\copyright}$) platform.

The SVM Results of their proposed algorithm are as follows:

For the experiment, three kinds of multi-class SVMs were constructed. Which were Winner-Takes-All SVM (WTA-SVM), Max-Wins-Voting SVM (MWV-SVM), and Directed Acyclic Graph SVM (DAG-SVM). In addition to that three kinds of kernels were chosen, i.e., Linear kernel (LIN), Homogeneous Polynomial kernel (HPOL), and Gaussian Radial Basis kernel (GRB). The experiment result were as follows:

Table 1 - Classification accuracy of SVMs

	LIN	HPOL	GRB
WTA-SVM	48.1%	61.7%	55.4%
MWV-SVM	53.5%	75.6%	88.2%
DAG-SVM	53.5%	70.1%	84.0%

Table 2 - Computation time of SVMs (in seconds)

	LIN	HPOL	GRB
WTA-SVM	8.439	9.248	11.522
MWV-SVM	1.680	1.732	1.917
DAG-SVM	0.489	0.403	0.563

The experiment conclusion was that, using of Max-Wins-Voting SVM (MWV-SVM) with Gaussian Radial Basis (GRB) kernel performed better in terms of classification but considering the computation speed, Directed Acyclic Graph SVM (DAG-SVM) with Gaussian Radial Basis (GRB) kernel performed better.

Automatic fruit classification using random forest algorithm

Cairo University, Faculty of Computers and Information, Cairo, Egypt Scientific Research Group in Egypt (SRGE) conducted a research titled “Automatic fruit classification using random forest algorithm” [5]. For this research, three fruits; i.e., Apples, Strawberry, and Oranges were analyzed and several features were extracted based on the fruits’ shape, color characteristics as well as Scale Invariant Feature Transform (SIFT). Their experiments were tested and evaluated using a series of experiments with 178 fruit images. Their experiment compared the accuracy of the Random Forest (RF) based algorithm with other different algorithms. The experiment result showed that the Random Forest (RF) based algorithm provides better accuracy compared to the other well-known machine learning techniques such as K-Nearest Neighborhood (K-NN) and Support Vector Machine (SVM) algorithms.

Automatic fruits classification system from a collection of images were carried out through a classification system which includes the following stages:

a) Preprocessing stage: in this stage, the images were resized to 90 x 90 pixels to reduce their color index.
b) Feature extraction stage: two feature extraction methods were used in this stage. The first one extracts the shape and color features. While, the second feature extract method uses the Scale Invariant Feature Transform (SIFT) [15].
c) Classification stage: in this stage, they implemented the Random Forests (RF) algorithm to classify the fruit image in order to recognize its name.

The Random Forest (RF) works as described in the algorithm below:

a) Draw M_{tree} bootstrap samples from the original data
b) For each of the bootstrap samples, grow an unpruned classification tree
c) At each internal node, randomly select n_{try} of the N predictors and determine the best split using only those predictor
d) Save tree as is, alongside those built thus far (Do not perform cost complexity pruning)
e) Forecast new data by aggregating the forecasts of the M_{tree} trees.

The selected fruit types are chosen to represent the similarities and differences between shape and color. In the following 3 experiments, 178 images were considered as input dataset and recognition system was run twice for 60% for training and 40% for testing, then 70% for training and 30% for testing. Then KNN, SVM and RF algorithms were applied. Three group of fruits were:

a) Orange and strawberry: fruits are different in both color and shape.
b) Apple and orange: fruits are different in color and similar in shape.
c) Apple and strawberry: fruits are different in shape and similar in color.

When the dataset is divided into 60% training and 40% testing, it was observed that strawberry image achieves high accuracy than orange image. Using shape and color as feature extraction archives the lower accuracy when classifying fruit images by KNN (71.42% orange and 72.72% strawberry) and RF (87.50% orange and 90.91% strawberry).

When system was run with 70% training and 30% testing, it achieved high accuracy when using SIFT as feature extraction with both KNN and SVM classifiers, whereas RF classifier achieved 100% accuracy when extract features by shape and color. Hence, their experiment concluded that RF based algorithm provides better accuracy compared to the other well-known machine learning techniques such as K-NN and SVM algorithms.

2.1 Requirement Analysis

The requirement analysis for this project is broken down into functional and nonfunctional requirements and each are discussed below.

2.1.1 Functional requirements

The functional requirements of this project are:

a) Implementation of an ensemble machine learning algorithm and save the trained data in a local storage so that we don’t have to train the system again.

b) Provide a web interface to test the data.

Functional requirements for the web interface to test the system is shown in the use-case diagram in Figure 3.

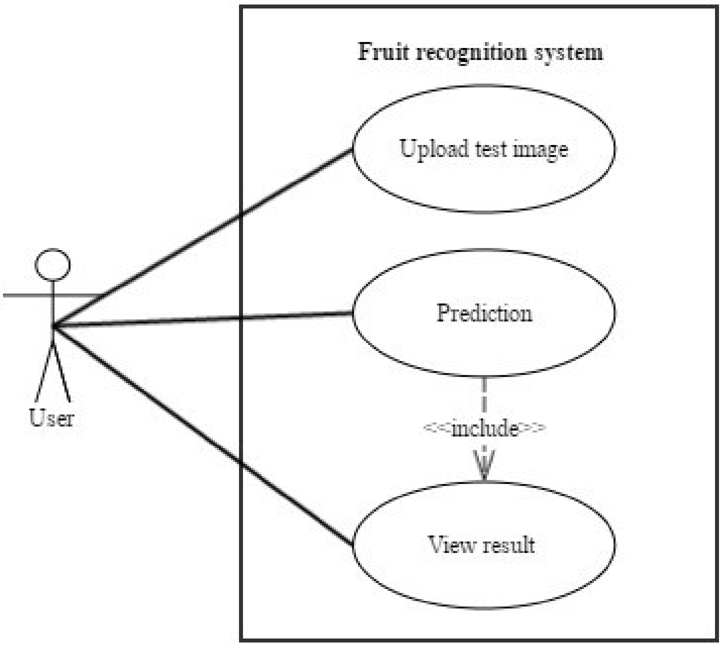


Figure 3 - Use case diagram for fruit recognition system

2.1.2 Non-functional requirements

The non-functional requirements of this project are:

a) Popular boosting algorithm AdaBoost, introduced in 1995 by Freund and Schapire [16] is the ensemble machine learning algorithm that will be implemented to train the system.

b) The trained model will be saved in the local file storage so that it can later be accessed and we can apply our prediction method to it.

c) A web interface will be developed to test the trained data. Web interface will be developed in Flask [17]. The web interface will contain an upload button through which user can upload an image. And another button "Predict" which runs the testing algorithm internally and renders the result in a web page. The rendered result will also have placeholders to show the extracted features of the image and histograms associated with it.

2.1.3 Project features

Result visualization is the key feature of the project. In visualization, user is prompted with the predicted fruit label and the probability of prediction accuracy of that fruit which is based on the input provided to the system. It is done in web interface by rendering the amount of accuracy percentage calculated by the classifier algorithm that is being implemented. The screenshots of the result visualization in the web interfaces are provided in the Appendix II.

2.2 Feasibility Analysis

After gathering of the required resource, whether the completion of the project with the gathered resource is feasible or not is checked using the following feasibility analysis.

2.2.1 Technical feasibility

The project is technically feasible as it can be built using the existing available technologies. The tools, modules and libraries needed to build the system are open source, freely available and are easy to use.

2.2.2 Operational feasibility

The project is operationally feasible as the user having basic knowledge about computer and Internet can use while concept of Machine learning is a plus point. Furthermore the system built in the project can be easily tested by users' if the computer have Internet access and browser is installed in computer.

2.2.3 Schedule feasibility

The schedule feasibility analysis is carried out using the CPM method. With CPM, critical tasks were identified and interrelationship between tasks were identified which helped in planning that defines critical and non-critical tasks with the goal of preventing time-frame problems and process bottlenecks. The CPM analysis was carried out as follows:

First, the activity specification table with WBS is constructed as shown in Table 3.

Table 3 - Activity specification with WBS

Activity	Time (days)	Predecessor
Data Collection (A)	1	-
Research on previous works (B)	5	-
Research on Machine Learning algorithms (C)	4	B
Image preprocessing (D)	2	A
Image feature extraction (E)	4	A, D
Machine training (F)	10	C, E
System testing (G)	8	A, F
Documentation (H)	15	B, F, G

Then, identification of critical path and estimates for each activity are analyzed with the help of network diagram which is based on the Table 3. The network diagram is shown in Figure 4.

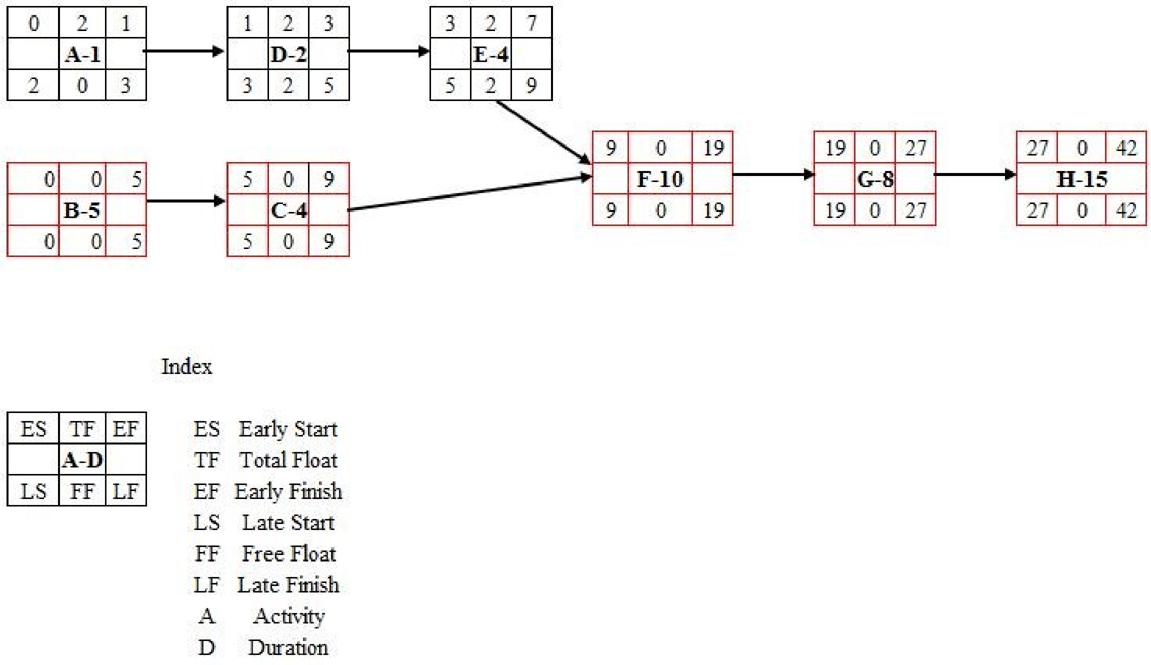


Figure 4 - Network diagram to identify Critical Path

As shown in Figure 4, it is observed that the critical tasks are (B) research on previous works, (C) research on machine learning algorithms, (F) machine training, (G) system testing and (H) documentation because it is clearly seen that the Early Finish (EF) and Late Finish (LF) of these activities are same. The total duration of the critical path (B-C-F-G-H) is 40 days, which is in the project deadline range. Hence, this project is feasible in terms of the schedule since, the project is completed in time if the critical tasks are carried out within the specified tasks duration in Table 3.

CHAPTER 3: METHODOLOGY

The methodology implemented while doing this project are discussed below.

3.1 Development Methodology

The waterfall development model was followed for this project because it is simple, easy to understand and to use it. Since, it is an individual project, it becomes easy to manage due to the rigidity of the model and each phase has specific deliverables and a review process. It was used because the model phases are processed and completed one at a time and these phases do not overlap. Also the requirements are well understood at the beginning of this small project, so, the waterfall model is helpful in this. In testing and validating the project, this model posed some difficulties such as it is very difficult to go back and change something that was not well-thought out in the concept stage.

3.2 Data Preparation

3.2.1 Data collection

Due to project deadline, manual data collection was not performed. Rather, the fruit image data sets were recycled from the previous research [18]. The data consist of 30 fruits categories and each category of fruit has approximately 30 images.

These images were used to train the system using AdaBoost classifier machine learning algorithm. The set of data is definitely not enough to cover wider range and classes of fruit, but due to the project being solely based on the implementation of the machine learning algorithm to automatically classify an image rather than being used as a business product and also because of the tight schedule, these data sets were considered enough for the purpose of the project. Some of the sample images used in this experiments are as shown in Figure 5. Other detailed samples are kept at Appendix I.

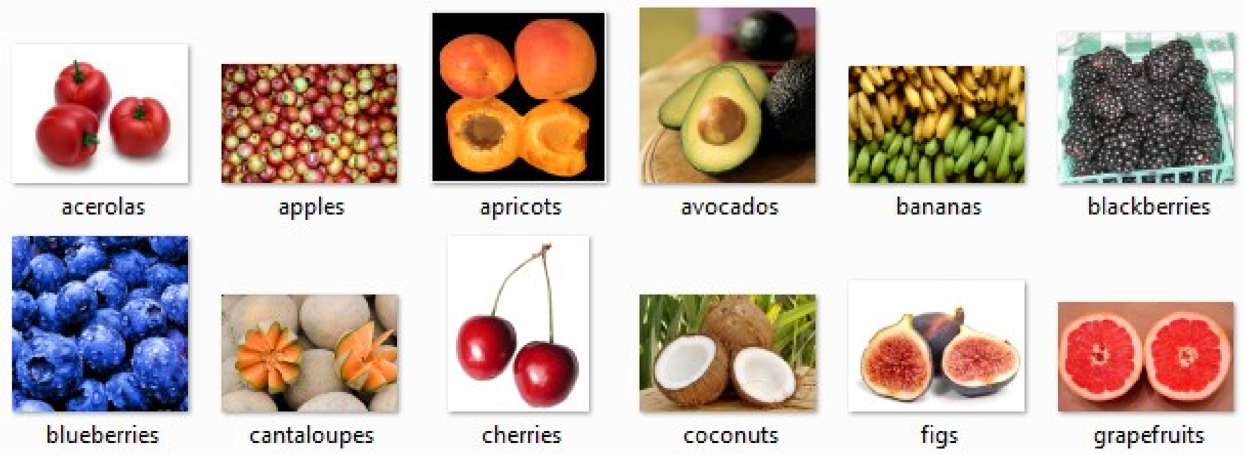


Figure 5 - Sample of fruit images

3.2.2 Data selection

There are around 971 image datasets of 30 fruits' categories in the collected data. Out of 30 fruit category, only five categories are selected for training purpose which has 120 images. Out of 120 images, 90% is used for the training purpose, whereas remaining 10% are used for the testing purpose. The fruit categories for this project are chosen randomly.

3.3 Algorithms Studied and Implemented

For this project, the ensemble machine learning algorithm is used to train the system. And three image processing algorithm for image feature extraction is used for feature extraction. These feature extraction algorithms are: Hue Histogram Feature extraction, Haar-Like Feature extraction and Edge Histogram Feature extraction algorithm. For machine learning, AdaBoost ensemble algorithm is used in this project for classification of fruits. AdaBoost algorithm was used because it can be employed to advance the functioning of any machine learning algorithm. It is best used with weak learners. Weak learners are the models that have achieved accuracy just above random chance on a classification problem.

3.3.1 Algorithms

Different algorithms are implemented in this project. Algorithms implemented are image processing, feature extraction algorithm and machine learning algorithm.

Image feature extraction algorithms are listed and described below:

a) Haar-Like Feature Extraction

This is used to generate Haar-Like features from an image. These Haar-Like features are used by the classifiers of machine learning to help identify objects or things in the picture.

a) Hue Histogram Feature Extraction

This feature extractor takes in an image, gets the hue channel, bins the number of pixels with a particular Hue, and returns the results. This feature extractor takes in a color image and returns a normalized color histogram of the pixel counts of each hue.

b) Edge Histogram Feature Extraction

This method takes in an image, applies an edge detector, and calculates the length and direction of lines in the image. It extracts the line orientation and length histogram.

The machine learning algorithm used in the project is described below:

In this project, a popular boosting algorithm AdaBoost, introduced in 1995 by Freund and Schapire [16] has been implemented to classify the fruit images. The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction.

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. AdaBoost can be used to boost the performance of any machine learning algorithm. The most suited and therefore most common algorithm used with AdaBoost are decision trees with one level. Because these trees are so short and only contains one decision for classification, they are often called decision stumps.

The AdaBoost Training algorithm is described below:

a) The initial weight for each instance in the training dataset weighted as:

$$\text{weight}(x_i) = 1/n$$

Where x_i is the i th training instance and n is the number of training instances.

b) One weak learning model trained as:

   i. A weak classifier (decision stump) is prepared on the training data using the weighted samples. Only binary (two-class) classification problems are supported, so each decision stump makes one decision on one input variable and outputs a +1.0 or -1.0 value for the first or second class value.

   ii. The misclassification rate for weak learner is calculated for the trained model. Traditionally, this is calculated as:

$$\text{error} = (\text{correct} - N) / N$$

   Where, error is the misclassification rate

   correct are the number of training instance predicted correctly

   N is the total number of training instances.

   iii. This generated error is then modified to use the weighting of the training instances. It is modified as:

$$\text{error} = \text{sum}(w(i) * t_{\text{error}}(i)) / \text{sum}(w)$$

Where, error is the weighted sum of the misclassification rate

w is the weight for training instance i

t_{error} is the prediction error for training instance i.

$t_{error} =$ 1 if misclassified.

$t_{error} =$ 0 if correctly classified.

iv. Then a stage value is calculated for the trained model which provides a weighting for any predictions that the model makes. The stage value for a trained model is calculated as follows:

$$stage = \ln((1\text{-}error) / error)$$

Where, stage is the stage value used to weight predictions from the model

ln() is the natural logarithm

error is the misclassification error for the model.

v. The training weights are then updated by giving more weight to incorrectly predicted instances, and less weight to correctly predicted instances.

The weight of one training instance (w) is updated using:

$$w = w * \exp(stage * t_{error})$$

Where, w s the weight for a specific training instance,

exp() is the numerical constant e or Euler's number raised to a power,

stage is the misclassification rate for the weak classifier

t_{error} is the error the weak classifier made predicting the output variable for the training instance

c) Weak models are added sequentially, trained using the weighted training data.

d) The process continues until a pre-set number of weak learners have been created (a user parameter) or no further improvement can be made on the training dataset.

e) Once completed, pool of weak learners is obtained where each has a stage value.

f) Making predictions in AdaBoost classifier:

i. Predictions are made by calculating the weighted average of the weak classifiers.

ii. For a new input instance, each weak learner calculates a predicted value as either +1.0 or -1.0.

iii. The predicted values are weighted by each weak learner's stage value.

iv. The prediction for the ensemble model is taken as a sum of the weighted predictions. If the sum is positive, then the first class is predicted, if negative the second class is predicted.

For example: 5 weak classifiers may predict the values [1.0, 1.0, -1.0, 1.0, and 1.0]. From a majority vote, it looks like the model will predict a value of 1.0 or the first class. These same 5 weak classifiers may have the stage values [0.2, 0.5, 0.8, 0.2, 0.9] respectively. Calculating the weighted sum of these predictions results in an output of -0.8, which would be an ensemble prediction of -1.0 or the second class.

Pseudocode for AdaBoost classifier algorithm.

```
Set uniform example weights.
for each base learner do
        Train base learner with weighted sample.
        Test base learner on all data.
        Set learner weight with weighted error.
        Set example weights based on ensemble predictions.
end for
```

3.4 System Design

The system is described into two models. Data modeling and process modeling. Both models are described and discussed below.

3.4.1 Data modeling

Data modeling is shown with the help of class diagram:

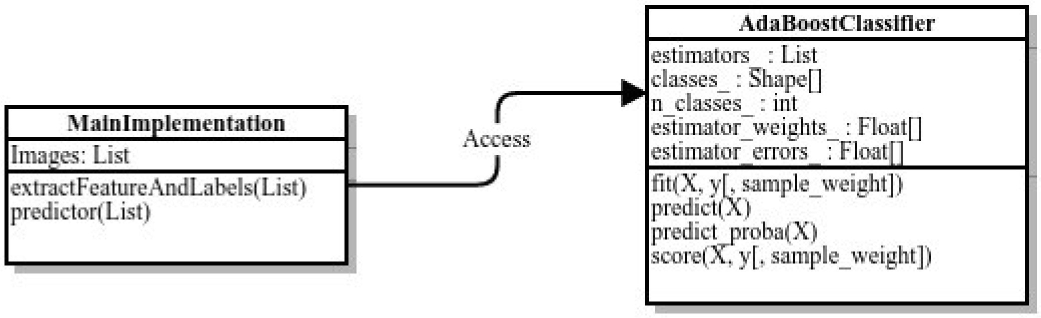


Figure 6 - Class diagram of the system

The data modelling is shown using the class diagram in Figure 6. There are two main classes i.e. AdaBoostClassifier which helps in implementation of AdaBoost machine learning algorithm. This class is used to fit the training samples and to predict the label for input data. Another major class is the MainImplementation. This class is used for the feature extraction from the images and to access the methods of AdaBoostClassifier. Details of this class and its major methods are described in the chapter 4, section 4.1.2.

3.4.2 Process modeling

The process modeling is shown with the help of event diagram and sequence diagram:

a) Event Diagram

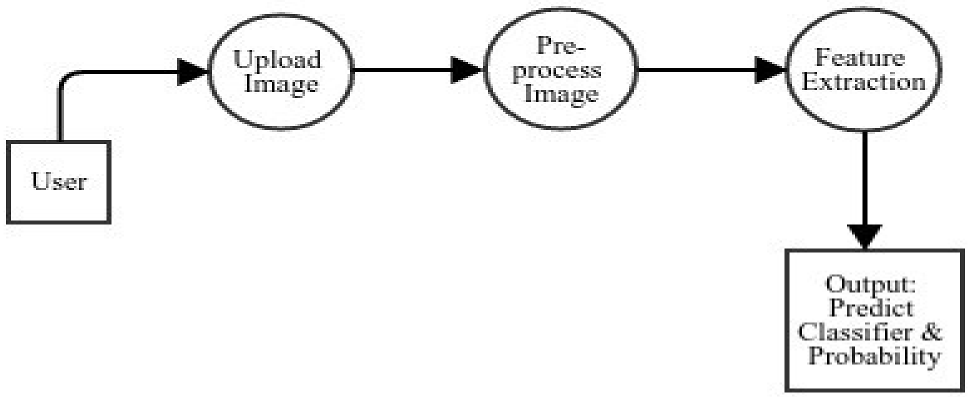


Figure 7 - Event diagram of the system

The Figure 7 explains the events that happens in the system. In this system, the user uploads a valid image in the server. The server than preprocesses the image and feature extraction is carried out. The extracted features are then passed into predict function to predict the class of the fruit.

b) Sequence diagram

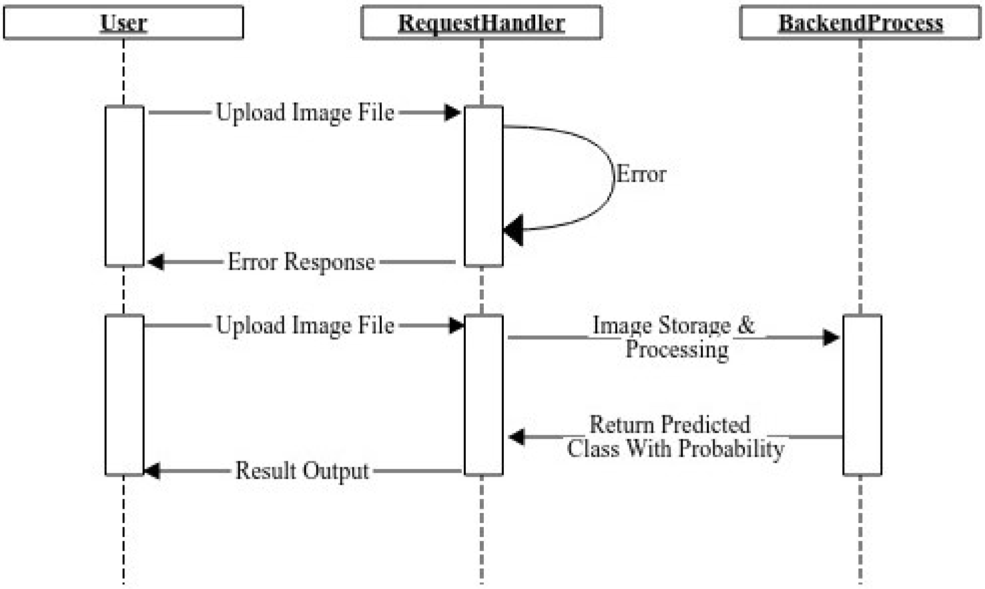


Figure 8 - Sequence diagram of the system

The process modeling is shown with the help of sequence diagram. The sequence diagram describes what happens in the system. As shown in Figure 8, first user uploads the image to the server. The middle tire (i.e) RequestHandler handles the request by the user. If there is some error in the process, a proper error message is thrown. If, upload is successful, then the image is stored in the server's storage for further processing. The BackendProcess of the server preprocesses the image, extracts features, fits into the trained model, and predicts the class and label and outputs to user via a web interface.

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

Automatic fruit classification is the ability to predict the class or label of fruit that has been provided to the system. For fruit classification, we need to implement the machine learning algorithm which is implemented by using a popular machine learning module written in Python built on SciPy.org [19] known as Scikit-Learn [20]. Scikit-Learn was used in this project because it is a simple and efficient tool for data mining and data analysis with different inbuilt and popular machine learning algorithms. Also, it is accessible to everybody, and is reusable in various contexts. It is open source, commercially usable with BSD license. Similarly, for image processing, image manipulation and feature extraction from the images, a simple yet powerful image processing module written in Python and C++ known as SimpleCV [21] was used.

In the implementation process, our collected data had around 971 image datasets of 30 categories of fruits. Out of 30 fruit categories, only five category are chosen at random for the training purpose which has 120 images. Out of 120 images, 90% of the images are used for the training purpose and remaining 10% images are used for the testing purpose. The abstract model of this project is shown in Figure 9. The abstract model shows the system architecture in higher abstraction level.

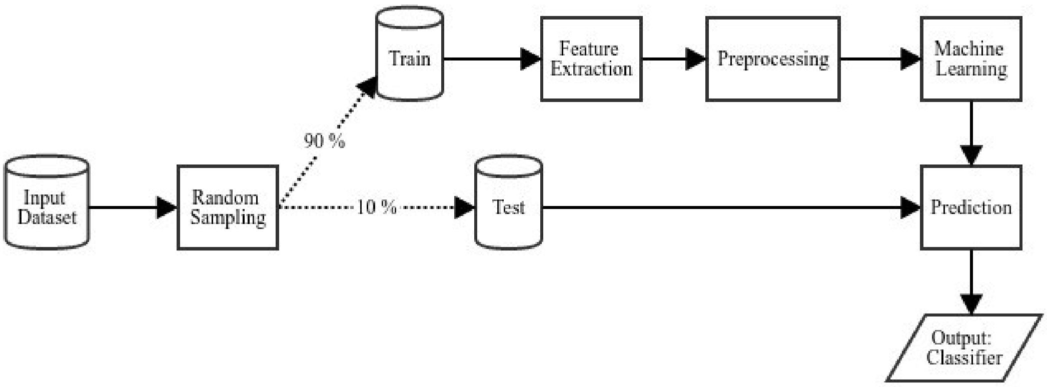


Figure 9 - Abstract model of system

The process flow chart for automatic fruit classification in this project shown in Figure 10.

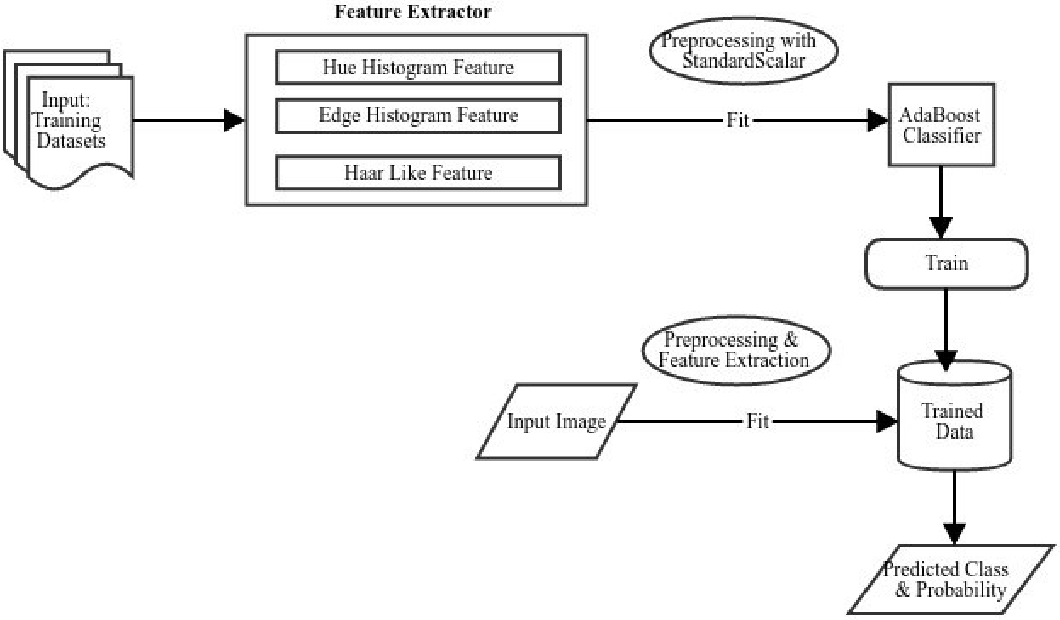


Figure 10 - Flow chart of the system

As shown in Figure 10, each image in the training samples are passed to extract the features of the image. These feature includes, Hue Histogram Feature, Haar-Like Feature and Edge Histogram Feature. The extracted features are first preprocessed or standardized using Standard Scalar. The preprocessed data are then fitted into AdaBoost classifier algorithm which is the final trained data. This trained data is saved in the local storage as an object file. Later to test the input image, feature extraction for the image provided is carried out. The trained data from local storage is loaded and prediction is carried out for the features of new image.

4.1.1 Tools used

Client Side:

1. HTML is used to display content in the browser.
2. CSS is used to properly align the HTML content.
3. Bootstrap CSS framework is used for beautifying the HTML elements to improve the user experience.

Server Side:

1. Python programming language is used to implement the core program logic.
2. Flask web framework is used for dynamic webpage generation and to display the predicted result in the browser as well as to handle page requests for image upload.
3. SciKit-Learn is used to implement the machine learning algorithm.
4. SimpleCV module is used to read and process images in the server side. It is used to read the image from local file storage and to perform image processing and feature extraction methods.

4.1.2 Listing and description major classes and methods

Some of the major classes in this project are described below:

a) AdaBoostClassifier

This class is responsible for the implementation of AdaBoost ensemble machine learning algorithm. The base estimator used in this algorithm is DecissionTreeClassifier. The parameters provided to this class are n_estimators which is maximum number of estimators at which boosting is terminated. Another parameter learning_rate of 0.7 is provided. The learning rate was chosen by iterating the training process which generates the maximum score while cross validating each training output. Some of the important methods of this class are:

i) fit(X, y, sample_weight=None)

Here, X is the training input samples. It is a matrix of input datasets which is feature of the image. y is the target values (class labels). It is a vector (1 Dimensional). Sample_weight is optional. If sample_weight is not provided then the sample weights are initialized to 1/n_samples. This method builds a boosted classifier from the training set (X, y).

ii) predict(X)

This method predicts classes for X (the training input samples or matrix). This method return the predicted classes y (the output vector).

iii) predict_proba(X)

This method predicts class probabilities for X. The predicted class probabilities of an input sample is computed as the weighted mean predicted class probabilities of the classifiers in the ensemble. This method returns the list of class probabilities of the input samples. The order of outputs is the same of that of the classes_ attribute.

b) MainImplementation

This is the main class that helps to access the AdaBoostClassifier class and its methods. Some of the important methods of this class are:

i) extractFeaturesAndLabels(image)

This method is responsible to read the image and extract features and labels from the image. This method implements HueHistogramFeatureExtractor(), HaarLikeFeatureExtractor() and EdgeHistogramFeatureExtractor() from the SimpleCV library to extract the image features. It returns a list which contains the features and labels of the images in the training sets.

ii) predictor(X)

This method is used to use the methods of the AdaBoostClassifier class. This methods takes X (features of input image). The input is used as a parameter for predict(X) and predict_proba(X) method in the AdaBoostClassifier class which eventually returns the predicted class and predicted probability for that class.

4.1.3 Analysis

The analysis for this project is carried out using cross validation score. The cross_validation module in the SciKit-Learn framework provides a method cross_validation_score() which takes in the input matrix and target vector and performs cross validation and returns array of scores of the estimator for each run of the cross validation. In this project, the cross validation score of 0.549 was obtained with the learning rate 0.7, 100 estimators and 10 cross validation (cv) folds.

4.2 Testing

During testing, 12 fruit’s images from the testing datasets were used. For the testing purpose following 4 approaches were developed. Testing was conducted using four test cases:

Test Case 1: Cross validating the trained data to choose the learning rate
Test Case 2: Uploading test image to the server
Test Case 3: Prediction result displaying in the web interface
Test Case 4: Overall system testing

4.2.1 Cross validating the trained data

In this test approach, the extracted features is cross validated to generate mean cross validation score with 100 estimators and learning rate in the range of 0.1 to 1.0. The cross validating process was iterated until the highest score was obtained. The result obtained is shown in the table below:

Table 4 - Mean cross validation score (w.r.t) learning rate

Learning rate	Mean cross validation score
0.1	0.506
0.2	0.519
0.3	0.543
0.4	0.533
0.5	0.542
0.6	0.545
0.7	0.549
0.8	0.543
0.9	0.530
1.0	0.426

As shown in Table 4, the highest mean cross validation score obtained is 0.549 when the learning rate is 0.7. Hence, 0.7 learning rate is chosen to train the system.

4.2.2 Uploading test image to the server

In this test approach, following test case is generated.

TC01 - Image Upload to the server	
Precondition: Fruit image is available in the local storage.	
Assumption: The image is not corrupted and is in standard extension like (png, jpg, jpeg)	
Test steps	1. Navigate to the index page 2. Choose image 3. Click upload button
Expected result: Image should be saved in the storage.	
Generated result: Image was saved in “temp” folder in the home directory.	

4.2.3 Prediction result displaying in the web interface

In this test approach, following test case is generated.

TC02 – Result display in the web interface	
Precondition: Uploading of fruit image to server is executed successfully.	
Assumption: The system is working without any error and user has uploaded a fruit’s image.	
Test steps	1. Navigate to the index page 2. Choose image 3. Click upload button 4. Click the “Predict” button
Expected result: The rendered result must contain the predicted label of fruit with its associated probability.	
Generated result: Predicted fruit name is displayed in the web interface. Result screenshots are mentioned in the Appendix B	

4.2.4 Overall system testing

For the system testing, 12 images (10% of the data sets) were taken into consideration for the testing. Obtained result is summarized in the Table 5 below.

Table 5 - Overall system testing

Correct Prediction	**Incorrect Prediction**
7	5

Overall prediction accuracy of the system is calculated as:

Accuracy = $\frac{7}{12} * 100 = 58.33$ %

CHAPTER 5: CONCLUSION, RECOMMENDATIONS AND FUTURE ENHANCEMENTS

5.1 Conclusion

This project aims to classify the fruit images based on its Haar-Like, Hue and Edge histogram features. The project is designed in such way that it reads image, extracts features, preprocesses it, implements machine learning algorithm and generate output based on the input provided. The project has been able to classify the fruit images based on the fruits features. The cross validation score obtained is 54.9% with learning rate of 0.7 and the prediction accuracy of the system is above 55%.

This result is not satisfactory since the cross validation score and probability of prediction accuracy is very less than what was expected. In some cases, the system doesn't predict the fruit images even the provided input falls under the training category. With this result, it can be concluded that the chosen ensemble machine learning algorithm is not suitable for fruit classification problems.

5.2 Recommendations

It is seen that the AdaBoost ensemble algorithm doesn't perform well for the fruit classification problem. If other ensemble machine learning algorithm such as Random Forest is chosen for these kinds of classification problem, satisfactory result can be obtained. Similarly, as for now only 5 fruits are taken as the training sample and only 3 features are extracted from each image. If more fruits data images are taken as training sample, then the prediction accuracy as well as the cross validation score might increase.

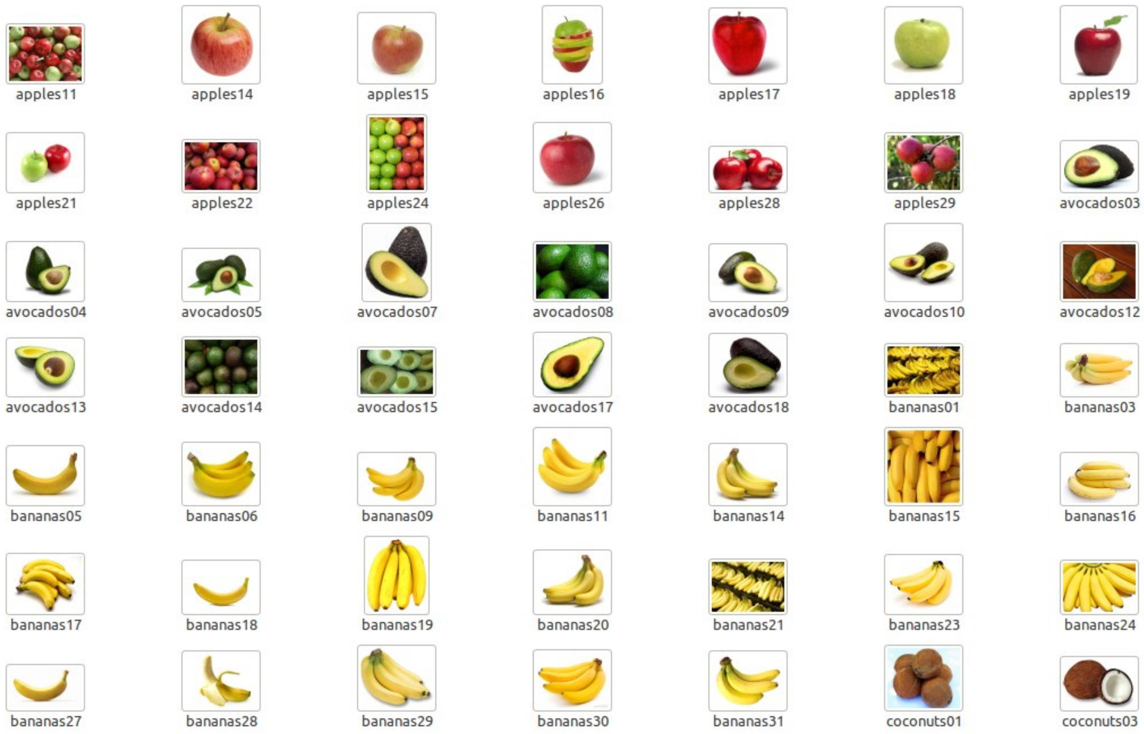
Extracting multiple features from the image has massive impact in the prediction and cross validation score in addition with choosing of different powerful machine learning algorithm might increase the prediction accuracy. However, implementation of neural network can produce more better and accurate results and will be faster as well.

5.3 Future Enhancements

For future enhancements to this project, the trained data and aforementioned algorithm can be tuned further in order to obtain the best results. Extraction of more features other than that are mentioned in the report above will assist in increasing the prediction accuracy. Implementation of better ensemble algorithm other than the AdaBoost algorithm can provide better results.

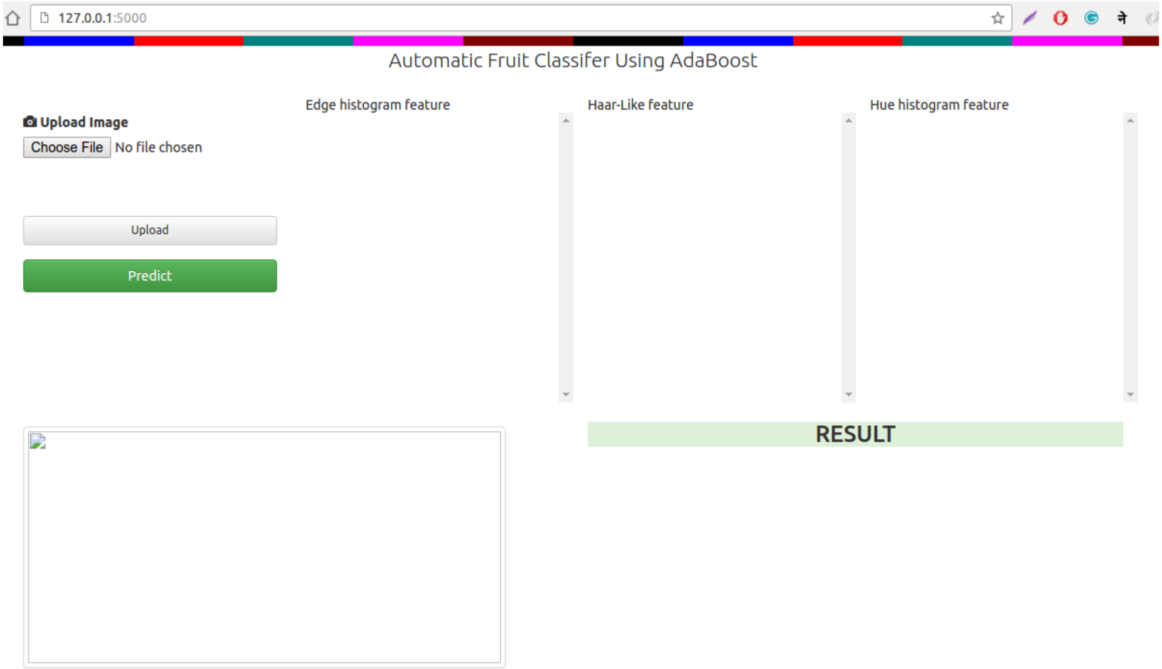
APPENDIX I

Sample of fruit images used for training



APPENDIX II

Snapshot of the landing page



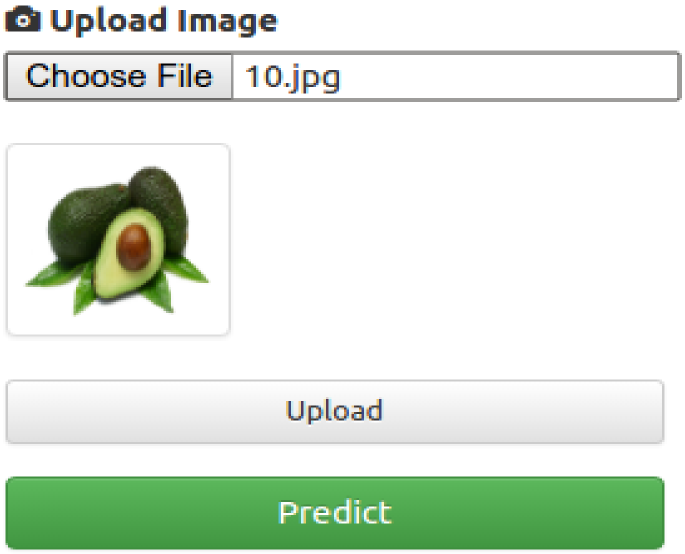
127.0.0.1:5000

Automatic Fruit Classifer Using AdaBoost

Upload Image

Choose File No file chosen

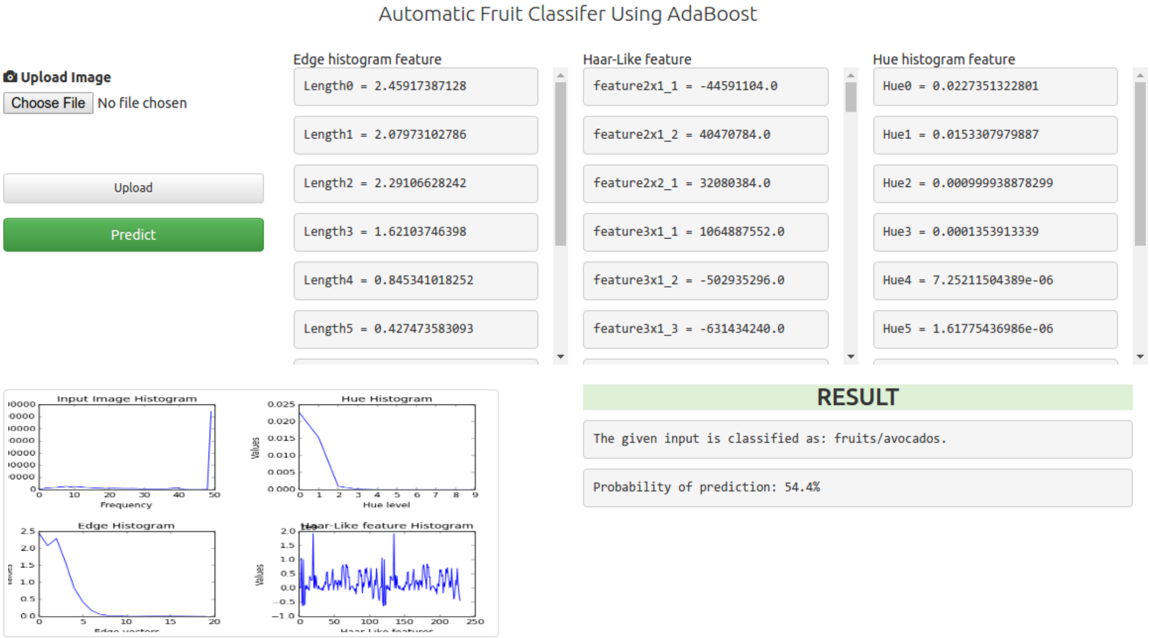
Selected image to predict



Upload Image

Choose File 10.jpg

Result visualization



Automatic Fruit Classifer Using AdaBoost

Edge histogram feature

Length0 = 2.45917387128

Length1 = 2.07973102786

Length2 = 2.29106628242

Length3 = 1.62103746398

Length4 = 0.845341018252

Length5 = 0.427473583093

Haar-Like feature

feature2x1_1 = -44591104.0

feature2x1_2 = 40470784.0

feature2x2_1 = 32080384.0

feature3x1_1 = 1064887552.0

feature3x1_2 = -502935296.0

feature3x1_3 = -631434240.0

Hue histogram feature

Hue0 = 0.0227351322801

Hue1 = 0.0153307979887

Hue2 = 0.000999938878299

Hue3 = 0.0001353913339

Hue4 = 7.25211504389e-06

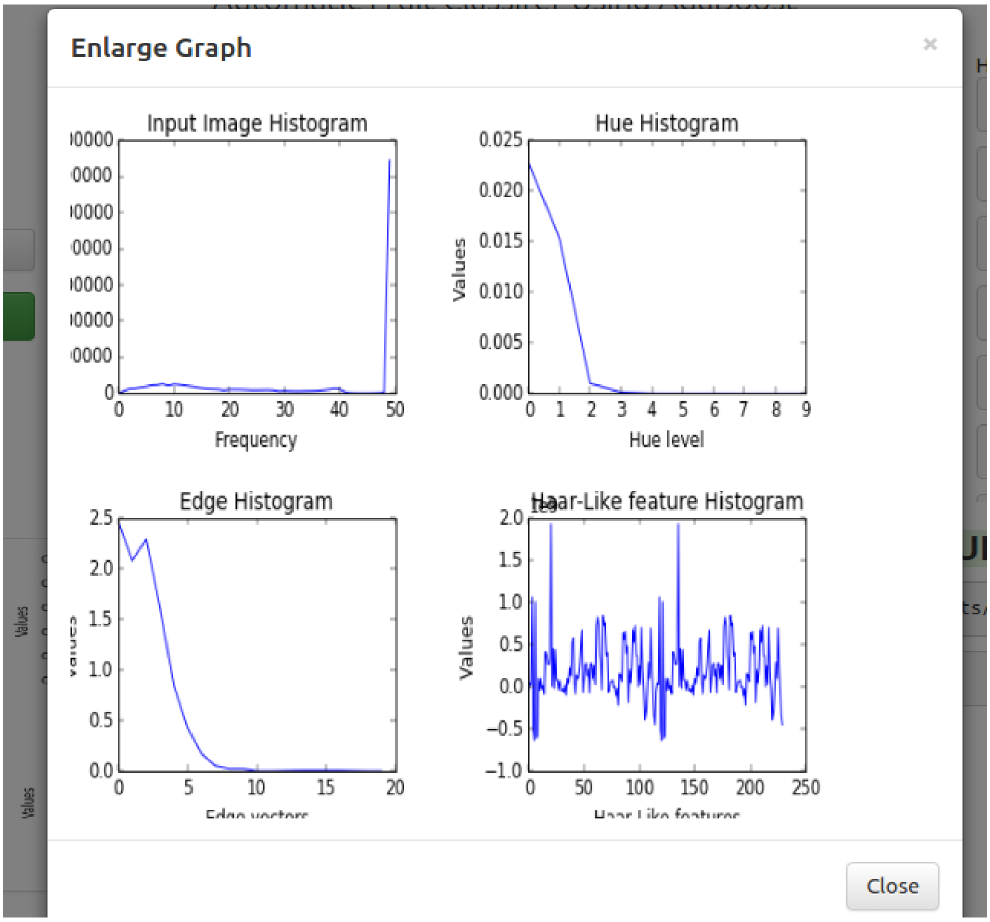
Hue5 = 1.61775436986e-06

RESULT

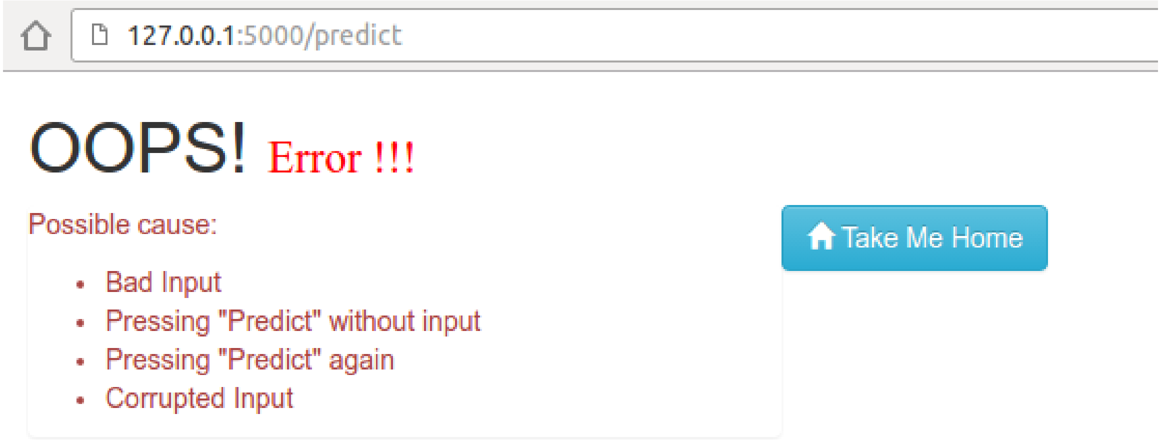
The given input is classified as: fruits/avocados.

Probability of prediction: 54.4%

Histogram generated based on testing image



Snapshot of the error page



127.0.0.1:5000/predict

OOPS! Error !!!

Possible cause:

- Bad Input
- Pressing "Predict" without input
- Pressing "Predict" again
- Corrupted Input

REFERENCES

[1] Patel H.N *et al.*, “Automatic Segmentation and Yield Measurement of Fruit using Shape Analysis”, International Journal of Computer Applications, Volume 45–No.7, May 2012

[2] Aasima Rafiq, *et al.* “Application of Computer Vision System in Food Processing- A Review”, PInt. Journal of Engineering Research and Applications, Vol. 3, Issue 6, Nov-Dec 2013

[3] Oikonomidis, A. A. Argyros and R. Boyle, “Deformable 2D shape matching based on shape contexts and dynamic programming” in ISVC 2009. LNCS, vol. 5876, 2009, Springer-Verlag, pp. 460-469

[4] A. Camargo and J. S. Smith, “An image-processing based algorithm to automatically identify plant disease visual symptoms,” Biosystems Engineering, vol. 102, January 2009, pp. 9-21

[5] Cairo University, Faculty of Computers and Information, Cairo, Egypt Scientific Research Group in Egypt (SRGE) “Automatic fruit classification using random forest al2gorithm,” in *2014 International Conference on Hybrid Intelligent Systems (HIS).*, Kuwait, 2014, pp. 164-168.

[6] Yudong Zhang and Lenan Wu “Classification of Fruits Using Computer Vision and Multiclass Support Vector Machine” School of Information Science and Engineering, Southeast University, Nanjing 210096, CH, Sensors 2012, 12

[7] "Haar-like features", Wikipedia, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Haar-like_features. [Accessed: 19- Aug- 2016].

[8] "Color histogram", Wikipedia, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Color_histogram. [Accessed: 19- Aug- 2016].

[9] "Edge detection", Wikipedia, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Edge_detection. [Accessed: 19- Aug- 2016].

[10] "AdaBoost", Wikipedia, 2016. [Online]. Available: https://en.wikipedia.org/wiki/AdaBoost. [Accessed: 19- Aug- 2016].

[11] D. Marr, “Vision”. W. H. Freeman and Company, 1982.

[12] Ming-Hsuan Yang, “Object Recognition”. University of California at Merced

[13] J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors. “Toward category-level object recognition”. Springer-Verlag, 2006.

[14] Kwak, N. “Principal Component Analysis Based on L1-Norm Maximization”. IEEE Trans. Patt. Anal. Mach. Int. 2008 , 30, 1672–1680

[15] D. G. Lowe, “Object recognition from local scale-invariant features,” In Computer Vision, 1999. The proceedings of the seventh IEEE international conference, Corfu, Greece. pp. 1150-1157.

[16] Y. Freund, and R. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”, 1997

[17] “Welcome | Flask (A Python Microframework)”, Flask.pocoo.org, 2016. [Online]. Available: http://flask.pocoo.org/. [Accessed: 19- Aug- 2016].

[18] M. Škrjanec. “Automatic fruit recognition using computer vision”, Bsc Thesis, (Mentor: Matej Kristan), Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2013.

[19] “Scientific Computing Tools for Python”, Scipy.org, 2016 [Online] Available: https://www.scipy.org/about.html [Accessed: 20- Aug- 2016].

[20] “scikit-learn: machine learning in Python — scikit-learn 0.17.1 documentation”, Scikit-learn.org, 2016. [Online]. Available: http://scikit-learn.org/stable/. [Accessed: 20- Aug- 2016].

[21] “SimpleCV”, Simplecv.org, 2016. [Online]. Available: http://simplecv.org/. [Accessed: 20- Aug- 2016].

BIBLIOGRAPHY

i. Rafael C. Gonzalez & Richard E. Woods: “Digital Image Processing”, Second Edition, Pearson Education, New Delhi, 2006

ii. Trevor Hastie, Robert Tibshirani & Jerome Friedman: “The Elements of Statistical Learning”, Second Edition, Springer

iii. Peter Harrington, “Machine Learning in Action”, Manning Publications Co., NY 11964, 2012

iv. David Julian, “Designing Machine Learning Systems with Python”, Packt publications, UK, 2016

v. Balazs Kegl, “Introduction to AdaBoost”, November 15, 2009