International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 1

Smart Fruit Classification using Neural Networks

K. Sandhiya, M. Vidhya, M. Shivaranjani, S. Saranya
Department of CSE, United Institute of Technology, Coimbatore, India

ABSTRACT

The objective of this project is to develop a system that determines the quality of the fruits which is an important helps the food industry to classify fruits based on specific factor in the food processing industry. quality features. Our system will give best performance when used to sort some brand of fruits. The fruit industry plays a vital role in a country's economic growth. They account for a fraction of the agricultural output produced by a country. It forms a part of the food processing industry. Fruits are a major source of energy, vitamins, minerals, fiber and other nutrients. They contribute to an essential part of our diet. Fruits come in varying shapes, color and sizes. Some of them are exported, thereby yielding profit to the industry.

Keywords: *Neural Network; Perceptron Algorithm; Fruits; Features; Classifiers*

I. INTRODUCTION

Fruit sorting and grading are performed before export. This determines the quality of the fruits which is an important factor in the food processing industry. The system is required to integrate hardware and software components to achieve the above objective. We have a conveyor system which the dates will pass throw and images will captured under an illumination chamber passes them PC which will classify the fruit and take the proper decision. The fruit industry plays a vital role in a country's economic growth. They account for a fraction of the agricultural output produced by a country. It forms a part of the food processing industry. Fruits are a major source of energy, vitamins, minerals, fiber and other nutrients.

They contribute to an essential part of our diet. Fruits come in varying shapes, color and sizes. Some of them are exported, thereb yielding profit to the industry. Fruit sorting and grading are performed before export. This

- This project is based on **neural networks**.
- Neural networks contains well designed, open source java library with small number of basic classes which correspond to basic NN concepts.
- It is small, well documented, easy to use, and very flexible neural network framework.

I. II. ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons.

Artificial neural networks are considered to be good alternatives to conventional statistical methods for the classification problems and pattern recognition problems. Artificial neural networks are considered to be good alternatives to conventional statistical methods for the classification problems and pattern recognition problems. Perceptron model is the basic model of ANN and is generally used for the classification of binary class problems. It is a fast and reliable network for the class of problems that it can solve.

- **Algorithm:-** PERCEPTRON ALGORITHM
- **Software:-** JAVA

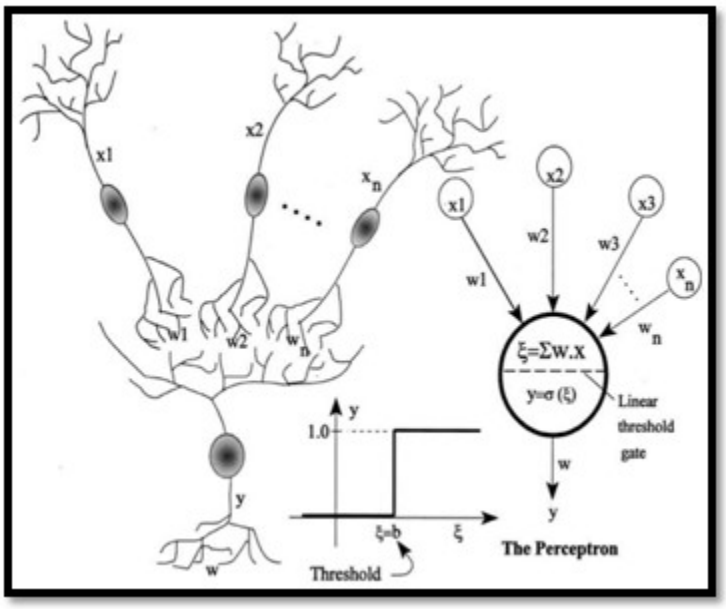


Fig 1: Artificial Neural Network

III. CHARACTERISTICS OF ANN

- A large number of very simple processing neuron-like processing elements.
- A large number of weighted connections between the elements.
- Distributed representation of knowledge over the connections.
- Knowledge is acquired by network through a learning process.

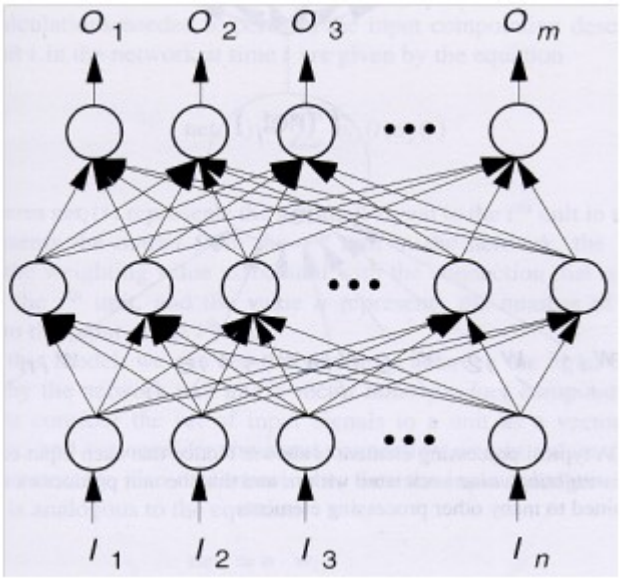


Fig 2: Characteristics of Ann

IV. PERCEPTRON NETWORK

The perceptron is an algorithm for supervised learning of binary classifiers. (functions that can decide whether an input, represented by a vector of numbers, belongs to some specific or not).

It is a type of linear classifier. All class of Neural Network needs to learn the environment in which that network has to perform. These neural networks learn via some rules. Being a part of neural network, there should be some learning rule for the perceptron.

A. FEED-FORWARD NETWORKS

Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organisation is also referred to as bottom-up or top-down.

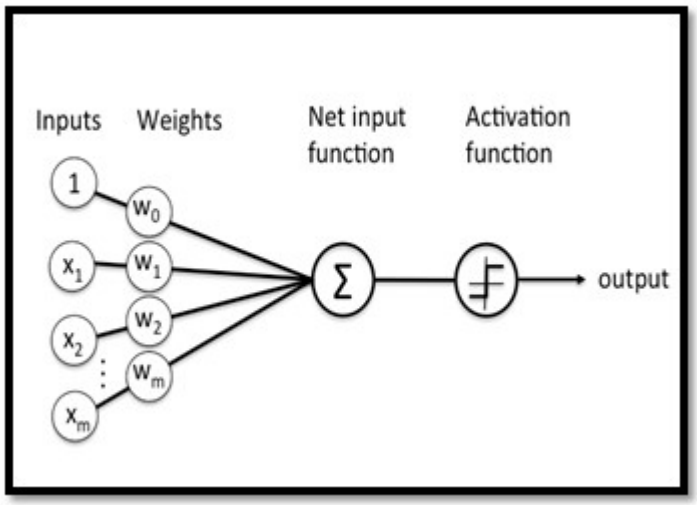
B. FEEDBACK NETWORKS

Feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organisations.

C. PERCEPTRON ALGORITHM

One of the oldest algorithms used in machine learning (from early 60s) is an online algorithm for learning a linear threshold function called the Perceptron Algorithm.

For simplicity, we'll use a threshold of 0, so we're looking at learning functions like: $w1x1 + w2x2 + ... + wnxn > 0$.We can simulate a nonzero threshold with a "dummy" input x0 that is always 1, so this can be done without loss of generality.



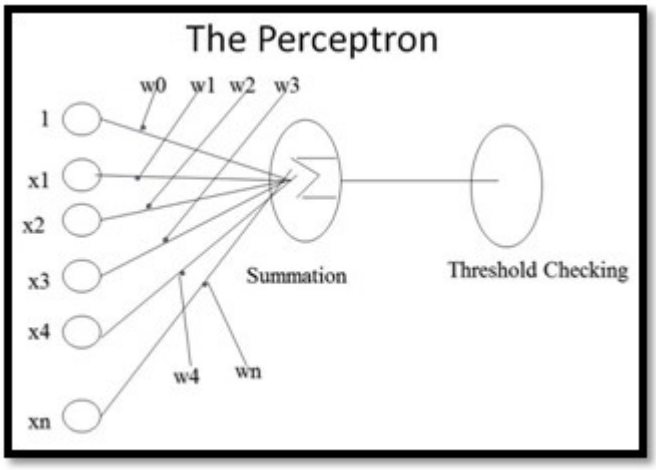


Fig 3: Perceptron in ANN

D. NETWORK LAYERS

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output". The activity of the input units represents the raw information that is fed into the network. The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units. This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

We also distinguish single-layer and multi-layer architectures. The single-layer organisation, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multi-layer organisations. In multi- layer networks, units are often numbered by layer, instead of following a global numbering.

V. FRUIT CLASSIFIER USING PERCEPTRON RULE

Perceptron model is the basic model of ANN and is generally used for the classification of binary class problems. It is a fast and reliable network for the class of problems that it can solve. The working of this model is very simple. The weighted sum of input signals is compared to a threshold to determine the neuron output. By learning rule we mean a procedure for modifying the weights and biases of a network. This procedure may also be referred to as a training algorithm. The purpose of the learning rule is to train the network to perform some task.

Fig 4: Perceptron Algorithm

A. EXPERIMENTATION

In this project we took a simple problem to classify two fruits Apple and Banana. The objective of our project is to classify objects according to their features.

We can now define a prototype banana and apple.

(Shape, texture, weight)

Shape: {0: round; 1: elliptical}

Texture: {0: smooth; 1: rough}

Weight: {1: <100 g; 1: > 100 g}

Apple < 2 and Banana > 2

Fig 5: Shape and Texture Identification

B. CLASSIFICATION

First, the fruits (Apple/ Banana) area was segmented from the background by thresholding and edge detection algorithms.

C. COLOR IDENTIFICATION

Color classification was to 12 levels according to Agrexco Carmel Standards. A Hue Saturation Value matrix was derived and identification was conducted on the hue matrix.

D. SHAPE AND SIZE

It was measured from the image by calculating the roundness of the date using the FFT of the distances from the edges to the date center. The rounder the object, the lower the frequency of the vector.

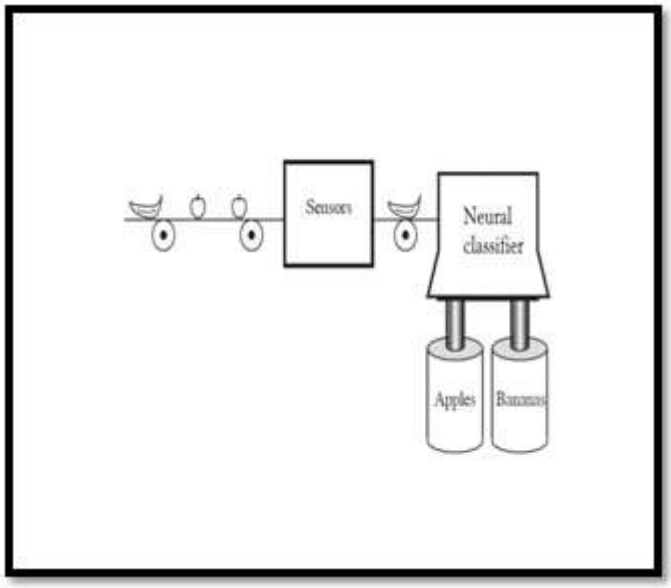


Fig 6: Apple/Banana Experimentation

VI. TRAINING THE NETWORK

In training set we can obtain some new weights by using perceptron rule. These new weights are used to identify the smart fruit classifier. New weight calculation coding is shown.

```
System.out.println("old weight"+"w[0]="+weight[0]+"w[1]="+weight[1]+"w[2]="+weight[2]);
iteration = 0;
do
{
        iteration++;
                        //loop through all instances (complete one epoch)
                for (p = 0; p < NUM_INSTANCES; p++)
{
// calculate predicted class
                                output = calculateOutput(theta,weights, shape[p], texture[p],weight[p]);
                                // difference between predicted and actual class values
        localError = eo[p] - output;
                        System.out.println(output);
        //System.out.println(eo[p]);
if(output==eo[p])
System.out.println(shape[p] + " " + texture[p]+"    "  +weight[p]+ " =" + output);
        else
                                {
        System.out.println("iteration starts");
                                        weights[0] +=  localError * shape[p];
                                weights[1] +=  localError * texture[p];
                weights[2]+=  localError * weight[p];
        globalError += (localError*localError);
System.out.println("============================================================");
System.out.println("new weights "+"w0="+weights[0]+" w1="+weights[1]+"w2="+weights[2]);
        System.out.println("============================================================");
                                                }
if(eo[p]==1){System.out.println("Banana");
}
else{System.out.println("apple");
}
}
```

Fig 7: Coding Implementation

A. TESTING THE TRAINED NETWORK

Now new weights are obtained

W1= 1.7, W2= 1.4, W3 = 0.7

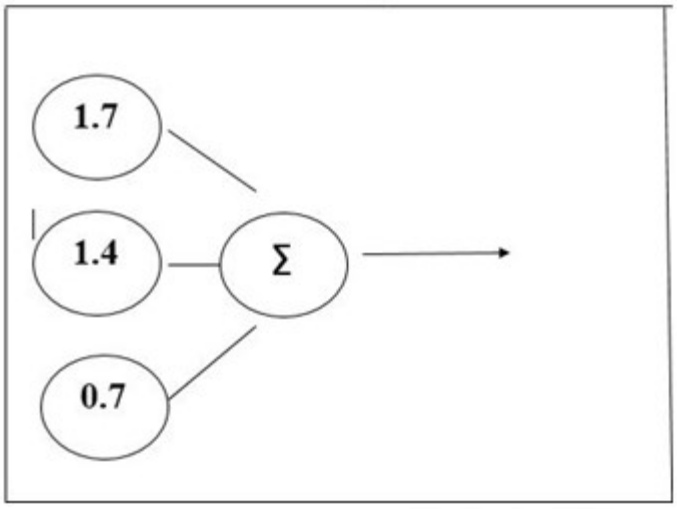


Fig 8: Initial Weight

Iteration begins…

= (w1 * x1 + w2 * x2 + w3 * x3)

= 1.7 * 0 + 1.4 * 1 + 0.7 * 0

= 0 + 1.4 + 0

= 1.4 < 2 (APPLE)

= 1.7 * 1 + 1.4 * 1 + 0.7 * 1

= 3.9 > 2 (BANANA)

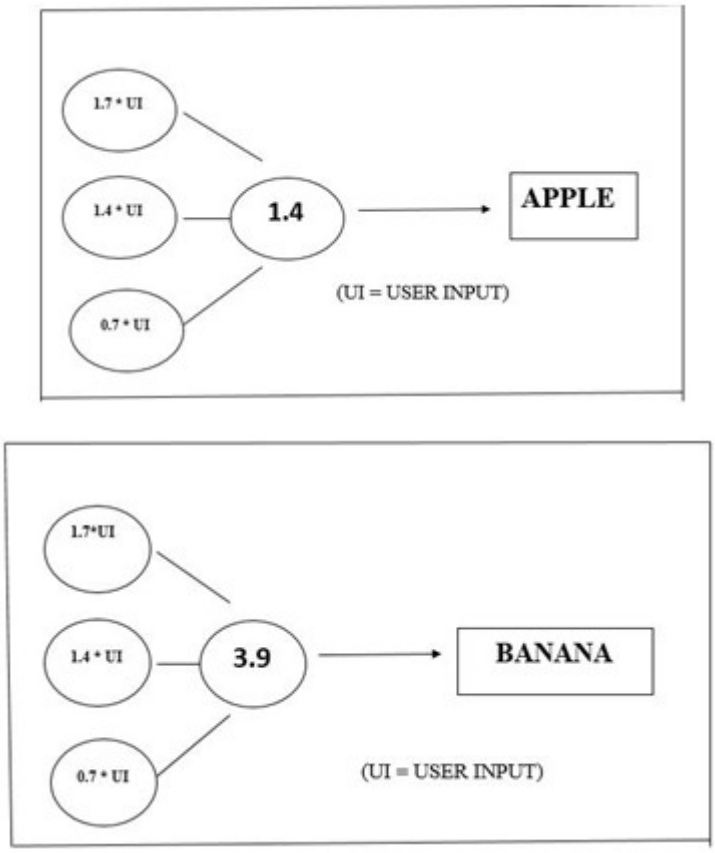
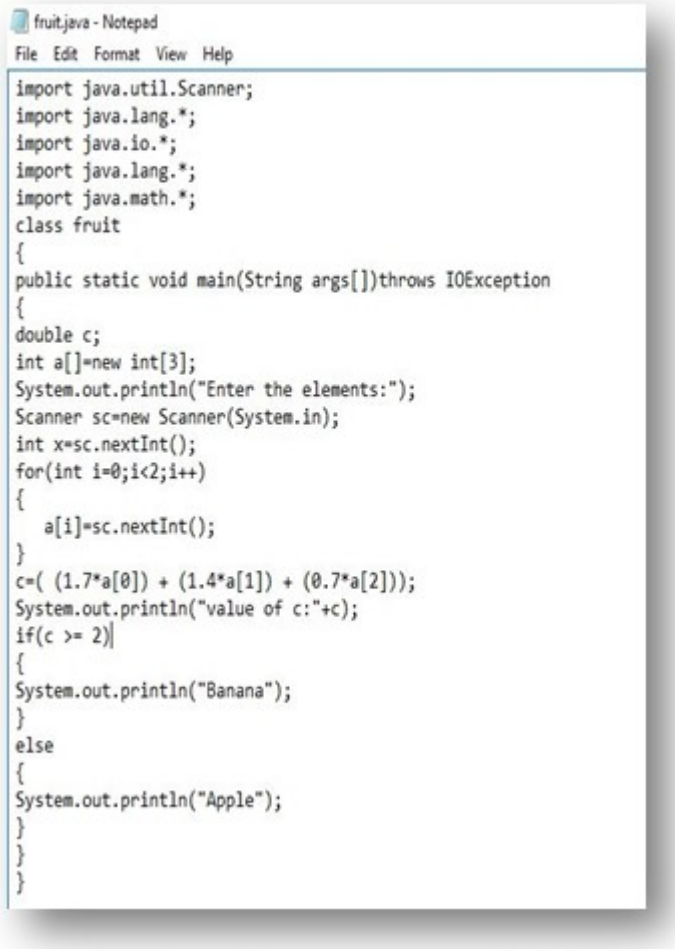


Fig 9: Diagramatic Representation

B. CODING AND EXECUTION PART

By using java we implement the coding to calculate old weight, new weight and easily identify the given type of fruit.



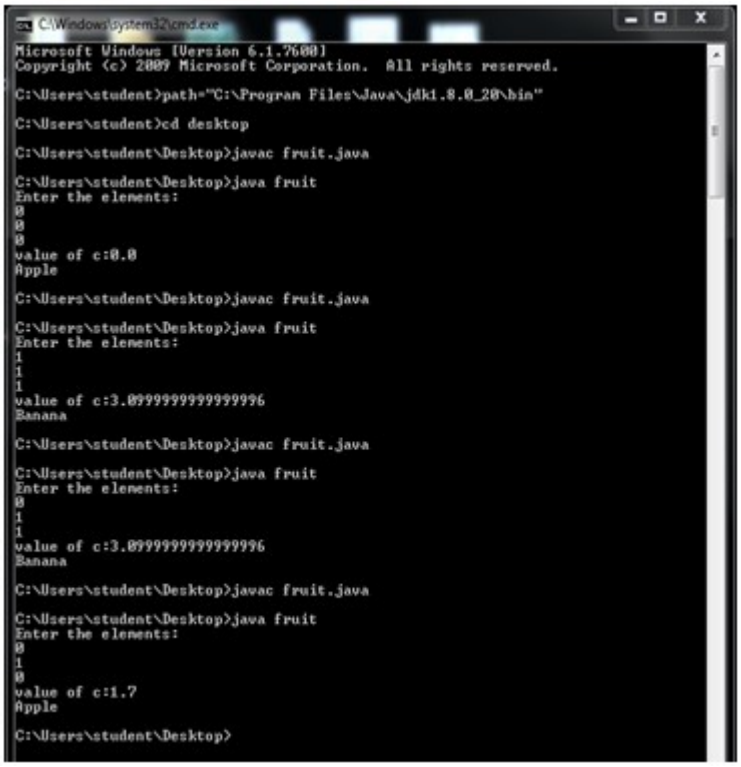
fruit.java - Notepad

File Edit Format View Help

```
import java.util.Scanner;
import java.lang.*;
import java.io.*;
import java.lang.*;
import java.math.*;
class fruit
{
public static void main(String args[])throws IOException
{
double c;
int a[]=new int[3];
System.out.println("Enter the elements:");
Scanner sc=new Scanner(System.in);
int x=sc.nextInt();
for(int i=0;i<2;i++)
{
   a[i]=sc.nextInt();
}
c=( (1.7*a[0]) + (1.4*a[1]) + (0.7*a[2]));
System.out.println("value of c:"+c);
if(c >= 2)
{
System.out.println("Banana");
}
else
{
System.out.println("Apple");
}
}
}
```

Fig 10: Coding Implementation

C. TESTING THE SOLUTION



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\student>path="C:\Program Files\Java\jdk1.8.0_20\bin"

C:\Users\student>cd desktop

C:\Users\student\Desktop>javac fruit.java

C:\Users\student\Desktop>java fruit
Enter the elements:
0
0
0
value of c:0.0
Apple

C:\Users\student\Desktop>javac fruit.java

C:\Users\student\Desktop>java fruit
Enter the elements:
1
1
1
value of c:3.0999999999999996
Banana

C:\Users\student\Desktop>javac fruit.java

C:\Users\student\Desktop>java fruit
Enter the elements:
0
1
1
value of c:3.0999999999999996
Banana

C:\Users\student\Desktop>javac fruit.java

C:\Users\student\Desktop>java fruit
Enter the elements:
0
1
0
value of c:1.7
Apple

C:\Users\student\Desktop>
```

Fig 11: Final Output

We conclude that Neural networks classifiers are best for fruit recognition with the combined features as though it gives more complexity will yield good accuracy. These are no proper methods to identify the defection from inside.

VII. CONCLUSION

Neural Networks can be pre-dominantly used for the classification of fruits as they result in a higher degree of accuracy, than compared to other classifiers. Neural Networks find applications in predicting weather, complex problem solving, pattern recognition, image matching, face recognition, and many others.

REFERENCES

[1] Suchitra A. Khoje, Dr. S. K. Bodhe, Dr. Alpana Adsul, "Automated Skin Defect Identification System for Fruit Grading Based on Discrete Curvelet Transform", International Journal of Engineering and Technology (IJET), Vol 5 No 4 Aug-Sep 2013.

[2] Woo Chaw Seng and Seyed Hadi Mirisaee, "A New Method for Fruits Recognition System", MNCC Transactions on ICT, Vol. 1, No. 1, June 2009.

[3] S. Arivazhagan, R. Newlin Shebiah, S. Selva Nidhyanandhan, L. Ganesan, "Fruit Recognition using Color and Texture features", Journal of Emerging Trends in Computing and Information Sciences(CIS Journal), Vol. 1, No. 2, Oct 2010.

[4] Dayanand Savakar, "Identification and Classification of Bulk Fruits Images using Artificial Neural Networks", International Journal of Engineering and Innovative Technology (IJEIT), Volume 1, Issue 3, March 2012.