An FPGA based Efficient Fruit Recognition System Using Minimum Distance Classifier

Harsh S Holalad, Preethi Warrier, Aniket D Sabarad
Dept of Electrical and Electronics Engg.,B V Bhoomaraddi College of Engg & Tech
Hubli-580031, India
*E-mail of the corresponding author: harsh.holalad@gmail.com

Abstract

The paper deals with a simple yet effective fruit identification system developed on an FPGA, SPARTAN 3(XC3S200-5PQ208) platform .The fruits under consideration were apple, banana, sapodilla and strawberry. Out of these selected fruits there were four different classes of apples, two different classes of sapodillas and one class each of the other two fruits. A total of 800 color images, 200 images of each fruit of size 64x64 were used for training.

The fruit identification success rate mainly depends on the feature vector and the Classifier used. The 3D feature vector incorporates two first order statistical features and the shape feature. Using the 3D feature vector the MATLAB analysis of The Minimum Distance Classifier (MID) fetched a success rate of 85%.The Verilog coded Hardware platform was developed by burning the COE file of a Test image generated by JAVA ECLIPSE IDE onto the IP core. The MATLAB results were verified using the Hardware Platform.

Keywords: RGB image, feature vector, MID, Verilog, FPGA, IP core, COE file**.**

1. Introduction

Fruit Recognition Systems that exist for fruit harvesting, tree yield monitoring,[2] disease detection and other operations use computer vision strategies that consider features like color, shape and texture for recognition. This paper suggests fruit recognition system design that uses a minimum distance classifier that imbibes first order statistical features along with shape feature for efficient fruit identification. FPGA based design for the above system has been simulated using Verilog.

Texture is a property that represents the surface and structure of an image.[5] Statistical methods analyze the spatial distribution of gray values, by computing local features at each point in the image, and deriving a set of statistics from the distributions of the local features. Depending on the number of pixels defining the local feature, statistical methods can be further classified into first-order (one pixel), second-order (two pixels) and higher-order (three or more pixels) statistics. The basic difference is that first-order statistics estimate the properties (e.g. mean and variance) of individual pixel values, ignoring the spatial interaction between image pixels, whereas second- and higher-order statistics estimate properties of two or more pixel values occurring at specific locations relative to each other. [3] The mean used in texture analysis has been calculated using the color feature, thereby showing an inter dependence of the features. The shape feature has been calculated by using the area and perimeter of the test image. Combining the above features, an efficient algorithm using minimum distance classifier has been designed. The initial analysis was done on MATLAB.

The motive behind the paper was to implement a real time fruit recognition system. This resulted in FPGA based hardware implementation. The Verilog simulations were carried out in Xilinx ISE 10.1 & ISIM. A COE file of the image was generated using JAVA Eclipse, which was later burnt onto the IP CORE and then ran through the identification algorithms.

The presented work deals with four different fruits, namely, apple (4 classes), banana, sapodilla (2 classes) and strawberry. The flow is structured as follows. The section 2 discusses the Feature Extraction. Section 3 talks about the Minimum Distance Classifier. The Section 4 deals with the Hardware Constrains of an FPGA experienced and Section 5 talks about the Verilog Results and FPGA output. Finally, Section 6 gives the Concluding remarks of this paper.

2. Feature Extraction

Features are the measurements which are used to discriminate between objects or data. They are the base on which Classification is done.The criterion for feature extraction is that the features should be position, scale and distance invariant. The features used are

2.1. First Order Statistical Features:

These features are the statistics calculated from the original pixel values without any [5] need of pixel relations or use of matrix transforms. Statistical measures such as mean, variance, skewness, kurtosis, average energy, entropy and dispersion fall into this class.

During the MATLAB analysis a trial and error approach was taken considering the hardware implementation into account. Different combinations of these features were tried and it was found out that mean and variance had a substantial impact on the efficacy and they were simple to be implemented as well. The mean and variance are expressed as shown below,

$$mean = \frac{1}{(m*n)} \sum_i^m \sum_j^n x(i,j) \text{.......................................} 1$$

$$variance = \frac{1}{(m*n)} \sum_i^m \sum_j^n (x(i,j) - mean)^2 \text{..............} 2$$

Where, *x (i,j)* is the pixel value of image *x* of size *m x n.*

2.2. Shape feature

Shape feature helps to determine the measure of roundness of an object. This feature makes the system more robust because this feature is [4] independent of the distance between the camera and the object, unlike area and perimeter.

To obtain the measure of roundness the area and the perimeter of the object has to be measured. Area can be measured by using segmentation techniques and edges of the object which is the perimeter can be obtained by going for various edge detection algorithms.

The shape feature of an object can be calculated using the relation,

$$S = \left({}^{area}/_{perimeter^2} \right) * 4\pi \text{.....................................} 3$$

2.2.1. Segmentation:

Image segmentation is used to segment an image into various parts. It is normally a pre-processing step and can be done to separate the main object under test from its background.

Of the different segmentation techniques like threshold based, edge based, and region based, the paper uses threshold based technique for segmentation.

Inter Means Algorithm: It is a threshold based automatic segmentation technique developed [7]& [8] by Ridler ,Calvard (1978) and Trussell (1979).This is an iterative algorithm which keeps on splitting the image histogram into two halves by changing the threshold value.

In Figure 1,

1 is the initial guessed threshold value.

2 is the new threshold value after first iteration

3 is the new threshold after second iteration.

4 is the final threshold value used to create a binary image.

The figure helps in better understanding of the Inter-means algorithm, which is as follows:

Step 1-Consider an arbitrary threshold *T*.

Step 2-Means of pixels above (M_a) and below (M_b) the threshold values are calculated.

Step 3-The new threshold is calculated as

$$T_{new} = {(M_a + M_b)}/{2} \text{...} 4$$

Step 4- The iteration is performed until the difference between the previous threshold (T) and the new threshold (T_{new}) is less than 1.

Once the threshold is calculated the image is then converted into a binary image. The count of white pixels is the area in pixels.

The Segmentation results are found in Figure 2.

2.2.2. Edge Detection:

Edge detection is used to identify and locate sharp discontinuities in an image.[6] The edges are characterized by abrupt changes in the pixel values. The various edge detection algorithms involve convolving the image with a 2-D matrix ranging from size 2x2 to 5x5 or greater for increased efficiency. But in this paper approximate Robert edge detection is used which is very simple to implement and is efficient.

.*Approximate Robert edge Detection*: It is a simple, quick to compute spatial Gradient measurement of an image. Calculating the approximate Gradient from the Pseudo-Convolution mask:

$$\begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix}$$

Using this mask the approximate magnitude is given by:

$$|G| = |P_1 - P_2| + |P_2 - P_3| \text{...} 5$$

The edges obtained, i.e. the count of white pixels is the perimeter in pixels.

The Edge Detection Results are obtained in Figure 2.

From equation 3, the shape feature is thus determined.

3. Minimum Distance Classifier

Image classifiers analyse the numerical properties of various image features and organize data into categories. Classification algorithms typically employ two phases of processing: training and testing. [1] In the initial training phase, characteristic properties of typical image features are separated and, based on these, a training class is created. In the subsequent testing phase, these feature-space partitions are used to classify image features. In supervised classification, statistical patterns are used for classification while in unsupervised classification, clustering algorithms are used. The minimum distance classifier is used to classify unknown image data to classes which minimize the distance between the image data and the class in multi-feature space.

The distance classifier [2] that has been implemented employs the Euclidean distance given by,

$$dk = \sqrt{\Sigma(\mathrm{X_t} - \mathrm{Y_t})^2{}_{\mathrm{j}}} \text{...6}$$

Where,

X_t :j^{th} feature of the t^{th} class from test sample.

Y_t:j^{th} feature of the t^{th} class from the centroids obtained by the test samples.

j=feature vector.

As mentioned above, MID has two phases training part and the testing part. The algorithms for both are

Training Algorithm:

Step1: Consider an image of a fruit belonging to a class (say apple).

Step 2: Find the green (from RGB) component of the images.

Step 3: Compute mean of the G component using equation 1.

Step 4: Compute the variance of the image using equation 2.

Step 5: Perform automatic segmentation and isolate the object.

Step 6: Count of all the white pixels is the area of the object.

Step 7: Perform edge detection on the segmented (binary) image.

Step 8: Count of all the white pixels gives the perimeter of the object.

Step 9: Find the shape (measure of roundness) from equation 3.

Step 10: Repeat the Steps from 1-10 for different images of a class.

Step 11: The centroids are found for each feature (mean, variance, shape). centroid is calculated by calculating the mean of each feature(means of 200 images of a class)

Step 12: Repeat the same for the remaining classes.

The obtained centroids will be used for the hardware implementation.

Testing Algorithm:

Step 1: The centroids of all the four fruits obtained from the MATLAB analysis are stored in the FPGA memory.

Step 2: The COE file of the test image is burnt on the IP core.

Step 3: Step 2-Step 9 from the training algorithm is performed.

Step 4: The Euclidean Distance between the test values obtained from Step 3 and that of the already stored Centroid values obtained from training are calculated using equation 5.

Step 5: Find out the minimum distance amongst all the distances (each class) and assign the test image to the class with minimum distance.

The Results of Minimum Distance classifier is in Table 1

4. Hardware Constraints

This part of the paper deals with the limitations of the FPGA board which were encountered:

4.1. No Provision for camera Interface:

The SPARTAN 3 family does not have the provision to interface a camera. This board has been used for developing the algorithm and testing it. Since the Camera interfacing was not possible on the board the test image coefficients were generated by software named JAVA ECLIPSE IDE and was burned on the memory (using IP Core Generator).

4.2. Concurrent operation

All the "*always @(posedgeclk)*" blocks in the *module* are executed at every positive edge of the clock. For the algorithm we have implemented involves more of sequential operations than the concurrent operations. Binary Flags were used as delays when there was a need for sequential operation.

4.3. Loop execution

For, Case, while and Repeat loops should be used meticulously in the code. The loop executions depend heavily on the instructions to be executed. It is best explained with an example,

Eg 1)

```
always@(posedge..clk)begin
 for(i=0;i<100;i=i+1)begin
   sum=t[i]+b[i];
end
end
```

Eg 2)

```
always@(posedge..clk)begin
  if(i<100)begin
    sum=t[i]+b[i];
end
end
```

Both the above examples will do the addition of two array elements and store it in the register sum. Only the code in Eg.2 will be Synthesized and Mapped into Gate Logic. The reason behind this is, the first example i.e. Eg.1 uses 100 adders to obtain the sum in one cycle, which will not be synthesized because the board is short of 100 adders. In the second example, the sum is calculated using just a single adder but the sum will be obtained after 100 positive edge clocks, hence it's synthesized.

4.4. Division

Multiple subtractions have been used to obtain the quotient (only the integer part, the fraction part is neglected).

Eg.7/2=3.5;a=7;b=2;c=0;

In veilog it would be 3

```
First cycle:a=a-b;//a=5
    c=c+1;//c=1
Second cycle:a=a-b;//a=3
    c=c+1;//c=2
Third cycle:a=a-b;//a=1
    c=c+1//c=3
Quotient=3;//it stops when a<b.
```

4.5. Procuring the Training data

As explained above that there's no provision for camera interfacing, this would mean training 800 images on the FPGA would be an uphill task and not to mention an unintelligent task to do. So the training data obtained in the

MATLAB analysis can be considered. Unlike VHDL, VERILOG does not support matrices, so the Verilog and the MATLAB results of a set of images were tested and were found to be the same.

This proves that the Verilog result and MATLAB results are same and hence the Centroids obtained in MATLAB can be directly considered for the Hardware development.

5. Verilog Results and FPGA Output

In Figure 4, the FPGA board, the first four LED's glowing indicates that the apple image is detected which is highlighted in the Red box. If in case a banana image was to be detected the next four LED's traversing from right to left highlighted in the Yellow box would glow and on sapodilla's detection the four LED's in the Brown box would glow. The LED's in the Pink box would glow if the input image was a strawberry (Figure 6).

The results of two fruit detections have been shown in this section.

5.1. *Apple*

The input test image is an Apple

The distance computed for various fruits are:

Dist_apple =801,061

Dist_banana= 2,881,034

Dist_chikoo= 856,466

Dist_strawberry = 1,682,609

In the Figure 3, the red oval part above shows the distance computed of the apple and the red oval part below shows that the classified image is an APPLE.

5.2. *Strawberry*

The input test image is a Strawberry

The distance computed for various fruits are:

Dist_apple = 1,347,922

Dist_banana= 14,086,609

Dist_chikoo=8,887,517

Dist_strawberry = 586,707

In the Figure 5, the red oval part above shows the distance computed of the apple and the red oval part below shows that the classified image is a STRAWBERRY.

6. Conclusion

The feature selection is always a tricky part of any recognition system. The features selected were position and distance invariant. Out of all the texture features only mean and variance were selected because of the substantial impact it played on the success rate. The images clicked are not from a fixed length, so area and perimeter cannot be chosen as the feature. Therefore area and perimeter were used to obtain the shape of the object (measure of roundness). The use of automatic segmentation (inter-means algorithm) has given a very good efficiency in calculating the area of the object. Even though the Approximate Gradient Robert Edge Detection technique is used, the results of Perimeter were not varying much from the inbuilt MATLAB Robert edge detector. The final feature vector is MEAN, VARIANCE and SHAPE. A set of 160 images, 40 from each class were tested on

XC3S200-5PQ208 and the success rate was found to be 85%.The board provides an excellent platform for developing and verifying the software analysis thus providing a firm ancillary towards implementing a real time fruit recognition system.

References

Richard O. Duda,Peter E.Hart,David G.Stork, "*Pattern Classification*", second edition.

Woo Chaw Seng, Seyed Hadi Mirisaee, "*A New Method for Fruits Recognition System*".

Hetal Patel, Dr.R.K.Jain, Dr.M.V.Joshi, "Fruit Detection Using Improved Multiple Features Based Algorithm" *International Journal of Computer Applications* (0975 – 8887), Volume 13– No.2, January 2011.

S.Arivazhagan, R.Newlin Shebiah, S.Selva Nidhyanandan, "Fruit recognition Using Color and Texture", *Journal of Emerging Trends in Computing and Information Sciences*, VOL. 1, NO. 2, Oct 2010.

G. N. Srinivasan, and Shobha G,"Statistical Texture Analysis".

Gonzalez & Woods, "*Digital image processing*",third edition.

John C.Russ,"*The Image Processing Handbook*", Sixth edition.

Harsh Holalad, Girish Shirigannavar, Manasa S Upadhyaya, "A Comparative Study of Various Image Segmentation Techniques and Classifiers for a Fruit Recognition System", Benision Education.

Harsh S Holalad, he is a student of Department of Electrical and Electronics Engineering and Technology, B.V.Bhoomaraddi College of Engineering and Technology,Hubli-580023,India.His areas of interest are Digital Signal Processing, Image Processing, Fuzzy logic, Pattern Recognition, MATLAB and HDL.

Preethi Warrier, she is a student of Department of Electrical and Electronics Engineering and Technology, B.V.Bhoomaraddi College of Engineering and Technology, Hubli-580023, India. Her areas of interest are Digital Image Processing, Fuzzy logic, HDL and MATLAB.

Aniket D Sabarad, he is a student of Department of Electrical and Electronics Engineering and Technology, B.V.Bhoomaraddi College of Engineering and Technology, Hubli-580023, India. His areas of interest are Digital Image Processing, Pattern Recognition, MATLAB and HDL.

Table 1: Minimum Distance Classifier results

Fruit	Test images	MATLAB result	VERILOG result
Apple	40	32/40	31/40
banana	40	39/40	39/40
sapodilla	40	33/40	33/40
strawberry	40	32/40	31/40
	160	136/160(85%)	134/160(83.5%)

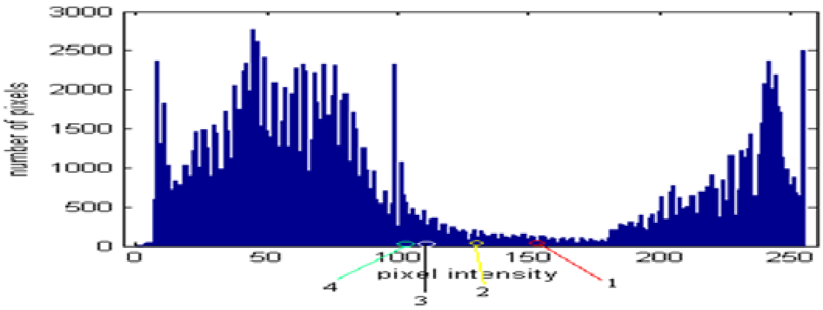


Figure 1.Histogram of a sample image

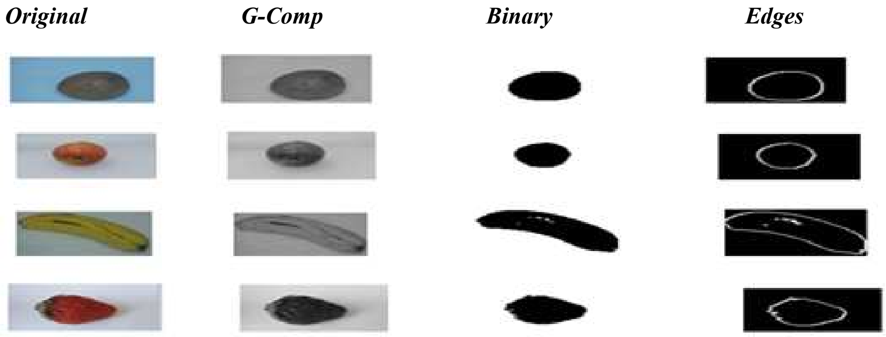
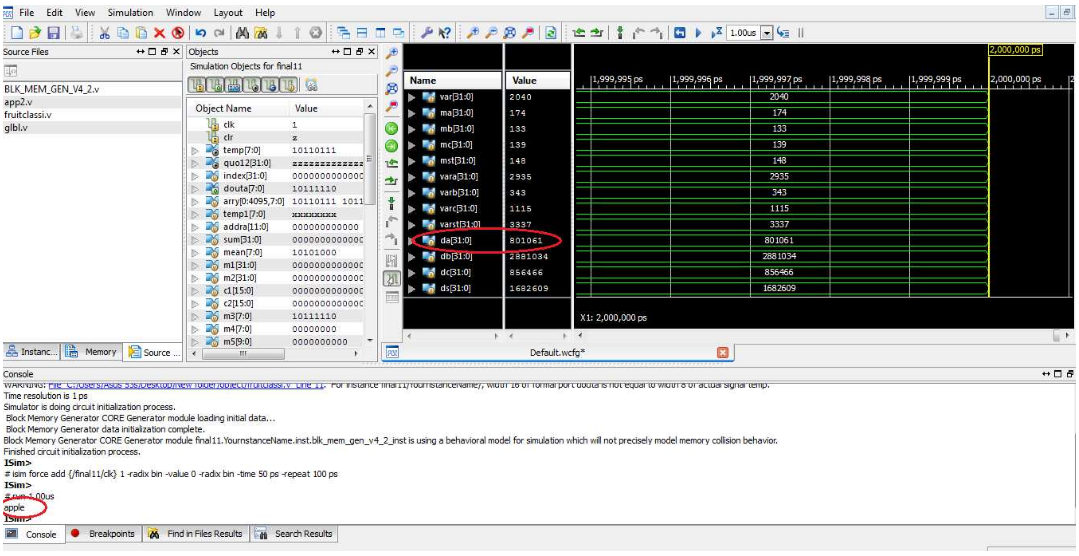


Figure 2: Segmented and Edge Detected Images



```
BLK_MEM_GEN_V4_2.v
app2.v
fruitclassi.v
glbl.v
```

```
Object Name          Value
clk                  1
clr                  z
temp[7:0]            10110111
quo12[31:0]          zzzzzzzzzzzzz
index[31:0]          0000000000000
douta[7:0]           10111110
arry[0:4095,7:0]     10110111  1011
temp1[7:0]           xxxxxxxx
addra[11:0]          000000000000
sum[31:0]            0000000000000
mean[7:0]            10101000
m1[31:0]             0000000000000
m2[31:0]             0000000000000
c1[15:0]             0000000000000
c2[15:0]             0000000000000
m3[7:0]              10111110
m4[7:0]              00000000
m5[9:0]              0000000000
```

```
Name          Value       |1,999,995 ps  |1,999,996 ps  |1,999,997 ps  |1,999,998 ps  |1,999,999 ps  |2,000,000 ps
var[31:0]     2040        2040
ma[31:0]      174         174
mb[31:0]      133         133
mc[31:0]      139         139
mst[31:0]     148         148
vara[31:0]    2935        2935
varb[31:0]    343         343
varc[31:0]    1115        1115
varst[31:0]   3337        3337
da[31:0]      801061      801061
db[31:0]      2881034     2881034
dc[31:0]      856466      856466
ds[31:0]      1682609     1682609

X1: 2,000,000 ps
```

```
WARNING: File "C:/Users/Asus 53s/Desktop/New folder/object/fruitclassi.v" Line 11. For instance final11/YournstanceName/, width 16 of formal port douta is not equal to width 8 of actual signal temp.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
 Block Memory Generator CORE Generator module loading initial data...
 Block Memory Generator data initialization complete.
Block Memory Generator CORE Generator module final11.YournstanceName.inst.blk_mem_gen_v4_2_inst is using a behavioral model for simulation which will not precisely model memory collision behavior.
Finished circuit initialization process.
ISim>
# isim force add {/final11/clk} 1 -radix bin -value 0 -radix bin -time 50 ps -repeat 100 ps
ISim>
# run 1.00us
apple
ISim>
```

Figure 3: Simulation Result of Apple Detection in ISIM

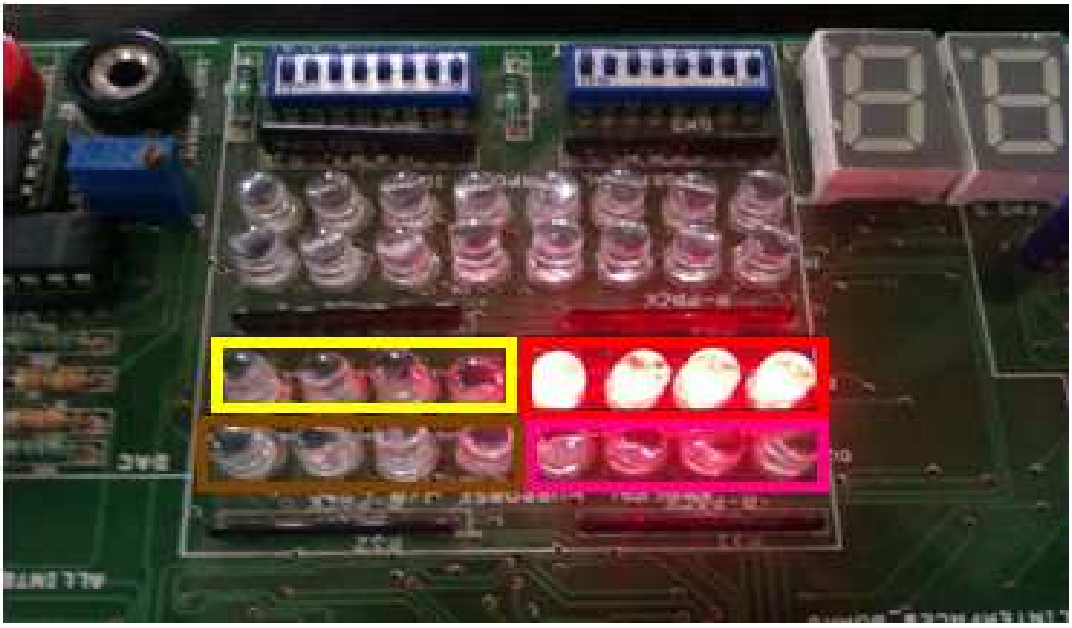
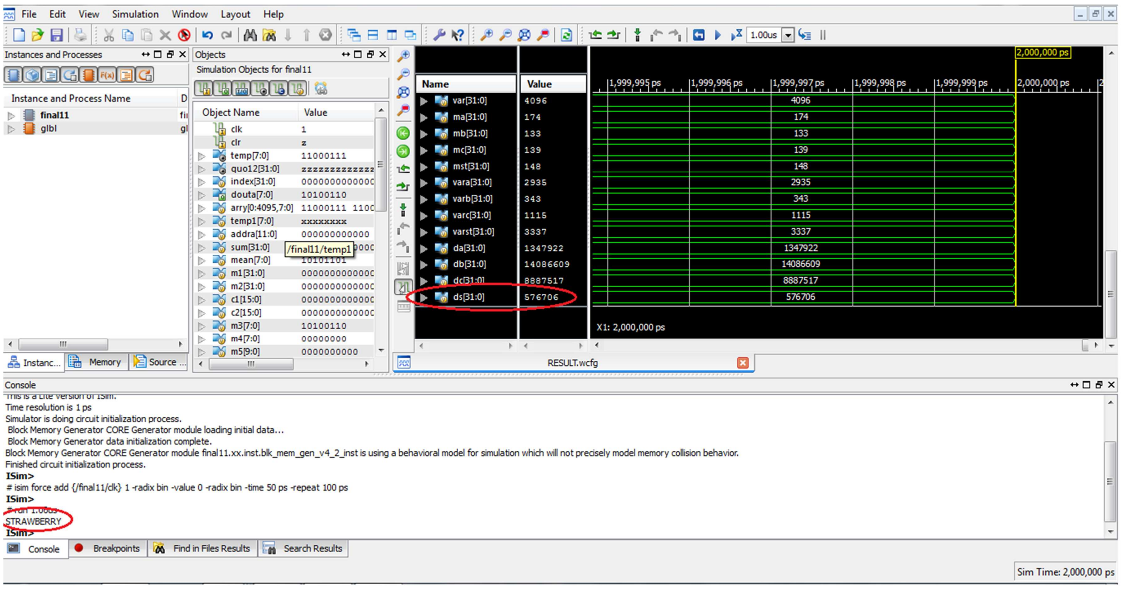


Figure 4: Led Output Image of Apple Detection on XC3S200



Object Name	Value
clk	1
clr	z
temp[7:0]	11000111
quo12[31:0]	zzzzzzzzzzzzz
index[31:0]	000000000000C
douta[7:0]	10100110
arry[0:4095,7:0]	11000111 1100
temp1[7:0]	xxxxxxxx
addra[11:0]	000000000000
sum[31:0]	000C
mean[7:0]	10101101
m1[31:0]	000000000000C
m2[31:0]	000000000000C
c1[15:0]	000000000000C
c2[15:0]	000000000000C
m3[7:0]	10100110
m4[7:0]	00000000
m5[9:0]	0000000000

Name	Value
var[31:0]	4096
ma[31:0]	174
mb[31:0]	133
mc[31:0]	139
mst[31:0]	148
vara[31:0]	2935
varb[31:0]	343
varc[31:0]	1115
varst[31:0]	3337
da[31:0]	1347922
db[31:0]	14086609
dc[31:0]	8887517
ds[31:0]	576706

```
This is a Lite version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
 Block Memory Generator CORE Generator module loading initial data...
 Block Memory Generator data initialization complete.
Block Memory Generator CORE Generator module final11.xx.inst.blk_mem_gen_v4_2_inst is using a behavioral model for simulation which will not precisely model memory collision behavior.
Finished circuit initialization process.
ISim>
# isim force add {/final11/clk} 1 -radix bin -value 0 -radix bin -time 50 ps -repeat 100 ps
ISim>
# run 1.00us
STRAWBERRY
ISim>
```

Figure 5: Simulation Result of Strawberry Detection in ISIM.

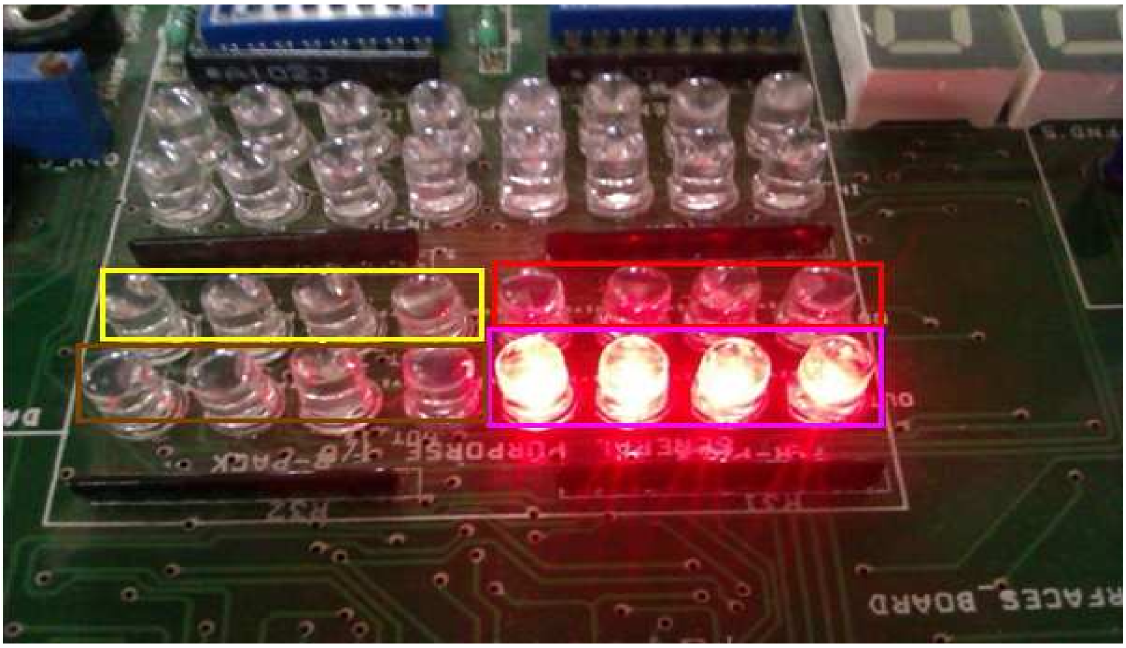


Figure 6: Led Output Image of Strawberry Detection on XC3S200