Image-Based Classification of Areca Nut

Submitted in partial fulfilment of the requirements of the degree of

Bachelor of Technology

By

Bala vamsi

Roll No:160030029

Supervisor:

Samarth Shankar Raut

॥ सा विद्या या विमुक्तये ॥

भारतीय प्रौद्योगिकी संस्थान धारवाड़

Indian Institute of Technology Dharwad

Department of Mechanical Engineering

INDIAN INSTITUTE OF TECHNOLOGY DHARWAD 2019

Approval Sheet

This report entitled **Image-Based classification of Areca nut** by **A.V.Bala Vamsi** is approved for the degree of Bachelor of Technology.

Examiners

Prof. D.Narasimha

Prof. Rajshekhar V Bhat

Advisor

Prof. Samarth S Raut

Chairman

Prof. Somashekara M A

Date: 03/12/2019

Place: Dharwad

Contents

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name: A.V.Bala vamsi

Roll No: 160030029

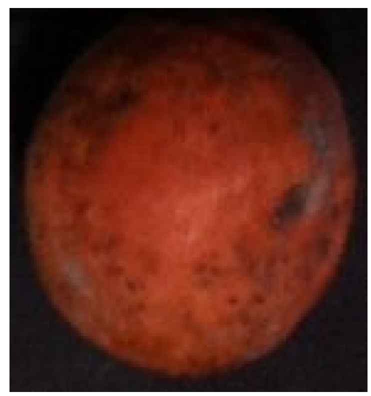
Signature:

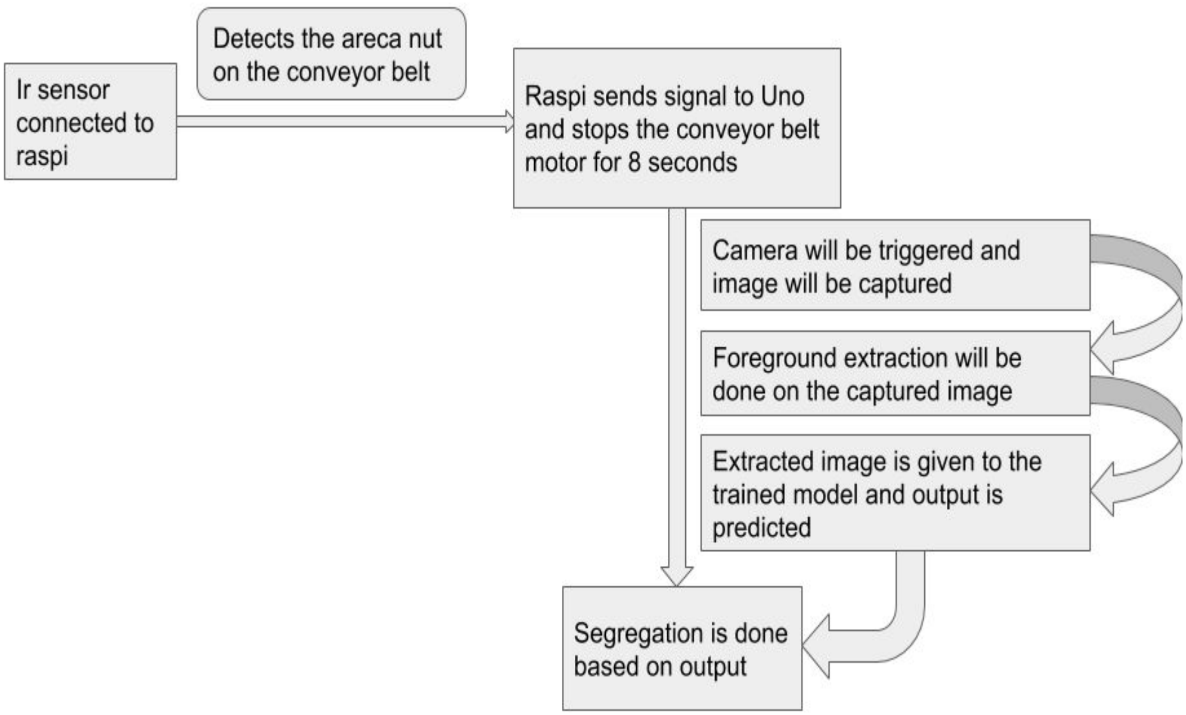
Date: 03/12/2019

Chapter 1

Introduction and Methodology

The areca nut palm is the source of common chewing nut, popularly known as betel nut or Supari. In India, it is extensively used by large sections of people and is very much linked with religious practices. India is the largest producer of areca nut and at the same time the largest consumer also. Major states cultivating this crop are Karnataka (40%), Kerala (25%), Assam (20%), Tamil Nadu, Meghalaya and West Bengal.

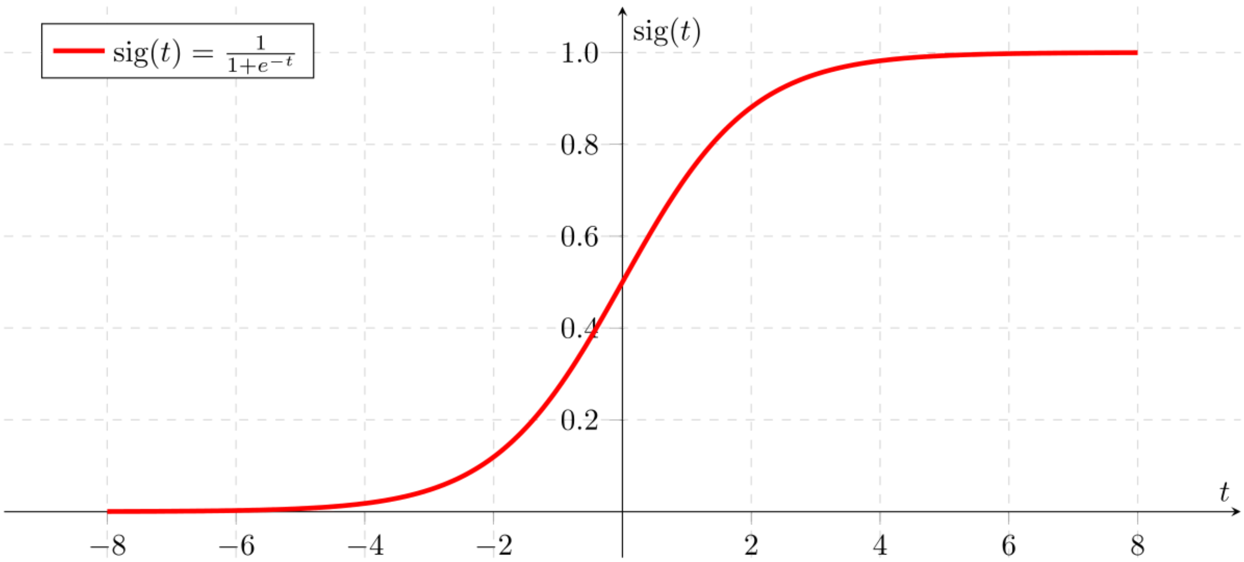



Detects the areca nut on the conveyor belt
Ir sensor connected to raspi
Raspi sends signal to Uno and stops the conveyor belt motor for 8 seconds
Camera will be triggered and image will be captured
Foreground extraction will be done on the captured image
Extracted image is given to the trained model and output is predicted
Segregation is done based on output

Chapter 2

Literature review

2.1) Logistic regression:



- $0 \leq h_w(x) \leq 1$
- $h_w(x) = g(w^T x)$, where, $g(z) = \frac{1}{1+e^{-z}}$
- $g(z)$ is sigmoidal or logistic function.
- $h_w(x) = \frac{1}{1+e^{-w^T x}}$
- $h_w(x) = p(y = 0|x, w)$ vs $h_w(x) = p(y = 1|x, w)$
- $p(y = 0|x, w) + p(y = 1|x, w) = 1.0$

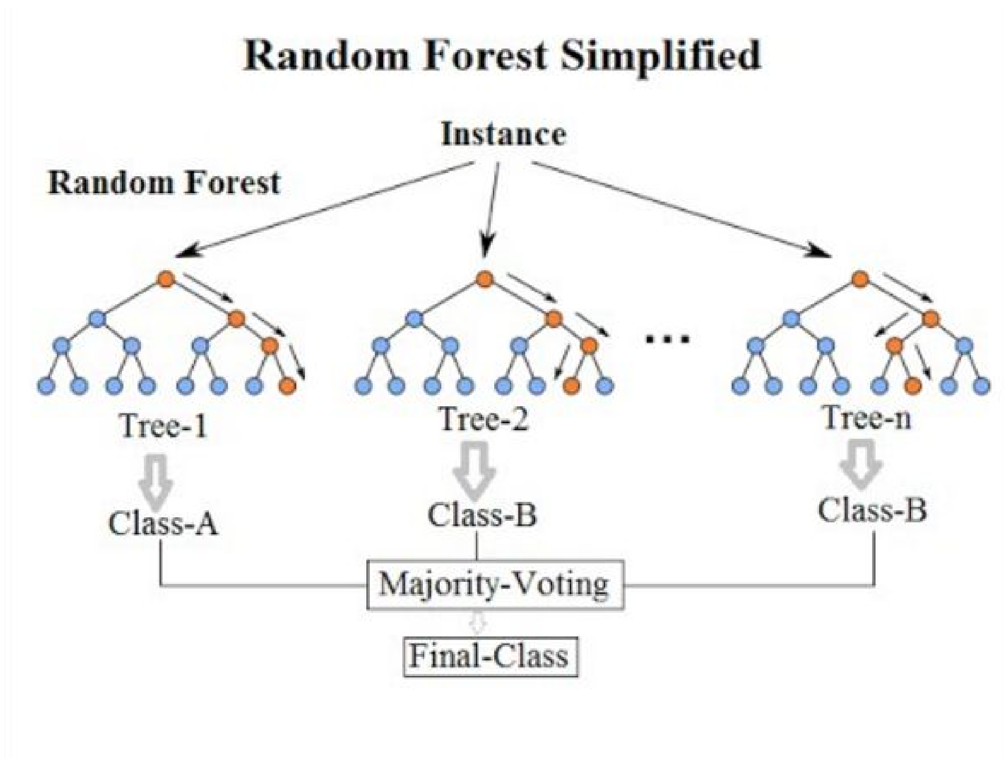
- Predict $y = 1$ when $h_w(x) = g(w^T x) \geq 0.5$
- Similarly, predict $y = 0$ when $h_w(x) = g(w^T x) < 0.5$
- When $z \geq 0$, then $g(z) \geq 0.5$ and $z < 0$, then $g(z) < 0.5$
- When $w^T x \geq 0$, then $g(W^T x) \geq 0.5$ and $w^T x < 0$, then $g(W^T x) < 0.5$

2.2)Support Vector Machine (SVM) :

- **SVM Cost function:**
$$\min_w C(w) = D \sum_{i=1}^{M} y^{(i)} cost_1(w^T x^{(i)}) + (1 - y^{(i)}) cost_0(w^T x^{(i)}) + \frac{1}{2} \sum_{j=1}^{N} w_j^2$$
- For a binary classifier we need, $h_w(x) = 1$, if $w^T x \geq 0$ and $h_w(x) = 0$ otherwise

- However what SVM cost function gives is the following:
- If $y = 1$, then $w^T x \geq 1$ and if $y = 0$, then $w^T x \leq -1$.
- Let D is very large value, $D >> 1 \implies$ for minimization first term should be 0 and cost function boils down to $\frac{1}{2} \sum_{j=1}^{N} w_j^2$.
- Let p is the magnitude or length of projection of vector x on vector w.
- $w^T x = p||w||$, p is a signed value, $+$ve if angle between x and w is $<$ 90 and -ve if it is $>$ 90

2.3)Random Forest classifier :



2.4) Bayes Classifier:

Bayes Theorem:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

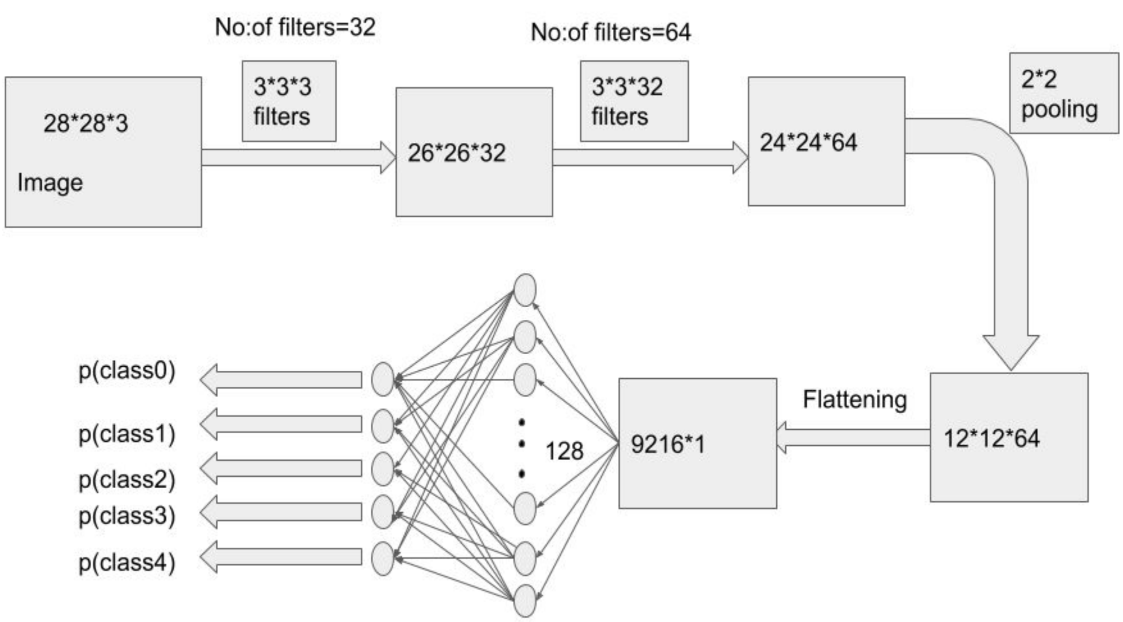
- $p(Y/X) = \frac{p(X/Y)p(Y)}{p(X)} = \frac{p((x_1, x_2, \ldots, x_N/Y)p(Y)}{p(X)}$, where X is an N dim vector.
- $p(x_1, x_2, \ldots, x_n/Y)$ is a joint prob of N dim of features and difficult to estimate.
- Even assuming each dim to be binary, 2^N possibilities.
- In Naive Bayes, a simplified assumption is made i.e., each dim is independent. Therefore, $p(x_1, x_2, \ldots, x_n/Y) = p(x_1/Y)p(x_2/x_1, Y) \ldots p(x_N/x_1, x_2, \ldots, x_{(N-1)}, Y)$
- In Naive Bayes, a simplified assumption is made i.e., each dim is independent. Therefore, $p(x_1/Y)p(x_2/x_1, Y) \ldots p(x_N/x_1, x_2, \ldots, x_{(N-1)}, Y) = p(x_1/Y)p(x_2/Y) \ldots p(x_N/Y)$
- Thus in Naive Bayes, $p(Y/X) = \frac{p((x_1, x_2, \ldots, x_N/Y)p(Y)}{p(X)} = \frac{p(x_1/Y)p(x_2/Y) \ldots p(x_N/Y)p(Y)}{p(X)}$.
- $p(Y/X) = \frac{\Pi_{i=1}^{N} p(x_i/Y)p(Y)}{p(X)}$ In case of binary, $2N$ possibilities in the NR.

For testing : $i = argmax_k p(X_i^{test}/Y = Y_k)p(Y = Y_k)$.

2.5) Convolution Neural network (CNN):

By multiplying an image with filters we extract features (like edges, finding circles in images etc)

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 32)        896

conv2d_2 (Conv2D)            (None, 24, 24, 64)        18496

max_pooling2d_1 (MaxPooling2 (None, 12, 12, 64)        0

dropout_1 (Dropout)          (None, 12, 12, 64)        0

flatten_1 (Flatten)          (None, 9216)              0

dense_1 (Dense)              (None, 128)               1179776

dropout_2 (Dropout)          (None, 128)               0

dense_2 (Dense)              (None, 5)                 645
=================================================================
Total params: 1,199,813
Trainable params: 1,199,813
Non-trainable params: 0
```

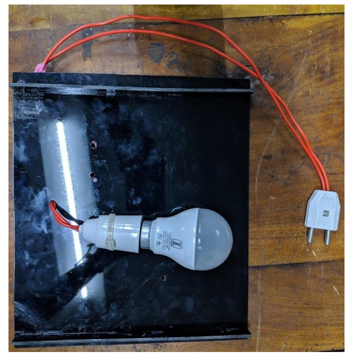


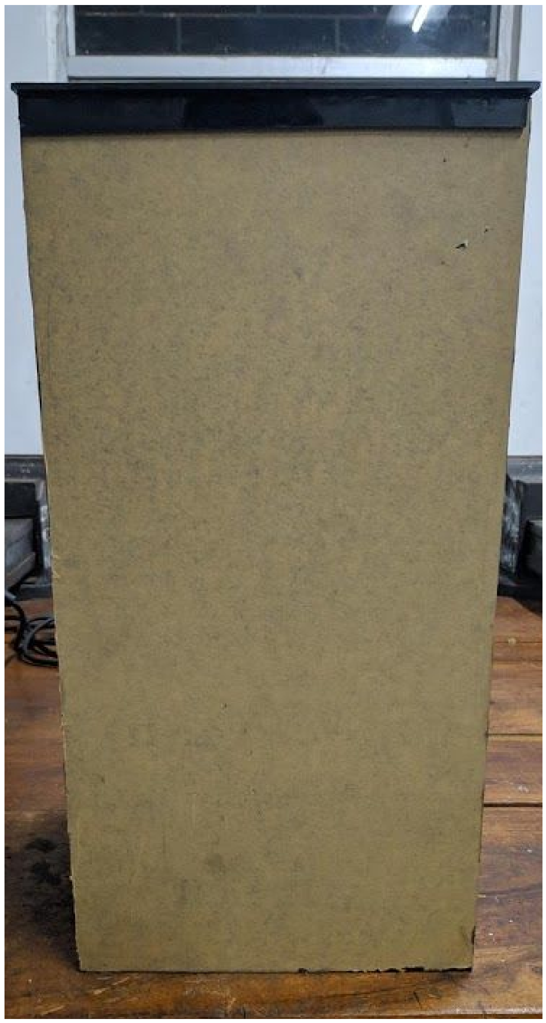
Chapter 3

Present investigation

3.1) Design of hood and its base:

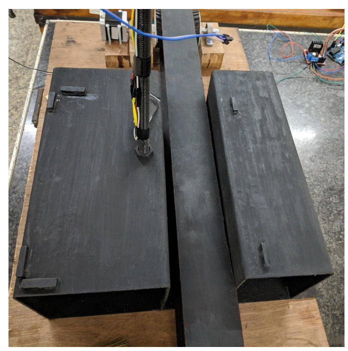
Black Acrylic material was chosen. Since the hood should be strong we didn't go for cardboard/thin sheet metal.





Mild steel was chosen for the base and appropriate dimensions were chosen such that conveyor belt can easily pass through it.

And there were stoppers provided on the base so that hood Sits in the same place every time



Adjustable height mechanism for the camera :

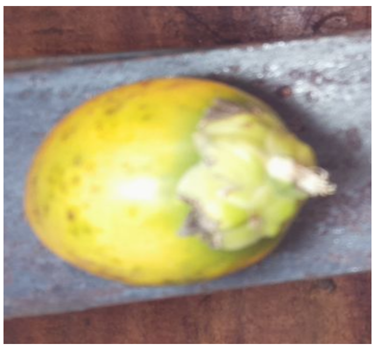
Stud of length 37 cms and 12mms dia is being used, which is fixed to the hood base with a nut.

Raspberry pi is fixed to a 4mm thick Acrylic sheet with holes drilled in it so that it facilitates for heat dissipation.

3.2)Finding Raspberry pi focus point and Building data set :

Finding Focus point: images were taken at different heights from the camera and found the height at which the camera takes good snapshots.

Height=20cms



Height=23cms

Height=27 cms

Height=30cms

It can be clearly seen that at 30cms the camera is taking good snapshots.

Building data set:

- Around 330*4 areca nut images were taken on 4 consecutive days post the harvesting day.
- And also the weight of each areca nut is noted down and compared with remaining 3 days data and found on an average areca nut weight decreases by an amount of 0.966 gms per day.
- Areca nuts were placed in trays and trays were numbered so that we can compare the weight of the same areca nut next day.
- A bash script was used for capturing images and noting down the weights of each areca nut from the terminal

	A	B	C	D	E	F	G
1	**Day 1**	**Day2**	**Day3**	**Day4**	**Day1-Day2**	**Day2-Day3**	**Day3-Day4**
2	28.7	27.72	26.79	25.78	0.98	0.93	1.01
3	28.16	27.26	26.57	25.75	0.9	0.69	0.82
4	25.51	24.65	24.06	23.31	0.86	0.59	0.75
5	25.83	25.08	24.53	23.85	0.75	0.55	0.68
6	25.28	24.36	23.73	22.96	0.92	0.63	0.77
7	33.5	32.25	30.96	29.97	1.25	1.29	0.99
8	24.49	23.04	21.54	19.21	1.45	1.5	2.33
9	31.98	30.72	29.91	28.86	1.26	0.81	1.05
10	28.46	27.55	26.72	25.68	0.91	0.83	1.04
11	31.88	30.69	29.67	28.45	1.19	1.02	1.22
12	30.43	29.18	28.25	27.26	1.25	0.93	0.99
13	23.02	22.32	21.45	20.12	0.7	0.87	1.33
14	23.87	23.18	22.58	21.88	0.69	0.6	0.7
15	26.7	25.94	25.28	24.54	0.76	0.66	0.74
16	27.09	25.98	24.96	23.92	1.11	1.02	1.04
17	26.33	25.46	24.54	23.37	0.87	0.92	1.17
18	26.97	26.17	25.45	24.87	0.8	0.72	0.58
19	32.92	31.92	30.84	29.92	1	1.08	0.92
20	29.19	28.43	27.66	27.12	0.76	0.77	0.54
21	27.09	25.98	24.89	24.06	1.11	1.09	0.83
22	21.62	20.8	20	19.44	0.82	0.8	0.56
23	25.79	25.03	24.3	23.9	0.76	0.73	0.4
24	34.92	33.75	32.67	31.54	1.17	1.08	1.13
25	29.63	28.39	27.24	26.08	1.24	1.15	1.16
26	27.39	26.51	25.59	24.67	0.88	0.92	0.92
27	25.45	24.49	23.53	22.65	0.96	0.96	0.88
28	28.31	27.06	25.94	24.81	1.25	1.12	1.13
29	26	25.08	24.15	23.22	0.92	0.93	0.93
30	27.41	26.27	25.23	24.27	1.14	1.04	0.96
31	27.56	26.77	26.01	25.28	0.79	0.76	0.73
32	27.35	26.26	25.23	24.23	1.09	1.03	1
33	28.34	27.39	26.46	25.5	0.95	0.93	0.96
34	29.65	28.53	27.51	26.46	1.12	1.02	1.05
35	24.33	23.33	22.44	21.55	1	0.89	0.89

Chapter 4

Results and Conclusions

4.1) Extraction Region of interest from the image :

- Since in our case the background of the image acquisition system is fixed we can use thresholding for extracting Region of interest(Areca nut) from the image.
- HSV (Hue, saturation, value) is more preferred over RGB because in RGB space all components (R component, Green component, Blue component) are correlated to the amount of lightning hitting the object and therefore with each other, image descriptions in terms of those components make object discrimination difficult.

Lower and Higher HSV values of interested region

lower = np.array([5, 80, 25])
upper = np.array([100, 255, 180])

- These lower and higher values were acquired by noting the areca nuts RGB value and converting them to HSV colour space and some trial and error are also used.

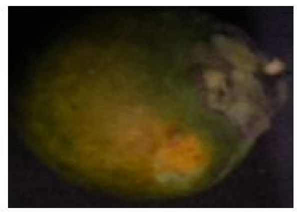
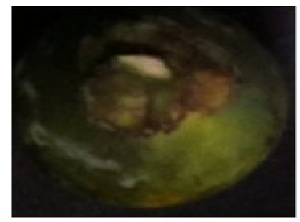
- And there is a function in OpenCV to find contours(a curve joining all points along the boundary which falls in the region of interested HSV limits) cv2.findContours()

- And the area of the contour can be found using cv2.contourArea()

- If the area is greater than some threshold value(in this case 8000) we can crop the image to the region of interest







4.2) Finding Accuracy of above ML Algorithms:

20% validation split was considered(out of the available data i.e 330*4 and model is trained with 80% data and tested with the remaining 20% data)

Epochs	**Batch size**	**Activation fn**	**optimizer**	**Accuracy**
30	30	relu	Adam	**95.6%**
30	30	tanh	Adam	**88.9%**
30	30	sigmoid	Adam	**40%**
30	30	relu	sgd	**88.17%**
40	30	relu	Adam	**91.04%**
60	30	relu	Adam	**90.32%**

- From the above table, we can see that Best optimizer is Adam.
- And best activation function is relu than sigmoid and tanh because it overcomes vanishing gradient descent problem (As we use generalised delta learning rule in backpropagation sigmoid and tanh have derivate for large x with implies no weight is changed)
- And increasing epochs after certain range causes overfitting to data which makes the model less accurate.
- And for very small no: of epochs underfitting to the data happens which also makes the model less accurate.
- And by increasing no: of filters in a convolution layer from (32,64,128) to (64,128,128) Accuracy decreased from 95% to 92% which may be due to overfitting.

So 95% validation accuracy has been achieved using CNN.

classifier	Accuracy
Logistic regression	**96.5%**
Convolutional neural network(CNN)	**95%**
Support Vector Machine(SVM)	**92%**
Random Forest	**86%**
Naive Bayes classifier	**55.55%**

Best of 96.5% accuracy is achieved using Logistic regression

Chapter 5

References

1) Class notes of Prof.SRM Prasanna
2) Machine learning video lectures by Prof. Andrew Ng, Stanford Uty.