**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnana Sangama”, Belagavi-590 018, Karnataka**



**Project Report**

**on**

**“GRADING AND SORTING OF ARECA NUT YIELD”**

**Submitted in partial fulfilment of the requirements for the award of the degree of**

**Bachelor of Engineering**

**in**

**Computer Science & Engineering**

|  |  |
| --- | --- |
|  | **Submitted by** |
| **USN** | **NAME** |
| **1BI16CS044** | **ELSON DSA** |
| **1BI16CS045** | **GANESH NAIDU N** |
| **1BI16CS054** | **H S SUNAG HEMMANNA** |
| **1BI16CS059** | **SOURAV PAI K** |
|  | Under the guidance of |
|  | **Prof. Y Shobha** |
|  | Associate Professor |
|  | Dept. of CS&E, BIT |



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**BANGALORE INSTITUTE OF TECHNOLOGY**

K. R. Road, V. V. Pura, Bengaluru-560 004

**2019-2020**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnana Sangama”,** Belagavi-590 018, Karnataka

**BANGALORE INSTITUTE OF TECHNOLOGY**

Bengaluru-560 004



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

***Certificate***

Certified that the Project work entitled **GRADING AND SORTING OF THE ARECANUT YIELD** carried out by

|  |  |
| --- | --- |
| **USN** | **NAME** |
| **1BI16CS044** | **ELSON DSA** |
| **1BI16CS045** | **GANESH NAIDU N** |
| **1BI16CS054** | **H S SUNAG HEMMANNA** |
| **1BI16CS059** | **SOURAV PAI K** |

bonafides student of Bangalore Institute of Technology in partial fulfillment for the award of Bachelor of Engineering in Computer Science & Engineering of the **Visvesvaraya** **Technological University, Belagavi** during the academic year 2019-20. It is certified that allcorrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

**Prof. Y Shobha**

Associate Professor Department of CS&E,BIT

**Dr. Asha T**

Professor and Head

Department of CS&E, BIT

**Dr. M .U Aswath** Principal, BIT

**External Viva**

**Name of the examiners, signature with date**

**1.**

**2.**

**DEDICATION**

Computer Science has been very close to my heart, it teaches all of us things which can be applied in other aspects of life, most of the developments in Computer Science are either motivated by the working of human body or its targeted at simulating the same, this project not only is for the partial fulfilment of the bachelor’s degree, it is for my love towards Computer Science.

Elson DSa

1BI16CS044

i

**ACKNOWLEDGEMENT**

The knowledge & satisfaction that accompany the successful completion of any task would be incomplete without mention of people who made it possible, whose guidance and encouragement crowned my effort with success. We would like to thank all and acknowledge the help we have received to carry out this final year project.

We would like to convey my sincere thanks to our college the **Bangalore Institute of**

**Technology, Dr. Aswath M U Principal and Dr. Asha T, Professor and Head of the Department for Computer Science and Engineering**, for being kind enough to provide anopportunity and platform to complete and present my final year project “Grading and Sorting of the Areca nut yield”.

We would also like to extend our thanks to our project coordinator **Prof. Nagamani D R**, Assistant Professor and **Prof. Shruthiba A** Assistant Professor, Department of Computer Science and Engineering for her support and guidance.

We are most humbled to mention the enthusiastic influence provided during the development and ideation phase by our guide **Prof. Y Shobha** Associate Professor, Department of Computer Science and Engineering for her ideas, time to time suggestions and co-operation showed during the venture and helping make my final year project a success.

We would also take this opportunity to thank my friends and family for their constant support and help. I'm very much pleasured to express my sincere gratitude to the friendly co-operation showed by all the staff members of Computer Science Department, BIT.

ELSON DSA (1BI16CS044)

GANESH NAIDU N (1BI16CS045)

H S SUNAG HEMMANNA (1BI16CS054)

SOURAV PAI K (1BI16CS059)

ii

**ABSTRACT**

With the emergence of Machine Learning and Digital Image processing techniques, the classification of the images is performed. Classifying images without human intervention is the need of the hour, since it reduces human labor and the cost for it. We use Digital Image Processing algorithms and Convolutional Neural Networks to extract the features and classify Areca nut.

iii

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CHAPTER 1 – INTRODUCTION** | | | **1-5** |  |
|  |  |  |  |  |
| 1.1 | Introduction | | 1 |  |
|  |  |  |  |  |
| 1.2 | Motivation | | 5 |  |
|  |  |  |  |  |
| **CHAPTER 2 - LITERATURE SURVEY** | | | **6-20** |  |
|  |  |  |  |  |
| 2.1 | Generative Adversarial Networks | | 6 |  |
|  |  |  |  |  |
| 2.2 | Image-to-Image Translation with Conditional Adversarial Networks | | 10 |  |
|  |  |  |  |  |
| 2.3 | Image Style Transfer Using Convolutional Neural Networks | | 14 |  |
|  |  |  |  |  |
| 2.4 | Landmark Assisted CycleGAN for Cartoon Face Generation | | 18 |  |
|  |  |  |  |  |
| **CHAPTER 3 - PROBLEM STATEMENT** | | | **21-21** |  |
|  |  |  |  |  |
| **CHAPTER 4 - SYSTEM REQUIREMENTS** | | | **22-26** |  |
|  |  |  |  |  |
| 4.1 | Functional & Non Functional Requirements | | 22 |  |
|  |  |  |  |  |
| 4.2 | Software Requirements | | 22 |  |
|  |  |  |  |  |
| 4.3 | Hardware Requirements | | 26 |  |
|  |  |  |  |  |
| **CHAPTER 5 - DESIGN** | | | **27-40** |  |
|  |  |  |  |  |
| **5.1** | Architecture | | 27 |  |
|  |  |  |  |  |
|  | **5.2.1** | Pre-processing | 28 |  |
|  |  |  |  |  |
| **5.2** | **5.2.2** | Discriminator | 32 |  |
|  |  |  |  |  |
|  | **5.2.3** | Generator | 33 |  |
|  |  |  |  |  |
| **5.3** | Use Case Diagram | | 35 |  |
|  |  | |  |  |
| **5.4** | Sequence Diagram | | 35 |  |
|  |  |  |  |  |
|  | **5.5.1** | Basic Error Functions | 37 |  |
|  |  |  |  |  |
|  | **5.5.2** | Error Function of Generator | 38 |  |
| **5.5** |  |  |  |  |
| **5.5.3** | Error Function of Discriminator | 38 |  |
|  |  |
|  |  |  |  |  |
|  | **5.5.4** | Algorithm | 38 |  |
|  |  |  |  |  |
| **CHAPTER 6 - IMPLEMENTATION** | | | **41-57** |  |
|  |  |  |  |  |

iv

|  |  |  |  |
| --- | --- | --- | --- |
| **6.1** | Model API | | 41 |
|  |  | |  |
| **6.2** | Flask Application | | 44 |
|  |  | |  |
| **6.3** | Pix2Pix Model Implementation | | 47 |
|  |  |  |  |
| **CHAPTER 7** | | **- TESTING & RESULTS** | **58-61** |
|  |  | |  |
| **7.1** | Unit Testing | | 58 |
|  |  |  |  |
| **7.2** | Results |  | 59 |
|  |  |  |  |
| **CHAPTER 8** | | **- APPLICATIONS** | **62-62** |
|  | |  |  |
| **CHAPTER 9** | | **- CONCLUSION & FUTURE WORK** | **63-63** |
|  |  | |  |
| **9.1** | Conclusion | | 63 |
|  |  | |  |
| **9.2** | Future Work | | 63 |
|  |  | |  |
| **CHAPTER 10 - BIBLIOGRAPHY** | | | **64-64** |
|  |  |  |  |

v

**LIST OF FIGURES**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Figure** |  |  | **Description** |  |  | **Page** |  |  |
|  | **No.** |  |  |  |  | **No.** |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **2.1** | |  |  | General workflow in GAN | | 6 | |  |  |
|  | |  |  |  | |  | |  |  |
| **2.2** | |  |  | Mode Collapse Problem | | 10 | |  |  |
|  | |  |  |  | |  | |  |  |
| **2.3** | |  |  | Choices for architecture of the generator. The “U-Net” is an encoder-decoder | | 12 | |  |  |
|  | |  |  |  | |  | |  |  |
| **2.4** | |  |  | Image representations in a CNN | | 15 | |  |  |
|  | |  |  |  | |  | |  |  |
| **2.5** | |  |  | Output of CycleGAN | | 19 | |  |  |
|  | |  |  |  | |  | |  |  |
| **5.1** | |  |  | Architecture | | 27 | |  |  |
|  | |  |  |  | |  | |  |  |
| **5.2** | |  |  | Pre-processing | | 29 | |  |  |
|  | |  |  |  | |  | |  |  |
| **5.3** | |  |  | Discriminator Network | | 32 | |  |  |
|  | |  |  |  | |  | |  |  |
| **5.4** | |  |  | UNet Generator Network | | 34 | |  |  |
|  | |  |  |  | |  | |  |  |
| **5.5** | |  |  | Use Case Diagram | | 35 | |  |  |
|  | |  |  |  | |  | |  |  |
| **5.6** | |  |  | Sequence Diagram | | 36 | |  |  |
|  | |  |  |  | |  | |  |  |
| **5.7** | |  |  | Generator Training | | 39 | |  |  |
|  | |  |  |  | |  | |  |  |
| **5.8** | |  |  | Discriminator Training | | 40 | |  |  |
|  | |  |  |  | |  | |  |  |
| **7.1** | |  |  | Image from Training Set | | 59 | |  |  |
|  | |  |  |  | |  | |  |  |
| **7.2** | |  |  | Image from Testing Set | | 59 | |  |  |
|  | |  |  |  | |  | |  |  |
| **7.3** | |  |  | Image 1 from Validation Set | | 60 | |  |  |
|  | |  |  |  | |  | |  |  |
| **7.4** | |  |  | Image 2 from Validation Set | | 60 | |  |  |
|  | |  |  |  | |  | |  |  |
| **7.5** | |  |  | Image with more than One Face | | 60 | |  |  |
|  | |  |  |  | |  | |  |  |
| **7.6** | |  |  | Image with No Faces | | 61 | |  |  |
|  | |  |  |  | |  | |  |  |
| **7.7** | |  |  | Image taken with Bad Camera Quality & Lighting | | 61 | |  |  |
|  | |  |  |  | |  | |  |  |
| **7.8** | |  |  | Image taken with Good Camera Quality & Lighting | | 61 | |  |  |
|  |  |  |  |  |  |  |  |  |  |

vi

**LIST OF TABLES**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Table** |  |  | **Description** |  |  | **Page** |  |  |
|  | **No.** |  |  |  |  | **No.** |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **4.1** | |  |  | Functional & Non Functional Requirements | | 22 | |  |  |
|  | |  |  |  | |  | |  |  |
| **7.1** | |  |  | Unit Testing | | 58 | |  |  |
|  |  |  |  |  |  |  |  |  |  |

vii

**CHAPTER 1**

**INTRODUCTION**

**INTRODUCTION**

* 1. **INTRODUCTION**

Areca nut is the crop grown mainly in South Asia and East Africa. In India, it is grown in coastal regions of Karnataka and Kerala. The word Areca is derived from the Kannada word Adike. For one kilogram of Areca, the market value is around rupees 300 as of 2020. The invaders to India took the crop from India to Europe in 1600s. India is the leader in production of Areca, with almost half of the world’s total production comes from India. It is also produced in countries like Bangladesh, Myanmar and Taiwan.

It is an annual crop. Areca nut gives huge profit to the farmers. Areca nut is the seed of a fruit. It is not a nut as the name indicates. Traditionally, they are chewed along with the betel leaves. They are used in medicines. But these also have adverse effect on human health. In 2003, a group sponsored by World Health Organization found that chewing Areca nuts can cause cancer. It is financially accessible in dried, relieved, and new structures. At the point when the husk of the new natural product is green, the nut inside is sufficiently delicate to be cut with an ordinary blade. In the ready organic product, the husk gets yellow or orange, and as it dries, the natural product inside solidifies to a wood-like consistency. At that stage, the areca nut must be cut utilizing an uncommon scissors-like shaper.

As a rule for biting, a couple of cuts of the nut are enveloped by a betel leaf alongside calcium hydroxide (slaked lime) and may incorporate clove, cardamom, catechu (kattha), or different flavors for additional enhancing. Betel leaf has a new, peppery taste, yet it can likewise be unpleasant to changing degrees relying upon the assortment.

Areca nuts are bitten with betel leaf for their belongings as a gentle stimulant, creating a warming uproar in the body and somewhat uplifted sharpness, in spite of the fact that the impacts shift from individual to individual.

The areca nut contains the tannins arecatannin and gallic corrosive; a fixed oil gum; a little terpineol; lignin; different saline substances; and three primary alkaloids—arecoline, arecaidine, and guvacine—all of which have vasoconstricting properties. The betel leaf bit alongside the nut contains eugenol, another vasoconstrictor. Tobacco leaf is frequently added to the blend, accordingly including the impact of nicotine.

In parts of India, Sri Lanka, and southern China, areca nuts are bitten alongside betel leaf, but on the other hand are utilized in the readiness of Ayurvedic and customary Chinese meds.

Powdered areca nut is utilized as a constituent in some dentifrices. Other customary uses incorporate the evacuation of tapeworms and other intestinal parasites by gulping a couple of teaspoons of powdered areca nut, inebriated as a decoction, or by taking tablets containing the extricated alkaloids. According to conventional Ayurvedic medication, biting areca nut and betel leaf is a decent cure against awful breath. Diplomat Edmund Roberts noticed that Chinese individuals would blend areca nut with Uncaria gambir during his visit to China in the 1830s. After biting a betelnut, the red buildup is by and large disagreement out. As needs be, places have restricted biting this nut to maintain a strategic distance from blemishes.

* 1. **MOTIVATION**
* Farmers face huge amount of loss due to the increase of the labour cost.
* Providing training for the classification task is difficult.
* Numbers of skilled workers are less.
* Huge loss occurs due to improper classification at the market level.

**CHAPTER 2**

**LITERATURE SURVEY**

**LITERATURE SURVEY**

**2.1 Survey of Grading Process for Agricultural Foods by Using Artificial Intelligence Technique**

In industrial world the currentdemand of the consumer is better quality of theproducts. The grading system is used to detect thequality of the products. This system have beenprocessed manually where it is not efficient andtime consuming. To overcome these problems andto reduce the labour requirements, the machinevision technique is developed for grading system.The hardware and software is needed for themachine vision system. The hardware process likecamera and the computer is needed to capture theimages of the products. Then the features of therespective images must be collected throughcomputer and analysed each and every featuresbased on the image processing techniques.Therefore the quality of the product can bedetermined easily and it reduces the time. To getan accurate value of the quality the Artificial Intelligence techniques are developed to improve he grading system. Mostly the industrial products of food industry like agricultural products are well developed by using this Artificial Intelligence techniques of the grading system. Thus the most relevant paper are collected and its drawbacks are listed on the below papers. The future work of the proposed system is the Artificial Intelligent technique are used as a classifier for the grading system to detect the accurate quality of the products. Where this proposed system is mostly applicable to detect the quality of the fruits but this system can be implemented to other industrial products in terms of safety of the human beings and also it is applicable to multiple products resulted in improving the perfect grading system.

The Artificial Intelligence technique (classifier) methods are:

* Neural Network (NN)
* Support Vector Machine (SVM)
* Back Propagation Neural Network (BPNN)
* Fuzzy Logic Based (FLB)
* Support Vector Regression (SVR)

|  |  |  |  |
| --- | --- | --- | --- |
| **Fruits** | **Features** | **Technique** | **Accur**  **-acy** |
| Mango | Colour of the image changes. | IR vision sensor and Gaussian Mixture Model. | Not specified. |
| Harumani mangoes | Shape, colour, weight of the mangoes. | Fourier Based separation  model. | 90%. |
| Cashew | Colour, texture, size and shapeof cashews. | Multi resolution Wavelet transform and AI(classifier)of SVM  andBPNN | 95%. |
| Cherrytomato | Colour, texture, shape (external and internal) characteristics | AI  technique of SVM and KNN  classifier. | Not specifi ed. |
| Peanut | Shape, texture  and colour of peanuts. | AI  technique of BPNN. | Not  specifi ed. |
| Apple | Skin or surface of the apple and colour. | AI  technique of FNN and SVM. | 89%. |
| Mango | Size, shape,  weight and surface defects of mangoes. | AI  technique of SVR, MADMand  FIL. | 87%. |

**Tab 2.1 Accuracy for different Fruits with Technique**

The grading system is more important in the current world demand in terms of health and the safe of the human beings. The grading process is more developed in many industrial to detect the quality of theproducts.Fromtheabovelistedpaperswherethe machine vision-based grading system is more applicable to the agricultural products especially in the fruits industries. To replace the manual grading the machine vision-based technique is used and it requires the hardware and software to analyse the quality of the products. The hardware like camera, conveyer belt, light sensors, speed control sensors is needed depend on the product we grade. The acquired images features are extracted and pre- processing the images based on image processing techniques. Finally, the suitable Artificial Intelligence technique is used for the classification and to analyse the approximate value of the quality detection. Mostly the Artificial intelligence technique is used in the agriculturalproducts.

**2.2 Machine vision based real time cashew grading and sorting system using SVM and back propagation neural network**

In this paper, a fully automated cashew grading system has been developed. Various external features of the cashew kernel such as color, texture, shape and size are extracted from the captured image using divergent techniques. All these features are necessary to grade the various cashew kernels efficiently. Texture Features are extracted from the co- occurrence matrix of the second level detail wavelet coefficients and contourlet coefficients. Classification performed using three texture features individually(Energy,contrast, homogeneity) is found to be efficient in contourlet domain so that it can reduce feature vector dimension. Both SVM and BPNN are used for performance analysis of the classifier with different kernel functions for SVM. It is found that classifier with RBF kernel function provides high accuracy (99.6%) than BPNN. and the accuracy of real time prototype cashew sorter is96.6%.

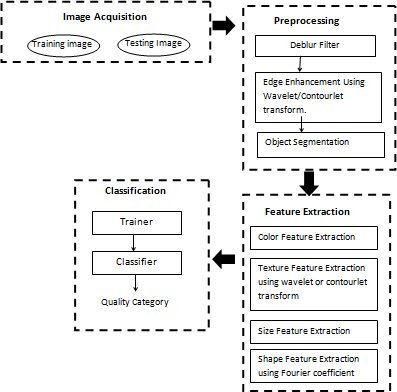
In today’s world consumers are greatly aware about quality of food products. So there is a great need to build automated quality management systems. Benefits of automating the quality management include reduced production cost and overall improvement in quality. Nowadays great deal of research is going on in the area of machine vision based grading of food products. Grading and sorting of cashew kernels are done manually in most of the countries which is time consuming and expensive. In this paper a real time classification system to automatically grade cashew kernels based on their color, texture, size and shape feature are presented. Multi resolutional wavelet and Contourlet transform are used for extracting texture features. The images of the cashew kernel are acquired using a Charge Coupled Device (CCD) camera, and then the images are pre-processed using an efficient background subtraction technique. Then various external features are extracted using machine learning techniques. For the experimental study, cashew kernels of five different varieties are collected. SVM and back propagation neural network classifiers are used and their performance in terms of accuracy isobserved.

DESIGNOFTHEPROPOSEDSYSTEM

The proposed system in this paper works in four phases namely image acquisition, pre-processing, feature extraction and classification as shown in Fig.1. By keeping the image acquisition phase same as that of existing works, the authors have focused on the remaining three stages to make the necessary enhancements. In preprocessing stage, the deburred image is passed through an edge enhancement process prior to object segmentation. In object segmentation, a comparison has been made with various object segmentation algorithms. In Feature Extraction stage, rather than taking texture features from spatial domain, they are extracted using co-occurrence matrix of the wavelet and contourlet coefficients of the input image. Also shape features are extracted using Fourier coefficient method. Finally, we have made a comparison using Back propagation neural network and SVM classifiers.

1. *Image Acquisition*

For capturing the image, a Basler scA1390-17g GigE camera with the Sony ICX267 CCD sensor that delivers 17 frames per second at 1.4MP resolution is used. An illumination chamber with 8 watt T5 Fluorescent Lamp lighting is also used. The Image acquisition toolbox of Matlab2015 software was used to capture the cashew kernel images. All the images of the cashew samples are of 1390 X 1038 sizes. The same lighting condition with same camera exposure and focus mode was used to acquire all images. A Kaiser RS 2 XA Camera Stand was used to place the camera. The object was placed at the base and the camera was placed at a height of 30cm above the base. To transport he captured image frames to the computer, D-LinkCat-61000Base-T Ethernet cable were used. The major units of this real time system consists of a conveying unit, a programmable logic controller unit ,solid-state relay switching devices, electronic circuitry ,air blowers, air compressor, sensors to trigger the camera as well as valves and signalingunit.Cashew kernel images were acquired using Image acquisition toolbox of Matlab 2015 software. Images of the entire cashew samples 1390 X1038 size were captured under same lighting condition and with same camera exposure and focus mode. The camera was placed on Kaiser RS 2 XA Camera Stand, at a height of 30 cm above the base where the object was placed. The image frames were transported to the computer via D-Link Cat-6 1000Base-T Ethernet cable. The real time system consisted of conveying unit, a programmable logic controller unit, solid-state relay switching devices, electronic circuitry, air blowers, air compressor, sensors to trigger the camera as well as valves and signaling unit.

  **Fig 2.2.1 Proposed System Design**

1. *ImagePre-processing*

Image preprocessing techniques can be applied to make the subsequent steps easier and error free. Certain samples were blurred and hence lucy filter has been applied to eliminate the blurring effect. Then a High pass sharpened image can be obtained using wavelet transform which provide good result for accurate segmentation. All the steps are shown inFig.2.

For the Comparative study, Edge enhancement can be done using second level contourlet transform using ‘haar’ pyramidal filter and 'pkva' Directional filter. Then Image segmentation techniques have been applied to split the pixels of the image in to two subsets: object area, and the background. In this work, a black-gray background has been used, aiming to choose backgrounds with different spectral characteristics than the cashew kernel. In this way, maximal contrast between the white/ivory cashew kernel and the background was achieved. In real-time experiments also the color of the conveyor belt where the cashew kernel is passing through, is chosen as black. In this work, a combined method which involves both genetic algorithm based thresholding and morphological processing techniquefor removing the background from the captured cashew image has been used.

1. *Feature Extraction*

The purpose of feature extraction is to reduce the original data set by measuring certain properties or features that distinguish one input pattern from another . The extracted features act as input to the classifier. Most significant features to assess the cashew grade are its color, texture, size and shape. In the present work, Texture features from the co- occurrence matrices of high frequency subbands are used since these subbands represents most clearest appearance of the changes between different textures. The various features extracted for subsequent classification were color, texture, shape and size. Since RGB is a poor choice for color analysis, HSV color moments such as mean, standard deviation and skewness were extracted as most of the color information is contained in these three moments .

The gray level co-occurrence matrix (GLCM) is a way of extracting second order statistical texture features . Out of these fourteen features, five of the textural features are considered to be most relevant for this proposed method. Those features are Energy, Contrast, Correlation and Homogeneity andentropy.

*Algorithm for Feature Extraction*

Step 1: Decompose input image using 2-D Wavelet or Contourlet transform after converting RGB image into grayscale.

Step 2: Derive Co-occurrence matrices for high frequency sub bands of wavelet or contourlet with 1 for distance and 0; 45; 90 and 135 degrees for ** andaveraged. Step 3: From these co-occurrence matrices, five Haralick texture features called Co-occurrence Texture features are extracted.

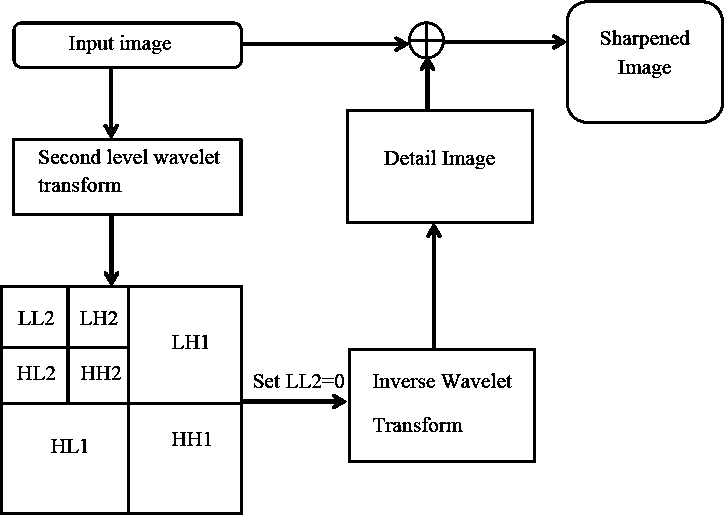
The size feature was calculated by counting the number of non-zero pixels in the segmented image and then normalizing it with the total number ofpixels

In order to extract the shape features, Fourier coefficient method has been used. This method involves the following steps to estimate the shape feature.

* 1. Estimate the outermost boundary points of the cashew kernel region, Let N be the total number ofpixels.
  2. Determine the centroid ( *xc*, *yc* ) of the kernelregion.
  3. Find Euclidian distance R(k) from each boundary point( *xk*, *yk* ) to thecentroid.
  4. Discrete Fourier Transform is applied to R(k),resulting one dimensional feature vector of the cashewkernel.

Only the first few coefficients are distinct and can be used to distinguish the difference between cashew kernel shapes. Therefore only the first fourier coefficient was used in this study.

Classification is the final stage in the cashew grading process. Support Vector Machine (SVM) is a powerful binary supervised classifier and accurate learning technique. It is very suitable for nonlinear classification. Here the basic idea is to map feature vectors nonlinearly to another space and learn a linear classifier there. The linear classifier in new space would be an appropriate nonlinear in the original space. Kernel functions effectively map the original feature vectors into higher dimensional space without explicit calculation. Many types of kernels are available for SVM. In this work, Linear, Polynomial, Quadratic and Radial Basis Function kernel has been used. For the classification purpose, a comparative study has been made with Back Propagation Neural Network and SVMclassifier.



**Fig.2.2.2 Edge Enhancement using wavelet transform**

|  |  |  |
| --- | --- | --- |
| Kernel Function | Accuracy (%) | Execution Time (Seconds) |
| Quadratic | 96 | 0.452 |
| Polynomial | 97.2 | 0.482 |
| Linear | 97.5 | 0.170 |
| RBF | 99.6 | 0.177 |

**TAB 2.2.1 Performance evaluation of SVM Classifier using different kernel functions**



|  |  |
| --- | --- |
| Classifier | Accuracy  (%) |
| SVM | 99.6 |
| BPNN | 97.7 |

**TAB 2.2.2 Performance evaluation of SVM Classifier with BPNN**

Though the classifier gave superlative results for cashew kernel grading, the performance of the system is slightly lowering when the speed of the conveyer belt is increasing. Future work includes incorporating techniques for identifying damaged cashews based on their color so that such cashews can be eliminated from the grading process.

**2.3 A Review of Application of Computer-vision for Quality Grading of Food Products**

The purpose of this paper is to review the various applications of computer-vision systems in classifying the food products based on various quality grading parameters. This paper has showcased how various classification techniques based on Decision Trees, Artificial Neural Networks, Digital Image Processing is being used for the classification of food products. This study covers a strong case for further application of computer-vision techniques in the field of quality grading of food products. This inquiry helps us to understand that researchers can contribute to the areas of application of Decision Trees, Artificial Neural Networks, Image Processing to aid computer-vision.

Grading is prerequisite for making any commodity marketable. In the food processing industry, there are many parameters to be considered when deciding whether a particular food product meets the required standards. Generally, with food, these parameters include visual and olfactory factors such as colour, texture, flavor etc. In food processing, this data has to be first extracted from the food products being analyzed. This information may be in the form of images of the food products. From this data, specific parameters such as texture and colour need to be extracted and compared with the set standards. If the quality parameters extracted fall within the range specified by the quality standards, then the given food product may be classified as marketable. If these parameters do not meet the required standards, then they can be classified as sub-standard or non- marketable. This paper discusses the various soft tools which help in this decision-making. This will help in building modelswhich can further assist in automating the inspection of food products for the food processing industry. Automation of this inspection will thus decrease the human error which may creep in during manual inspection of foodproducts.Quality grading primarily consists of a classification of the food products by classifying based on various quality parameters. This idea is to see that each of the values of the quality parameters is ranked based on the well-defined threshold limits.

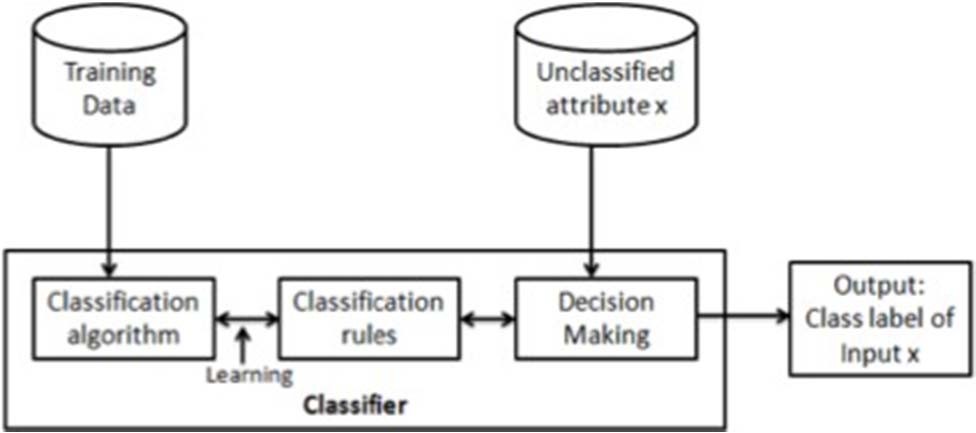
1. *Meaning of Classification*

Classification is the process of identifying various items and then assigning them to groups with similar characteristics for the ease of use.

1. *Steps taken for Classification*

Classification starts by collecting a data set whose class assignments are already known. Then a classification model is built based on this observed data. Various attributes of the received data and the target class assignments act as predictors. Predictors are parameters which are likely to influence the behavior of results. In predictive modeling the following steps areused:

*Learning:* The Classification algorithm analyses the training data. Training data is pre-processed data that contains records whose class labels are known. The algorithm creates a relationship between to determine the class label of a given tuple using the training data. The firststepofclassificationstudiesthefunctiony=f(x). which can in- turn estimate the class label y of a given tuple x. Typically, this is represented in the form of mathematical formulae, decision trees, or classification rules. This is called the inductive step of classification.

* *Classification:* Once the rules are established in the learning phase, test data (containing records with unknown class tags) are used to anticipate the accuracy of these classification rules. If it is considered acceptable, the rules can be applied to the classification of new data tuples (flowchart shown in figure 1). This is called the deductive stage ofclassification.

**Fig 2.3.1 Steps of Classification**

In short, data is collected, and relevant predictors are identified, and then a statistical model is constructed which establishes relationships between the predictors and results (classification rules). Further, predictions are made, and the model is validated when any additional data is received.

This model can then be applied to a different data set in which the class assignments are unknown. In the simplest type of classification (binary classification), the target attribute can fall in one of 2 possibleclasses.

Classifications are discrete. Continuous, floating-point values would indicate a numerical, rather than a certain target. A predictive model with a numerical target uses a regression algorithm, not a classification algorithm.

1. *Assessment of Classification and Prediction*

Table 1 defines the various criteria for the assessment of classification and prediction.

Evaluation of the performance of a classification model is dependent on the number of test data records correctly and incorrectly predicted by the model. These numbers are tabulated in a confusion matrix . Where accuracy is defined as the ratio of the number of correct predictions to the total number of predictions and the error rate is the ratio of the number of wrong predictions to the total number of predictions.

There are some measures like coefficient p-values and R2 values which measure the overall goodness of fit of the classification models. These statistical measures express how well the training data fit the model and do not explicitly indicate how well the model will perform on future data.

1. CLASSIFICATION USING DECISIONTREES

As the name suggests, decision tree-based classifications have a tree-shaped structure. They partition data sets into successively smaller subsets while simultaneously incrementally developing an associated decision.

A Decision Tree includes root node, branch and leaf node. The labeled diagram of a decision tree can be seen in figure 2.



**Fig2.3.2 Decision Tree**

1. *Advantages of Decision Trees:*

They do not require any domain knowledge. Therefore, they are appropriate knowledge discovery.

1. The representation of rules in tree form is intuitive and therefore intuitive to humans.
2. They can handle multidimensionaldata.
3. Both the stages of decision trees are simple andfast.
4. *Tree Pruning*

This process is utilized to remove any abnormalities from the training data. After the pruning process, the trees are much smaller and comparatively less complex.

The following are the stages of pruning:

* Pre-pruning: In this stage, the tree is pruned by terminating its construction in the earlystages.
* Post-pruning: This stage comprises the removal ofsub- trees from fully growntrees.

1. *Attribute SelectionMeasures*

In decision trees, there is always the question of how to split the given data sets into smallersubsets.

Attribute selection measures can be utilized to rank attributes based on their ability to split the tuples at a given node. A comparison is made and only the attribute with the best score for the measure is selected as the splitting attribute for the given tuples.

**2.4 Possible approaches to arecanut sorting / grading using computer vision: A brief review**

Different image processing-based techniques are taken into consideration in this paper for the classification of Arecanut with various characteristics such as color, shape and texture etc. Computer vision systems have been used extensively in the real world for inspection and evaluation purposes as they can provide faster, efficient, healthful, precise and clear assessment. Till today classification of Arecanut isdone manually with the skills developed by long practice, invariably wrought with human errors. Therefore there is adire need for the technology that minimizesmanpowerandefficientlyincreasesproductivity. Error free assessment of Arecanut can bestbeachieved through computer vision systems. Still therearecertain challenges, such as, relatively the growth or upwakeof computer vision technologyincommercialsectors isvery slow. Even though researchersproduceefficient,adequate and accuratealgorithms,processingspeeds didn’t meet the modern-daymanufacturingrequirements.However, thereview reveals thatalthough being culturallyandeconomicallyimportant,work done in respect of Arecanut is inadequate and still much needs to bedone.

Computer science is involved to the greater extent in agricultural and food science these days. Many Artificial Intelligence and soft computing techniques and technologies are used for classification and defect detection of various products and thus helps in *Bette*r quality product for the end users. In this paper we focus on the standing of Arecanut in global and Indian market and usage of computer vision and image processing in an Arecanut classification and grading system. It is essential to take into consideration cultural and economic importance of Arecanut to determine the importance of computer vision technology for Arecanut. There areso many challenges to face in order to develop a system for automatic classification of Arecanut using images. Depending on the category and the region they are grown; several varieties of Arecanut are subject to significant difference in color, texture and shape. Various methods are used to process Arecanut mainly focusing on the external appearance of the product. Solution for classifying/grading Arecanut can be developed using its color, size and texture. We havealso quoted the important work accomplished in respect of Arecanut from the Computer vision perspective and on some other fruits as well. The main motto of this article is to provide in-depth introduction to Arecanut, Computer Vision, need and applications of vision based technology in classification and grading of Arecanut.

Agriculture plays a significant role in socio- economic development of the country. It is Indian economy’s backbone and it contributes18.5% of the gross domestic product. It accounts for ten percent (10%) of total exports of the country. Over 60 % of India’s land area is arable making it the second largest country in terms of total arable land. About 50% of the Indian workforce depend upon Agriculture in the country . Being the major contributor for the primary livelihood of mankind, it is a traditional occupation pursued by the majority of population. A stable Agricultural sector assures a nation with food, source of income and source of employment. As the country develops in various cultivation technologies intherecent years, the overall cultivation areas and yields for agricultural produces have increased rapidly, results in greater market value. Although we have a great chance to stand as a major exporter of agricultural produce, in global scenario our share is very low because of reasons like, post-harvest losses in handling and processing various products, unscientific methods that we fallow in our trades and procurement technologies, lack of in depth knowledge in preservation of products and quality evaluation measures. In our country with ever-increase in population there is always a demand for good quality products with greater outlooks. For everyagriculturalproduct there is an urge for the concentrated quality determination, which is faster reliable and accurate. Before exporting food and agricultural produce, guaranteeing product identification and quality product is one of the most significant and challenging tasks of the industry. The classification and grading techniques that we do manually, being used to distinguish between different types of fruits and vegetables fully rely on human efforts. Since these techniques involve greater human intervention, they are subject to human errors. As humans are subjected to tiredness and due to the shortage of man power, automated system needs to be incorporated to minimize the work, reduce the process time anderrors.

In the last two decades, computer vision is getting a lot of attention of research community. Computer vision based solutions are fast, economic, consistent, which has gone into many different industrial sectors. This Computer Vision includes various steps like image acquisition, pre-processing, enhancement and understanding images and, in general, we can say a high-dimensional data from the real world will be converted to numerical or symbolic information. Totally Computer Vision solutions aim at mimicking the human vision system these days.

‘Arecanut', is botanically popular as Areca catechu. It is a tropical plant that we can find in the regions of South East Asia. Areca belongs to Arecaceae family and the palm tree species. Arecanut (Arecanut catechu) is a common masticator nut, consumed by all sections of the population, region, religion, cutting across caste, age and gender in India. In Indian subcontinent the fruit or nut of this Areca tree ispopular as betel nut or supari. In the Western Ghats, Eastern Ghats, East and North Eastern regions of India, Arecanut or betelnut (*Areca catechu*) is grown as an important commercial crop. In India it has its own importance in the social, cultural and religious celebrations and has huge impact on economic life of people. It is also a commodity of commercial, economic and conventional importance. It contributes about 21,000 cores of rupees to Gross National Product. Its economic importance is such that 6.25 million people in India earn their lively hood through areca industries.

ARECANUT- A GLOBAL SCENARIO

Globally it is initially cultivated inIndia, China,Burma, Indonesia, Myanmar and Bangladesh. India topsin the production where China and Bangladesh stood next to India. In 2013-14 the production of Arecanut from an area of over 9.5 lakh hectares in the world was over 13 lakh tones. India ranks first when it comes to both area (47%) and production (54%) of Arecanut. China (20% in production and 5% in area ) ,and Bangladesh (9% in production and 17% in area and), and Indonesia( 14% production and 14% in area) are the major Areca producing countries in the world. Myanmar and Thailand also contributes to Areca production but in a very smaller scale. The global productivity of Arecanut is about 1.25 tons/ha. Indian productivity almost matches with the world productivity of (1.20 tons/ha). In the mid-eighties major expansion happened in Areca growing area and Areca growing stared in many of the countries of the world. China stands first with a productivity of 4164.76 kg/ha, where Myanmar stands with a productivity of 2264.15 kg/ha and the third position taken by Thailand. Although India ranks first in global production, it is ranked 4th in terms of productivity,whereas Malaysia, Bangladesh and Indonesia stands in 5th, 6th and 7th positions following India in the 4th.

# CLASSIFICATION OF ARECANUT

# Arecanut can be classified in to various categories depending on the countries and different regions of it.As it can be used for several purposes, many individual industries differentiate Arecanut into several types according to their need and usage. Maturity, color, glossy appearance, moisture content, weight, size, shape, texture etc. determine the grade and class of Arecanut. Arecanut is basically classified in to its two categories, that is, raw Arecanut and processed Arecanut. It is also termed as with husk and without husk respectively. Raw Arecanut is further classified into four categories that is, *Hasa, Bette, Gorabalu* and *Chali).* The hierarchical structure is given in figure 5*. Hasa* is a premature state of Arecanut and it weighs less but it is costliest among the varieties of Arecanut. *Bette* is a transition state of nut between immature to mature Arecanut. *Gorabalu* is matured Arecanut and it can be easily identified with its properties of color, hardness etc. Whereas *Chali* is t ypical raw Arecanut which is dried along with its husk and later on used after removing its outer shell.Processed Arecanut is classified in to manytypes based on its applications. General classification is in to 5 types that is, *Hasa*, *Bette*, *Rash iIdi*, *Gorabalu* and *Chali*. Further *Hasa*, *Bette* and *Chali* are again classified in to 7 of its categories, whereas *Rashi* and *Idi* are classified in to 3 different categories. *Gorabalu* is classified in to 4 different categories.The different varieties of Arecanut can be seen in the Figure 5.1, 5.2, 5.3, 5.4, 6.1, 6.2, 6.3, 6.4, 6.5.

USES OF ARECANUT

Arecanut is mainly used in medicinal filed for the preparation of various medicines. The habit of chewing Arecanut is almost common in Indian subcontinent.Although, production of Arecanut concentrated in few states of the country, the commercial product has it’s importance throughout the nation. There are mainly two kinds of processed Arecanut, *Chali* (ripe sun dried nuts) and *red boiled type* (tender or mature nuts). Dakshina Kannada and parts of Uttara Kannada districts of Karnataka produce this *Chali* kind of Arecanut. This *Chali* Arecanut is mainly used in making of scented *supari* and it has huge commercial demand in Northern India. There is a huge market demand for varieties of red boiled nuts that are produced in other states of the country. Almost 20 per cent of total areca production in the country is consumed as ripe fruit . As per the various markets throughout the country, there are more than 150 types of Arecanut are produced, these areca types differs in maturity, processing conditions and taste characteristics. The husk of nuts which is an primary sub-product of the Arecanut industry, is mainly used in the preparation of particle, paper and board etc. Where the leather industry uses the *‘Chogaru’*, a by-productobtainedfrom the tender Arecanut for converting hides into skins as it is rich in Tannins .

POSSIBLEAPPROACHESFOR ARECANUT

1. Based on Color

Color is the most significant feature in defining the visual quality of any fruit. Most of the available technology determines color of fruit by comparing the fruit color with the existing predefined reference colors . Arecanut can be classified with the help of color features. In raw Arecanut color plays a prominent role as most of the raw Arecanut classificationwillbedonebasedoncoloronly.Forexample *Hasa*or *Api* will be Greenish in color, whereas *Bette* or *Idi* will be in a transition state from Greenish to Yellowish and *Gorabalu* will be Yellowish in color. So by extracting the mean color values orby extracting color histograms we can classify raw Arecanut with the help of color features.

1. Based on Shape

Shape means graphical data that contains location, size and rotational effects are filtered out. The cost of many products is directly related to theirsize.Arecanut can be graded and classified based on shape. In classification of raw Arecanut although a shape feature can be used to differentiate various types of Arecanut but we cannot rely on it completely, we should also consider some supported features as well. However, for processed Arecanut shape feature can be a very good descriptor in classifying different types of Arecanut as there are lot of differences in the shapes of processedArecanut.

1. Based on Texture

Textures are the important features in computer vision, which will help to partition the images into different regions of interest and then classify the regions . The model for analyzing texture selection extracting the feature critical in classification based on texture. Numerous techniques have been proposed for obtaining the texture. An imagetexture is a set of metrics calculated in image processing designed to quantify the obtained texture of an image. Image texture gives us information about the spatial arrangement of color or intensities in an image or selected region of animage.

Like in the case of Arecanut, Texture features are best suited for classification of fruits. We can consider Texture as a single feature or along with any shape or color features in classification and grading of Arecanut. As the color and shape of varieties of Arecanut are similar we cannot rely only on color and shape features. Since textures of Arecanut are unique, we can consider texture as a major descriptor forArecanut.

1. Hybrid Approach

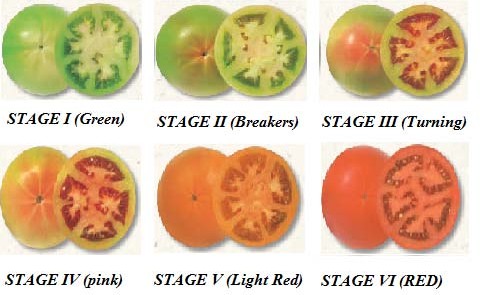
As there is a huge challenge in handling classification and grading of areca nut with color, shape and texture features alone, we can use hybrid approachbycombiningabovefeatures.Some researchers have already developed a few technologies to classify Arecanut by combining color, shape and texture features and they are giving convincing results as well. An example for Hybrid approach is, usingshapefeatures as primary classifier before considering color, which is prominent in Areca, as the grading parameter, results in easy and better classification with less error and high accuracy.

**2.5 Machine vision based automatic sorting of cherry tomatoes**

In this work a two phase sorting based on maturity and quality for cherry tomatoes has been developed. We have collected different tomatoes at different maturity stages with defective and non defectives. First phase grading was based on color while the second phase used color, texture and shape characteristics of tomatoes. Quality based classification has been performed using KNN based SVM classification. Thisclassifier combines KNN and SVM classifier to improve the performance in terms of accuracy and computational time.

Tomatoes will be harvested at the earlier stage for longer market life. Therefore while grading tomatoes, maturity as well the quality need to be considered. In this paper automatic and non destructive grading of cherry tomatoes based on maturity and quality has been proposed. This algorithm comprises two phases of grading. First phase grades tomatoes in terms of maturity and the second phase grades matured tomatoes in terms of quality. Using the concept of change in external color during different maturity stages, a color based maturity estimation algorithm has been proposed. And to classify the tomato in the second phase, texture, color and shape features correlated to external and internal characteristics are extracted from the surface of tomato. To extract the tomato surface alone from the background and leaves a new color based segmentation based on Euclidean distance has also been proposed. Then the extracted features are given to K- Nearest Neighbor based Support Vector Machine classifier to classify the matured fruit into three classes Class I, II and III. This classifier outperforms both SVM and KNN classifier in terms of accuracy and computationtime.

Cherry tomato is one of the major vegetables consumed in Asian fresh-cut markets. On account of market growth, the sorting technique requires non-destructive, effective, less time consuming and higher quality standards. Cherry tomatoquality gradingisusuallyperformedbasedonsizesorting.Thissorting technique is deficient in satisfying consumer demands inAsian countries. Color, size, surface defects, firmness, and sugar/acid contents are important factors in evaluating the overall quality and market value of the vegetables. Above all, the color of the cherry tomatoes is a major decision factor for consumers in choosing what product to purchase. Additionally tomato coloration follows a progression with ripening. Ripening stages of mature tomato categorized as green, breaker, turning, pink, light red and red are shown in figure 1. The fruit will continue to change color even after it is cut from the plant. In addition to this, tomatoes will be picked at green stage for maximum market life. The farmer’s job is not done when the harvest is in. It should be delivered to the marketplace, where the end-consumer could be located down the road or on a different continent. The longer a product takes to get to market or the longer a product sits in a warehouse or on a shelf, the morelikelyitistodeterioratetoapointwhereitcannotbesoldin that market. So for long distance travel, first three stages of tomatoescanbesentorinotherwordslaststagetomatoshould be sold first because it has less market life. So sorting in terms of maturity as well as quality isessential.



**Figure 2.5 USDA ripening stage of tomatoes**

Machine vision has also been used in the judgment of tomato quality classification to ensure good quality seedlings were transplanted. More to the point it has been found that computer vision inspection of food products, was more consistent,efficientandcosteffective.DhanabalandDababrata Samantha used image processing algorithms for defect detection and sorting of tomatoes . A new approach which uses histogram and stem value to detect damaged tomatoes has been proposed. The overall accuracy was about 96.5%. Mehrdad et al analyzed tomato quality as good, fair and bad. Thefeaturesoftomatoareextractedinfuzzyformandgivento three classifiers namely LVQ, MLP and SVM. SVM classifier showed better performance compared to other classifier . Polder et al used Independent component analysis for tomato sorting. It is suitable for in-line calibration and only less datais required for calibration . Wang et al proposed a tomato sorting algorithm using visible and near infrared images. They used the color changes in the ripening of tomatoes to sort the tomatoesintofivematuritystages.Theaccuracyachievedis about96%.Existingmethodscaneffectivelyassessthematurity of tomatoes. But these algorithms had not considered the damaged or diseased tomatoes into account for grading. Though some of them have done for defect classification but the maturity estimation has not been performed. Stage 6 tomatoeshaveshortermarketlifecomparedtoother.Therefore toimprovethemarketlifematuritybasedsortingisessential.

In some methods different spectral information has been used to measure the internal characteristics. But by usingparticular external measure we can estimate internal characteristics also. Because both external and internal attributes of cherry tomato are interrelated. The objective of this work is to grade cherry tomatoes pertaining to maturity as well as quality which must be non destructive, cost effective, efficient and less time consuming.

PROPOSED SYSTEM

In this paper color, texture and shape changes due to different maturity changes has been used for quality grading.Firststepisto classify query tomato into matured and immature. Then the mature tomato is classified into class I, II and III. Class I tomatoes are stage 5 tomatoes, class II are stage 6 tomatoes whereas class III are damaged or rottentomatoes.

1. MATURITY

The first step is to classify the tomato into mature and immature. In this work, we classify stage 1, 2, 3 and 4 tomatoes as immature and stage 5 and 6 as mature tomatoes. For stage 5 tomatoes more than 60% of the surface is red whereas for stage 6 type more than 90% of the surface is red . We used this concept to check the maturity oftomatoes.

* + 1. First the tomato object is extracted from the background and total area isdetermined.
    2. The area of surface with red pixels isestimated.
    3. The ratio of red pixel area to the total area is obtained.
    4. If the ratio is greater than 0.6, the tomato is classified as mature, otherwise immature.

1. SEGMENTATION

Once the tomato is classified as mature, it is necessary to classify into class I,II and III. For this the outer surface of the tomato from the leaf needs to be extracted. Euclidean distance based color image segmentation has been proposed to extract the surface. In this algorithm for every pixel in the image, the Euclidean distance is computed with the agent pixel. The agent pixels are the R, G and B values of red, blue and white pixels.The agent pixels for red, green and white are ,, . The flow of this color segmentation algorithm is shown in fig 3 where dg, dr, dw are the Euclidean distance computed with respect to red, green and whiterespectively.

1. FEATURE EXTRACTION

Next step is to represent the tomato surface in terms of features. The surface of tomato is uniform in color and texture. It has smooth and shiny external appearance. Thetomato characteristilike color, texture and shape will change with respect to different maturity conditions. So the texture, color and shape features correlated to this external property of tomato are extracted.

1. *Texture Features:*Texture describes the characteristic of the intensity surface of the image. It contains information about the structural arrangement of tomato surface. The texture features like Entropy, Angular second Moment (ASM), Inverse Difference Moment (IDM) and variance are extracted from the gray level co-occurrence matrix and are shown in table 1. It depicts how often different combinations of gray levels co-occur in an image.
2. *Color features:*Color features like color ASM and color Inverse difference moment are extracted from the color level co occurrence matrix to measure the external color characteristics of cherry tomatoes. The GLCM is estimated for red band alone instead of gray values, and it is named as color level co occurrence matrix. Red band contains more information with respect to tomato compared to blue and green bands. So the CLCM is calculated for RED band alone. And for Class I and Class II the color of tomato is uniformly distributed over the surface, whereas the damages or defectsin class III fruit will make tomato surface as heterogeneous. So the color ASM feature value is higher for class I and class II tomatoes whereas class III tomatoes has high color IDM feature value.
3. Shape Features:Shape features like area, perimeter, circularity and aspect ratio are extracted to describe the shape of tomato in terms of features. When the tomato is defected or rotten it becomes over soft and its shape is distorted. So shape features also play an important role in sorting of tomatoes.Aspect ratio is function of the largest diameter to the smallest diameter. The aspect of good quality tomatoes will be approximately equalto 1.Circularity of tomato is a function of perimeter and area of surface of tomato.
4. *KNN-SVMCLASSIFIER*

The features extracted above are given to KNN-SVM classifier as proposed in . The computation time for KNN classifierisgoodbutitisnotasefficientasSVM.ButtheSVMis classifier which is not suitable for grading. So these two classifiers have been combined to improve the computation time as well as accuracy. The algorithm for KNN-SVM algorithm is given as,

1. Compute the texture, color and shape features for thedatabase of tomato images.

2. For the query fruit estimate the same features.

3. Compute distance of the query tomato features to alltraining features and pick the nearest Kneighbors.

4. If all the K neighbors have the same labels, then thequery tomato is labeled to the corresponding class andexit, otherwise compute the pair wise distancesbetween K neighbors.

5. Convert the distance matrix to a kernel matrix usingthe kernel trick formula and is given as,

K(x,y) = ½(d(x,0)+d(y,0)-d(x,y))

6.Apply multiclass SVM for the kernel matrix and usethe resulting class to label the query.

**CHAPTER 3**

**PROBLEM STATEMENT**

**PROBLEM STATEMENT**

“Implementing the software to categorize the areca nut yield into 3 categories that will help the farmer to speed up the process of separation of the yield.”

**Input:** Images of the areca nut yield from the cultivated crop.

**Output:** Classified Areca nut yield.

**CHAPTER 4**

**SYSTEM REQUIREMENTS**

**SYSTEM REQUIREMENTS**

* 1. **Functional & Non Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Requirements** | **Non-Functional Requirements** | |  |
|  |  | |  |
| Extract the areca nut from the image. | **Usability:**The GUI is required to be easily usable. | |  |
| Identify the category of the peeled areca nut. | **Reliability:**The system is expected to make correct categorisation of areca. | |  |
| To integrate the above in user interface | **Performance:**The model is required to classify correctly in minimum response time and minimum percentage error. | |  |
| **Tab 4.1 Functional & Non Functional Requirements** | |  |

**4.2 Software Requirements**

**Google Colaboratory**

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just [a few lines of code](https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/quickstart/beginner.ipynb). Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including [GPUs and TPUs](https://colab.research.google.com/notebooks/intro.ipynb#using-accelerated-hardware), regardless of the power of your machine. All you need is a browser.

Colab is used extensively in the machine learning community with applications including:

* Getting started with TensorFlow
* Developing and training neural networks
* Experimenting with TPUs
* Disseminating AI research
* Creating tutorials

**Google Drive**

Google Drive is a [file storage](https://en.wikipedia.org/wiki/File_hosting_service) and [synchronization service](https://en.wikipedia.org/wiki/File_synchronization) developed by [Google](https://en.wikipedia.org/wiki/Google). Launched on April 24, 2012, Google Drive allows users to store files on their servers, synchronize files across devices, and [share files](https://en.wikipedia.org/wiki/File_sharing). In addition to a [website](https://en.wikipedia.org/wiki/Web_application), Google Drive offers apps with offline capabilities for [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and [macOS](https://en.wikipedia.org/wiki/MacOS) computers, and [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) and [iOS](https://en.wikipedia.org/wiki/IOS) smartphones and tablets. Google Drive encompasses [Google Docs](https://en.wikipedia.org/wiki/Google_Docs), [Google Sheets](https://en.wikipedia.org/wiki/Google_Sheets), and [Google Slides](https://en.wikipedia.org/wiki/Google_Slides), which are a part of an [office suite](https://en.wikipedia.org/wiki/Office_suite) that permits collaborative editing of documents, spreadsheets, presentations, drawings, forms, and more. Files created and edited through the office suite are saved in Google Drive. In Google drive we store all the training and testing images. Google colab takes input from these folders and performs the training and does the validation on the test Images. Google drive also saves our model.

**TENSOR FLOW**

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 on November 9, 2015. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors.

**Python IDLE**

Python is a [multi-paradigm programming language](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language). [Object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) and [structured programming](https://en.wikipedia.org/wiki/Structured_programming) are fully supported, and many of its features support [functional programming](https://en.wikipedia.org/wiki/Functional_programming) and [aspect-oriented programming](https://en.wikipedia.org/wiki/Aspect-oriented_programming) (including by [metaprogramming](https://en.wikipedia.org/wiki/Metaprogramming)and [metaobjects](https://en.wikipedia.org/wiki/Metaobject) (magic methods)). Many other paradigms are supported via extensions, including [design by contract](https://en.wikipedia.org/wiki/Design_by_contract) and [logic programming](https://en.wikipedia.org/wiki/Logic_programming).

Python uses [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing) and a combination of [reference counting](https://en.wikipedia.org/wiki/Reference_counting) and a cycle-detecting garbage collector for [memory management](https://en.wikipedia.org/wiki/Memory_management). It also features dynamic [name resolution](https://en.wikipedia.org/wiki/Name_resolution_(programming_languages)) ([late binding](https://en.wikipedia.org/wiki/Late_binding)), which binds method and variable names during program execution.

Python's design offers some support for [functional programming](https://en.wikipedia.org/wiki/Functional_programming) in the [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)) tradition. It has filter, map, and reduce functions; [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension), [dictionaries](https://en.wikipedia.org/wiki/Associative_array), sets, and [generator](https://en.wikipedia.org/wiki/Generator_(computer_programming)) expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)) and [Standard ML](https://en.wikipedia.org/wiki/Standard_ML).

The language's core philosophy is summarized in the document *The*[*Zen of Python*](https://en.wikipedia.org/wiki/Zen_of_Python) (*PEP 20*), which includes [aphorisms](https://en.wikipedia.org/wiki/Aphorism) such as:

* Beautiful is better than ugly.
* Explicit is better than implicit.
* Simple is better than complex.
* Complex is better than complicated.
* Readability counts.

In order to run the python script locally we need python idle installed in the system. When the API is called the python script is executed and the output is fed to the front end. The python IDLE is also needed for the image processing of the arecanut.

**React JS**

React (also known as React.js or ReactJS) is an [open-source](https://en.wikipedia.org/wiki/Open-source) [JavaScript library](https://en.wikipedia.org/wiki/JavaScript_library) for building [user interfaces](https://en.wikipedia.org/wiki/User_interfaces). It is maintained by [Facebook](https://en.wikipedia.org/wiki/Facebook) and a community of individual developers and companies.

React can be used as a base in the development of [single-page](https://en.wikipedia.org/wiki/Single-page_application) or mobile applications. However, React is only concerned with rendering data to the [DOM](https://en.wikipedia.org/wiki/Document_Object_Model), and so creating React applications usually requires the use of additional libraries for [state management](https://en.wikipedia.org/wiki/State_management) and routing. [Redux](https://en.wikipedia.org/wiki/Redux_(JavaScript_library)) and React Router are respective examples of such libraries.

**Bootstrap**

Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to [web apps](https://en.wikipedia.org/wiki/Web_Apps)). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all [HTML elements](https://en.wikipedia.org/wiki/HTML_element). The result is a uniform appearance for prose, tables and form elements across [web browsers](https://en.wikipedia.org/wiki/Web_browser). In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of [jQuery](https://en.wikipedia.org/wiki/JQuery) plugins. They provide additional user interface elements such as [dialog boxes](https://en.wikipedia.org/wiki/Dialog_box), [tooltips](https://en.wikipedia.org/wiki/Tooltip), and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. While the latter always fills the width of the web page, the former uses one of the four predefined fixed widths, depending on the size of the screen showing the page:

* Smaller than 576 pixels
* 576–768 pixels
* 768–992 pixels
* 992–1200 pixels
* Larger than 1200 pixels

Once a container is in place, other Bootstrap layout components implement a CSS Flexbox layout through defining rows and columns.

A precompiled version of Bootstrap is available in the form of one CSS file and three JavaScript files that can be readily added to any project. The raw form of Bootstrap, however, enables developers to implement further customization and size optimizations. This raw form is modular, meaning that the developer can remove unneeded components, apply a theme and modify the uncompiled [Sass](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)) files.

**Node JS**

Node.js is an [open-source](https://en.wikipedia.org/wiki/Open-source_software), [cross-platform](https://en.wikipedia.org/wiki/Cross-platform), [JavaScript](https://en.wikipedia.org/wiki/JavaScript) runtime environment that executes JavaScript code outside a [web browser](https://en.wikipedia.org/wiki/Web_browser). Node.js lets developers use JavaScript to write command line tools and for [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting)—running scripts server-side to produce [dynamic web page](https://en.wikipedia.org/wiki/Dynamic_web_page) content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying [web-application](https://en.wikipedia.org/wiki/Web_application) development around a single programming language, rather than different languages for server- and client-side scripts. Node.js allows the creation of [Web servers](https://en.wikipedia.org/wiki/Web_server) and networking tools using [JavaScript](https://en.wikipedia.org/wiki/JavaScript) and a collection of "modules" that handle various core functionalities. Modules are provided for [file system](https://en.wikipedia.org/wiki/File_system) I/O, networking ([DNS](https://en.wikipedia.org/wiki/Domain_Name_System), [HTTP](https://en.wikipedia.org/wiki/HTTP), [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol), [TLS/SSL](https://en.wikipedia.org/wiki/Transport_Layer_Security), or [UDP](https://en.wikipedia.org/wiki/User_Datagram_Protocol)), [binary](https://en.wikipedia.org/wiki/Binary_file) data (buffers), [cryptography](https://en.wikipedia.org/wiki/Cryptography) functions, [data streams](https://en.wikipedia.org/wiki/Stream_(computing)), and other core functions. Node.js's modules use an API designed to reduce the complexity of writing server applications.

JavaScript is the only language that Node.js supports natively, but many [compile-to-JS](https://en.wikipedia.org/wiki/Source-to-source_compiler) languages are available. As a result, Node.js applications can be written in [CoffeeScript](https://en.wikipedia.org/wiki/CoffeeScript), [Dart](https://en.wikipedia.org/wiki/Dart_(programming_language)), [TypeScript](https://en.wikipedia.org/wiki/TypeScript), [ClojureScript](https://en.wikipedia.org/wiki/ClojureScript) and others.

Node.js is primarily used to build network programs such as Web servers. The most significant difference between Node.js and [PHP](https://en.wikipedia.org/wiki/PHP) is that most functions in PHP [block](https://en.wikipedia.org/wiki/Asynchronous_I/O) until completion (commands only execute after previous commands finish), while Node.js functions are [non-blocking](https://en.wikipedia.org/wiki/Asynchronous_I/O) (commands execute [concurrently](https://en.wikipedia.org/wiki/Concurrent_computing) or even in [parallel](https://en.wikipedia.org/wiki/Parallel_computing), and use [callbacks](https://en.wikipedia.org/wiki/Callback_(computer_programming)) to signal completion or failure).

**4.3 Hardware Requirements**

# Quad core Intel Core i7

# 16 GB RAM

# GTX 980 graphic cards

**CHAPTER 5**

**DESIGN**

**DESIGN**

**5.1 Architecture**

**Fig 5.1 Architecture**

The architecture has 4 basic parts:

**Input:**

The peeled Areca nut should be given to the system. Lighting conditions are standardized, so that their classification is not compromised. There shouldn’t be ambiguity in the background of the Areca nut. If there are two backgrounds in the same picture, the Areca nut picture will not be perfectly extracted.

**Preprocessing:**

The areca picture is being fed to the software, which has noise in it. Since the dataset is limited, enhancing of the data set needs to be done. For this we perform augmentation. Augmentation is the process of creating images from the existing images by rotating with certain angle as necessary. This also enhances the training because the areca nut can be fed at different angles. Images are standardized using rescale so that there is no ambiguity in training.

**Convolutional Neural Networks:**

The pre processed images are stored in Google Drive in training and testing folders. Since we have used Google Colab for the training, it takes input from the Google drive to train and construct the model. The model is saved in the Google drive and it is downloaded to the local system.

**Categorizing the Areca nut:**

Front End is developed using React JS and Bootstrap is used for good look and feel for the system. Node JS server is run at the back end and when the request is accepted, the python program runs in the backend and gives the category of the Areca nut.

**5.2 Modules**

**5.2.1 Preprocessing**

Initially, the areca nut is raw and it needs to be peeled to yield good market value. Before peeling the areca nut needs to be dried for almost 40 days. It is fed to the peeling machine and the peeled Areca nut is extracted. Here the categorization of the Areca nut takes place, which is done by human labor. But here we are replacing human labor with the system that categorizes Areca nut based on the features. Following is the process through which the Areca nut goes through. First, the peeled Areca nut is fed to the system. Following changes takes places.

1. Resizing: The pictures are standardized to particular scale to 300\*300 pixels. All these pictures are of RGB color model.
2. Augmentation: The images are augmented to enhance the limited dataset. The augmentation transforms limited images into many images by flipping horizontally and vertically, rotating at a certain angle. Augmentation also resolves the problem of inclination when giving the test images, because the Areca can be aligned in any direction.
3. Grab-Cut: This function identifies the background and foreground pixels and helps to extract only the Areca from the background. This eliminates the issue of having multiple backgrounds during training and test environments.

**5.2.2 Convolutional Neural Networks**

The pre processed Areca nut pictures are stored in the Google Drive. Training folder is specified and the images are converted to array of size 32\*32\*3, where 32\*32 is the size of the image and 3 is the representation of RGB array. The labels are converted to categorical representation i.e class 1 is denoted as and class 3 is denoted as . This representation is needed so that it is fed to the Sequential model. The model is trained and stored in the Google Drive, which is downloaded to the local machine.

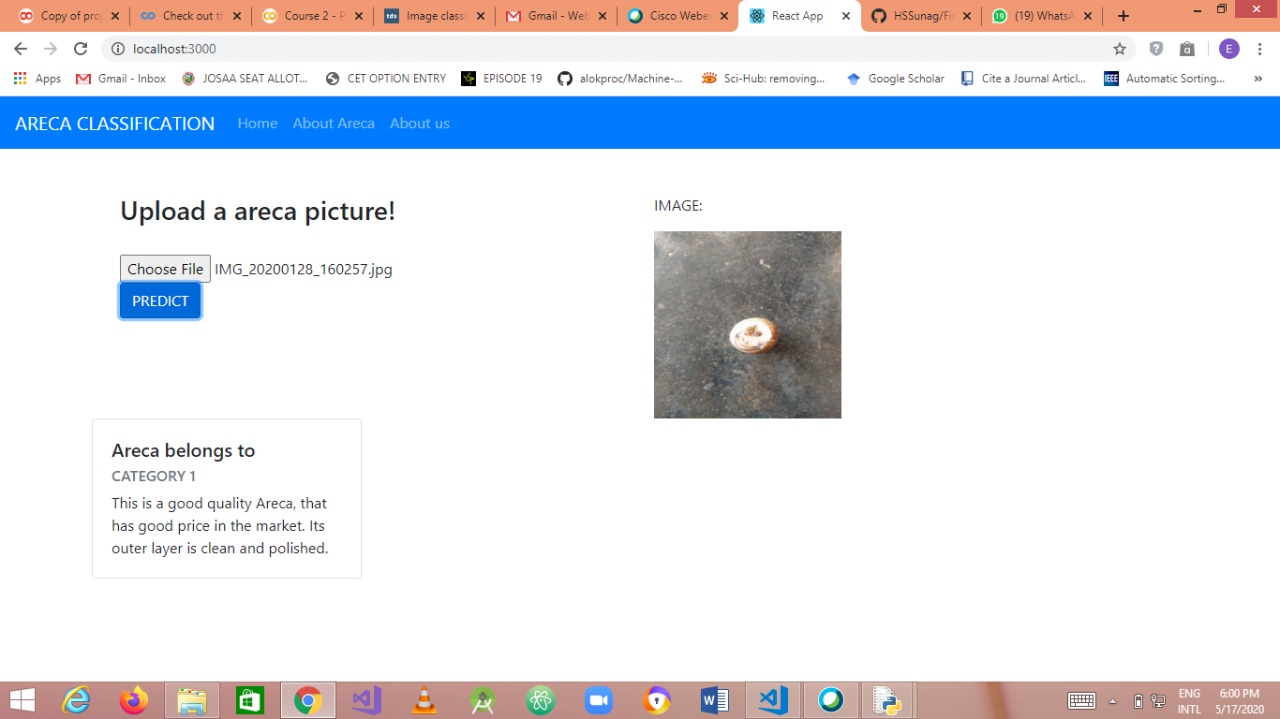
**Fig 5.2.2 Process of categorizing the Areca nut**

**5.2.3 Back End**

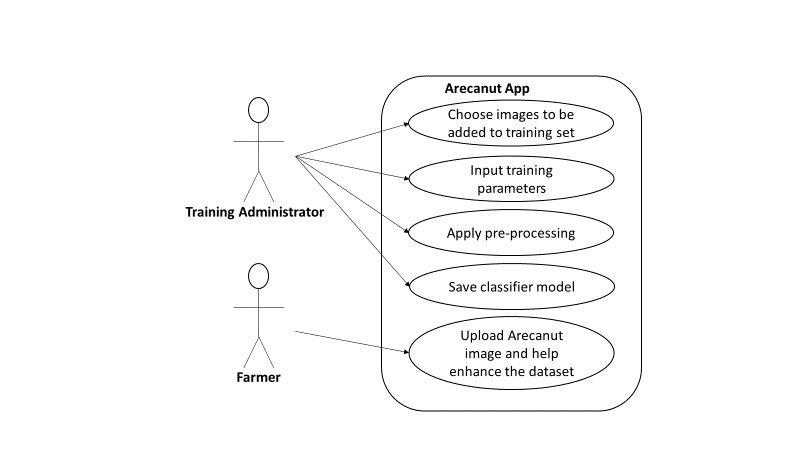
Back End is designed using Node JS framework. Here we specify an API or a endpoint which takes the image as input and gives category of the Areca nut as the output. When the request is made to the endpoint it executes the local python script and outputs the category of the Areca nut, which is fed to the Front End for displaying the same.

**5.2.4 Front End**

Front End is designed using React JS framework. To have good look and feel of the website, bootstrap framework is being used. Here the user inputs the image to be classified by browsing the file system. When he uploads the Areca picture, it is sent to the server and category is given back using response JSON. Category card is displayed in the front end, along with its information.



` **Fig 5.2.4 Snapshot of the Front End**

**5.3 Use-Case Diagram**

**Fig 5.3 Use Case Diagram**

The use-case diagram of the project consists of two actors on the scene: The Training Administrator and The Farmer.

**Training Administrator**

* The Training Administrator performs the operations that build the classifier model.
* He/she chooses the images that need to be added to the training set and the test set.
* He/she selects the proportion of dataset that need to be provided as training set and the test set.
* He/she inputs the different parameters on which the images have to be classified. i.e., provide the training parameters.
* He/she applies the pre-processing on the dataset of images to enhance the dataset.
* Finally, he/she saves the best classifier model generated into the server.

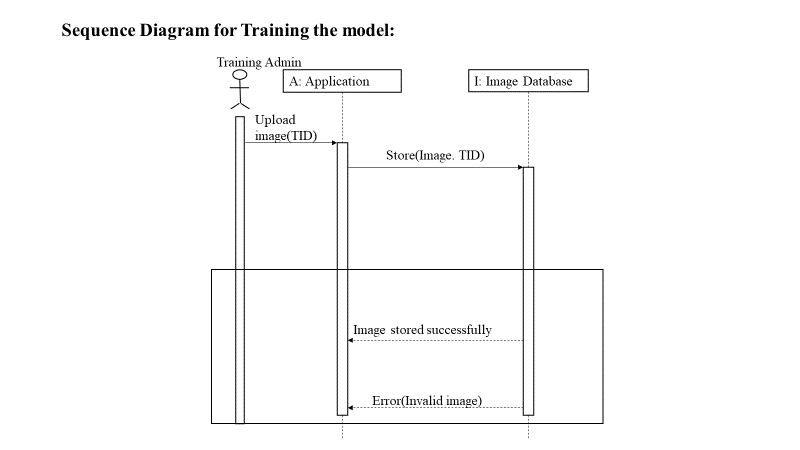
**Farmer/User**

* The farmer uploads the image of the Arecanut and gets the resulting category.
* He/she helps enhance the dataset by uploading new images onto the server.

**5.4 Sequence Diagram**

The sequence diagram consists of two perspectives one for each of the actors defined in use-case diagram, namely, one for training the model and one for farmer/user.

**Sequence diagram for training the model:**

****

**Fig 5.4 Sequence Diagram for training the model**

* The above figure shows the interactions between the Training administrator and the application and between the application and the database.
* First, the training administrator uploads the images one-by-one into the application.
* The application then checks if the image of valid size and checks various parameters to authenticate that the image is of Arecanut. Then, it stores the image into the Image database.
* If the Image is of invalid format or if the storing of image fails, then the Image database returns an error to the application.

**Sequence diagram for Farmer:**

****

* The sequence diagram for the Farmer/user consists of the interactions between the Farmer and the application and between the application and the Classifier.
* The Farmer uploads the image of the Arecanut to the application for test.
* The application interacts with the classifier by sending the image for classification.
* The classifier identifies and classifies the Arecanut image into a category and returns the category to the application which will be shown to the user.
* If the uploaded image is of invalid format, then the classifier returns an error and the error will be shown to the user.

**5.5 Algorithm Design**

**5.5.1 Resizing function**

The Areca images are resized to 300\*300 pixels for standardizing the picture. The pseudo code goes as follows:

new\_width = 300

new\_height = 300

for f in os.listdir('.'):

if f.endswith('.jpg'):

i=Image.open(f)

fn, fext = os.path.splitext(f)

img = i.resize((new\_width, new\_height), Image.ANTIALIAS)

img.save('F:\Agumented1\{}\_300{}'.format(fn,fext))

**Fig 5.5.1 Rescaling of the Areca Image**

**5.5.2 Grab-Cut function**

Grab-Cut identifies the background images and foreground images and masks the background image to extract only the Areca nut.This is done by following function:

for f in os.listdir('.'):

if f.endswith('.jpg'):

img = cv2.imread(f)

mask = np.zeros(img.shape,np.uint8)

fn, fext = os.path.splitext(f)

bgdModel = np.zeros((1,65),np.float64)

fgdModel = np.zeros((1,65),np.float64)

rect = (50,50,180,170)

cv2.grabCut(img,mask,rect,bgdModel,fgdModel,5,cv2.GC\_INIT\_WITH\_RECT)

mask2 = np.where((mask==2)|(mask==0),0,1).astype('uint8')

img = img\*mask2

im = Image.fromarray(img)

im.save('F:\Agumented1\Extract1\{}\_C300{}'.format(fn,fext))

**Fig 5.5.2 Image of Areca after Grab-Cut**

**5.5.3 Augmentation function**

Enhancing of the dataset needs to be done which is done by flipping and rotating at certain degrees. Sample code goes as follows:

* **Horizontal flip**

fn, fext = os.path.splitext(f)

hoz\_flip = image.transpose(Image.FLIP\_LEFT\_RIGHT)

hoz\_flip.save(‘F:\arecaimg\{}\_fr{}'.format(fn,fext))

* **Vertical flip**

fn, fext = os.path.splitext(f)

hoz\_flip = image.transpose(Image. FLIP\_TOP\_BOTTOM)

hoz\_flip.save(‘F:\arecaimg\{}\_fr{}'.format(fn,fext))

* **Degree flip - Flip the image to a different angle**

flip\_90=image.rotate(90)

flip\_90.save(‘F:\arecaimg\{}\_r90{}'.format(fn,fext))

flip\_180=image.rotate(180)

flip\_180.save('F:\arecaimg\{}\_r180{}'.format(fn,fext))

flip\_270=image.rotate(270)

flip\_270.save('F:\arecaimg\{}\_r270{}'.format(fn,fext))

|  |  |
| --- | --- |
| WhatsApp Image 2020-06-05 at 7.41.08 PM.jpeg | WhatsApp Image 2020-06-05 at 7.41.09 PM.jpeg |
| WhatsApp Image 2020-06-05 at 7.41.10 PM.jpeg | WhatsApp Image 2020-06-05 at 7.41.10 PM (1).jpeg |



**Fig 5.5.3 Augmented Areca Images**

**5.5.4 Training Algorithm**

1. To build the model we need to create the architecture using Sequential() .
   1. From keras.models import Sequential from keras.layers import Dense, Flatten,
   2. Conv2D, MaxPooling2D model = Sequential()
2. Next we add the first layer, a convolution layer to extract features from the input image, and create 32 5 x 5 ReLu convoluted features also known as feature maps. Since this is the first layer we must input the dimension shape which is a 32 x 32 pixel image with depth = 3 (RGB).
   1. model.add(Conv2D(32, (5, 5), activation='relu', input\_shape=(32,32,3)))
3. The next layer will be a pooling layer with a 2 x 2 pixel filter to get the max element from the feature maps. This reduces the dimension of the feature maps by half and is also known as sub sampling.
   1. model.add(MaxPooling2D(pool\_size=(2, 2)))
4. Create one more convolution layer and pooling layer like before, but without the input\_shape.
   1. model.add(Conv2D(64, (5, 5), activation='relu’))
   2. model.add(MaxPooling2D(pool\_size=(2, 2)))
5. Add a flattening layer, to reduce the image to a linear array also known as a one 1-Dimension vector to feed into and connect with the neural network.

model.add(Flatten())

1. Now create a neural network where the first layer has 1000 neurons and the activation function ReLu.
   1. model.add(Dense(1000, activation='relu'))
2. Create the last layer of this neural network with 3 neurons (one for each label) using the softmax function.
   1. model.add(Dense(3, activation='softmax'))

**5.5.5 Classification Algorithm**

1. Read the command line argument.

img = cv2.imread(sys.argv)

1. Resize the image to 32\*32.

img = cv2.resize(img,(32,32))

1. Populate the mask with zeroes

mask = np.zeros(img.shape,np.uint8)

1. Define the background model

bgdModel = np.zeros((1,65),np.float64)

1. Define the foreground model

fgdModel = np.zeros((1,65),np.float64)

1. Define the targe rectangle

rect = (5,5,19,18)

1. Use Grabcut to extract Areca nut

cv2.grabCut(img,mask,rect,bgdModel,fgdModel,5,cv2.GC\_INIT\_WITH\_RECT)

1. Multiply mask with the image to get Areca

mask2 = np.where((mask==2)|(mask==0),0,1).astype('uint8')

img = img\*mask2

1. Convert Image to Array

img\_array = keras.preprocessing.image.img\_to\_array(img)

1. Normalize the image array

img\_array = img\_array/255.0

1. Expand the dimensions for classification

img = np.expand\_dims(img\_array, axis = 0)

1. Load the model

model=load\_model('fine.h5')

1. Predict the image

a = model.predict\_classes(img)

1. Print the category

print(a)

**CHAPTER 6**

**IMPLEMENTATION**

**IMPLEMENTATION**

**6.1 Resizing**

Resizing helps to have standardized form of the images so that all preprocessing is done fairly and helps in the objective without any ambiguity.

from PIL import Image

import os

new\_width = 300

new\_height = 300

for f in os.listdir('.'):

if f.endswith('.jpg'):

i=Image.open(f)

fn, fext = os.path.splitext(f)

img = i.resize((new\_width, new\_height), Image.ANTIALIAS)

img.save('F:\8th sem\Project\Dataset\Category1\Agumented1\{}\_300{}'.format(fn,fext))

**6.2 Augmentation**

Since there is a scarcity of the dataset, augmentation is necessary to enhance the dataset by flipping horizontally and vertically, rotating with respect to a particular angle.

# create flipped versions of an image

from PIL import Image

import os

from matplotlib import pyplot

# load image

for f in os.listdir('.'):

if f.endswith('.jpg'):

image=Image.open(f)

# horizontal flip

fn, fext = os.path.splitext(f)

hoz\_flip = image.transpose(Image.FLIP\_LEFT\_RIGHT)

hoz\_flip.save('F:\8th sem\Project\Dataset\Category 1\Agumented1\Extract1\Final1\{}\_fr{}'.format(fn,fext))

# vertical flip

ver\_flip = image.transpose(Image.FLIP\_TOP\_BOTTOM)

ver\_flip.save('F:\8th sem\Project\Dataset\Category1\Agumented1\Extract1\Final1\{}\_ftb{}'.format(fn,fext))

#degree flippe

flip\_90=image.rotate(90)

flip\_90.save('F:\8th sem\Project\Dataset\Category1\Agumented1\Extract1\Final1\{}\_r90{}'.format(fn,fext))

flip\_180=image.rotate(180)

flip\_180.save('F:\8th sem\Project\Dataset\Category1\Agumented1\Extract1\Final1\{}\_r180{}'.format(fn,fext))

flip\_270=image.rotate(270)

flip\_270.save('F:\8th sem\Project\Dataset\Category1\Agumented1\Extract1\Final1\{}\_r270{}'.format(fn,fext))

**6.3 Extraction of Areca**

When we give areca for classification, there may be different backgrounds and hence it is necessary to give only the areca which doesn’t contain the background with which it is captured.

from PIL import Image

import os

import numpy as np

import cv2

from matplotlib import pyplot as plt

for f in os.listdir('.'):

if f.endswith('.jpg'):

img = cv2.imread(f)

mask = np.zeros(img.shape,np.uint8)

fn, fext = os.path.splitext(f)

bgdModel = np.zeros((1,65),np.float64)

fgdModel = np.zeros((1,65),np.float64)

rect = (50,50,180,170)

cv2.grabCut(img,mask,rect,bgdModel,fgdModel,5,cv2.GC\_INIT\_WITH\_RECT)

mask2 = np.where((mask==2)|(mask==0),0,1).astype('uint8')

img = img\*mask2

im = Image.fromarray(img)

im.save('F:\8th sem\Project\Dataset\Category 1\Agumented1\Extract1\{}\_C300{}'.format(fn,fext))

**6.4 Training the data**

The data is trained using convolutional neural networks.

import matplotlib.pyplot as plt

import cv2

from google.colab import drive

# Accessing My Google Drive

drive.mount('/content/drive',force\_remount=True)

import numpy

import keras

import os

import tensorflow as tf

def images\_to\_array(dataset\_dir, image\_size):

dataset\_array =

dataset\_labels =

class\_counter = 0

classes\_names = os.listdir(dataset\_dir)

for current\_class\_name in classes\_names:

class\_dir = os.path.join(dataset\_dir, current\_class\_name)

images\_in\_class = os.listdir(class\_dir)

print("Class index", class\_counter, ", ", current\_class\_name, ":" , len(images\_in\_class))

for image\_file in images\_in\_class:

if image\_file.endswith(".jpg"):

image\_file\_dir = os.path.join(class\_dir, image\_file)

img = keras.preprocessing.image.load\_img(image\_file\_dir, target\_size=(image\_size, image\_size))

img\_array = keras.preprocessing.image.img\_to\_array(img)

img\_array = img\_array/255.0

dataset\_array.append(img\_array)

dataset\_labels.append(class\_counter)

class\_counter = class\_counter + 1

dataset\_array = numpy.array(dataset\_array)

dataset\_labels = numpy.array(dataset\_labels)

return dataset\_array, dataset\_labels

train\_dir = "drive/My Drive/Final Datasets"

image\_size = 32

train\_dataset\_array, train\_dataset\_array\_labels = images\_to\_array(dataset\_dir=train\_dir, image\_size=image\_size)

print("Training Data Array Shape :", train\_dataset\_array.shape)

numpy.save("train\_dataset\_array.npy", train\_dataset\_array)

numpy.save("train\_dataset\_array\_labels.npy", train\_dataset\_array\_labels)

'''test\_dir = "drive/My Drive/DataSet/Testing/test"

test\_dataset\_array, test\_dataset\_array\_labels = images\_to\_array(dataset\_dir=test\_dir, image\_size=image\_size)

print("Test Data Array Shape :", test\_dataset\_array.shape)

numpy.save("test\_dataset\_array.npy", test\_dataset\_array)

numpy.save("test\_dataset\_array\_labels.npy", test\_dataset\_array\_labels)'''

from sklearn.utils import shuffle

train\_dataset\_array, train\_dataset\_array\_labels = shuffle(train\_dataset\_array, train\_dataset\_array\_labels,random\_state=0)

from keras.utils import to\_categorical

y\_train\_one\_hot = to\_categorical(train\_dataset\_array\_labels)

print(len(train\_dataset\_array))

print(len(train\_dataset\_array\_labels))

from keras.models import Sequential

from keras.layers import Dense, Flatten, Conv2D, MaxPooling2D, Dropout

model = Sequential()

model.add(Conv2D(32, (5, 5), activation='relu', input\_shape=(32 ,32 ,3)))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Conv2D(64, (5, 5), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Flatten())

model.add(Dense(1000, activation='relu'))

model.add(Dense(3, activation='softmax'))

model.compile(loss='categorical\_crossentropy',

optimizer='adam',

metrics=)

hist = model.fit(train\_dataset\_array, y\_train\_one\_hot,

batch\_size=10, epochs=10, validation\_split=0.10)

plt.plot(hist.history)

plt.plot(hist.history)

plt.title('Model accuracy')

plt.ylabel('Accuracy')

plt.xlabel('Epoch')

plt.legend(, loc='upper left')

plt.show()

print(hist.history)

model.save('areca4.h5')

**6.5 Front End Code(React JS)**

import axios from 'axios';

import React,{Component} from 'react';

import { useState } from 'react';

import ReactDOM from 'react-dom';

import { Button } from 'react-bootstrap';

import { Navbar } from 'react-bootstrap';

import { Nav } from 'react-bootstrap';

import { Form } from 'react-bootstrap';

import { FormControl } from 'react-bootstrap';

import 'bootstrap/dist/css/bootstrap.min.css';

import { Container } from 'react-bootstrap';

import { Image } from 'react-bootstrap'

import { Row } from 'react-bootstrap'

import { Modal } from 'react-bootstrap'

import { Col } from 'react-bootstrap'

import { Card } from 'react-bootstrap'

function Areca() {

const = useState(false);

const handleClose = () => setShow(false);

const handleShow = () => setShow(true);

return(

<div>

<Nav.Link href="#features" onClick={handleShow}>About Areca</Nav.Link>

<Modal show={show} onHide={handleClose}>

<Modal.Header closeButton>

<Modal.Title>About ARECA</Modal.Title>

</Modal.Header>

<Modal.Body>Areca Nut is the seed of the areca palm, which grows in much of the tropical specific,South East and South Asia.

The areca nut is not a true nut, but rather the seed of a fruit categorized as a berry. It is commercially available in dried, cured, and fresh forms. When the husk of the fresh fruit is green, the nut inside is soft enough to be cut with a typical knife. In the ripe fruit, the husk becomes yellow or orange, and as it dries, the fruit inside hardens to a wood-like consistency. At that stage, the areca nut can only be sliced using a special scissors-like cutter.

In 2017, world production of areca nut was 1.3 million tonnes, with India providing 54% of the total. As other leading producers, Myanmar, Indonesia, Bangladesh and Taiwan combined contributed 38% of the world total (table).

</Modal.Body>

<Modal.Footer>

<Button variant="primary" onClick={handleClose}>DONE</Button>

</Modal.Footer>

</Modal>

</div>

);

}

function AboutUs() {

const = useState(false);

const handleClose = () => setShow(false);

const handleShow = () => setShow(true);

return(

<div>

<Nav.Link href="#pricing" onClick={handleShow}>About us</Nav.Link>

<Modal show={show} onHide={handleClose}>

<Modal.Header closeButton>

<Modal.Title>About ARECA</Modal.Title>

</Modal.Header>

<Modal.Body><h6>Developed by</h6><br />Elson DSA<br />HS Sunag Hemmanna<br />Ganesh Naidu N<br />Sourav Pai K

</Modal.Body>

<Modal.Footer>

<Button variant="primary" onClick={handleClose}>DONE</Button>

</Modal.Footer>

</Modal>

</div>

);

}

class App extends Component {

constructor(){

super();

this.state = {

// Initially, no file is selected

selectedFile: null,

urlFile: null,

category: null

};

this.onFileChange = this.onFileChange.bind(this);

this.getCategory = this.getCategory.bind(this);

}

getCategory() {

let file = this.state.selectedFile;

let formdata = new FormData();

formdata.append('image',file);

formdata.append('name','somefile');

this.setState({ category: "Loading......."})

axios.post("http://localhost:3001/api/category",formdata)

.then((response) => {

this.setState({category: response.data.cat})

})

.catch((error)=>{

console.log(error);

})

}

// On file select (from the pop up)

onFileChange(event) {

// Update the state

this.setState({ selectedFile: event.target.files, urlFile: URL.createObjectURL(event.target.files) });

};

// File content to be displayed after

// file upload is complete

fileData = () => {

if (this.state.selectedFile) {

return (

<div>

<p>IMAGE:</p>

<Image src={this.state.urlFile} height="200" width="200"></Image>

</div>

);

} else {

return (

<div>

<br />

<h4>Choose before Pressing the Upload button</h4>

</div>

);

}

};

categoryData = () => {

if (this.state.category==0) {

return (

<div>

<Row>

<Card style={{ width: '18rem' }}>

<Card.Body>

<Card.Title>Areca belongs to</Card.Title>

<Card.Subtitle className="mb-2 text-muted">CATEGORY 1</Card.Subtitle>

<Card.Text>

This is a good quality Areca, that has good price in the market. Its outer layer is clean and polished.

</Card.Text>

</Card.Body>

</Card>

</Row>

</div>

);

}

if (this.state.category==1) {

return (

<div>

<Row>

<Card style={{ width: '18rem' }}>

<Card.Body>

<Card.Title>Areca belongs to</Card.Title>

<Card.Subtitle className="mb-2 text-muted">CATEGORY 2</Card.Subtitle>

<Card.Text>

This is a medium quality Areca, that has average price in the market. Its outer layer is hairy and cracks are visible.

</Card.Text>

</Card.Body>

</Card>

</Row>

</div>

);

}

if (this.state.category==2) {

return (

<div>

<Row>

<Card style={{ width: '18rem' }}>

<Card.Body>

<Card.Title>Areca belongs to</Card.Title>

<Card.Subtitle className="mb-2 text-muted">CATEGORY 3</Card.Subtitle>

<Card.Text>

The quality of this areca is degraded and it has least price.

</Card.Text>

</Card.Body>

</Card>

</Row>

</div>

);

}

}

render() {

return (

<div>

<Navbar bg="primary" variant="dark" >

<Navbar.Brand href="#home">ARECA CLASSIFICATION</Navbar.Brand>

<Nav className="mr-auto">

<Nav.Link href="#home">Home</Nav.Link>

<Areca />

<AboutUs />

</Nav>

</Navbar>

<br/>

<br/>

<Container fluid="md">

<Row>

<Col>

<h3>

Upload a areca picture!

</h3>

<br/>

<div>

<input type="file" onChange={this.onFileChange} /> <br/>

<Button variant="primary" onClick={this.getCategory} >PREDICT</Button>{' '}

</div>

</Col>

<Col>

{this.fileData()}

</Col>

</Row>

<Row>

{this.categoryData()}

</Row>

</Container>

</div>

);

}

}

ReactDOM.render(<App />, document.getElementById('root'))

**6.6 Back End Code(NODE JS)**

const express = require('express')

const app = express()

const multer = require('multer');

const upload = multer({ dest: 'uploads/' });

const { exec }= require('child\_process')

const bodyParser = require('body-parser');

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({extended: true}));

const cors = require("cors");

app.use(cors());

// // app.use(express.static(\_\_dirname, 'public'));

app.post('/api/category', upload.single('image'), (req, res) => {

var ans={};

console.log("This is reached")

var filepath = req.file.path.split("\\");

exec(`cd C:/Users/DEll/AppData/Local/Programs/Python/Python36 && python last.py C:/Users/DEll/robofriends/${filepath}/${filepath}`, (error,stdout,stderr) =>{

console.log("This is also reached")

if(stdout){

console.log("category:"+stdout);

ans = stdout;

console.log(ans);

res.send(ans);

}

})

});

app.listen(3001, () => console.log('Example app listening on port 3000!'))

**6.7 Python Code in Local System**

This code is executed when node api is called and this api gives the category of the areca based on the input image.

import keras

import sys

import numpy as np

from keras.models import load\_model

import cv2

argumentList = sys.argv

img = cv2.imread(sys.argv)

img = cv2.resize(img,(32,32))

mask = np.zeros(img.shape,np.uint8)

bgdModel = np.zeros((1,65),np.float64)

fgdModel = np.zeros((1,65),np.float64)

rect = (5,5,19,18)

cv2.grabCut(img,mask,rect,bgdModel,fgdModel,5,cv2.GC\_INIT\_WITH\_RECT)

mask2 = np.where((mask==2)|(mask==0),0,1).astype('uint8')

img = img\*mask2

img\_array = keras.preprocessing.image.img\_to\_array(img)

img\_array = img\_array/255.0

img = np.expand\_dims(img\_array, axis = 0)

model=load\_model('fine.h5')

a = model.predict\_classes(img)

print(a)

**CHAPTER 7**

**TESTING & RESULTS**

**TESTING AND RESULTS**

**7.1 Working Cases**

**Fig 7.1 Images of Working Cases**

These test examples have proper lighting and hence the classification is proper. The reasons for the proper classification will be:

* Constant lighting
* No ambiguity among the test examples

**7.2 Not Working Cases**

**Fig 7.2 Images of Not Working Cases**

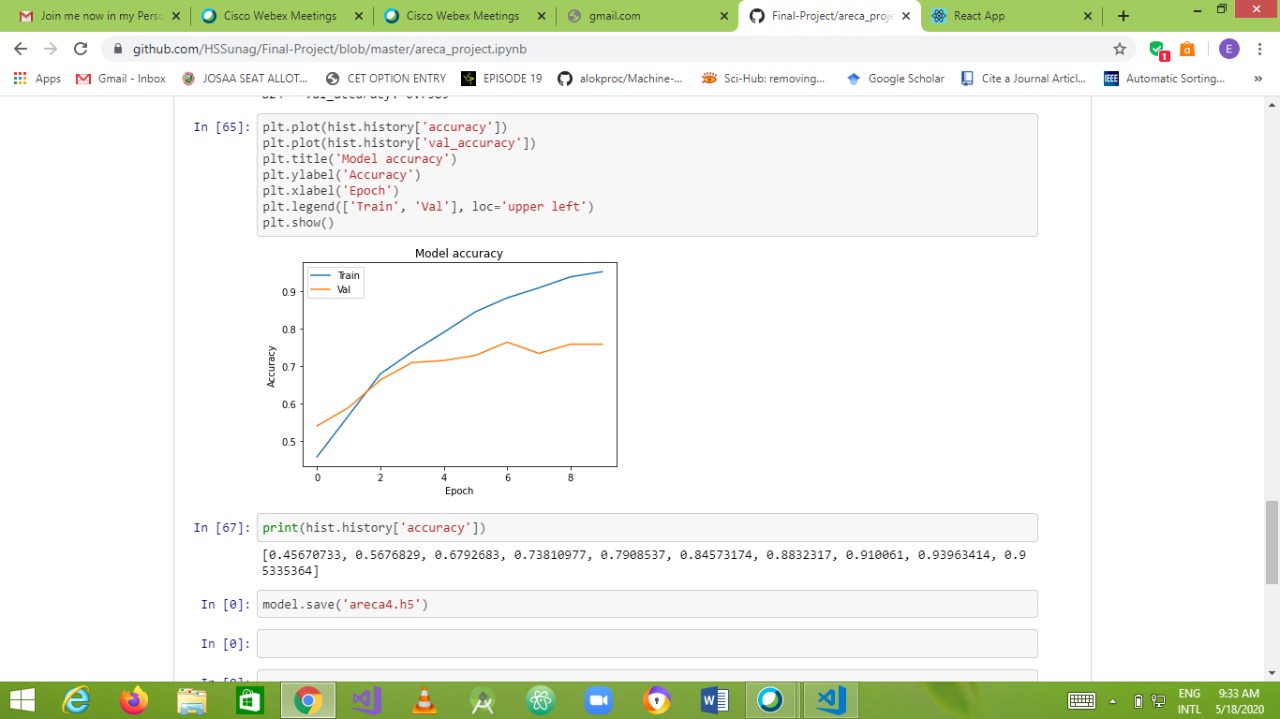
The above images are not classified correctly because of certain reasons as mentioned below.

First Image has two background by which the preprocessing module fails to detect the actual background and hence gives deformed areca picture and categorizes wrongly.

Second Image has extensive brightness due to which the picture is bright and looks as if the areca belongs to the first category, but it is actually the second category areca.

**7.3 Results**

We trained the model using Convolutional Neural Networks and obtained a test result of 75 percent accuracy. The sequential model was being used. Two convolution layers and two Max-pooling layers were used to obtain the results.



**Fig 7.3 Accuracy vs Epoch graph**

**CHAPTER 8**

**APPLICATIONS**

**APPLICATIONS**

1. Better classification of the areca nut yield.
2. Human labour can be reduced.
3. Speed of the classification is increased.
4. Low cost of implementation.

**CHAPTER 9**

**CONCLUSION & FUTURE WORK**

**CONCLUSION & FUTURE WORK**

**9.1 Conclusion**

The farmers had to depend on middlemen and labourers to decide the quality of their Arecanut yield. This project makes the farmers independent and allows them to divide the yield into 3 categories namely, CAT1, CAT2 & CAT3. The project considers all the real-world grading measures like shape, color and size of the Arecanut. The dataset images are pre-processed using Augmentation tools. And the model is trained with 70% of the images in the dataset. This model is tested with remaining 30% of the images in the dataset. The model with highest accuracy is stored in the back-end. The front-end is simple for the farmers to understand and consists of uploading the image.This image is processed at the back-end and the CATEGORY of the uploaded image is displayed.

**9.2 Future Work**

* The project can be made to work on more number of CATEGORIES of Arecanut.
* Different types of fruits and vegetables can also be categorized by varying the classifying conditions.
* A mobile app can also be created in future.

**CHAPTER 10**

**BIBLOGRAPHY**

**BIBLOGRAPHY**

**A.CONFERENCE**

[1]  Elakkiya, N., Karthikeyan, S. and Ravi, T. (2018). Survey of Grading Process for Agricultural Foods by Using Artificial Intelligence Technique. 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA).

[2]  Shyna, A. and George, R. (2017). Machine vision based real time cashew grading and sorting system using SVM and back propagation neural network. 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT).

[3]  Shetty, D., Dinesh Acharya, U., Malarout, N., Gopakumar, R. and PJ, P. (2019). A Review of Application of Computer-vision for Quality Grading of Food Products. 2019 International Conference on Automation, Computational and Technology Management (ICACTM).

[5] V.Pavithra, R.Pounroja, B.Sathya bama, “Machine vision based automatic sorting of cherry tomatoes”,(ICECS),ISBN.4799-7226,pp.271275,IEEE 2015.

[6]  Nandi, C., Tudu, B. and Koley, C. (2016). A Machine Vision Technique for Grading of Harvested Mangoes Based on Maturity and Quality. IEEE Sensors Journal, 16(16), pp.6387-6396.

[7]  Kumar, R., Rajpurohit, V. and Nargund, V. (2017). A neural network assisted machine vision system for sorting pomegranate fruits. 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT).

[8] Riaz Ahmed Sheikh, Jian Ping Li, Asif Khan and Imran Khan(2016). Content Based grading of Fresh fruits using Markov Random Field. International conference on computing for Sustainable global development.