MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY

BHOPAL, INDIA, 462003

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Minor Project Report

On

FRUIT SHELF LIFE PREDICTOR

Submitted by:

Prateek Verma **Scholar No: UG/FT 171112291**

Ammar Alavi **Scholar No: UG/FT 171112292**

Mayank Verma **Scholar No: UG/FT 171112260**

Under the Guidance of

BHOLANATH ROY SIR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Session: 2019-20

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY

BHOPAL, INDIA, 462003

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project report carried out on “**FRUIT SHELF LIFE PREDICTOR**” by the 3rd year students:

Prateek Verma **Scholar No: UG/FT 171112291**

Ammar Alavi **Scholar No: UG/FT 171112292**

Mayank Verma **Scholar No: UG/FT 171112260**

Have successfully completed their project in partial fulfillment of their Degree in Bachelor of Technology in Computer Science and Engineering.

BHOLANATH ROY SIR

(Minor Project Mentor)

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY

BHOPAL, INDIA, 462003

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We, hereby declare that the following report which is being presented in the Minor Project Documentation Entitled as “**FRUIT SHELF LIFE PREDICTOR**” is an authentic documentation of our own original work and to the best of our knowledge. The following project and its report, in part or whole, has not been presented or submitted by us for any purpose in any other institute or organization. Any contribution made to the research by others, with whom we have worked at Maulana Azad National Institute of Technology, Bhopal or elsewhere, is explicitly acknowledged in the report.

Prateek Verma **Scholar No: UG/FT 171112291**

Ammar Alavi **Scholar No: UG/FT 171112292**

Mayank Verma **Scholar No: UG/FT 171112260**

ACKNOWLEDGEMENT

With due respect, we express our deep sense of gratitude to our respected guide and coordinator Dr. Dhirendra Pratap Singh, for his valuable help and guidance. We are thankful for the encouragement that he has given us in completing this project successfully.

It is imperative for us to mention the fact that the report of the minor project could not have been accomplished without the periodic suggestions and advice of our project guide Bholanath Roy Sir and project coordinators Dr. Dhirendra Pratap Singh and Dr. Jay Trilok Chaudhary.

We are also grateful to our respected director Dr. N. S. Raghuwanshi for permitting us to utilize all the necessary facilities of the college.

We are also thankful to all the other faculty, staff members and laboratory attendants of our department for their kind cooperation and help. Last but certainly not the least, we would like to express our deep appreciation towards our family members and batch mates for providing the much-needed support and encouragement.

ABSTRACT

Recently computer vision has gained attraction by many researchers and practitioners, because it can be used to control various applications and devices. With the upcoming era of IOT devices, computer vision applications and devices have been in common trend.

The project is about creating an application which can predict fruit shelf life. Initially we propose to create a simple application which can predict the shelf life of banana fruit.

We propose to use digital image processing to recognize the color of a banana.

However, few companies in the agriculture industry use computer vision because of the non-uniformity of sellable produce. The small number of agriculture companies that do utilize computer vision use it to extract features for size sorting or for a binary grading system: if the piece of fruit has a certain color, certain shape, and certain size, then it passes and is sold. If any of the above criteria are not met, then the fruit is discarded. This is a highly wasteful and relatively subjective process.

CONTENTS

1. INTRODUCTION

Computer vision is one of the fastest growing and one of the most researched disciplines in this day and age. Computer vision is the hardware and software involved in capturing and analyzing an image with a computer. It has found a home on inspection lines because of its speed, accuracy, reliability, and objectivity. Computer vision excels on systems such as inspection lines at finding defects in well-defined objects. When a computer vision system has an ideal representation of what a, say, electronic component looks like, it is very easy to detect deviations from the norm. However, a field that is still in its infancy is applying computer vision to non-uniform objects. The purpose of this thesis is to present a process that applies existing computer vision techniques to build a predictive model of the shelf-life of fruit.

Increasingly, computer vision is being used in the agriculture industry. Agriculture is a good fit for computer vision because of the sheer volume of fruit that most factories process every day and the time, effort, and money that goes into quality control and distribution decisions. Today, some factories use computer vision to detect for bruises and other defects that would detract customers. This type of inspection falls under the broader category of fruit quality grading. In most agricultural businesses, this grading is done by humans. Essentially, the grading is a rating on a scale from one to five of the external quality of the fruit. Fruits with ratings of three and above are sold while those with ratings of two and below are discarded. This type of grading is a binary process that is beginning to be (somewhat reliably) performed by computers. Another, much more advanced, application is to use computer vision to classify distinct categories 2 of fruit with similar characteristics and place all of one group of fruit into one room, dose the entire room with ethylene, and then sell the entirety of that group when it is ripe. However, this takes up time, space, and money to store and wait for the fruit. It is also frequently wasteful and inaccurate.

The purpose of this thesis is to present a process that uses computer vision to extract color features of a large group of the fruit and then builds a predictive model of the shelf-life for the fruit (specifically bananas for the purposes of this thesis). While the thesis demonstrates the process on bananas, the process can theoretically be repeated for any climacteric fruit that changes colors distinctly as it ripens. Essentially, a camera would scan each piece of fruit, extract features, and then plug the values into the created model to get an accurate prediction of the shelf-life of that particular piece of fruit. This would allow fruit with a shorter shelf-life to be shipped across the state whereas fruit with a longer shelf-life would be shipped across the country. This would save money and space. It would also allow more fruit to be sold because fruit that may have been given an unacceptable rating would still be able to be sold (if the fruit is deemed to ripe to be shipped, it can still be sold locally).

Fundamentally, instead of making distribution decisions based on current features (as is done in the industry today), this thesis proposes a model which allows current features to be used to predict future features, which are used to inform distribution decisions. Finally, the proposed technique would be advantageous because it allows for an objective, accurate, and concrete shelf-life prediction system as opposed to the current fruit grading procedure. This thesis develops an experimental process to undergo and the subsequent statistical analysis to build a predictive model of color features against shelf-life. The thesis uses bananas as a proof 3 of concept of the process. There are two experiments: the first one captures images of a few samples of fruit and then destructively measures sugar content. The purpose of the second experiment is to correlate color representations with sugar content to determine the end point of shelf-life. A sugar content of 23% indicates a ripe banana, which, for the purposes of this thesis will serve as the last day of data collection. The color of many samples of bananas on the first day that their sugar content reaches or exceeds 23% is recorded and the color range is then set as the end color of the second experiment. The second experiment involves capturing images of many samples of the fruit each day until it becomes spoiled. The color features are extracted every day using computer vision. For statistical analysis, each fruit is plotted to measure color as a function of time. Postprocessing then occurs with different representations of color being calculated (using existing computer vision algorithms). Then, for each fruit, shelf-life is calculated at a given color value. This is repeated for each selected color and each fruit. Finally, shelflife as a function of color can be modeled (with the corresponding standard deviation at each color) to ultimately give a model of shelf-life based on color.

2. PROPOSED WORK

The objective of this thesis is to develop a process that acts as the framework to build a predictive model of fruit shelf-life based on features extracted by computer vision.To that end, the following section outlines each step to take in the process and justifies each step.

The process proposed in this thesis develops certain discrete steps to undertake to build a model that predicts shelf-life based on empirically measured color features. The first step is to undertake an experiment that correlates color to sugar content (and, therefore, literature values of ripeness). The process uses color features to estimate sugar content to estimate ripeness. That is, sugar content is predicted by color, so **color is the criterion for shelf-life**. Using sugar content to estimate ripeness is different than using ethylene content to measure ripeness (most of the papers in the Literature Review use ethylene content measurements). The reason that this approach was chosen is simple: cost without loss of quality. By using the cost-effective method of subjectively measuring sugar content, the process allows for research to be done that may not have been economically viable using ethylene content. Research done by Tapre and Jain, 2012 and Soltani et al., 2010, have correlated sugar content to ripeness, validating it as an effective technique. In this thesis, twenty bananas are chosen as proof of concept of the process.

We use the research done by Faculty of California that studied the ripeness is correlated to the color of the banana and found out that the bananas were ripe at a certain hue value which was found close to **23~24**, hence we use this hue value as our end point to say a banana is ripe. The process uses 60 bananas and takes a picture of each banana on every day of the experiment. Each day, after the picture is taken of each banana, computer vision is used to extract features and the color of every banana is recorded every day. The first day that the experimental banana falls into the “ripe” color range (determined in experiment one) indicates the end of life for that specific banana. Because we are interested in distribution decisions, the first day of ripeness serves as the subjective end of life for the project. The justification of this is that retail stores will not accept bananas after they are ripe because they will not be able to sell them.

After large scale data collection, **RGB color coordinates (the default color representation) are transformed into HSI color coordinates**. While RGB (red, green, blue) coordinates are valuable and ubiquitous, they are very sensitive to lighting conditions and vary from camera to camera. It is, therefore, necessary to convert the extracted RGB coordinates to HSI (Hue, Saturation, Intensity) coordinates. This conversion deemphasizes the issue of lighting sensitivity because the image is deconstructed into color (hue) and intensity (brightness) instead of being deconstructed into red, green, and blue pixels. This simplifies and standardizes color processing by separating the color dimension from the other dimensions of an image. Also, HSI is standardized to not vary between devices as RGB does. That is, because of differences in the amount and quality of RGB filters between imaging devices, RGB coordinates of the same image can vary between different devices. However, HSI values will be the same across devices because RGB coordinates are translated to sRGB (standard red, green, blue) before being converted to HSI, which is calculated to not take RGB filter differences between devices into account. Standard RGB accomplishes this by finding the largest coordinate of the red, green, and blue values and uses that as a non-zero reference baseline, while the other two colors become zero. This allows for a richer and more accurate description of color.

The algorithms to transform RGB to HSI are presented in the "Methodology" section of this project. We calculate the day at which the banana reaches the hue value of 23 at which it is ripe. After we get the hue values of the different bananas at different days, we use **hue as the main feature to make a predictive model** that classifies a banana into the corresponding day using hue value of the banana. Then we find the **difference between the predicted day of the banana and the day of ripening**, this tells us the remaining shelf life of the banana.

3. METHODOLOGY & WORK DESCRIPTION

WORK DESCRIPTION

3.1.1 Camera

A simple camera or mobile camera attached to the system is needed to capture the photos of banana fruit. The camera will change the visual data into digital data. A good camera along with good lighting & preferred angle of about 45 degree is a must for better results.

3.1.2. Background Subtraction for Data Processing

Separating the banana image utilizing basic and static cameras is an algorithm-based approach. This approach's sole purpose is to identify the object (banana fruit) from the stationary context, in our case its side. This is achieved by fixing the camera at a certain angle with the frame of reference. The background image which is white sheet will be the reference point. Subtraction of the image background is done using OpenCV Python by identifying the region of interest (ROI). The area of the image containing the banana fruit will be the region of interest. Subsequent steps of image preprocessing include eradicating the noise, adjusting the color tones (if necessary) and post-processing. And the various factors which include uneven skin tone, reflections, erratic lighting and blurred objects leads the background subtraction to difficult tasks.

3.1.3. Conversion of RGB to HSI Color Model

The major objective of this conversion is to extract the Hue value for each banana fruit and for this RGB color coordinates must be transformed to HSI color coordinates in order to build the models. Each pixel in a given (and consistent between individual bananas) region of interest was classified using RGB coordinates and then converted to HSI. The consistent region of interest was 300 pixels by 300 pixels. A python script was written to automate the conversion between RGB and HSI shown below:

```python
import numpy as np
import math
import cv2
import csv
daylist = [["Day", "Hue"]]
for j in range(1,6):
    for i in range(20):
        path = "images/Day" + str(j) +"/Day" + str(j) + "_" + str(i) +
".jpg"
        image = cv2.imread(path)
        color = image[300,300]
        region= image[0:299,0:299]
        b,g,r = np.mean(region, axis=(0,1))
        r_ = r/255
        g_ = g/255
        b_ = b/255
        Cmax = max(r_,g_,b_)
        Cmin = min(r_,g_,b_)
        delta = Cmax-Cmin
        H=0
        # Hue Calculation
        if delta == 0:
            H = 0
        elif Cmax == r_ :
            H = 60 * (((g_ - b_)/delta) % 6)
        elif Cmax == g_:
            H = 60 * (((b_ - r_)/delta) + 2)
        elif Cmax == b_:
            H = 60 * (((r_ - g_)/delta) + 4)

        print(H)
        daylist.append(["Day"+str(j), H])
with open('HueValue.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerows(daylist)
```

Average hue is calculated by first finding the RGB coordinates of every pixel in the 300 pixel by 300 pixel region of interest (which is the same region of interest between and among bananas).

Then, each pixel in the region of interest is converted from RGB to HSI (using the script above). Finally, the average hue value is calculated by dividing the sum of the hue values of the pixels in the region of interest by the total number of pixels (for this thesis, 90000 pixels) in the region of interest.
Hue maximum is the largest hue value in the region of interest and hue minimum is the smallest hue value in the region of interest. All calculation is done by Python script.

METHODOLOGY

This section will introduce the method for predicting shelf life using hue values is introduced above. The criteria for choosing fruit as well as the exact steps that must be taken are elaborated upon. The required software and hardware as well as the procedure for this experiment are discussed.

There are two fundamental features that the fruit of interest must possess in order to work in the proposed model building process. The first feature is that the fruit must be capable of ripening after it is picked. The post-harvest ripening is a characteristic of climacteric fruit. If the fruit does not ripen after harvesting, there would be no shelf-life model to build.

The second feature is that the fruit must exhibit noticeable changes in color (or some other feature such as size or shape) as it ripens. That is, computer vision must be able to detect the color (or other feature) changes as the fruit ripens. The fruit used in this thesis is bananas because they have very distinct unripe (yellowish-green), ripe(yellow), and spoiled (brown-black) states. Bananas are also climacteric, so they will ripen after they are harvested.

Bananas were also chosen as the fruit for this thesis because they are one of the most important agricultural commodities (in terms of units sold and widespread distribution). Other fruits that the process presented in this thesis could work with are: apples, melons, tomatoes, avocado, blackberries, etc. It is important to note that fruit grown in different conditions (soil composition, temperature, humidity, sun exposure, etc.) may exhibit different color features. The results that this thesis presents are valid only for the specific bananas that were tested. The following section outlines the experimental processes that were undertaken.

Method:

1. Label each banana so that it can be distinguished at a later time.
2. Set up the experiment by putting the first banana on the blank sheet of paper. Place the piece of fruit directly under the camera.
3. Configure the camera using the viewfinder so that the image acquired only captures the entire fruit.
4. Measure and record the distance of the camera from the banana. Ensure that this distance is consistent for every image acquisition.
5. Set up a repository for the images on the computer that the camera is attached to.
6. Capture the image.
7. Save the image to the computer.
8. Repeat steps 1-7 for each day until the hue reaches the end-point hue established from above Python script.

		
Day 1 Hue value=44.164	Day 3 Hue value=35.269	Day 5 Hue value=22.133

Computer Vision Feature Extraction:

1. Load the image into OpenCV Python framework and navigate the file path to the folder with the acquired images.

2. Choose a consistent 300 pixel by 300 pixel square region of interest to perform processing.
3. Using the RGB values feature in OpenCV to extract and record color features.
4. Convert RGB to HSI. This should be done for every banana in the sample.
5. Compute Hue average value and this should be done for every banana in the sample.
6. Repeat analysis for each acquired image.

Prediction Using Machine Learning Model

For predicting the Shelf-Life of the banana fruit based on the Hue value as an attribute we have used the two different machine learning models which are described below :

1. K Nearest Neighbor

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method proposed by Thomas Cover used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression. Here we are using KNN as a classification model:

- In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.
- In our case the **classes will be days ranging from Day1 to Day5** and the feature on which we will classify the banana will be hue.
- We will be using 5 nearest neighbors.

Here is the code snippet for KNN:

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from mlxtend.plotting import plot_decision_regions

# Importing the dataset
dataset = pd.read_csv('HueValueFinal.csv')
X = dataset.iloc[1:, 1:].values
y = dataset.iloc[1:, 0].values
```

```
dataset = pd.read_csv('TestHueValue.csv')
X_test = dataset.iloc[:, 1:2].values
y_test = dataset.iloc[:, 0].values

X_train = X
y_train = y

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Fitting classifier to the Training set
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric =
'minkowski', p=2)
classifier.fit(X_train,y_train)


# Predicting the Test set results
y_pred = classifier.predict(X_test)

from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,y_pred)*100)

plot_decision_regions(X, y, clf = classifier, legend = 2)
plt.show()

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix, classification_report
cm = confusion_matrix(y_test, y_pred)
print(cm)
report = classification_report(y_test, y_pred)
print(report)
```

2. Polynomial regression

In statistics, **polynomial regression** is a form of regression analysis in which the relationship between the independent variable *x* and the dependent variable *y* is modelled as an *n*th degree polynomial in *x*. Polynomial regression fits a nonlinear relationship between the value of *x* and the corresponding *y*. Polynomial regression is considered to be a special case of multiple linear regression.

- In our case we have y variable representing the **number of days** and the x variable representing the **hue value** of banana fruit corresponding to the respective day.

- Here in our model we have used the *n*th degree polynomial in *x* as 2, which fits satisfactory in our dataset.
- After we make our model using polynomial regression, the predicted value obtained is in the form of a floating number which we have to round off to the nearest integer value to get our corresponding day.

Here is the code snippet for polynomial regression:

```
# Polynomial Regression

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('HueValueFinal.csv')
X = dataset.iloc[1:, 1:].values
y = dataset.iloc[1:, 0].values

dataset = pd.read_csv('TestHueValue.csv')
X_test = dataset.iloc[:, 1:2].values
y_test = dataset.iloc[:, 0].values

X_train = X
y_train = y

# Training the Linear Regression model on the whole dataset
from sklearn.linear_model import LinearRegression

# Training the Polynomial Regression model on the whole dataset
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 2)
X_poly = poly_reg.fit_transform(X_train)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y_train)

# Visualising the Polynomial Regression results (for higher resolution
and smoother curve)
```

```
X_grid = np.arange(min(X_train), max(X_train), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_grid, lin_reg_2.predict(poly_reg.fit_transform(X_grid)),
color = 'blue')
plt.title('Day Prediction (Polynomial Regression)')
plt.xlabel('Hue Value')
plt.ylabel('Day')
plt.show()

# Predicting a new result with Polynomial Regression
y_pred = lin_reg_2.predict(poly_reg.fit_transform(X_test))
y_pred=np.array(y_pred).round()
print(y_test)
print(y_train)

from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,y_pred)*100)

import sklearn.metrics as sm
print("Mean absolute error =", round(sm.mean_absolute_error(y_test,
y_pred), 2))
print("Mean squared error =", round(sm.mean_squared_error(y_test,
y_pred), 2))
print("Median absolute error =", round(sm.median_absolute_error(y_test,
y_pred), 2))
print("Explain variance score =",
round(sm.explained_variance_score(y_test, y_pred), 2))
print("R2 score =", round(sm.r2_score(y_test, y_pred), 2))
```

TOOLS & TECHNOLOGY USED

- Software Requirements:

  The following softwares were used for this project:

Operating System	:	Microsoft® Windows® 10 (64 bit)
Python 3.7.0	:	Programming Language for Machine Learning.
IDE	:	Spyder
Library	:	OpenCV (digital image processing), Numpy (data handling), Matplotlib (graph plotting), Sklearn (ML models)

- Hardware Requirements:

  The following hardware configuration were used and at least required to run the various softwares for this project:

System	:	Intel Compatible PC
Processor	:	Intel® Core™ i5 CPU and above.
Memory	:	4 GB+ RAM
Hard Disk Drive	:	10 GB

5. RESULT ANALYSIS

- We applied Polynomial Regression and k-nearest neighbors on our dataset.
- The hue values of banana images ranges from 48 to 20.
- As the banana ripens, its hue value decreases from day 1 to day 5.
- The below table shows the accuracy of both models in percentage -

Polynomial Regression	K-nearest neighbors
68.33%	73.33%

- We can see that K-nearest neighbors gives us a better accuracy than polynomial regression therefore we plan to use K-NN as our classification model moving forward.

ANALYSIS OF POLYNOMIAL REGRESSION:

Accuracy	68.33333333333333
Mean absolute error	0.32
Mean squared error	0.32
Median absolute error	0.0
Explain variance score	0.84
R2 score	0.84

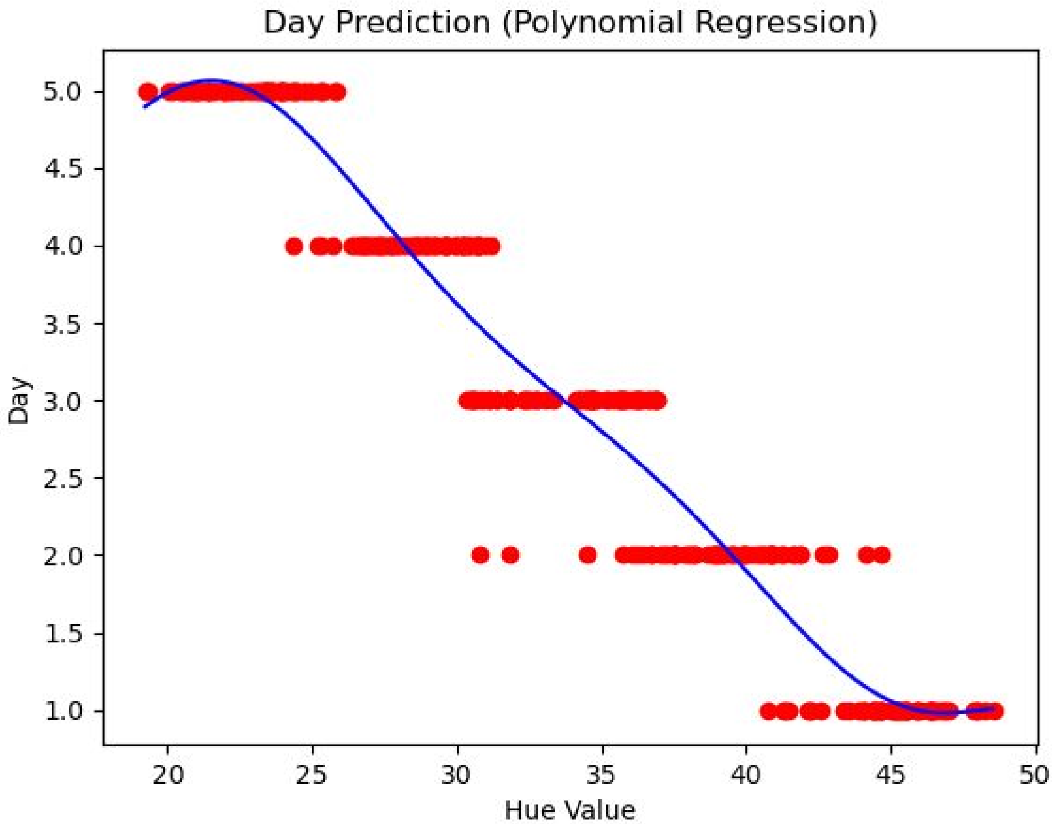

Fig. Generated Chart for polynomial regression

ANALYSIS OF K NEAREST NEIGHBORS:

Actual / Predicted	Day1	Day2	Day3	Day4	Day5
Day1	10	2	0	0	0
Day2	2	6	4	0	0
Day3	0	2	9	1	0
Day4	0	0	2	9	1
Day5	0	0	0	2	10

Confusion matrix

	precision	recall	f1-score	support
Day1	0.83	0.83	0.83	12
Day2	0.60	0.50	0.55	12
Day3	0.60	0.75	0.67	12
Day4	0.75	0.75	0.75	12
Day5	0.91	0.83	0.87	12
accuracy			0.73	60
Macro avg	0.74	0.73	0.73	60
Weighted avg	0.74	0.73	0.73	60

Result on testing data

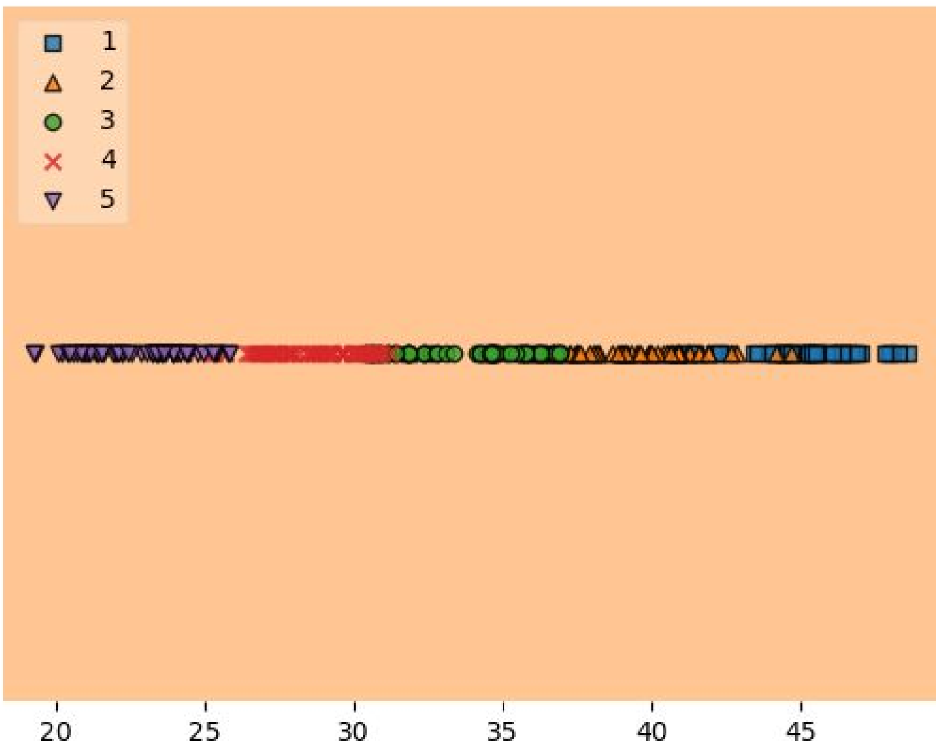

Fig. Generated Chart for k-nearest neighbors

6. CONCLUSION

Ultimately, this project has demonstrated proof of concept of a process to use computer vision and machine learning to build a predictive model of shelf-life of fruit. Computer vision is becoming widespread in many industries and is, therefore, a burgeoning field of research.
Computer vision holds many advantages to it’s human counterpart: it is faster, more reliable, cheaper in the long-term, and more accurate.

Naturally, computer vision is starting to become implemented in other industries. One of the most important fields that **computer vision is starting to move into is the agriculture industry.** Computer vision excels at doing a well defined process many times. This lends well to industries such as electronics production because every piece of one electronic component must look the same.

This project describes how an image of the component leads to a decision to sell or discard the component.

From this project we come to the conclusion that the hue value can be used as a reliable feature for predicting the shelf life of the fruit. When we use the K-Nearest Neighbor (KNN) classification model the result obtained is better than Polynomial Regression. Therefore, we plan on using the KNN Classification model moving forward.

7. REFERENCES

RESEARCH PAPERS:-

- Using Computer Vision Techniques to build a predictive model of **Fruit Shelf-Life** by **Nandan G. Thor** (California Polytechnic State University, San Luis Obispo, June 2017)

WEBSITE:

- www.wikipedia.com - WIKIPEDIA
- www.researchgate.net - RESEARCHGATE
- www.docs.opencv.org - OPENCV
- www.googlescholar.com - GOOGLE SCHOLAR
- www.numpy.org - NUMPY