iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71687

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Support Vector Machine for Detection of Tomato Ripeness Using Feature Extraction - HSV, RGB and LBP

Dattatray Sadashiv Arade
Assistant Professor, Department of BBA(C.A.), Tuljaram Chaturchand College, Baramati , Pune, Maharashtra.

Abstract: ***Accurate detection of tomato ripeness is crucial for post-harvest processes, reducing waste, and ensuring the consistent quality product delivered which maintaining quality in agricultural production. This study investigates two feature extraction techniques for detection of tomato ripeness using a Support Vector Machine classifier: (1) color-based features combining HSV and RGB, and (2) texture-based features using LBP (Local Binary Patterns). The experimental results show that color based feature approach achieves a high classification accuracy of 98.88%, whereas the texture based method which records an accuracy of 82.3%. The superior performance of color features indicates their strong ability to capture the visual changes in color associated with different ripeness stages. While texture features provide useful information, they are less effective alone in distinguishing ripeness variations. The cross-validation results for color based feature show that the model achieves a high average accuracy of 95.51% with a low standard deviation of ±2.36%, indicating strong and consistent performance across different data splits.***

Keywords: Tomato Ripeness, Support Vector Machine, Feature Extraction, Classification, Cross-Validation

I. INTRODUCTION

Support Vector Machine is a powerful supervised machine learning algorithm widely used for classification and regression tasks. Its primary objective is to find the optimal hyperplane that best separates data points belonging to different classes in a high-dimensional feature space. The optimal hyperplane is defined as the one that maximizes the margin the distance between the hyperplane and the nearest data points from each class, known as support vectors. Tomato ripeness detection is essential in agriculture and food processing, where traditional manual inspection methods are often subjective and inefficient. The growing need for automated, accurate, and scalable solutions has led to the adoption of computer vision and machine learning techniques. This study investigates the effectiveness of Support Vector Machine classifiers trained on color based features extraction derived from combining HSV and RGB, and texture features extracted using Local Binary Patterns. The study aims is to determine the most reliable method for accurately classifying tomato ripeness and applying overfitting method to generalize levels in real-world scenarios.

II. LITERATURE REVIEW

[1] The paper "Identification of Ripeness of Tomatoes" proposes a cost-effective system using image processing and machine learning to assess tomato ripeness and quality. Images of tomatoes are captured using a Raspberry Pi, and features such as color, texture, size, and shape are extracted. Color features are analyzed using the HSV model, while texture features are extracted using the Gray Level Co-occurrence Matrix (GLCM). Defect detection is performed using the K-means clustering algorithm. The extracted features are stored in a CSV file and classified using a Support Vector Machine (SVM) model. The system categorizes tomatoes into ripe (red), unripe (green), and defective (black spots), and outputs ripeness percentage, size, and defect status. With 80% of data used for training and 20% for testing, the system provides accurate classification results. The authors propose future hardware integration using a conveyor belt and Pi camera for large-scale, real-time tomato sorting in agricultural industries.

[2] The integration of machine learning techniques in determining tomato ripeness offers a significant advancement in agricultural practices by enabling accurate, efficient, and non-destructive evaluation. Traditional methods like visual inspection and colorimetric analysis are limited by human error and environmental factors, whereas machine learning models, including SVM, YOLOv5, CNNs, and hyperspectral imaging, have demonstrated high accuracy in classifying ripeness stages. Studies have shown accuracies exceeding 90%, with some combinations like YOLOv5m and ResNet101 achieving up to 100% prediction accuracy. These models analyze features such as color, texture, and size to automate sorting and grading processes. Furthermore, integrating these

techniques with IoT and smart farming technologies enables real-time monitoring, optimizing harvesting schedules and reducing post-harvest losses. The review highlights the practical benefits of machine learning in precision agriculture, supply chain management, and food quality control. Continued innovation and collaboration are essential for enhancing model performance and facilitating widespread adoption across the agricultural sector.

III. OBJECTIVES

A. Objective of this Research

1) To develop a feature extraction method for detection of tomato ripeness by combining color based features (HSV and RGB) and texture based features (LBP).

2) To build a Support Vector Machine classifier that predicts tomato ripeness with cross validation method for arising overfitting condition.

3) To evaluate the performance of the Support Vector Machine classifier.

IV. DETAILS OF FEATURE EXTRACTION

A. Feature Extraction

Feature extraction means converting raw data like an image into meaningful numeric values called features. It represent important information that a machine learning model can understand and use.

1) Color Based Feature Extraction-

a) HSV Hue (H), Saturation (S), and Value (V)

The mean HSV feature extraction method involves computing the average values of the Hue (H), Saturation (S), and Value (V) components of an image after converting it from RGB to HSV color space. This technique is widely used in image analysis and computer vision because it provides a compact yet informative representation of an image's overall color characteristics. The mean Hue represents the dominant color tone, the mean Saturation indicates the intensity or vividness of the color, and the mean Value reflects the overall brightness of the image. These features are especially useful in scenarios where color plays a significant role in distinguishing between different classes or conditions, such as in object recognition, crop health monitoring, and image classification tasks.

HSV stands for Hue, Saturation, Value:

$$\mu H = \frac{1}{N}\sum_{i=1}^{n} Hi$$

$$\mu S = \frac{1}{N}\sum_{i=1}^{n} Si$$

$$\mu V = \frac{1}{N}\sum_{i=1}^{n} Vi$$

where:

N = total number of pixels

Hi,Si,Vi components of the i-th pixel

Hue (H): color type (e.g., red, blue, yellow), measured in angles (0–179 in OpenCV)

Saturation (S): color intensity (0–255)

Value (V): brightness (0–255)

HSV Mean Calculation:

We'll convert the image to HSV using OpenCV:

hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

Assume the HSV values after conversion example values:

Pixel (H, S, V)

(15, 128, 200)

(17, 126, 180)

So:

H mean: (15 + 13 + 17 + 16) / 4 = 61 / 4 = 15.25

S mean: (128 + 127 + 126 + 129) / 4 = 510 / 4 = 127.5

V mean: (200 + 210 + 180 + 190) / 4 = 780 / 4 = 195.0

Final Output Example:

HSV Means - H: 15.25, S: 127.50, V: 195.00

2) RGB Red (R), Green (G), and Blue (B)

RGB feature extraction using the mean involves calculating the average intensity values of the Red (R), Green (G), and Blue (B) channels in an image. This method provides a simple yet effective summary of the image's overall color composition. Each of the RGB channels captures different aspects of color: the Red channel reflects the intensity of red tones, the Green channel captures green tones, and the Blue channel represents blue tones. By computing the mean of each channel, we obtain a 3-dimensional feature vector

$$\mu R = \frac{1}{N}\sum_{i=1}^{n} Ri$$

$$\mu G = \frac{1}{N}\sum_{i=1}^{n} Gi$$

$$\mu B = \frac{1}{N}\sum_{i=1}^{n} Bi$$

Where:

Ri,Gi,Bi are the red, green, and blue intensities of the i-th pixel.

By computing the mean (average) of each channel:

R_mean: the overall "redness" of the image

G_mean: the overall "greenness"

B_mean: the overall "blueness"

These give a general idea of the image's color tone.

RBG image (img):

Pixel (B, G, R)

(100, 150, 200)

(130, 140, 180)

Convert this into NumPy format:

```
img = np.array([[[100, 150, 200], [120, 160, 210]],
                [[130, 140, 180], [110, 155, 190]]], dtype=np.uint8)
```

RGB Mean Calculation:

Since OpenCV loads in BGR format:

R channel: 200, 210, 180, 190

⇒ r_mean = (200 + 210 + 180 + 190) / 4 = 780 / 4 = 195.0

G channel: 150, 160, 140, 155

⇒ g_mean = (150 + 160 + 140 + 155) / 4 = 605 / 4 = 151.25

B channel: 100, 120, 130, 110

⇒ b_mean = (100 + 120 + 130 + 110) / 4 = 460 / 4 = 115.0

Final Output Example:

RGB Means - R: 195.00, G: 151.25, B: 115.00

3) Local Binary Pattern (LBP)

Local Binary Pattern (LBP) is a widely used feature extraction technique in image processing and computer vision, particularly for texture analysis. It works by comparing each pixel in a grayscale image with its neighboring pixels, typically in a 3×3 window. For each center pixel, the surrounding pixel values are compared to it: if a neighbor's intensity is greater than or equal to the center pixel, it is assigned a value of 1; otherwise, it is assigned a 0. These binary values are then combined in a clockwise order to form an 8-bit binary number, which is converted to a decimal value and used to represent the texture of that pixel. Once the LBP values for all pixels are computed, a histogram of these values is created. This histogram serves as the LBP feature vector, effectively capturing the texture patterns of the image. LBP is popular due to its simplicity, computational efficiency, and robustness to illumination changes. It has been successfully applied in tasks such as face recognition, image classification, and medical image analysis.

$$LBP_{P,R} = \sum_{P=0}^{P=1} S(I_p - I_c).2^P$$

Where:

P: Number of neighbors (e.g., 8 for a 3×3 window)

R: Radius from the center pixel (e.g., 1 for adjacent neighbors)

I_c: Intensity of the center pixel

I_p: Intensity of the p-th neighbor

s(x): Thresholding function

$$S(x) = \begin{cases} 1, if\ x \geq 0 \\ 0, if\ x < 0 \end{cases}$$

Step-by-Step LBP Calculation

Grayscale Image Patch (3×3)

30 80 45

20 **50** 60

90 10 70

The center pixel (in bold) has a value of **50**.

Compare each neighboring pixel to the center pixel:

If the neighbor ≥ 50, assign **1**

If the neighbor < 50, assign **0**

30 (0) 80 (1) 45 (0)

20 (0) 50 60 (1)

90 (1) 10 (0) 70 (1)

Now arrange these binary values in **clockwise order**, starting from the top-left:

0 1 0 1 1 1 0 0

Convert this binary number to decimal:

$LBP=0\cdot2^0+1\cdot2^1+0\cdot2^2+1\cdot2^3+1\cdot2^4+1\cdot2^5+0\cdot2^6+0\cdot2^7=92$

The LBP value for the center pixel is **92**.

V. RESEARCH METHODOLOGY

1) Image Acquisition

Publically available online dataset on Kaggle website of Tomato images at different ripeness stages (green, turning, and red) will be used. A dataset of consist of total 3560 images (ripe 1975 and unripe 1585) will be compiled.

2) Image Preprocessing

The acquired images will be resized to a uniform dimension and split the dataset into ripe and unripe.

3) Feature Extraction

Two categories of features will be extracted:

a) Color Features:
RGB: Average values of Red, Green, and Blue channels.
HSV: Average values of Hue, Saturation, and Value components. Both combined together
b) Texture Features:
Local Binary Pattern (LBP): To compute the LBP value, the image is first converted to grayscale and average LBP value is calculated. The resulting feature vectors will be normalized to ensure scale uniformity.

4) Classification Using SVM

The extract color features by combining HSV and RGB values and texture features LBP mean values will be used to train an SVM classifier. The dataset will be split into training (80%) and testing (20%) sets. A grid search with cross-validation will be used to tune the SVM parameters (kernel type, C, and gamma values). Different kernel functions (linear, RBF, polynomial) will be evaluated to identify the best-performing configuration.

5) Performance Evaluation

The model's performance will be assessed using standard classification metrics like Accuracy, Precision, Recall, F1-Score, Confusion Matrix.

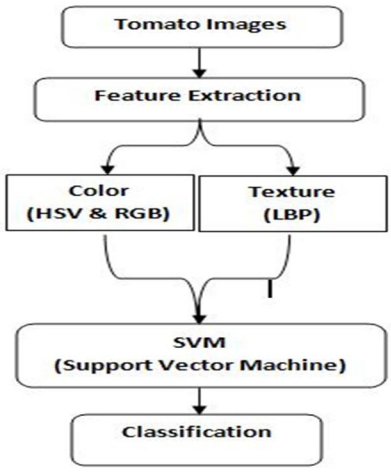


Fig: Research Methodology

VI. RESULT AND FINDINGS

A. Development of Feature Extraction Approach for Tomato Ripeness Detection

The feature extraction approach combining color features HSV and RGB valuable information about the dominant color and brightness of the tomatoes, while the LBP texture features captured the surface patterns and textures indicative of ripeness.

B. Evaluation of SVM Classifier Performance

a) Combining HSV & RGB :
Accuracy: 0.9887640449438202

	precision	recall	f1-score	support
Unripe	0.99	0.99	0.99	348
Ripe	0.99	0.99	0.99	364
accuracy			0.99	712
macro avg	0.99	0.99	0.99	712
weighted avg	0.99	0.99	0.99	712

Cross-Validation Accuracy: 95.51% (+/- 2.36%)
Predicted Ripeness: Unripe
The Support Vector Machine classifier, trained on combining the feature HSV and RGB achieved an impressive classification accuracy of 98.87%. The cross-validation results show that the model achieves a high average accuracy of **95.51%** with a low standard deviation of **±2.36%,** indicating strong and consistent performance across different data splits. The SVM's ability to find an optimal hyperplane that separates the data in the high-dimensional feature space contributed to its high accuracy. The effectiveness of the SVM classifier highlights its potential for use in real-world tomato ripeness detection applications.

b) LBP (Local Binary Patterns)
Accuracy: 0.8207282913165266
Cross-Validation Accuracy: 41.28% (+/- 2.24%)
['Unripe']
The tomato ripeness detection model using LBP (Local Binary Patterns) texture features achieved a moderate accuracy of 82.07%**,** but the cross-validation accuracy of only 41.28% (±2.24%) indicates poor generalization. This suggests that texture information alone is insufficient to capture the full variability of ripeness, especially under different lighting and surface conditions.

C. Impact of Feature Extraction on Classification Accuracy

Feature extraction played a crucial role in enhancing classification accuracy. The reduction in dimensionality through feature extraction allowed the SVM model to focus on the most relevant information, significantly improving computational efficiency and model performance. By isolating important patterns related to color and texture, feature extraction minimized noise and irrelevant data, leading to more accurate predictions. The feature extraction based on color and texture is more robust and reliable foundation for the SVM model. The results confirm the importance of careful feature selection and extraction in image processing tasks, especially for applications like ripeness detection where subtle variations in visual characteristics are critical.

D. Overfitting

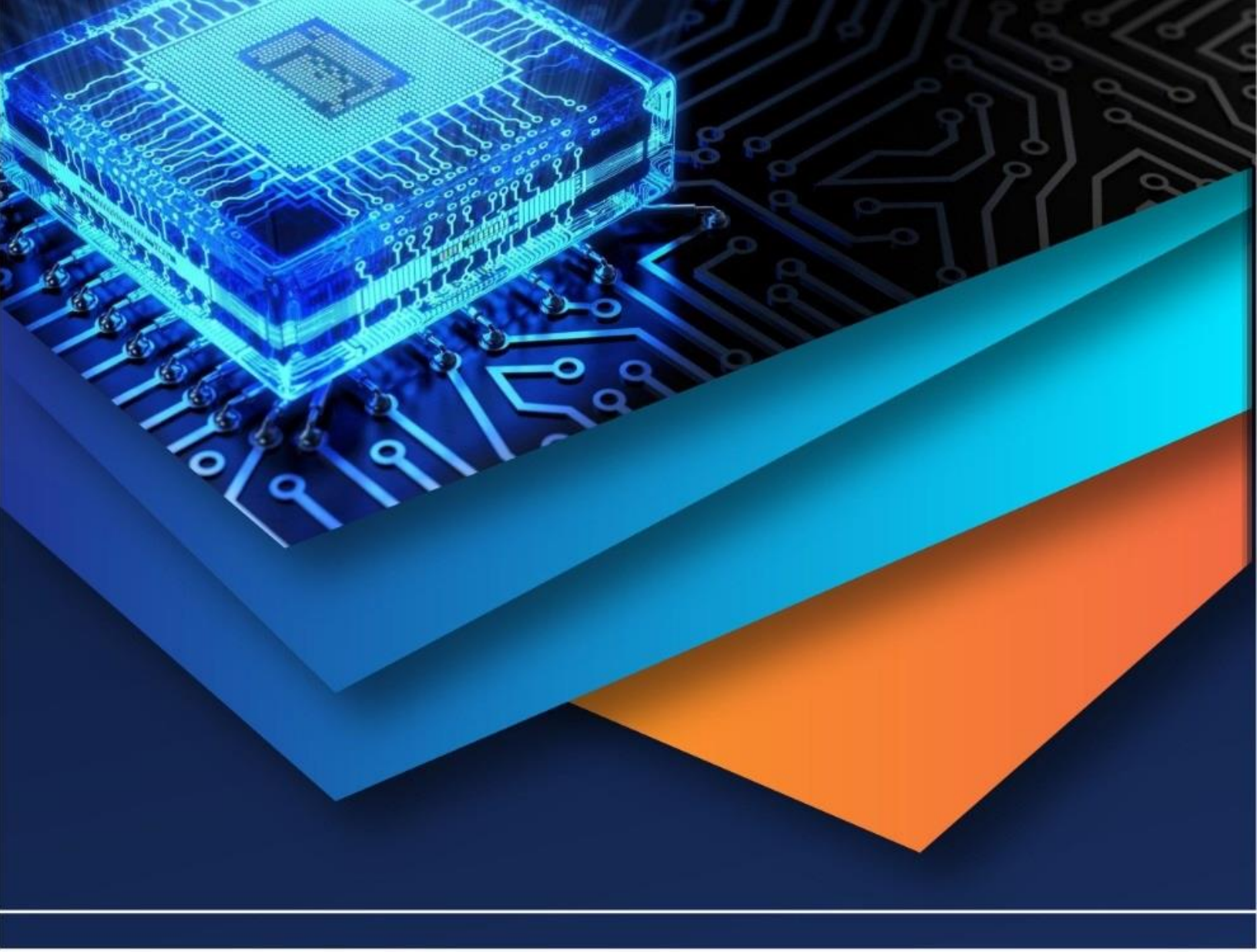
Overfitting occurs when a machine learning model performs exceptionally well on training data but fails to generalize to unseen data, leading to poor real-world performance. Cross-validation is an effective technique to detect and mitigate overfitting by splitting the dataset into multiple folds and evaluating the model on different subsets. This process ensures the model is not just learning noise or patterns specific to one part of the data. By averaging the results across folds, cross-validation provides a more reliable estimate of the model's true performance. It helps in selecting models and tuning parameters that generalize better to new, unseen data.

VII. CONCLUSION

Using HSV and RGB color features with a Support Vector Machine worked very well for detecting tomato ripeness. This method reached a high accuracy of 98.82%. It showed that color features are much better at recognizing ripeness than texture features like LBP, which only gave 82.07% accuracy. The results prove that color changes are the most helpful signs for checking how ripe a tomato is. This project shows that using color-based features can help build accurate and reliable systems to check tomato ripeness in farms and food processing, saving time and reducing human effort.

REFERENCES

[1] Vaibhav G B, 2019, Identification of Ripeness of Tomatoes, International Journal of Engineering Research & Technology (IJERT)
[2] Jesie B. Abad,2024, Integration Of Machine Learning Techniques In Determining Tomato Ripeness: A Literature Review, International Journal of Engineering Applied Sciences and Technology, 2024 Vol. 8, Issue 09, ISSN No. 2455-2143
[3] Rini Nuraini1 ,2023, Tomato Ripeness Detection Using Linear Discriminant Analysis Algorithm with CIELAB and HSV Color Spaces, Building of Informatics, Technology and Science (BITS) Volume 5, No 2, September 2023
[4] Dhiren R. Patel, 2019, Texture Classification of Machined Surfaces Using Image Processing and Machine Learning Techniques, FME Transactions VOL. 47, No 4, 2019 ▪ 871
[5] Priyanka Paygude, 2020, Image Processing Using Machine Learning, Ijsdr | Volume 5 Issue 9
[6] Winda Astuti, 2018, Automatic fruit classification using support vector machines: a comparison with artificial neural network, doi:10.1088/1755-1315/195/1/012047, IOP Conf. Series: Earth and Environmental Science



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY