

COMPARATIVE ANALYSIS OF MATURE TOMATO DETECTION BY FEATURE EXTRACTION AND MACHINE LEARNING FOR AUTONOMOUS GREENHOUSE ROBOTS

Hakkı Alparslan İLGİN¹, Fevzi Anıl AYDEMİR¹, Berkay CEDİMOĞLU¹,
Muhammet Nurullah AYDIN¹, Hasan SİLLELİ²

¹Department of Electrical and Electronics Engineering, Ankara University,
Ankara, TÜRKİYE






²Department of Agricultural Machinery and Technologies Engineering,
Ankara University, Ankara, TÜRKİYE

ABSTRACT. Accurate detection of tomatoes grown in greenhouses is important for timely harvesting. In this way, it is ensured that mature tomatoes are collected by distinguishing them from the unripe ones. Insufficient light, occlusion, and overlapping adversely affect the detection of mature tomatoes. In addition, it is time consuming for people to detect mature tomatoes at certain periods in large greenhouses. For these reasons, high-performance automatic detection of tomatoes by greenhouse robots has become an increasingly studied area today. In this paper, two feature extraction methods, histogram of oriented gradients (HOG) and local binary patterns (LBP), which are effective in object recognition, and two important and commonly used classifiers of machine learning, support vector machines (SVM) and k-nearest neighbor (kNN), are comparatively used to detect and count tomatoes. The HOG and LBP features are classified separately and together by SVM or kNN, and the success of each case are compared. Performance of the detection is improved by eliminating false positive results at the postprocessing stage using color information.

1. INTRODUCTION

In recent years, studies in the field of agriculture which are supported by artificial intelligence have increased. Therefore, the development of fruit harvesting robots that fulfill this purpose has accelerated. These robots can detect and recognize a fruit

Keywords. Tomato detection, harvesting robots, machine learning, smart agriculture.

✉ ilgin@ankara.edu.tr – Corresponding author;  0000-0003-0112-4833
✉ anilaydemir9989@gmail.com;  0000-0002-6530-3040
✉ berkaycedimoglu@gmail.com;  0000-0002-2179-9566
✉ naydinn6@gmail.com;  0000-0002-4026-9739
✉ Hasan.Silleli@agri.ankara.edu.tr;  0000-0003-2242-3402.

© 2023 Ankara University
Communications Faculty of Sciences University of Ankara Series A2-A3: Physical Sciences and Engineering

autonomously with computer vision. On the other hand, developing an artificial visual system as successful as human perception is not an easy task [1, 2].

There are many studies that have been carried out for the visual system of harvesting robots and fruit detection. In [1], mature tomatoes were detected by using color and HOG features. Arivazhagan et al. [3] proposed a method for fruit detection using hue, saturation, value (HSV) color space and texture features. Bulanon et al. [4] developed a method for apple recognition using luminance and red, green and blue color difference. Liu et al. [5] used a simple linear iterative clustering method to segment apple images followed by color feature extraction to detect apples. Drg-Drb color index was utilized to segment apples from their surroundings in [6]. There are also various studies that use techniques other than color space models. Tanigaki et al. [7] used a three-dimensional (3D) vision sensor which is equipped with red and infrared lasers to locate and recognize the fruits and obstacles, prevented by sunlight. Ji et al. [8] utilized a support vector machine (SVM) classifier, for apple recognition. Circular Gabor texture analysis for feature extraction, and neural network with SVM classifier were used in [9] to detect peach fruit. Song et al. [10] proposed bag-of-words (BoW) model and novel statistical approach for locating and counting the pepper fruits. Histogram of oriented gradients (HOG) and local binary patterns (LBP) were used in [11] to detect plants.

In this paper, tomato detection is mainly carried out in three steps which are brightness and contrast enhancement as preprocessing, classification of features, and elimination of false positives as postprocessing. HOG and LBP features were classified separately and together with SVM or kNN, and the results of feature classification combinations were compared. Thus, six different cases for tomato detection were studied. Images are first preprocessed by means of illumination and contrast enhancement to improve the results in terms of detection success under different conditions. Naïve Bayes classifier (NBC) is then used to classify the pixels of the image as black and white, which represent possible background and tomato pixels, respectively. Morphological operations are applied to the black and white image to combine neighboring big groups of possible tomato pixels. Vertical and horizontal limits of those pixels give the coordinates of the region of interest (ROI) on the enhanced image, where the search for tomatoes is carried out with 64×64 sliding windows at a certain step size. LBP and HOG features of sliding windows in ROI are extracted to be used separately and together in detection process via SVM or kNN. At the postprocessing stage, false color removal (FCR) method is utilized to eliminate false positive results. Lastly, non-maximum suppression (NMS) is used to merge overlapped detections of the same tomato.

In the next section, preprocessing and obtaining the ROI is explained. Feature extraction and classification methods are presented in section 3. Postprocessing with

FCR and non-maximum suppression is discussed in section 4. Experimental results and conclusion are given in last two sections.

2. PREPROCESSING AND ROI

At the first stage of the preprocessing, contrast and lighting enhancement is applied to reduce the negative effect of illumination and contrast conditions such as light fluctuation and low contrast which may reduce the success of the detection process. There are various methodologies for contrast enhancement such as general histogram equalization (GHE), local histogram equalization (LHE), singular value equalization (SVE), dynamic histogram equalization (DHE) and contrast limited adaptive histogram equalization (CLAHE) [12]. CLAHE is frequently used in image enhancement applications where contrast and illumination are of prime importance [13, 14]. In this paper, before the training and detection processes, illumination and contrast enhancement algorithms were performed for all images. Since hue, saturation, intensity (HSI) is the one of the most suitable color spaces for these processes, original images in red, green, blue (RGB) color space are converted to HSI color space. Hue (H), saturation (S) and intensity (I) components of HSI color space are obtained as follows, respectively:

$$H = \begin{cases} \theta, & B \leq G \\ 360 - \theta, & B > G \end{cases} \quad (1)$$

where θ is angle in degrees and is given as

$$\theta = \cos^{-1} \left(\frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \quad (2)$$

$$S = 1 - \frac{3}{R + G + B} [\min(R, G, B)] \quad (3)$$

$$I = \frac{R + G + B}{3} \quad (4)$$

where R , G and B are red, green and blue components of the original RGB color space. The illuminance is enhanced by applying the natural logarithm to the intensity component I [1]. CLAHE is applied to the modified intensity component for contrast enhancement. Since CLAHE enhances the contrast by correlating the intensities of neighboring pixels, less noise is seen in the output image compared to AHE. In this way, brightness bursts in the image are prevented [15-17]. After illumination and

contrast enhancement, the image is converted from the HSI color space back to the RGB color space by the equations given below:

$$G = I(1 - S) \quad (5)$$

$$B = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad (6)$$

$$R = 3I - (G + B) \quad (7)$$

An example for the enhancement by natural logarithm and CHALE is given in Fig. 1, where an original greenhouse video frame and illumination and contrast-enhanced video frame are shown, respectively.



FIGURE 1. Video frame color enhancement (a) Original frame (b) Enhanced frame.

In Fig. 1 (a) a dark video frame with low contrast captured in a greenhouse is shown. After contrast and illumination enhancement, it becomes brighter with enhanced contrast for each color component as shown Fig. 2. Red, green and blue color histograms of the original and enhanced frames are shown in Fig. 2 (a), (b) and (c), respectively.

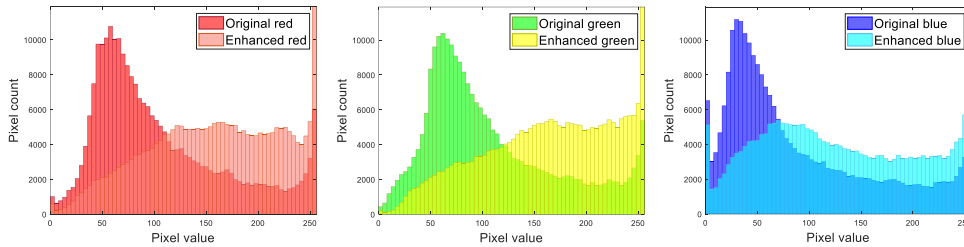


FIGURE 2. Color histograms of the original and enhanced frames in Fig. 1.

As shown in the histograms in Fig. 2, colors are distributed more evenly in the enhanced image than in the original one. Therefore, enhanced image has better contrast.

After illumination and contrast enhancement, ROI, which is the scanning region is obtained using NBC and morphological operations, which is explained in the next subsection.

2.1. Region of Interest (ROI). Images are cropped before they are scanned for tomatoes. Cropped areas are called ROI, which has been investigated in studies such as [1, 18, 19]. ROI contributes to the extraction of a functional feature by reducing the number of background pixels in the image and positively affects the success of object detection and computational complexity. The most critical point of ROI acquisition is the extraction of color features. These features are used in the training of the NBC model and also in the classification of the pixels to determine the pixel labels as either tomato or background in the image during the pixel detection process to acquire ROI. There are three color features related to red color components used in NBC [1], which are given as

$$c_1 = R - G \quad (8)$$

$$c_2 = R - B \quad (9)$$

$$c_3 = \frac{R}{R+G+B} \quad (10)$$

The first two features show how high or low red is relative to green and blue, respectively. The closer the third feature is to one, the more dominant the red color is compared to the other colors. After color features are calculated for training, threshold value for all three is obtained empirically. In Fig. 3, sorted color features are shown. As can be seen in the graphs, drastic change of the slopes of the curves begins around the sixty thousandth value for each of the three ordered color features and increases to higher values than its normal course. Therefore, an empirical threshold determination was performed by taking into account the slope change points for each feature, and accordingly, the pixels above the thresholds were labeled as tomatoes, and the rest as background.

After training NBC, a binary image where white and black pixels represent tomato and background regions, respectively, are obtained by classifying the pixels of the enhanced RGB image. Then some morphological operations such as erosion and dilation are applied to the binary image to obtain ROI precisely [20]. Gaps

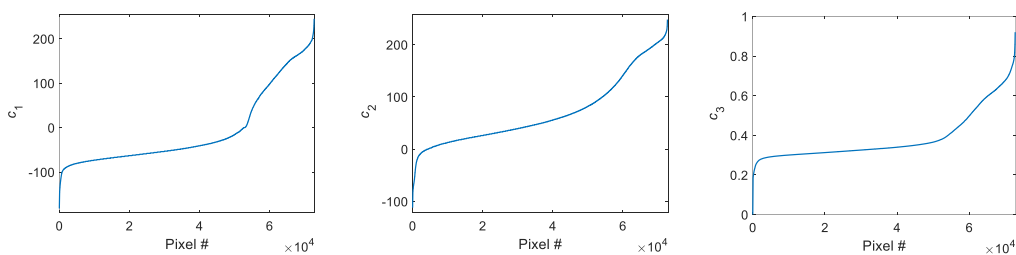
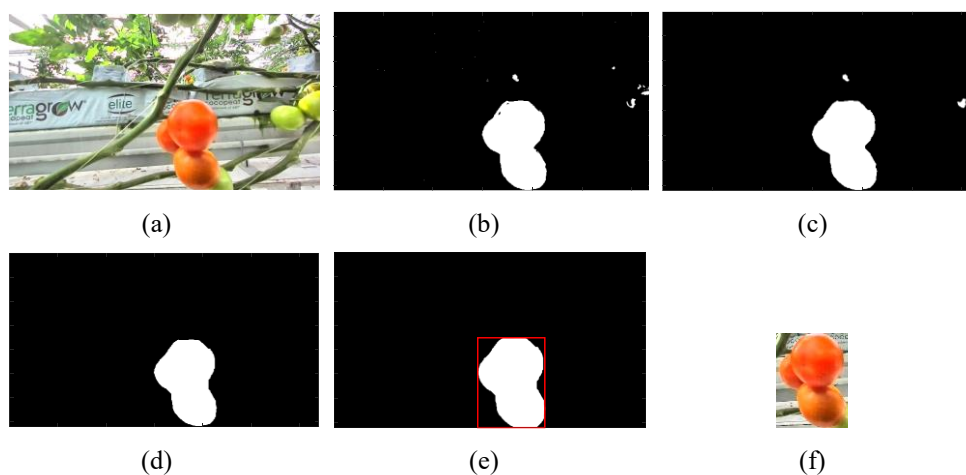
FIGURE 3. Sorted color features, c_1 , c_2 and c_3 .

FIGURE 4. The process of obtaining ROI (a) Enhanced image, (b) Binary image through NBC (c) Filling in large white pixel groups and removing small white pixel groups, (d) Removing white pixel groups other than the largest ones and filling the remaining white pixel groups with small discs, (e) Dilated white pixel group with red borders; (f) Cropped image (ROI).

between and within large groups of white pixels are filled, while small white pixel groups are removed through morphological operations. Finally, the binary image is cropped using the vertical and horizontal endpoints of white pixels, giving ROI. Binary image after pixel classification by NBC and a series of morphological operations to obtain ROI are shown in Fig. 4.

As seen in Fig. 4, NBC classifies pixels as white for tomato and black for background. Then, morphological operations give endpoints to crop the enhanced image. After cropping operation, ROI, which is the search area is obtained. Since it

is mostly smaller than the enhanced image, it reduces the processing time and increases the detection success.

In the next stage of the algorithm, tomatoes are searched in the ROI. For this purpose, a 64×64 sliding window is shifted in the search region with a certain vertical and horizontal step size. HOG and/or LBP features of each sliding window in the ROI are extracted. Then, the features of the sliding windows are classified using SVM or kNN as tomato or background. After the search process is finished, the ROI is downscaled by 10% and the same operations continue for 64×64 sliding window which is shifted by the same step size in the downscaled ROI. This is to ensure that large tomatoes that could not be detected in the previous scale can be detected in the downscaled ROIs, as tomatoes can be of different sizes. The downscaling process of ROI is repeated until its size is close to the size of the sliding window.

Feature extraction and classification are explained in detail in the next section.

3. FEATURE EXTRACTION AND CLASSIFICATION

For humans, detection and classification of objects are simple and effortless acts. However, it is difficult and complicated task for machines and robots to detect an object. In order to overcome this difficulty, classifier algorithm must use features, such as HOG, LBP, Bag of Visual Words (BoVW) and scale-invariant feature transform (SIFT), of the object to be detected. HOG feature extraction method was firstly used in pedestrian detection by Dalal and Triggs [21], and then has been widely used in object detection. HOG is a successful descriptor especially when used with SVM. LBP is also a widely used robust method for describing the texture properties of objects [22, 23]. In the LBP algorithm, the pixels of an image are labeled with decimal numbers called LBP codes, which encode the local structure around each pixel. The values obtained by subtracting each pixel from the eight pixels in its neighborhood are compared with a threshold value, and 1 and 0 are assigned for values greater and less than the threshold, respectively. The opposite is also possible. After the matrix containing 0's and 1's is obtained, the binary bit sequence is created by moving clockwise with the first digit in the upper left. The decimal equivalent of this bit sequence corresponds to the LBP value of the center pixel.

In this paper, besides using the HOG and LBP features separately, we use them together to increase the classification success. HOG features were extracted with 4×8 pixel cells, 2×2 cell blocks and 10 orientation bins. Using the labeled training data, SVM outputs an optimal hyperplane which categorizes the samples [24]. The results of SVM classification are compared with the results of kNN, which is also widely used in vision systems [25]. Generally, it is applied in pattern recognition and

data mining for classification thanks to its easy use and low error rate. It takes less effort to implement according to other classification techniques, and classifies unknown or new data by calculating the distance among existing and new data, and then checking the k -close neighbors, where k is number of neighbors. In our case, uniform distance was used, where all data points in each neighborhood were weighted equally. In addition, since our dataset are not very large, and SVM usually gives better results with fewer outliers than deep learning approach for relatively small data sets, deep learning models and methods are not used for classification.

3.1. SVM Training and Detection Processes. Before detection, SVM classifier is trained for HOG and LBP features separately. Features are extracted from a set of 64×64 pixel-resolution enhanced tomato and background training images to classify whether an image is tomato or not. These features are combined with corresponding labels to train SVM classifier. For classification, 64×64 images are also enhanced before extracting HOG and LBP features.

In the search area specified by the ROI, 64×64 sliding window is shifted with the fixed step size of 16 pixels in both vertical and horizontal directions. At each step, HOG and LBP features of the sliding window are extracted, and classified by SVM. When the sliding window is shifted all over the ROI, the ROI is downscaled with a scaling factor of 1.1. Then, tomatoes are searched in the downscaled ROI with the same processes with 64×64 sliding window. The final downscaled ROI is greater than or equal to 64×64 pixels in size.

4. POSTPROCESSING WITH FCR AND NMS

False and multiple positive detections are eliminated by FCR and NMS, respectively. FCR is used to remove false positive detections using color information [26]. During the search process in ROI, if sliding window is marked as tomato, then the result is checked by FCR whether it is false positive or not. If it is false positive, it is marked as background. For this aim, 64×64 -pixel sling window is binarized using the equation given below [1]:

$$I(x, y) = \begin{cases} 1, & 0.16R(x, y) - 0.093G(x, y) - 0.037B(x, y) - 11.032 \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where $I(x, y)$ is the binarized, namely black or white, pixel of the sliding window image with x and y vertical and horizontal pixel coordinates, respectively. If the equation is greater than or equal to zero, the pixel is classified as 1, which is tomato, otherwise it is classified as 0, which is background. After obtaining the binarized sliding window, the ratio of white pixels to the total number of pixels is calculated.

If it is less than a predetermined threshold, the label of the sliding window is switched to background from tomato [1]. Threshold value is determined empirically. All sliding windows in the original and all downscaled ROIs are classified as tomato or background by SVM or kNN classifier. After removing false positive detections by FCR, there may be more than one true positive results of the same tomato because of the sliding windows and downscaled ROIs. Therefore, NMS is applied as the last step of the process to eliminate positive over-detections [27]. NMS is based on the comparison of overlapping positive detections according to the classifier prediction score value and the selection of the detection with the highest score [28]. Confidence value and overlap threshold are important parameters used in NMS. The intersection over union (IOU) value of any of two sliding windows containing the detected tomatoes is calculated. After comparison with overlap threshold, over-detections of the same tomato are eliminated, leaving a single detection result. Thus, overlapping detections are filtered out, and the sliding window containing the highest score is selected as the only detection that includes the tomato. In this paper, best detection results were achieved when overlap threshold and confidence value are 0.27 and 0.7, respectively.

An example for elimination of unripe tomato detection by FCR and discarding over-detections by NMS is shown in Fig. 5. In both images, green bounding boxes are the results of tomato detections of sliding windows before FCR and NMS. In the left image, there are two tomato detections as the final result shown by red bounding boxes without applying FCR after NMS. As seen in this figure, when FCR is not used, unripe tomato is detected along with the mature one. However, FCR removes the detection of unripe tomato before NMS as shown in the image at the right.



FIGURE 5. Detection results with NMS and without FCR (left) and with FCR (right).

In the next section, test set, experimental results and comparison of the results are given.

5. EXPERIMENTAL RESULTS

In this paper, experiments were performed on NVIDIA Jetson AGX Xavier Developer Kit with Volta GPU w/512 CUDA Cores, 8-Core ARM v8.2 64 Bit CPU, 32 GB 256-Bit LPDDR4 RAM, and Ubuntu 18.04 operating system using Python 3.8 programming language. We used a public greenhouse tomato dataset from [29] with images sized 202×360 pixels in the experiments.

In the first stage of the experiments, tomatoes and background images were cropped in the size of 64×64 pixels from the data set for training. To expand the SVM training set, cropped images were rotated by 90, 180, and 270 degrees, and added to the training set beside other cropped images. Before the training, the cropped images were enhanced in terms of illumination and contrast. Then, HOG and LBP feature extraction were applied to the enhanced images. Finally HOG and LBP features were trained through SVM classification.

After training process, HOG and LBP features of the tomato and background test images were also extracted. Then test images were classified by SVM or kNN using only HOG, only LBP and HOG and LBP features together. To compare the performance of the feature extraction and classification algorithms, four metrics, recall, precision, F_1 and accuracy, were calculated from the confusion matrix. Classification of a 64×64 -sized image may result in four different detections, which are the elements of the confusion matrix as shown in Table 1.

TABLE 1. Confusion matrix.

Real Label	Background	TN	FP
	Tomato	FN	TP
		Background	Tomato
		Predicted Label	

True positive (TP) and true negative (TN) detections represent correct detections of tomato and background, respectively. On the other hand, false positive (FP) and false negative (FN) are undesirable results. FP is the detection of the background as

a tomato. Also, FN means that the detection result is the background, even though the actual image is a tomato.

In the first part of the experiments, confusion matrix results for classification of total of 450 images with the size of 64×64 , 150 of which are tomatoes and 300 of which are backgrounds, are given in Table 2. As seen in this table, the best result was obtained when HOG and LBP features were classified with SVM. In this case, all 300 backgrounds and 147 of 150 tomatoes were predicted correctly. Also, recall, precision, F_1 and accuracy metrics given in Table 3 were calculated from the confusion matrices in Table 2. The worst results were obtained when HOG features were classified by kNN, except the precision result of this approach is among the highest since there is no background misclassification.

TABLE 2. Confusion matrices of feature extraction and classification methods.

Feature	Classifier	
	SVM	kNN
HOG	$\begin{bmatrix} 300 & 0 \\ 5 & 145 \end{bmatrix}$	$\begin{bmatrix} 300 & 0 \\ 21 & 129 \end{bmatrix}$
LBP	$\begin{bmatrix} 298 & 2 \\ 7 & 143 \end{bmatrix}$	$\begin{bmatrix} 300 & 0 \\ 12 & 138 \end{bmatrix}$
HOG+LBP	$\begin{bmatrix} 300 & 0 \\ 3 & 147 \end{bmatrix}$	$\begin{bmatrix} 300 & 0 \\ 19 & 131 \end{bmatrix}$

TABLE 3. Metrics for detection results (%).

Feature	Classifier	Recall	Precision	F_1	Accuracy
HOG	SVM	96.67	100	98.31	98.89
LBP		95.33	98.62	96.95	98.00
HOG+LBP		98.00	100	98.99	99.33
HOG	kNN	86.00	100	92.47	95.33
LBP		92.00	100	95.83	97.33
HOG+LBP		87.33	100	93.24	95.78

In the second part of the experimental studies, tomatoes are searched in images with the size of 202×360 consisting total of 149 tomatoes. The search is carried out by means of 64×64 sliding windows on the ROI cropped from an 202×360 image from the test set. First, HOG and/or LBP feature extraction are performed for each sliding window which is shifted with a step size of 16 from left to right and top to bottom in the original resolution ROI. Then, the features of the sliding windows

in the ROI are classified by SVM or kNN whether they are tomatoes or not. ROI is then downscaled by the factor of 1.1, and the same operations for sliding windows in the reduced sized ROI are repeated. Processes are stopped when the ROI reaches its minimum size. Detection results for 75 images with 149 tomatoes are given in Table 4.

TABLE 4. Detection results for 75 images with 149 tomatoes.

Feature	Classifier	Number of Tomatoes	Tomato Detections	Missed Detections	Over-Detections
HOG	SVM	149	139	17	7
LBP		149	196	10	57
HOG+LBP		149	146	15	12
HOG	kNN	149	128	27	6
LBP		149	233	7	91
HOG+LBP		149	135	23	9

As seen in Table 4, best detection result in terms of missed detections is obtained when LBP features are classified with kNN. However, at the same time LBP with kNN gives the worst result in terms of over-detections. Total tomato detection with these two methods is 233, which is the worst of all. On the other hand, HOG with kNN gives the best result in terms of over-detections. However, missed detections of this combination is the worst of all. When we consider missed and over-detections together, HOG and SVM combination achieves the best result, which is followed by HOG+LBP and SVM combination where missed detections result is better.

An example of tomato detection by classifying HOG and LBP features with SVM and kNN is given in Figure 5 (a) and (b), respectively. As seen in this figure, while all tomatoes were detected with SVM, one of the tomatoes were missed by kNN.



FIGURE 5. An example of the classification of HOG and LBP features by (a) SVM and (b) kNN for tomato detection.

6. CONCLUSION

In this paper, a comparative study was conducted for tomato detection with HOG and LBP feature extraction and SVM and kNN classifier algorithms. Before the detection process, images were enhanced to improve detection success. At the postprocessing stage, false positive results were eliminated by FCR. Finally, NMS was used to select a single detection result out of multiple overlapping detections for a tomato. In the first experiment, the detection success measured by various metrics was investigated for the images with a size of 64×64 pixels. The best result was obtained for the classification of HOG and LBP features by SVM. In the second experiment, where tomatoes were searched in a ROI extracted from full-sized images, the best results were obtained for the classification of HOG features by SVM, which is followed by HOG and LBP features classified by SVM. The kNN classifier, on the other hand, gives different results. In other words, while it gives the best result for missed detections for classifying LBP features, it obtains the worst result for over detections. In addition, while HOG classification by kNN achieves the best result for over detections, it obtains the worst result in terms of missed detections. In general, classification by SVM gives better results than kNN.

Author Contribution Statements The authors equally worked on the study. All authors read and approved the final copy of the manuscript.

Declaration of Competing Interests The authors declare that there is no conflict of interest regarding the publication of manuscript.

Acknowledgement The research was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant No 7201372.

REFERENCES

- [1] Liu, G., Mao, S., Kim, J. H., A mature-tomato detection algorithm using machine learning and color analysis, *Sensors* 2019, 19 (2023), <https://doi.org/10.3390/s19092023>.
- [2] Onishi, Y., Yoshida, T., Kurita, H. et al., An automated fruit harvesting robot by using deep learning, *Robomech J.*, 6 (13) (2019).
- [3] Selvaraj, A., Shebiah, N., Nidhyananthan, S., Ganesan, L., Fruit recognition using color and texture features, *J. Emerg. Trends Comput. Inf. Sci.*, 1 (2010), 90-94.
- [4] Bulanon, D. M., Kataoka, T., Ota, Y., Hiroma, T., AE-Automation and emerging technologies: A segmentation algorithm for the automatic recognition of fuji apples at harvest, *Biosyst. Eng.*, 83 (4) (2002), 405-412.
- [5] Liu, X., Zhao, D., Jia, W., Ji, W., & Sun, Y., A detection method for apple fruits based

- on color and shape features, *IEEE Access*, 7 (2019), 67923–67933.
- [6] Mao, W., Ji, B., Zhan, J., Zhang, X. and Hu, X., Apple location method for the apple harvesting robot, *2009 2nd International Congress on Image and Signal Processing*, (2009),1-5.
- [7] Tanigaki, K., Fujiura, T., Akase, A., Imagawa, J., Cherry-harvesting robot, *Comput. Electron. Agric.*, 63 (1) (2008), 65-72.
- [8] Ji, W., Zhao, D., Cheng, F., Xu, B., Zhang, Y., Wang, J., Automatic recognition vision system guided for apple harvesting robot, *Comput. Elec. Eng.*, 38 (5) (2012), 1186-1195.
- [9] Kurtulmus, F., Lee, W.S., Vardar, A., Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network, *Precision Agric.*, 15 (2014) 57-79.
- [10] Song, Y., Glasbey, C. A., Horgan, G. W., Polder, G., Dieleman, J. A., van der Heijden, G. W. A. M., Automatic fruit recognition and counting from multiple images, *Biosyst. Eng.*, 118 (2014), 203-215, <https://doi.org/10.1016/j.biosystemseng.2013.12.008>.
- [11] Islam, M. A., Yousuf, Md. S. I., Billah, M. M., Automatic plant detection using HOG and LBP features with SVM, *Int. J. Comput.*, 33 (1) (2019), 26-38.
- [12] Hummel, R. A., Image enhancement by histogram transformation, *Comput. Graph. Image Process.*, 6 (1977), 184-195.
- [13] Ketcham, D. J., Lowe, R. W. and Weber, J. W., Real-time image enhancement techniques, *Seminar on Image Processing*, (1976), 1-6.
- [14] Pizer, S. M., Intensity mappings for the display of medical images, *Functional Mapping of Organ Systems and Other Computer Topics*, Society of Nuclear Medicine (1981).
- [15] Pizer, S. M., Amburn, E. P., Austin, J. D., et al., Adaptive histogram equalization and its variations, *Comput. Vis. Graph. Image Process.*, 39 (1987), 355-368.
- [16] Maison, Lestari, T., Luthfi, A., Retinal blood vessel segmentation using gaussian filter, *J. Phys.: Conf. Ser.*, 1376 (2019), 012023, <https://doi.org/10.1088/1742-6596/1376/1/012023>.
- [17] Umri, B. K., Utami, E. and Kurniawan, M. P., Comparative analysis of CLAHE and AHE on application of CNN algorithm in the detection of Covid-19 patients, *2021 4th Int. Conf. on Inf. and Comm. Tech. (ICOIACT)*, (2021), 203-208, <https://doi.org/10.1109/ICOIACT53268.2021.9563980>.
- [18] Wang, Q., Lu, Y., Zhang, X., Hahn, J., Region of interest selection for functional features, *Neurocomputing*, 422 (2021), 235-244.
- [19] Zhang, L., Sun, Q. and Zhang, J., Region of interest extraction via common salient feature analysis and feedback reinforcement strategy for remote sensing images, *GI Science Remote Sens.*, 55 (5) (2018), 745-762.
- [20] Vogt, P., Riitters, K. H., Estreguil, C., Kozak, J., Wade, T. G., Wickham, J. D. , Mapping spatial patterns with morphological, *Image Process.*, 22 (2) (2007), 171-177.
- [21] Dalal, N. and Triggs, B., Histograms of oriented gradients for human detection, *2005*

- IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, (2005), 886-893.
- [22] Tuncer, T. and Avcı, E., Yerel ikili örüntü tabanlı veri gizleme algoritması: LBP-LSB, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 10 (1) (2017) 48-53.
- [23] Chen, J., Kellokumpu, V., Zhao, G., Pietikainen, M., RLBP: Robust Local Binary Pattern, *Proceedings British Machine Vision Conference 2013*, 122 (2014),1-11, <http://dx.doi.org/10.5244/C.27.122>.
- [24] Cortes, C. and Vapnik, V., Support-Vector Networks, *Mach. Learn.*, 20 (1995), 273-297, <http://dx.doi.org/10.1007/BF00994018>.
- [25] Kuang, Q. and Zhao, L., A practical GPU based KNN algorithm, *Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCST '09)*, (2009), 151-155.
- [26] Li, H., Chen, L., Removal of false positive in object detection with contour-based classifiers, *2010 IEEE International Conference on Image Processing*, (2010), 3941-3944.
- [27] Rothe, R., Guillaumin, M., Van Gool, L., Non-maximum suppression for object detection by passing messages between windows, *Computer Vision ACCV 2014*, 9003 (2015), 290-306, https://doi.org/10.1007/978-3-319-16865-4_19.
- [28] Salscheider, N. O., FeatureNMS: Non-maximum suppression by learning feature embeddings, *25th International Conference on Pattern Recognition (ICPR)*, (2020), 7848-7854.
- [29] Liu, G., Mao, S., Open tomatoes dataset, (2019). Available at: <https://github.com/pandalgx/Tomato-dataset>. [Accessed February 2023].