

Received 5 December 2023, accepted 8 January 2024, date of publication 11 January 2024,
date of current version 19 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3352745

RESEARCH ARTICLE

Tomato Quality Classification Based on Transfer Learning Feature Extraction and Machine Learning Algorithm Classifiers

HASSAN SHABANI MPUTU¹, AHMED ABDEL-MAWGOOD²,
ATSUSHI SHIMADA³, (Member, IEEE), AND
MOHAMMED S. SAYED^{1,4}, (Member, IEEE)

¹Department of Electronics and Communications Engineering, Egypt-Japan University of Science and Technology (EJUST), New Borg El-Arab City, Alexandria 21934, Egypt

²Institute of Basic and Applied Sciences, Egypt-Japan University of Science and Technology (EJUST), New Borg El-Arab City, Alexandria 21934, Egypt

³Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan

⁴Department of Electronics and Communications Engineering, Zagazig University, Zagazig 44519, Egypt

Corresponding author: Hassan Shabani Mputu (hassan.mputu@ejust.edu.eg)

This work was supported by the Egypt-Japan University of Science and Technology.

ABSTRACT The demand for high-quality tomatoes to meet consumer and market standards, combined with large-scale production, has necessitated the development of an inline quality grading. Since manual grading is time-consuming, costly, and requires a substantial amount of labor. This study introduces a novel approach for tomato quality sorting and grading. The method leverages pre-trained convolutional neural networks (CNNs) for feature extraction and traditional machine-learning algorithms for classification (hybrid model). The single-board computer NVIDIA Jetson TX1 was used to create a tomato image dataset. Image preprocessing and fine-tuning techniques were applied to enable deep layers to learn and concentrate on complex and significant features. The extracted features were then classified using traditional machine learning algorithms namely: support vector machines (SVM), random forest (RF), and k-nearest neighbors (KNN) classifiers. Among the proposed hybrid models, the CNN-SVM method has outperformed other hybrid approaches, attaining an accuracy of 97.50% in the binary classification of tomatoes as healthy or rejected and 96.67% in the multiclass classification of them as ripe, unripe, or rejected when Inceptionv3 was used as feature extractor. Once another dataset (public dataset) was used, the proposed hybrid model CNN-SVM achieved an accuracy of 97.54% in categorizing tomatoes as ripe, unripe, old, or damaged outperforming other hybrid models when Inceptionv3 was used as a feature extractor. The performance metrics accuracy, recall, precision, specificity, and F1-score of the best-performing proposed hybrid model were evaluated.

INDEX TERMS Grading, feature extraction, machine learning algorithms, tomato, image preprocessing.

I. INTRODUCTION

The most common and familiar vegetable in our daily lives is the tomato, an essential horticultural crop in all regions of the World. Tomatoes play a significant role in the food industry. According to the data from FAOSTAT [1] and TILASTO [2], the worldwide production of tomatoes in 2021 reached 189 million tons. Sorting and grading

these tomatoes is a heavy and tiresome task. Fruits have several qualities, depending on the parameters. The sorting and grading of tomatoes is a crucial process that ensures the quality and safety of the products for consumers. The grading process involves sorting tomatoes into different quality grades based on external features like size, shape, color, ripeness, and defects. Traditional methods used for grading tomatoes include a manual approach by trained experts. Manual grading has several challenges, such as the time-consuming process and the significant amount of labor

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.

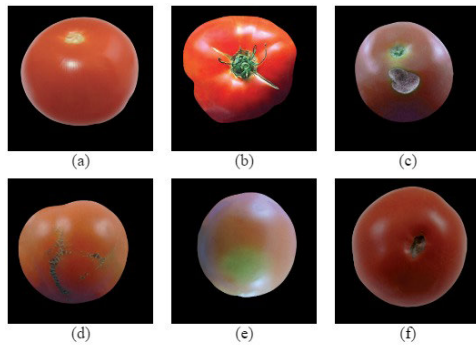


FIGURE 1. Tomato defects: (a) defect-free (b) cracks (c) pest (d) skin cracks (e) sunburn (f) end rot.

required. Thus, each tomato is individually inspected and sorted by a human expert. In most cases, sorting and grading differ from expert to expert, and performance can be affected by fatigue, subjectivity, or even human error. This method is expensive and inefficient in terms of performance because it does not handle large volumes of tomatoes [3].

The Organization for Economic Cooperation and Development (OECD) [4] and the United States Department of Agriculture (USDA) [5] have established standards for tomatoes intended for consumption. These standards focus on tomatoes' quality, classification, and grading for domestic and international markets. They often cover aspects such as size, color, shape, ripeness, and defect tolerance allowed in tomatoes. The aim of establishing standard criteria is to facilitate fair and transparent trade practices among member countries. According to these standards, this affects the fruit's appearance, quality, and marketability. Some of the typical defects include cracks on the tomato's surface, scars caused by wounds or marks, sunburn due to excessive exposure to sunlight, insect damage from pest penetrations, discoloration that impacts the tomato's appearance, blemishes or irregular marks, distortions leading to misshapen tomatoes, bruises resulting from impact or pressure, and blossom end rot with brown. These defects can pose challenges to growers, distributors, and consumers, and adhering to standardized practices is essential to maintain quality and meet market demands. Figure 1 shows some of the tomato defects.

Machine learning has found extensive effectiveness in numerous applications, ranging from quality assessment, detection of diseases in crops, the precise recognition of handwritten digits to facilitate natural language processing, and the accurate identification of audio and speech patterns [6], [7], [8], [9], [10]. Recently, there has been a significant use of machine learning in the food industry, for sorting and grading vegetables or fruits to tackle the challenges of human errors, subjectiveness, labor costs, time-consuming, and increased performance [11], [12], [13]. Within the quality analysis of the tomato, many different approaches were implemented. Many studies have been conducted on developing and implementing machine learning systems

for tomato sorting and grading. Various image processing techniques, such as color-based segmentation, morphological operations, and edge detection, have been used for tomato sorting and grading. Moreover, shallow and deep learning algorithms, such as CNN, SVM, KNN, and RF have been employed to classify tomatoes into different quality grades based on color.

Traditional machine learning methods require manual selection and extraction of features, while Deep Learning, such as CNN, can be hindered by longer training times [14], [15]. The surveyed research demonstrated enhanced performance, but this improvement often comes at the cost of increased system complexity, including longer training periods and larger model memory footprint. Despite the advantages of improved performance, researchers must strike a balance between achieving better results and managing the increased computational demands and resource requirements associated with the more complex models employed in Deep Learning approaches.

There are many studies for fruit and vegetable quality assessment, the following points spotlight the uniqueness of this study:

- 1) This study is uniquely tailored to address the characteristics, attributes, and challenges associated with tomato quality assessment.
- 2) This study proposes a model with the capability to effectively operate in real-world scenarios across various backgrounds as the dataset was collected in uncontrolled environmental conditions with varying light sources.
- 3) This study introduces the hybrid model between CNN and traditional machine learning algorithms for tomato image classification. The CNNs are efficient in feature extraction tasks with minimal computational resources as opposed to traditional machine learning while traditional machine learning requires less training time compared to CNN. In this regard, CNN was deployed for tomato image feature extraction, and traditional machine-learning algorithms were deployed to speed up training significantly and classification. To the best of our knowledge, this is the first model attempting tomato classification using hybrid algorithms (shallow and deep learning).

The rest of this paper is organized as follows: Section II introduces the related work, Section III introduces the framework of the proposed method, and Section IV presents results and discussions. Finally, Section V presents the conclusion and future work.

II. RELATED WORK

Significant efforts have been directed toward developing automated systems for sorting and grading fruits and vegetables, focusing on evaluating external attributes, including shape, size, and color. An algorithm for identifying defects in orange fruits presented in a study [11]. The red, green,

and blue (RGB) and near-infrared (NIR) images captured by a dual-sensor camera were used. The thresholding approach followed by the voting technique was used for color combination for defect detection. The algorithm attained an overall accuracy of 95%. Another study [16] proposed a comparative study on different deep learning algorithms (ResNet, VGGNet, GoogleNet, and AlexNet) for fruit classification. Also, the study utilized the you only look once (YOLO) algorithm to detect the region of interest to increase the performance of fruit grading.

The study in [17], proposed a system that employs image processing techniques for vegetable and fruit quality grading and classification. The system classifies by extracting external features (color and texture). A comparative study of two machine learning algorithms for vegetable and fruit classification was presented in [12]. The SVM achieved a superior accuracy of 94.3% over the KNN. The study in [18] introduced a hybrid model for weed identification in winter rape fields. The study compared five models, the hybrid model between VGGNet with SVM achieved a higher accuracy of 92.1% than others. The study in [19] performed similar work to [18] by adding the residual filter network for feature enhancement into the hybrid network (CNN-SVM), with the proposed model attaining an overall accuracy of 99% against others.

Classification of appearance quality in red grapes using transfer learning with convolutional neural networks was introduced in [20]. The investigation employed the transfer learning technique with four pre-trained networks, namely VGG19, Inceptionv3, GoogleNet, and ResNet50, to achieve its objectives. Notably, ResNet50 demonstrated the highest accuracy, reaching 82.85% in the precise categorization of red grapes into three distinct quality categories. To further enhance performance, the research employed ResNet50 for feature extraction and SVM for classification. The ResNet50-SVM model excelled, achieving the highest accuracy of 95.08% in effectively categorizing red grapes into the aforementioned categories.

In a study conducted by [13] within the domain of fruit quality assessment, a comparative analysis was undertaken involving CNN and Vision Transformers (ViT). The findings of the study revealed that the CNN model demonstrated superior performance compared to the ViT model. Specifically, the CNN model achieved a remarkable accuracy of 95% in classifying apple fruits into four distinct classes and banana fruits into two classes, surpassing the accuracy of 93% attained by the ViT model. The study in [21] performed similar work to [13] for olive disease classification, employing CNN and ViT for feature extraction purposes and utilizing SoftMax as a classifier. The study achieved notable results by employing a fusion of ViT and VGG-16 for feature extraction, attaining an average accuracy of 97% for binary and multiclass classification. However, according to [22], ViT requires a large training dataset of more than 14 million images to outperform CNN models like ResNet's.

A system for tomato grading based on machine vision was presented in [23]. The system attained an average accuracy of 95% for detecting calyx and stalk from the tomatoes and an accuracy of 97% in classifying tomatoes into healthy and defective classes, with radial basis function-support vector machines (RBF-SVM) outperforming other classifiers. In this study, it was observed that the training set was also used for testing. In [24], the study presented an advanced tomato detection algorithm that utilizes enhanced hue saturation and value (HSV) color space and watershed techniques for efficient fruit separation. The algorithm achieved an overall accuracy of 81.6% in red fruit detection.

The study in [15] proposed tomato ripeness detection and classification using VGG-based CNN models. The proposed model involved transfer learning and fine-tuning techniques of VGG-16 to classify tomatoes into ripe and unripe classes. The average accuracy of 96% was attained by the model. The study in [25] introduced an automated grading system designed for the assessment of tomato ripeness through the application of deep learning methodologies. The investigation leveraged transfer learning with Resnet18, achieving an impressive average validation accuracy of 93.85% in the precise categorization of tomatoes into ripe, under-ripe, and over-ripe.

In [26], the study suggested a system for classifying tomatoes into three categories based on maturity: immature, partially mature, and mature. The system was accomplished by applying deep transfer learning while utilizing five pre-trained models. Among these models, VGG-19 demonstrated the highest accuracy of 97.37% compared to the others. A tomato classification system based on size was presented in [27] with a system utilizing thresholding, machine learning, and deep learning techniques based on area, perimeter, and enclosed circle radius features. The machine learning techniques showed the best performance, with an average accuracy of 94.5% achieved by the SVM. In [28], the study developed an automatic tomato detection method. The SVM algorithm was used as a classifier and achieved an average accuracy of 90%, outperforming others. A computer vision-based grading and Sorting system of tomatoes into defected and non-defected was presented in [29]. The task was performed using a backpropagation neural network algorithm, attaining an accuracy of 92%.

III. MATERIALS AND METHODS

In this section, we present the proposed framework for tomato classification, as shown in Figure 2. The proposed system includes five major sections: image acquisition, preprocessing, transfer learning, feature extraction, and classifier.

A. IMAGE ACQUISITION

The images were acquired in an uncontrolled lighting environment, as illustrated in Figure 4, which maintained a fixed distance of 20 cm between the table surface and

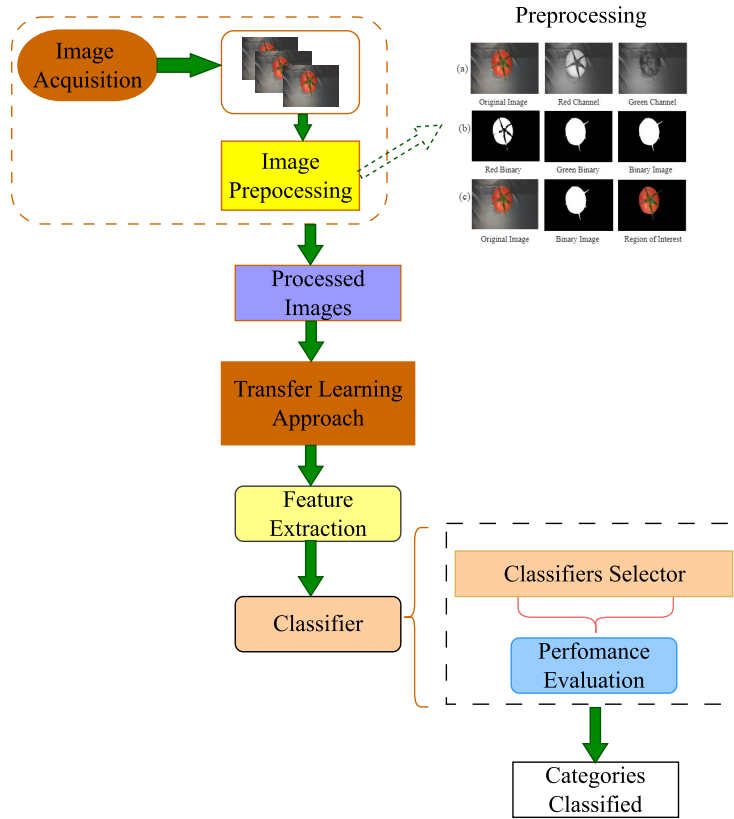


FIGURE 2. An overview of the proposed system for tomato classification.

the camera. The acquisition system employed a single-board computer NVIDIA Jetson Tx1 [30] with an onboard camera equipped with an RGB sensor capable of capturing images in JPEG format at 640×480 pixels resolution. The system operated at 30 frames per second, capturing tomato images one at a time. The sum of 600 tomatoes with different shapes, sizes, colors, and defects collected from the local market was washed and dried. The image acquisition involved capturing four images (four sides of the tomato i.e., top, bottom, rear, and front as shown in Figure 3) from each tomato, resulting in a total of 2400 images. The idea of capturing four images from each tomato is to mimic the concept of scanning a tomato rolling in the rotating mechanical support like conveyor belts or rollers as described in [31]. The images were then categorized following OECD [4] and USDA [5] standards as described in Table 1. The datasets are made public and are available at [32].

B. IMAGE PREPROCESSING

The captured tomato images have non-uniform illumination due to the light’s reflection, which leads to contrast variation. These variations of the contrast in the images may impact the performance of the classification model and increase computational complexity. To overcome this problem, we performed segmentation and background cancellation to get the region

TABLE 1. Distribution of the tomato’s dataset.

Batch One		Batch Two	
Healthy	1600	Ripe	800
		Unripe	800
Reject	800	Reject	800
Total	2400		2400

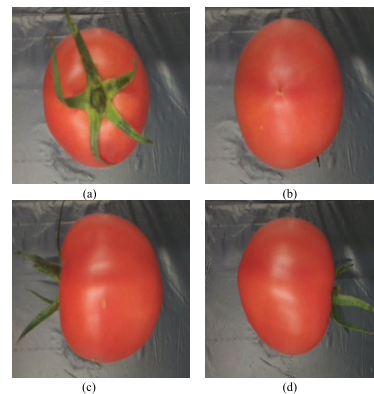


FIGURE 3. Sides of tomato: (a) top (b) bottom (c) front (d) rear.

of interest (ROI) before training our dataset. Red and green channels of the image were extracted, followed by simple

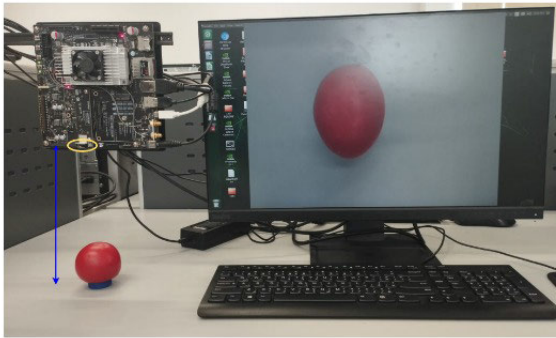


FIGURE 4. Setup of the image acquisition.

thresholding using the Otsu method [33] to obtain two separate binary images (masks).

In the Otsu method, the threshold value for image binarization is determined by maximizing the between-class variance. First, the histogram of the grayscale or individual channel of the image is computed, representing the distribution of pixel intensities. The histogram is then normalized, and cumulative probabilities and means are calculated. The global mean, representing the average intensity of the entire image, is obtained. Next, for each possible threshold value, the between-class variance is computed by evaluating the separation between two classes: the class below the threshold and the class above it. The threshold that maximizes this between-class variance is identified as the optimal threshold for image binarization. By effectively maximizing the separation between foreground and background pixel intensities, the Otsu method provides an automated and robust way to determine the threshold value for accurate image segmentation and object extraction. A simplified mathematical representation of the Otsu thresholding is shown in equation (1).

$$B(x, y) = \begin{cases} 1, & p(x, y) > thr \\ 0, & \text{Otherwise} \end{cases} \quad (1)$$

where, $B(x, y)$, $p(x, y)$, thr , x , y represent binary image, image pixel, threshold pixel value, and the coordinates of the pixel respectively.

The “OR” logical operator fused the generated masks, followed by morphological operations to remove gaps and holes to get the final mask that includes both color features. In addition, we performed background cancellation by multiplying the original image with its respective binary image (mask), which helps identify specific ROI. Thus, the images are taken for training. Figure 5 demonstrates the preprocessing techniques investigated in our work for the case of the tomato image with a plain background and Figure 6 demonstrates the capability of the preprocessing techniques to segment the tomato image with a complex background of rollers similar to the rotating mechanical support in an industrial setup.

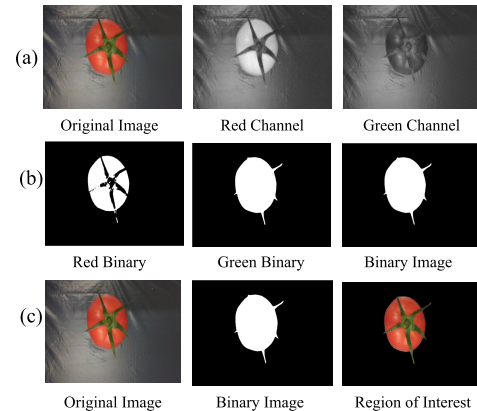


FIGURE 5. Image preprocessing: (a) channels extraction (b) thresholding and binary logical combination (c) background cancellation.

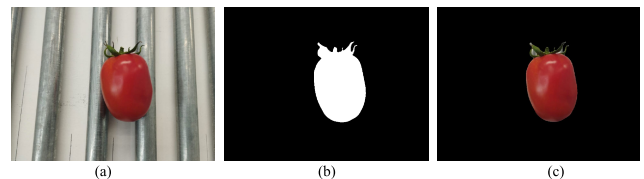


FIGURE 6. Image preprocessing: (a) original image (b) binary image (c) region-of-interest.

C. DATA AUGMENTATION

Data augmentation is a machine learning and computer vision technique to increase the training data available for a model to learn from [34]. It involves applying various transformations to the existing data to create new, slightly altered versions of the original data. Data augmentation aims to improve the generalization and robustness of machine learning models by exposing them to a more extensive and diverse set of examples during training. By introducing variations in the training data, the model can learn to be more flexible and adaptable and handle new and unseen inputs during inference. We augmented our training set by applying rotation, reflection, translation, and scaling.

D. TRANSFER LEARNING

Transfer learning [35] addresses the challenges associated with training deep learning networks when the amount of available data is limited. Instead of starting from scratch, transfer learning uses pre-trained deep learning networks customized for the specific task. The pre-trained network can be utilized as a feature extractor or for end-to-end classification tasks by carefully adjusting some parameters. This study used a pre-trained network as a feature extractor and classify the extracted features using traditional machine learning classifiers. We evaluated four pre-trained networks, namely MobileNetv2 [36], Inceptionv3 [37], ResNet50 [38], and AlexNet [39], for transfer learning in our proposed approach. We analyzed the performance of these networks

in terms of accuracy before deploying them for feature extraction. We then proceed with fine-tuning the selected pre-trained networks. Fine-tuning aims to teach the network to recognize classes that were not trained before. It involves removing the last convolutional and classification layers of the adopted pre-trained network to match the number of classes in the building classification challenge. To prevent overfitting, We froze some network layers, apply batch normalization, and dropout during training.

E. FEATURE EXTRACTION

In our proposed system, the selected pre-trained networks were used to extract the deep features from the last convolutional layers, or dense layers of the network [40]. Usually, these networks are designed with more convolutional layers to increase network performance. A set of weight layers cascade with another layer separated by the activation layer such as a rectified linear unit (ReLU). Thus, features were extracted from the deepest layer of the network and used to train traditional machine learning classifiers.

F. CLASSIFIERS

Machine learning classifiers [41] are algorithms trained on labeled datasets to learn patterns and relationships between input features and corresponding output labels. The goal of a machine learning classifier is to predict the correct label for a new input based on the relationships learned from the training dataset. Machine learning classifiers include decision trees, random forests, logistic regression, support vector machines, k-nearest neighbors, and neural networks. The choice of a machine learning classifier depends on the specific characteristics of the dataset and the desired performance metrics. A labeled dataset is typically divided into training and validation sets to train a machine learning classifier. The training set is used to train the classifier, and the validation set is used to evaluate the classifier's performance on new, unseen data. The classifier's performance is measured using accuracy, precision, recall, specificity, and F1 score metrics. This study uses support vector machines, random forests, k-nearest neighbor's classifiers, and principal component analysis as features dimensionality reduction.

1) SUPPORT VECTOR MACHINE – SVM

The support vector machine [28] is highly effective in image classification and regression tasks because it can handle dimensionality issues. This algorithm employs support vectors for determining the coordinates of individual observations. It can produce accurate outcomes even in high-dimensional spaces, such as when the number of dimensions exceeds the number of samples.

2) RANDOM FOREST - RF

The random forest classifier is a popular machine learning algorithm that belongs to the ensemble learning family of methods [41]. It is widely used for classification and

regression tasks in which the goal is to make accurate predictions. The random forest algorithm builds an ensemble of decision trees by randomly selecting a subset of features and data samples to train each tree. During training, the trees learn to classify the input based on splitting rules. When making a prediction, the input is passed through each tree, and the final prediction is determined by aggregating the individual tree predictions, often using majority voting. random forest is known for its ability to handle high-dimensional data, provide accurate predictions, and identify important features for the task at hand.

3) K-NEAREST NEIGHBOR -KNN

The K-Nearest Neighbors (KNN) is a machine learning algorithm for classification and regression tasks. A simple but effective algorithm finds the k-nearest data points in the dataset to a new data point. Then it assigns the majority class label among those KNN as the predicted label for the new data point [42]. KNN works well with linear and non-linear decision boundaries and can be applied to various problem domains such as image recognition, text classification, and recommendation systems. However, KNN can be sensitive to the choice of K and the distance metric used and may be computationally expensive for large datasets. The distance metric used to measure the similarity between data points can vary depending on the problem domain. KNN may not handle large datasets such as image classification, so it is recommended to use principal component analysis [43] for feature dimensionality reduction before utilizing the KNN classifier.

IV. RESULTS AND DISCUSSION

A. EXPERIMENTAL SETUP

The training and testing of the proposed model has been performed on 128GB of RAM, NVIDIA GeForce RTX 2080 Titan with 11GB of RAM, and an Intel (R) Xeon (R) Silver 4114 CPU 2.20 GHz PC using the MATLAB R2022b release. Our experiment deployed four pre-trained networks, namely MobileNetv2 [36], Inceptionv3 [37], ResNet50 [38], and Alex-Net [39], for feature extractions. These features were then classified using three traditional machine learning classifiers namely SVM, RF, and KNN. Also, a dataset of 2400 tomato images was used in our experiments, with the first batch having healthy and reject classes and the second batch having ripe, unripe, and reject classes, as described in Table 1. The dataset was randomly divided into 70% (1680 images) for training, 10% (240 images) for validation, and 20% (480 images) for testing. The standard hyperparameters used in this experiment were 0.005 learning rate, 32 minibatch sizes, and 60 epochs.

B. DISCUSSION

To evaluate the performance of our proposed model, we performed four different experiments using our dataset and one experiment using a public dataset available on [44] as

described in experiments 1 to 5. The performance was evaluated based on training time, testing time, accuracy, recall, precision, specificity, and F-1 score calculated according to equation (2) to (6).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Recall(R) = \frac{TP}{TP + FN} \quad (3)$$

$$Precision(P) = \frac{TP}{TP + FP} \quad (4)$$

$$Specificity = \frac{TN}{TN + FP} \quad (5)$$

$$F1 - score = 2 \left(\frac{RP}{R + P} \right) \quad (6)$$

where TP, TN, FP, and FN represent true positive, true negative, false Positive, and false negative, respectively.

1) EXPERIMENT 1: PERFORMANCE OF THE PROPOSED MODEL USING THE BINARY DATASET

Table 2 summarizes the performance of the Transfer Learning, CNN-SVM, CNN-RF, and CNN-KNN models on the binary classification (healthy and reject) regarding accuracy with corresponding training and testing times. In Table 2, first performed transfer learning to evaluate the performance of our selected pre-trained networks, where an average accuracy of 95.52% has been attained for classifying tomatoes into binary classes. The performance of transfer learning gives a starting point for determining whether a network can be deployed for feature extraction. Among the proposed hybrid models, CNN-SVM attained the highest accuracy of 97.50% when Inceptionv3 was used as a feature extractor. Furthermore, on average the training time of 4.45 seconds for 1680 images, and the testing time averaged 2.43 milliseconds per image for the proposed hybrid model CNN-SVM compared to other models.

2) EXPERIMENT 2: PERFORMANCE OF THE PROPOSED MODEL USING THE MULTICLASS DATASET

We evaluated our model using batch two datasets (the multiclass), as shown in Table 1. The model's performance is summarized in Table 3, where transfer learning was first performed to evaluate the performance of our selected pre-trained networks for feature extraction, where an average accuracy of 94.95% has been obtained for multiclass classification. Among the proposed hybrid models, the highest accuracy of 96.67% was attained by CNN-SVM when Inceptionv3 was used as a feature extractor. On average the training time of 4.54 seconds for 1680 images and the testing time of 2.44 milliseconds per image was recorded for the proposed CNN-SVM model compared to other models.

3) EXPERIMENT 3: PERFORMANCE OF THE PROPOSED MODEL IN LIGHTWEIGHT CNN ON THE BINARY DATASET

We evaluated our proposed model in lightweight pre-trained networks, i.e., ShuffleNet (5.4MB) [45], MobileNetv2

(20MB) [36], and EfficientNetB0 (13MB) [46], using the binary datasets (healthy and reject). The model's performance is summarized in Table 4, where an average accuracy of 92.12% has been obtained for transfer learning as we evaluate the feature extractor. The highest accuracy of 95.21% was achieved by CNN-SVM when MobileNetv2 was used as a feature extractor. On average the training time of 5.68 seconds for 1680 images and the testing time of 3.28 milliseconds per image was recorded for the proposed hybrid model CNN-SVM compared to others.

4) EXPERIMENT 4: PERFORMANCE OF THE PROPOSED MODEL IN LIGHTWEIGHT CNN USING THE MULTICLASS DATASET

Also, we evaluated our proposed model in lightweight pre-trained models using the multiclass dataset (ripe, unripe, and reject). The model's performance is summarized in Table 5, where an average accuracy of 93.13% has been obtained for transfer learning for feature extractor evaluation. Among the proposed hybrid models, CNN-SVM attained the highest accuracy of 94.37% when MobileNetv2 was used as a feature extractor. On average the training time of 5.72 seconds for 1680 images and the testing time of 3.30 milliseconds per image was recorded for the proposed hybrid model CNN-SVM compared to others.

5) EXPERIMENT 5: PERFORMANCE OF THE PROPOSED MODEL USING THE PUBLIC DATASET

We evaluated our model using the public dataset, which has four classes, i.e., ripe, unripe, old, and damaged, available on [44]. The model's performance is summarized in Table 6, where an average of 95.38% accuracy has been obtained for transfer learning for feature extraction evaluation. Among the proposed hybrid models, CNN-SVM attained the highest accuracy of 97.54% when Inceptionv3 was used as a feature extractor. Furthermore, on average the training time of 4.20 seconds for 1423 images, and the testing time of 2.68 milliseconds per image was recorded for the proposed hybrid model CNN-SVM compared to others. However, the performance of our proposed model in the public dataset is slightly higher compared to our dataset, this is because the inter-class color feature in the public dataset is large compared to our dataset as shown in Figure 12.

C. PERFORMANCE METRICS OF THE PROPOSED MODEL

Figures 7 to 11 show the confusion matrices of the best-performing proposed model in each experiment. From each confusion matrix, we computed respective standard performance metrics i.e., recall, precision, specificity, and F-1 score calculated according to equations (3) to (6) as summarized in Table 7.

D. IMAGE ANALYSIS AND DESIGN OPTIMIZATION

Within the scope of our investigation, the examination of image features involved aspects related to both appearance and resolution within the dataset. Our analysis revealed that

TABLE 2. Performance of the proposed method on the binary datasets.

Proposed Models	Performance	Feature Extractor Networks			
		AlexNet	Resnet50	MobileNetV2	InceptionV3
Transfer Learning	Accuracy (%)	93.7500	95.6250	95.4167	97.2917
	Training Time(sec)	368.3	1563.1	1779.2	3376.1
	Testing Time (sec)	0.8035	0.8955	1.3564	2.7081
CNN-SVM	Accuracy (%)	94.3750	95.4167	95.2083	97.5000
	Training Time(sec)	3.4701	4.4316	4.1569	5.7310
	Testing Time (sec)	0.8325	1.2346	1.0443	1.5559
CNN-RF	Accuracy (%)	91.4583	95.0000	95.2083	96.4583
	Training Time(sec)	3.6949	4.5802	4.3089	5.8760
	Testing Time (sec)	0.9148	1.3034	1.1186	1.6327
CNN-KNN	Accuracy (%)	93.1250	93.7500	93.3333	95.6250
	Training Time(sec)	3.4415	4.4015	4.1012	5.7068
	Testing Time (sec)	0.8318	1.2300	1.0434	1.5573

TABLE 3. Performance of the proposed method using multiclass datasets.

Proposed Models	Performance	Feature Extractor Networks			
		AlexNet	Resnet50	MobileNetV2	InceptionV3
Transfer Learning	Accuracy (%)	92.7083	96.2500	94.7917	96.0417
	Training Time(sec)	369.7	1551.5	1695.1	3484.7
	Testing Time (sec)	0.8516	0.8868	1.3335	1.9373
CNN-SVM	Accuracy (%)	91.6667	96.2500	94.3750	96.6667
	Training Time(sec)	3.7295	4.3227	4.0638	6.0280
	Testing Time (sec)	0.8659	1.1094	1.1102	1.5935
CNN-RF	Accuracy (%)	92.9167	95.6250	92.2917	96.2500
	Training Time(sec)	3.9512	4.4345	4.2100	6.1593
	Testing Time (sec)	0.9459	1.1784	1.1847	1.6688
CNN-KNN	Accuracy (%)	84.1667	80.6250	68.3333	69.3750
	Training Time(sec)	3.6597	4.2348	3.9851	5.9484
	Testing Time (sec)	0.8642	1.1081	1.1087	1.5925

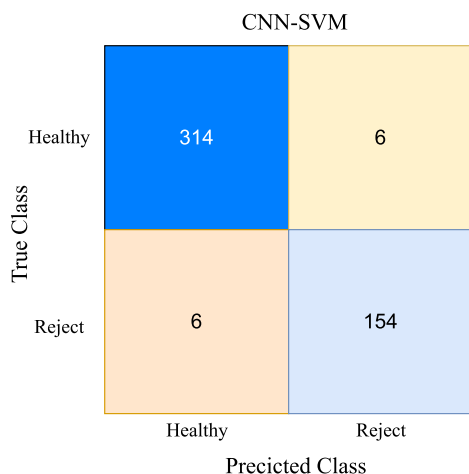


FIGURE 7. Best performing proposed model confusion matrix for experiment 1.

Inceptionv3 outperforms Resnet50 when handling images with closely aligned appearance characteristics, such as

those present in our dataset, which exhibits inter-class color features that are in proximity. Conversely, Resnet50 excels with images characterized by slightly broader inter-class color features, as found in public datasets. Notably, Figure 12 provides visual representations of inter-class color features from both our dataset and the public dataset.

The resolution of our dataset stands at 640×480 pixels, whereas the public dataset is at 256×256 pixels. Our analysis further demonstrates that Inceptionv3 attains superior performance with our dataset due to its input resolution (299×299 pixels) closely aligning with our dataset’s dimensions [47], in contrast to ResNet50 (224×224 pixels). Conversely, ResNet50 performs marginally better than Inceptionv3 with the public dataset, as its input resolution closely approximates the resolution of the public dataset [48].

In the process of model optimization, we conducted feature extraction from various layers, observing a consistent improvement in classification accuracy with the

TABLE 4. Performance of proposed model in lightweight CNN using the binary datasets.

Proposed Models	Performance	Feature Extractor Networks		
		ShuffleNet	MobileNetV2	EfficientNetB0
Transfer Learning	Accuracy (%)	92.5833	95.0167	93.1667
	Training Time(sec)	1256.7	1779.2	5526.1
	Testing Time (sec)	0.7017	1.3564	2.4128
CNN-SVM	Accuracy (%)	92.5833	95.2083	94.1250
	Training Time(sec)	4.2963	4.1569	8.5849
	Testing Time (sec)	1.1695	1.0443	2.5099
CNN-RF	Accuracy (%)	92.5000	95.2083	90.6250
	Training Time(sec)	4.4779	4.3089	8.8104
	Testing Time (sec)	1.2398	1.1186	2.5919
CNN-KNN	Accuracy (%)	93.5417	93.3333	88.9583
	Training Time(sec)	4.2736	4.1012	8.5617
	Testing Time (sec)	1.1693	1.0434	2.5098

TABLE 5. Performance of proposed model in lightweight CNN on the multiclass datasets.

Proposed Models	Performance	Feature Extractor Networks		
		ShuffleNet	MobileNetV2	EfficientNetB0
Transfer Learning	Accuracy (%)	94.5833	94.7917	90.0000
	Training Time(sec)	1263.6	1695.1	5576.6
	Testing Time (sec)	0.7110	1.3335	2.1346
CNN-SVM	Accuracy (%)	94.1917	94.3750	90.0000
	Training Time(sec)	4.6281	4.0638	8.4685
	Testing Time (sec)	1.2253	1.1102	2.4164
CNN-RF	Accuracy (%)	93.	92.2917	89.1667
	Training Time(sec)	4.8151	4.2100	8.6797
	Testing Time (sec)	1.2992	1.1847	2.4970
CNN-KNN	Accuracy (%)	68.9583	68.3333	67.5000
	Training Time(sec)	4.5592	3.9851	8.4019
	Testing Time (sec)	1.2245	1.1087	2.4152

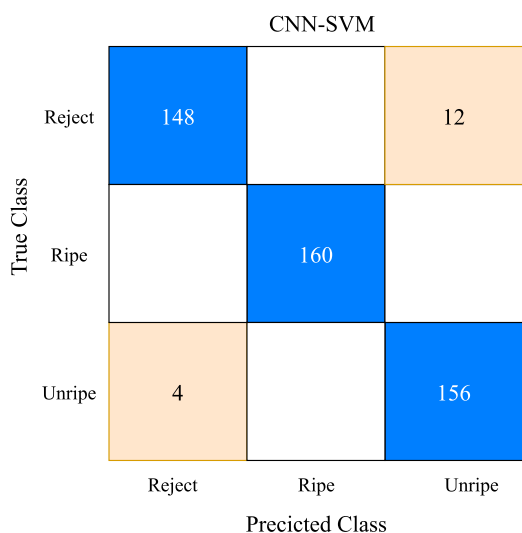


FIGURE 8. Best performing proposed model confusion matrix for experiment 2.

utilization of progressively deeper layers used for feature extraction [49]. Additionally, significant improvements in

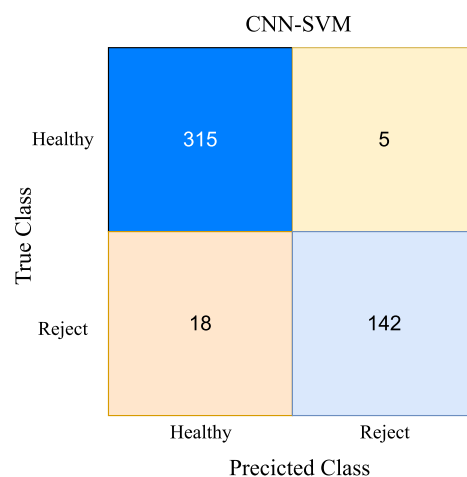


FIGURE 9. Best performing proposed model confusion matrix for experiment 3.

classification accuracy were achieved by employing classical machine learning classifiers in the hybrid model, as opposed to the softmax classifier used during transfer learning [50].

TABLE 6. Performance of the proposed model on the public datasets.

Proposed Models	Performance	Feature Extractor Networks			
		AlexNet	Resnet50	MobileNetV2	InceptionV3
Transfer Learning	Accuracy (%)	93.1034	97.0443	93.3498	98.0296
	Training Time(sec)	326.3	4725.7	1316.3	2838.0
	Testing Time (sec)	0.9996	0.7801	1.1975	1.9672
CNN-SVM	Accuracy (%)	91.6256	97.5369	93.1034	97.5369
	Training Time(sec)	3.3623	4.1313	3.6140	5.7025
	Testing Time (sec)	0.8030	1.0849	0.9550	1.5042
CNN-RF	Accuracy (%)	91.3793	96.7980	93.1034	97.0443
	Training Time(sec)	3.4553	4.2124	3.7086	5.7983
	Testing Time (sec)	0.8822	1.1614	1.0396	1.5805
CNN-KNN	Accuracy (%)	61.0837	73.1527	63.7931	80.7882
	Training Time(sec)	3.2160	4.0332	3.5227	5.6206
	Testing Time (sec)	0.7985	1.0826	0.9546	1.5022

TABLE 7. Performance metrics of the best-performing hybrid.

Experiment	Recall (%)	Precision (%)	Specificity (%)	F1-score (%)
1	98.12	98.12	96.25	98.12
2	96.67	96.74	98.28	96.66
3	98.44	94.59	96.60	96.48
4	94.38	94.43	97.07	94.38
5	98.28	98.34	99.43	98.27

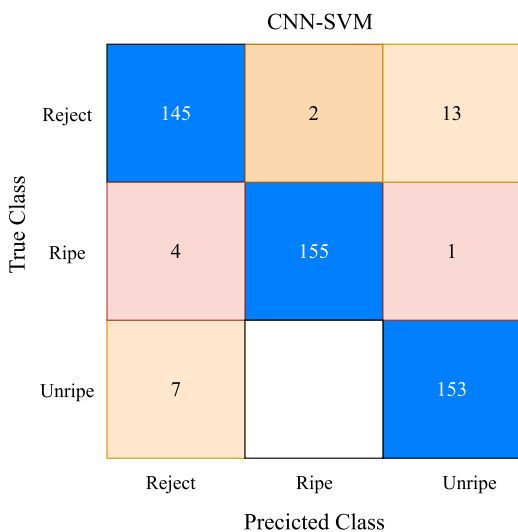


FIGURE 10. Best performing proposed model confusion matrix for experiment 4.

E. LIMITATIONS OF THE STUDY

While our model demonstrates superior performance, it is important to acknowledge the following remarkable limitations.

- 1) The performance of our model will be affected by the minimal inter-class color feature difference.

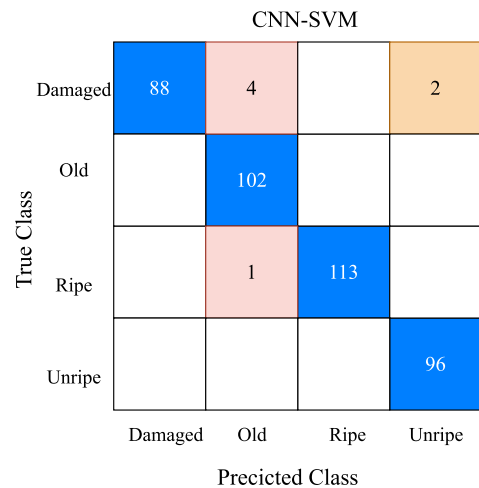


FIGURE 11. Best performing proposed model confusion matrix for experiment 5.

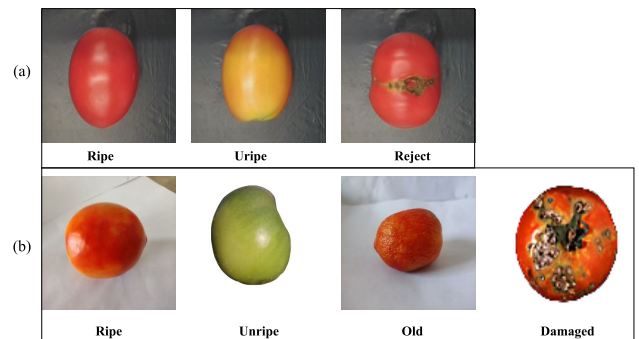


FIGURE 12. Dataset classes: (a) Our dataset (b) public dataset.

- 2) The performance of our model will be poor when images of low intensity are used.

F. PERFORMANCE COMPARISON

We investigated the state-of-the-art (SOTA) works [20], [25], [26] performed in the field of fruit and vegetable

TABLE 8. Performance comparison of our proposed model with SOTA.

Model	Accuracy of the models in our dataset (%)	
	Binary class	Multiclass
ResNet50-SVM [20]	95.42	94.25
Resnet18 [25]	93.75	92.50
VGG19 [26]	94.17	93.13
Proposed model (Inceptionv3-SVM)	97.50	96.67

quality assessment. We deployed their proposed model to our dataset [32] for binary and multiclass classification. Table 8 summarises the performance comparison of our proposed model with SOTA in terms of accuracy. From Table 8, it was observed that the proposed model performs better compared to SOTA.

V. CONCLUSION

We presented a new approach for tomato quality grading based on the external image features. The proposed approach utilized the pre-trained networks for feature extraction and traditional machine learning algorithms as the classifiers (hybrid model). We took advantage of fine-tuning techniques in the pre-trained networks to make the networks suitable for the deep layers to learn and concentrate on the complex and significant features of the tomato images. Also, we analyzed and performed various image preprocessing techniques for feature enhancement and performance improvement in our proposed method. Features obtained from these networks are then classified by SVM, RF, and KNN classifiers. We vividly demonstrated the performance of various fine-tuned networks and the highest accuracy attained by Inceptionv3. Later, Inceptionv3 was considered further for feature extraction and utilized in our classifiers.

Among the proposed hybrid models, the CNN-SVM method outperformed others with an accuracy of 97.50% in the binary classification of tomatoes into healthy and reject and an accuracy of 96.67% in the multiclass classification of tomatoes into ripe, unripe, and reject when Inceptionv3 was used as a feature extractor. The hybrid CNN-SVM method outperformed other models by achieving an accuracy of 97.54% with Inceptionv3 as a feature extractor once deployed in a public dataset to categorize tomatoes into ripe, unripe, old, and damaged. However, as compared to the public dataset, the classification accuracy of the proposed model in our dataset could potentially be improved if the difference in color characteristics between classes in the dataset were slightly higher. The investigation revealed that the proposed model outperforms the SOTA. Furthermore, the proposed model can operate in different backgrounds of varying light sources.

This study is subject to certain limitations, notably the potential for overfitting when training a model with a dataset containing images, particularly those with low intensity and minimal inter-class color features. In the future study, we plan to extend the application of our model to various crops and

items while concurrently employing an in-depth analysis of image feature characteristics as a complementary approach, aimed at deriving sound and comparable conclusions, thereby enhancing the robustness of our findings beyond the post-training assessment method.

Furthermore, we will investigate optimization techniques for model size reduction without hurting accuracy to be suitable for hardware implementation and deploy it for real-time inference. For model deployment, the system may require multiple cameras or tomato rotating mechanical support to facilitate capturing images at different angles.

ACKNOWLEDGMENT

The authors would like to thank the Egypt-Japan University of Science and Technology (E-JUST) and the TICAD7 scholarship program for graciously providing them with the essential research facilities that were instrumental in the successful execution of their study.

CONFLICT OF INTEREST

The authors have no conflict of interest.

REFERENCES

- [1] FAO. (2021). *Tomatoes Production*. [Online]. Available: <https://www.fao.org/faostat/en/#data/QCL/visualize>
- [2] Tilasto. (2021). *World: Tomatoes, Production Quantity*. [Online]. Available: <https://www.tilasto.com/en/topic/geography-and-agriculture/crop/tomatoes/tomatoes-production-quantity/world?countries=World>
- [3] R. Nithya, B. Santhi, R. Manikandan, M. Rahimi, and A. H. Gandomi, "Computer vision system for mango fruit defect detection using deep convolutional neural network," *Foods*, vol. 11, no. 21, p. 3483, Nov. 2022.
- [4] "International standards for fruit and vegetables-tomatoes," OECD, Org. Econ. Cooperation, Paris, France, Tech. Rep., 2019.
- [5] "United states consumer standards for fresh tomatoes," USDA, United States Dept. Agricult., Washington, DC, USA, Tech. Rep., 2018.
- [6] M. M. Khodier, S. M. Ahmed, and M. S. Sayed, "Complex pattern jacquard fabrics defect detection using convolutional neural networks and multispectral imaging," *IEEE Access*, vol. 10, pp. 10653–10660, 2022.
- [7] J. Amin, M. A. Anjum, R. Zahra, M. I. Sharif, S. Kadry, and L. Sevcik, "Pest localization using YOLOv5 and classification based on quantum convolutional network," *Agriculture*, vol. 13, no. 3, p. 662, Mar. 2023.
- [8] S. R. Shah, S. Qadri, H. Bibi, S. M. W. Shah, M. I. Sharif, and F. Marinello, "Comparing inception v3, VGG 16, VGG 19, CNN, and ResNet 50: A case study on early detection of a Rice disease," *Agronomy*, vol. 13, no. 6, p. 1633, Jun. 2023.
- [9] S. Ahlawat and A. Choudhary, "Hybrid CNN-SVM classifier for hand-written digit recognition," *Proc. Comput. Sci.*, vol. 167, pp. 2554–2560, Jan. 2020.
- [10] A. Copiaco, C. Ritz, S. Fasciani, and N. Abdulaziz, "Scalogram neural network activations with machine learning for domestic multi-channel audio classification," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2019, pp. 1–6.
- [11] A. M. Abdelsalam and M. S. Sayed, "Real-time defects detection system for orange citrus fruits using multi-spectral imaging," in *Proc. IEEE 59th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Oct. 2016, pp. 1–4.
- [12] N. K. Bahia, R. Rani, A. Kamboj, and D. Kakkar, "Hybrid feature extraction and machine learning approach for fruits and vegetable classification," *Pertanika J. Sci. Technol.*, vol. 27, no. 4, pp. 1693–1708, 2019.
- [13] M. Knott, F. Perez-Cruz, and T. Defraeye, "Facilitated machine learning for image-based fruit quality assessment," 2022, *arXiv:2207.04523*.

- [14] T. Lu, B. Han, L. Chen, F. Yu, and C. Xue, "A generic intelligent tomato classification system for practical applications using DenseNet-201 with transfer learning," *Sci. Rep.*, vol. 11, no. 1, p. 15824, Aug. 2021.
- [15] S. R. N. Appe, G. Arulsevi, and G. Balaji, "Tomato ripeness detection and classification using VGG based CNN models," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 1, pp. 296–302, 2023.
- [16] Y. Fu, M. Nguyen, and W. Q. Yan, "Grading methods for fruit freshness based on deep learning," *Social Netw. Comput. Sci.*, vol. 3, no. 4, p. 264, Jul. 2022.
- [17] J. S. Tata, N. K. V. Kalidindi, H. Katherapaka, S. K. Julakal, and M. Banothu, "Real-time quality assurance of fruits and vegetables with artificial intelligence," *J. Phys., Conf. Ser.*, vol. 2325, no. 1, Aug. 2022, Art. no. 012055.
- [18] T. Tao and X. Wei, "A hybrid CNN–SVM classifier for weed recognition in winter rape field," *Plant Methods*, vol. 18, no. 1, p. 29, Dec. 2022.
- [19] Y. Chen, H. Sun, G. Zhou, and B. Peng, "Fruit classification model based on residual filtering network for smart community robot," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–9, Mar. 2021.
- [20] Z. Zha, D. Shi, X. Chen, H. Shi, and J. Wu, "Classification of appearance quality of red grape based on transfer learning of convolution neural network," *Tech. Rep.*, 2023.
- [21] H. Alshammari, K. Gasmii, I. Ben Ltaifa, M. Krichen, L. Ben Ammar, and M. A. Mahmood, "Olive disease classification based on vision transformer and CNN models," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–10, Jul. 2022.
- [22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
- [23] D. Ileri, E. Belal, C. Okinda, N. Makange, and C. Ji, "A computer vision system for defect discrimination and grading in tomatoes using machine learning and image processing," *Artif. Intell. Agricult.*, vol. 2, pp. 28–37, Jun. 2019.
- [24] M. H. Malik, T. Zhang, H. Li, M. Zhang, S. Shabbir, and A. Saeed, "Mature tomato fruit detection algorithm based on improved HSV and watershed algorithm," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 431–436, 2018.
- [25] S. Malhotra and R. Chhikara, "Automated grading system to evaluate ripeness of tomatoes using deep learning methods," in *Proc. 2nd Int. Conf. Inf. Manag. Mach. Intell. (ICIMMI)*. Cham, Switzerland: Springer, 2021, pp. 129–137.
- [26] N. Begum and M. K. Hazarika, "Maturity detection of tomatoes using transfer learning," *Meas., Food*, vol. 7, Sep. 2022, Art. no. 100038.
- [27] R. G. de Luna, E. P. Dadios, A. A. Bandala, and R. R. P. Vicerra, "Size classification of tomato fruit using thresholding, machine learning, and deep learning techniques," *AGRIVITA J. Agricult. Sci.*, vol. 41, no. 3, pp. 586–596, Oct. 2019.
- [28] G. Liu, S. Mao, and J. H. Kim, "A mature-tomato detection algorithm using machine learning and color analysis," *Sensors*, vol. 19, no. 9, p. 2023, Apr. 2019.
- [29] S. Kaur, A. Girdhar, and J. Gill, "Computer vision-based tomato grading and sorting," in *Advances in Data and Information Sciences*, vol. 1. Cham, Switzerland: Springer, 2018, pp. 75–84.
- [30] B. D. Learning, "A performance and power analysis," NVidia, Santa Clara, CA, USA, White Paper, Nov. 2015.
- [31] Z. Labs. (2019). *Optical Fruit Grading and Sorting Machine*. [Online]. Available: <https://zentronlabs.com/systems/optical-fruit-grading-and-sorting-machine/>
- [32] H. Mputu et al. (2023). *Tomato Fruits Dataset for Binary and Multiclass Classification*. Accessed: Oct. 10, 2023. [Online]. Available: <https://data.mendeley.com/datasets/x4s2jz55dx/1>
- [33] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
- [34] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *Proc. Int. Interdiscipl. PhD Workshop (IIPhDW)*, May 2018, pp. 117–122.
- [35] M. Hussain, J. J. Bird, and D. R. Faria, "A study on CNN transfer learning for image classification," in *Proc. UK Workshop Comput. Intell.*, Nottingham, U.K. Cham, Switzerland: Springer, 2019, pp. 191–202.
- [36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
- [37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
- [38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [40] M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, "Feature extraction using convolution neural networks (CNN) and deep learning," in *Proc. 3rd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2018, pp. 2319–2323.
- [41] C. Crisci, B. Ghattas, and G. Perera, "A review of supervised machine learning algorithms and their applications to ecological data," *Ecolog. Model.*, vol. 240, pp. 113–122, Aug. 2012.
- [42] Y.-I. Cai, D. Ji, and D. Cai, "A KNN research paper classification method based on shared nearest neighbor," in *Proc. NTCIR*, vol. 336, 2010, p. 340.
- [43] A. Mackiewicz and W. Ratajczak, "Principal components analysis (PCA)," *Comput. Geosci.*, vol. 19, pp. 303–342, Mar. 1993.
- [44] N. Q. Khang. (2021). *Tomatoes Dataset*. [Online]. Available: <https://www.kaggle.com/datasets/enalis/tomatoes-dataset>
- [45] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
- [46] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
- [47] J.-W. Feng and X.-Y. Tang, "Office garbage intelligent classification based on inception-v3 transfer learning model," *J. Phys., Conf. Ser.*, vol. 1487, no. 1, Mar. 2020, Art. no. 012008.
- [48] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou, "Fixing the train-test resolution discrepancy," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
- [49] J. Y. Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 53–61.
- [50] X. Qi, T. Wang, and J. Liu, "Comparison of support vector machine and softmax classifiers in computer vision," in *Proc. 2nd Int. Conf. Mech., Control Comput. Eng. (ICMCCE)*, Dec. 2017, pp. 151–155.



HASSAN SHABANI MPUU received the B.Sc. degree in telecommunications engineering from the University of Dar es Salaam, Tanzania, in 2014. He is currently pursuing the master's degree with the Department of Electronics and Communications Engineering, Egypt-Japan University of Science and Technology, Egypt. Since 2021, he has been a Research Student with the Department of Electronics and Communications Engineering, Egypt-Japan University of Science and Technology. His research interests include the field of artificial intelligence and embedded systems.



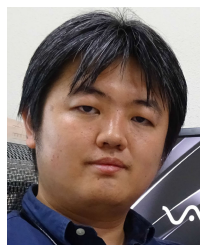
AHMED ABDEL-MAWGOOD was born in Alexandria, Egypt. He received the Ph.D. degree from the Biological Science Department, Purdue University, USA. He moved to the Research and Development Department and the Nucleic Acids Research Department, Promega Corporation, Wisconsin, USA. He worked on the development of new kits for the nucleic acids detection. He was a Professor with the Egypt Japan University of Science and Technology, in 2019. He is currently a Biotechnology Program Coordinator and an Academic Advisor with the Biotechnology Laboratory. He is also the Director of the Food Safety and Quality Management Diploma which is approved by SCU. His research interests include the application of biotechnology in the everyday life, specifically develop new products using the microbes isolated from the environment, biodegradation of the agriculture waste and soil contaminated with hydrocarbons, and collaboration with engineering to develop new instruments, such as conventional PCR.



MOHAMMED S. SAYED (Member, IEEE) received the B.Sc. degree in electronics and communications engineering from Zagazig University, Zagazig, Egypt, in 1997, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Calgary, Calgary, Canada, in 2003 and 2008, respectively. He is currently a Professor with the Department of Electronics and Communications Engineering, Egypt-Japan University of Science and Technology, Egypt.

He has one U.S. patent and seven contributions to the ITU/MPEG video coding standards, two of them included in the H.264 standard's reference hardware model. He has authored or coauthored more than 110 international publications. He managed and participated in several national/international research and development projects. His research interests include digital systems design, embedded systems, system-on-chip, machine vision, video coding, and wireless body area networks. He received several awards both in Canada and in Egypt. He acted as a reviewer of several IEEE TRANSACTIONS and other journals.

...



ATSUSHI SHIMADA (Member, IEEE) received the M.E. and D.E. degrees from Kyushu University, Fukuoka, Japan, in 2007. He is currently a Professor with the Faculty of Information Science and Electrical Engineering, Kyushu University. From 2015 to 2019, he was also a JST-PRESTO Researcher. His current research interests include learning analytics, pattern recognition, media processing, and image processing. He was a recipient of the MIRU Interactive Presentation Award, in

2011 and 2017, MIRU Demonstration Award, in 2015, Background Models Challenge 2012 (The First Place), in 2012, PRMU Research Award, in 2013, SBM-RGBD Challenge (The First Place), in 2017, ITS Symposium Best Poster Award, in 2018, JST PRESTO Interest Poster Award, in 2019, IPSJ/IEEE-Computer Society Young Computer Researcher Award, in 2019, CELDA Best Paper Award, in 2019, and MEXT Young Scientist Award, in 2020.