A CNN-ELM Classification Model for Automated Tomato Maturity Grading

John Paul Tan Yusiong

Division of Natural Sciences and Mathematics, University of the Philippines Visayas
Tacloban College, Tacloban City, Leyte, Philippines
E-mail: jtyusiong@up.edu.ph

Abstract. Tomatoes are popular around the world due to their high nutritional value. Tomatoes are also one of the world's most widely cultivated and profitable crops. The distribution and marketing of tomatoes depend highly on their quality. Estimating tomato ripeness is an essential step in determining shelf life and quality. With the abundant supply of tomatoes on the market, it is exceedingly difficult to estimate tomato ripeness using human graders. To address this issue and improve tomato quality inspection and sorting, automated tomato maturity classification models based on different features have been developed. However, current methods heavily rely on human-engineered or handcrafted features. Convolutional neural networks have emerged as the preferred technique for general object recognition problems because they can automatically detect and extract valuable features by directly working on input images. This paper proposes a CNN-ELM classification model for automated tomato maturity grading that combines CNNs' automated feature learning capabilities with the efficiency of extreme learning machines to perform fast and accurate classification even with limited training data. The results showed that the proposed CNN-ELM model had a classification accuracy of 96.67% and an F1-score of 96.67% in identifying six maturity stages from the test data.

Keywords: *automated tomato maturity grading; CNN-ELM; convolutional neural networks; extreme learning machines; hybrid classification model; tomato classification.*

1 Introduction

Tomatoes are popular around the world due to their high nutritional value. Tomatoes are also one of the most widely cultivated and profitable crops on the planet. Tomatoes contain essential vitamins and minerals that are beneficial to human health, including vitamin C, potassium, lycopene, beta-carotene, and dietary fibers. As a result, both production and consumption continue to increase. Tomatoes are sold fresh in markets as ingredients for a variety of dishes and processed into a range of products, including sauce, juice, paste, and ketchup [1]. The distribution and marketing of tomatoes are heavily reliant on their quality. However, since tomatoes ripen quickly, it is crucial to determine their ripeness properly to guarantee shelf life and quality. When tomatoes ripen, their shape, color, size, and texture change. The color is one of the most obvious

Received May 5th, 2021, Revised September 17th, 2021, Accepted for publication December 3rd, 2021.

characteristics closely associated with their ripeness or maturity and is easy to recognize by humans. When tomato tissue matures, the pigments, internal texture, and chemical composition change [2]. Human graders can easily assess the maturity level of tomatoes based on visual perception and practical experience. However, with the abundance of tomatoes on the market, determining tomato ripeness has become exceedingly difficult.

To address this issue and improve tomato quality inspection and sorting, automated tomato maturity classification models have been created that use a variety of features to determine the level of tomato maturity. Automated tomato maturity classification is a method that utilizes digital image processing, computer vision, and machine learning techniques to categorize tomatoes according to their maturity. Automated tomato maturity classification can be done on images using various features and then classifying them using different methods. Current methods use surface color descriptors derived from different color models, texture, shape, GLCM, and colorimetric properties [3-17]. These methods also employ different algorithms for ripeness classification, including Support Vector Machines (SVMs) [4-7], Linear Discriminant Analysis (LDAs) [6, 7], Naïve Bayes [8,9], Artificial Bee Colony-trained ANNs [10], the Levenberg–Marquardt NN algorithm [11], Backpropagation neural networks [12-14], K-Nearest Neighbor (KNN) [15], KNN-SVM [16], and decision trees [17]. There are three phases in the current automated tomato maturity classification methods: preprocessing, feature extraction, and classification. The feature extraction phase is laborious, challenging, and time-consuming because it heavily relies on human-engineered or handcrafted features.

Convolutional neural networks (CNNs) [18] have emerged as the preferred technique for general object recognition problems due to their ability to automatically identify and extract useful features from input images. According to a recent study [19], CNNs also work well in automated tomato maturity classification. However, there are two significant drawbacks to using CNNs to grade tomato maturity automatically. Firstly, CNN training requires backpropagation of the classification error, which requires significant training time and data. Secondly, the amount of training data must be sufficiently large to avoid overfitting, whereas previous research on automated tomato maturity classification relied on datasets containing just a few hundred of tomato images for each maturity stage. To address these issues, this paper proposes a hybrid CNN-ELM classification model for automated tomato maturity grading that takes advantage of CNN's capability in feature detection and extraction, and ELM's ability to manage limited datasets. The CNN-ELM architecture integrates the capability of CNNs in automatic feature learning with the reliability of extreme learning machines (ELMs) [20] to perform fast and accurate classification even

with sparse training data. In the CNN-ELM architecture, ELMs replace CNN's final fully connected layer and softmax classification layer [21-23].

The following are the major contributions of this work:

1. It is the first study to demonstrate the use of a CNN-ELM architecture for automated tomato maturity grading. This hybrid classification model makes use of both convolutional neural networks and extreme learning machines.
2. The proposed method employs a supervised learning framework to train the proposed hybrid architecture, which incorporates convolutional neural networks to automatically identify and extract features and extreme learning machines to achieve fast and accurate classification even with small training data.
3. It was demonstrated that the proposed method can identify six maturity stages from the test data. The experimental results showed that combining CNN and ELM increased generalization performance.

The remainder of this paper is organized as follows. Section 2 reviews recent studies on automated tomato maturity classification. Section 3 describes the proposed CNN-ELM architecture. Section 4 presents the experiments and results of the proposed method. This paper is concluded by Section 5.

2 Related Work

This section discusses recent approaches to the automated tomato maturity classification problem. Previous studies have shown that the various techniques used for estimating tomato maturity consist of three phases: preprocessing, feature extraction, and classification. The preprocessing phase of the different methods is almost identical and consists of a series of digital image processing operations, but they differ in the feature extraction method and the classification algorithm they employ.

In two separate studies [4,5], El-Hariri, *et al.* used a PCA-SVM model. To create a feature vector for each image in the dataset, the feature extraction phase employs two color descriptors based on the HSV color model and the PCA algorithm. Finally, for the classification phase, the proposed method classifies tomatoes into one of five maturity classes using SVM. El-Bendary, *et al*. [6] used the same framework as [4,5] but introduced an LDA algorithm as a classifier. A 250-image dataset was used to evaluate their proposed model, and the results showed that the PCA-LDA model successfully classified the tomato images in the dataset. On the other hand, Garcia, *et al.* [7] presented an automated tomato ripeness identification system using the CIE L*a*b* color model for the features vectors and SVM for the classifier. The feature extraction phase involves

generating a features vector for each image based on the CIE L*a*b* color model, while the classification phase involves labeling each image into one of six tomato ripeness classes. They validated their proposed approach using 900 sample images; the results showed an accuracy of 83.39% on the test data.

Kusuma and Setiadi [8] and Ceh-Varela and Hernandez-Chan [9] demonstrated that a Naïve Bayes tomato classifier could replace a manual classification procedure using color histograms, but their methods for extracting histogram features differed. Kusuma and Setiadi [8] extracted six histogram features from grayscale images. Their study used a dataset of 100 images to train a Naïve Bayes classifier to predict whether a tomato image is raw, mature, or rotten; the trained classifier had a 76% accuracy score. Ceh-Varela and Hernandez-Chan [9] used the HSV color model to extract histogram features from the hue component. They designed a Naïve Bayes classifier that predicts whether an image is a red tomato, a green tomato, or a yellow tomato. The experiments used a dataset of 37 images and the K-fold cross-validation method. The results show that the trained model achieved 96% accuracy.

Opeña and Yusiong [10], Astrianda and Mohamad [11], Kassem *et al*. [12], Wan *et al*. [13], and Kaur *et al.* [14] showed that artificial neural networks could be trained to estimate tomato maturity. Opeña and Yusiong [10] trained artificial neural networks using the artificial bee colony algorithm for this task. The RGB, HSI, and CIE L*a*b* color models were used during the feature extraction phase. Five color features were chosen from these three color models, and a color features vector was created for each tomato image. ANN training was performed during the classification phase to obtain a trained ANN classifier using the features vectors as inputs. A dataset of 600 tomato images was used in the experiments. The results suggest that an automated tomato classification method based on an ABC-trained ANN classifier can be used instead of a manual classification procedure to reduce the likelihood of misclassification. In comparison, Astrianda and Mohamad [11] generated features vectors for the CIE L*a*b*, YCbCr, and HSV color models. The Levenberg–Marquardt algorithm was used to train three networks to classify tomatoes in images as ripe or unripe. The experiments used a dataset of 70 sample tomato images to compare the performance of the three models. The results show that the network trained with the CIE L*a*b* color model outperformed the other two models.

Kassem, *et al.* [12], Wan, *et al.* [13], and Kaur, *et al*. [14] used the backpropagation algorithm to train ANNs for tomato maturity grading. Kassem, *et al.* [12] extracted 12 color features from the CIE L*a*b* color model, Wan, *et al*. [13] generated a five-feature vector from the RGB and HSI color models, and Kaur, *et al*. [14] extracted 13 features using the RGB color model and shape attribute. The number of maturity classes can vary as well. Kassem, *et al.* [12]

trained an ANN classifier to distinguish between green, pink, and red tomato images. Wan, *et al.* [13] trained an ANN classifier to differentiate between red, orange, and green tomato images. Kaur, *et al*. [14] trained an ANN classifier to distinguish between defective and non-defective tomato images. During the experiments, Kassem, *et al.* [12] used 237 tomato images, Wan, *et al.* [13] used 150 images, and Kaur, *et al*. [14] utilized 53 images. Regardless, all ANN classifiers trained with the backpropagation algorithm obtained a remarkable accuracy score.

The K-Nearest Neighbor (KNN) algorithm can also be used to predict tomato maturity, as shown by Indriani, *et al.* [15] and Pavithra, *et al.* [16]. Indriani, *et al.* [15] used texture and color analysis to generate the features vector for each image, which are then fed into the KNN classifier. Their experiments used a dataset of 100 images to develop a KNN classifier that can classify a tomato image into one of five maturity classes. In contrast, Pavithra, *et al.* [16] trained a KNN-SVM classifier using color, shape, and texture features to predict whether a tomato image belongs to one of three classes. The results of these two models revealed that KNN can achieve excellent classification performance.

Goel, *et al*. [17] proposed a new methodology for classifying tomato images into six maturity classes based on the RGB color model, fuzzy logic, and decision trees. The approach is known as Fuzzy Rule-Based Classification (FRBCS). To create and evaluate the performance of FRBCS, a dataset of 116 sample images was used. The results showed that the proposed method attained a ripeness classification accuracy of 94.29% on the test images.

These automated tomato maturity classification methods used various digital image processing, computer vision, and machine learning techniques. Furthermore, these traditional approaches relied heavily on human expertise to manually derive a variety of features from different color models, shapes, and texture attributes. The number of images in the dataset, the classification algorithms employed, and the number of maturity or ripeness classes also varied. While automated tomato classification systems can perform remarkably, the key bottleneck is the heavy reliance on human expertise to identify the features used as input of the classification algorithm. A method that can automatically detect and extract useful features from input images is needed to solve this bottleneck. A convolutional neural network's feature learning capability automatically identifies and extracts features from input images.

To enhance the generalization ability of the CNN-based tomato classification method, this paper proposes a hybrid CNN-ELM classification model that combines CNNs' automated feature learning capability with the efficiency of extreme learning machines to perform quick and accurate classification even with

limited training data. The CNN-ELM architecture has been used to solve image classification problems [21], such as maritime ship recognition [22] and age and gender classification [23]. This hybrid approach has been shown to increase accuracy and performance [21-23].

3 CNN-ELM for Automated Tomato Maturity Grading

This section discusses in detail the proposed supervised method for the automated tomato maturity grading system. A training dataset of tomato images with the corresponding ground-truth maturity labels is necessary for the proposed method. The supervised framework for tomato maturity grading is depicted in Figure 1, which consists of two training phases and a test phase.

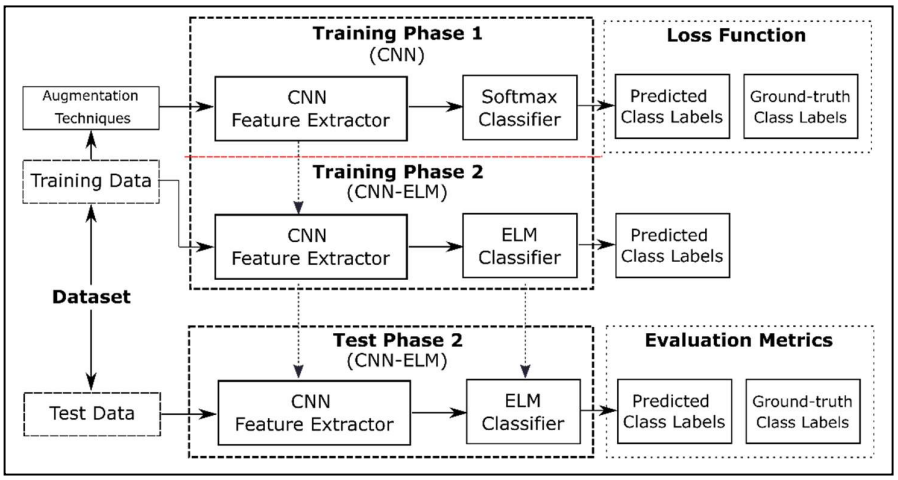

Figure 1 The supervised framework for tomato maturity grading using the proposed CNN-ELM architecture.

3.1 Hybrid CNN-ELM Network Architecture

There are two training phases, as described in Figure 1. The first phase entails training an end-to-end CNN classifier over many epochs. The CNN classifier is based on the PeleeNet architecture [24] and consists of a feature extractor and a softmax classifier. PeleeNet is a DenseNet derivative with optimizations for limited memory and computing resources. In this study, PeleeNet's initial weights were generated at random.

The CNN feature extractor part obtained in the first training phase is reused in the second phase to create a CNN-ELM architecture. Unlike the first training phase, which requires many epochs, CNN-ELM training is achieved in one single

pass over the training data due to the structure of the ELM network. The tomato dataset was used in both training phases, while the data augmentation methods were only employed in the first training phase. Figure 2 depicts the general structure of CNN-ELM for automated tomato maturity grading based on PeleeNet.

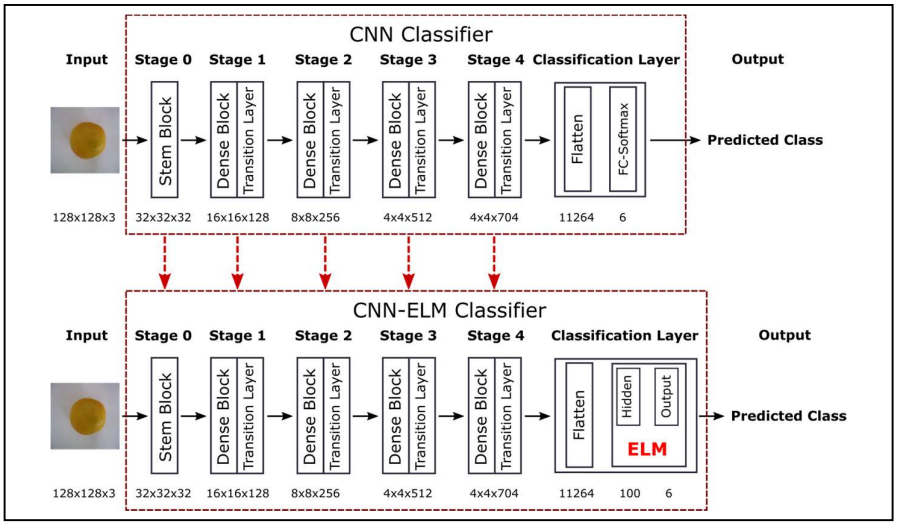

Figure 2 The CNN classifier based on PeleeNet is pre-trained on the tomato dataset. Afterward, the fully connected and softmax layers are replaced by an ELM network, while the other trained layers are reused to create a new model, the CNN-ELM classifier.

The original PeleeNet architecture uses a fully connected layer and a softmax layer to classify the generated feature maps. On the other hand, the proposed hybrid model modifies the classification layer of PeleeNet by replacing the fully connected layer and softmax layer with a single-layer ELM with 100 neurons in the hidden layer and 6 output neurons because the tomato dataset has 6 classes.

The neurons in the ELM use hyperbolic tangent (tanh) as the activation function. Although the number of ripeness classes in previous studies' tomato datasets varies, using one of these datasets to re-train the CNN-ELM classifier is fast and straightforward since CNN-ELM training can be completed in one single pass over the training data due to the structure of the ELM network. On the other hand, re-training a CNN classifier will take several epochs if the number of ripeness classes is changed.

3.2 Loss Function

A loss function is used to train the model. In this study, the cross-entropy loss function is employed, which is widely used for classification problems. For a multi-class classification problem, the loss function, as defined in Eq. (1), calculates the cross-entropy value between the probability vector predicted by the model and the ground truth, typically in the form of a one-hot encoded vector, where only one class label is true.

$$CE = -\sum_{i=1}^{C} T_i \log(S_i) \tag{1}$$

T_i and S_i denote the ground-truth class label and the CNN score for each class i in C, where C indicates the total number of classes. Before calculating the cross-entropy loss, an activation function such as softmax is applied to the scores.

3.3 Evaluation Metrics

This section discusses the evaluation metrics used in this work in detail. Evaluation metrics are used to quantify the performance of trained models, that is, how well each model predicts unseen instances. There are four standard evaluation metrics for classification problems: accuracy, recall, precision, and F1-score [25]. As defined in Eq. (2), accuracy is the ratio between true outcomes and the total number of cases analyzed. Precision is the ratio between true positives and all the classifier's positive predictions, as defined in Eq. (3). As defined in Eq. (4), recall is the ratio between true positives and all samples that should have been classified as positive. The F1-score is a metric that strikes a balance between precision and recall. A good F1 score should have a low rate of false positives and false negatives. Eq. (5) defines the F1-score metric.

$$Accuracy = \frac{TP+T}{TP+TN+FP+FN} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$F_1 - score = 2 * \frac{Precision*Recall}{Precision+R} \tag{5}$$

3.4 Dataset

The tomato dataset introduced in [10] was used in this study. During the image acquisition phase, tomatoes were bought from the local market. Each tomato was placed on top of a plain white paper that served as the background, and a Sony digital camera was used to obtain the tomato images. The tomato images included in the dataset were resized to an image resolution of 200 by 200 to speed up the training process. No additional pre-processing was done on the acquired images.

The tomato dataset includes 600 tomato images, 100 images for each stage of maturity: green (G), breakers (B), turning (T), pink (P), light red (LR), and red (R). These six tomato maturity classes are based on the USDA color classification and are widely used for fresh tomatoes. A detailed description of each tomato maturity class is presented in the work of Garcia *et al*. [7]. Figure 3 shows sample tomato images for each maturity class. Furthermore, the tomato images in the dataset were split into two sets. Four hundred twenty images, or seventy images per maturity class, were used as the training set, and one hundred eighty images, or thirty images per maturity class, were used as the test set.

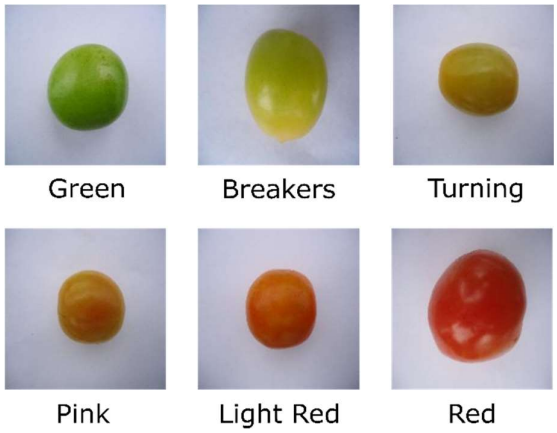

Figure 3 Sample tomato images for each maturity stage or class.

4 Experimental Results and Discussions

The CNN-ELM model for automated tomato maturity grading was implemented using Tensorflow [26]. A single Nvidia GTX 1080 Ti GPU equipped with 11 GB of memory was used to train the model. The recent Padam optimizer [27] with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$, and p = 0.125 was used during the initial training phase to train the model for 750 epochs by minimizing the loss as defined in Eq. (1). The initial learning rate was $\lambda = 10^{-3}$ and was used for the first 450 epochs; afterward, it was reduced by half every 150 epochs. Additionally, during the first training phase, the input images were resized from 200 by 200 to 128 by 128, employing a batch size of 16, and performing online data augmentation on the input images to increase the quantity and diversity of the training data. The data augmentation methods used in this work were: random erasing, flipping, cropping, and affine transformation [28]. The second training phase was simpler than the first. It only reduced the input images from their original resolution to

128 by 128 and provided the CNN-ELM model with training data in one single pass.

4.1.1 Results and Discussion

This section discusses the experiments and results of the proposed CNN-ELM model for automated tomato maturity grading. Three experimental setups were designed to evaluate the CNN-ELM model. The first experiment entailed conducting the first training phase to evaluate the impact of integrating an online data augmentation module on the performance of the CNN model and using the test set to evaluate the trained CNN-only model. This setup trained two different models, with and without data augmentation, to classify tomato images into one of six maturity classes. In the second experiment, the feature extraction component of the CNN model trained without data augmentation was used, and the number of maturity classes was varied. Then four CNN-ELM models were evaluated separately with two, three, five, and six maturity classes, respectively. The third experiment setup was like the second experiment, but in this setup, the feature extraction component of the CNN model trained with data augmentation was used to train four CNN-ELM models with different maturity classes. The experimental setups used to evaluate the proposed approach are summarized in Table 1, where the maturity classes enclosed in parenthesis are considered a single maturity class.

Table 1 Experimental setups to evaluate the CNN-ELM models.

Setup	Training Phase 1	Training Phase 2	# of Maturity Classes	Classes
1	CNN without online data augmentation (A)	None	6	G, B, T, P, LR, R
	CNN with online data augmentation (B)	None	6	G, B, T, P, LR, R
2	CNN feature extractors from setup 1(A)	ELM	6	G, B, T, P, LR, R
		ELM	5	(G, B), T, P, LR, R
		ELM	3	(G, B), (T, P), (LR, R)
		ELM	2	(G, B, T), (P, LR, R)
3	CNN feature extractors from setup 1(B)	ELM	6	G, B, T, P, LR, R
		ELM	5	(G, B), T, P, LR, R
		ELM	3	(G, B), (T, P), (LR, R)
		ELM	2	(G, B, T), (P, LR, R)

Tables 2, 3, and 4 summarize the quantitative results from the various experimental setups. The impact of the online data augmentation module on the performance of the CNN model can be seen in Table 2. The CNN model trained with augmented data achieved an accuracy rate of 94.44%, while the CNN model trained without augmented data only achieved an accuracy rate of 87.22%. These

findings indicate that performing data augmentation during the training phase of a CNN improves the model's accuracy because data augmentation techniques allow for a substantial increase in the size and diversity of data by introducing slightly modified copies of available data.

Table 2 Setup 1: Impact of online data augmentation during CNN training.

Model	CNN without augmentation	CNN with augmentation
Correctly classified	157/180	170/180
Accuracy	87.2222%	94.4444%
Recall	88.1485%	94.7163%
Precision	87.2222%	94.4444%
F1-score	87.6829%	94.5802%

Table 3 Setup 2: Quantitative results of the CNN-ELM models using the feature extractor of the CNN model trained without augmentation.

Number of Maturity Classes	6	5	3	2
Correctly classified	168/180	169/180	175/180	177/180
Accuracy	93.3333%	93.8889%	97.2222%	98.3333%
Recall	93.6864%	93.1644%	97.2678%	98.3393%
Precision	93.3333%	94.0000%	97.2222%	98.3333%
F1-score	93.5095%	93.5803%	97.2450%	98.3363%

Table 4 Setup 3: Quantitative results of the CNN-ELM models using the feature extractor of the CNN model trained with augmentation.

Number of Maturity Classes	6	5	3	2
Correctly classified	174/180	174/180	177/180	179/180
Accuracy	96.6667%	96.6667%	98.3333%	99.4444%
Recall	96.7990%	96.1588%	98.4127%	99.4505%
Precision	96.6667%	96.6667%	98.3333%	99.4444%
F1-score	96.6731%	96.4121%	98.3730%	99.4475%

Furthermore, a comparison of the findings in Table 3 and Table 4 reveals that the CNN model's feature extraction functionality affects the CNN-ELM model's efficiency. Indeed, all CNN-ELM models that used the feature extraction component of the CNN model trained with data augmentation outperformed all CNN-ELM models that used the CNN model's feature extraction component trained without data augmentation. The results indicate that, in a hybrid setup, the success of the initial model can have a substantial impact on the performance of subsequent models. More precisely, the quality of the CNN feature extraction component affects the CNN-ELM model's efficiency. However, as shown in Tables 3 and 4, the CNN-ELM models outperformed the CNN-only models presented in Table 2, demonstrating that combining two models can improve performance. Finally, since the CNN-ELM training can be performed in a single pass over the training data due to the ELM network's architecture, training several CNN-ELM models with different maturity classes is straightforward, since only

the ELM model's output layer is modified during training, while the ELM network's previous layer and CNN feature extraction layers remain unchanged.

Finally, Table 5 compares the CNN-ELM model's performance to past studies that used other techniques in order to provide future researchers with insights into which model works best for tomato maturity classification. However, it should be noted that these models differ not only in their techniques but also in other aspects, such as the dataset used to evaluate the model, the number of maturity classes the model can identify, and the handcrafted features used by the conventional machine learning methods. It can also be observed that color is widely used in constructing the features vector, but the actual color descriptors vary, and these variations affect the model's performance.

Table 5 A comparative analysis of the different models for tomato maturity classification.

Model	Number of images in the dataset	Features	Number of maturity classes	Accuracy score
ABC-NN [10]	600	Color	6	98.19%
CNN-ELM	600	Learned features	6	96.67%
FRBCS [17]	116	Color	6	94.29%
SVM [7]	900	Color	6	83.39%
KNN [15]	100	Texture, color	5	100.00%
CNN-ELM	600	Learned features	5	96.67%
SVM [4]	230	Color	5	92.72%
OAO-SVM [5]	250	Color	5	91.20%
OAO-SVM [6]	250	Color	5	90.80%
OAA-SVM [5]	250	Color	5	85.60%
OAA-SVM [6]	250	Color	5	84.80%
LDA [6]	250	Color	5	84.00%
CNN-ELM	600	Learned features	3	98.33%
BPNN [13]	150	Color	3	99.31%
BPNN [12]	237	Color	3	97.90%
Naïve Bayes [9]	37	Color	3	96.00%
Naïve Bayes [8]	100	Color	3	76.00%
KNN-SVM [16]	-	Texture, color, shape	3	-
CNN-ELM	600	Learned features	2	99.44%
LM [11]	70	Color	2	96.00%
ANN [14]	53	Color	2	92.00%

5 Conclusion

This paper described a supervised learning framework for automated tomato maturity grading based on a CNN-ELM hybrid architecture. The proposed hybrid model capitalizes on the feature learning capabilities of convolutional neural networks and the fast and accurate classification performance of extreme learning

machines even with minimal training data. The experimental findings showed that the proposed CNN-ELM classifier outperformed the CNN-only classifier based on the standard evaluation metrics. Moreover, the experimental results revealed that the CNN-ELM model is flexible and reliable because it can be reconfigured easily to accommodate a different set of maturity classes without negative effect on the performance.

Acknowledgement

This is a professorial chair output of the author as an awardee of the Diamond Jubilee (DJ) Professorial Chair in Computer Science covering the period from July 1, 2010, to June 30, 2011.

References

[1] Bergougnoux, V., *The History of Tomato: From Domestication to Biopharming*, Biotechnology Advances, **32**, pp. 170-189, 2014.

[2] Qin, J. & Lu, R., *Measurement of The Optical Properties of Fruits and Vegetables Using Spatially Resolved Hyperspectral Diffuse Reflectance Imaging Technique*, Postharvest Biology and Technology, **49**, pp. 355-365, 2008.

[3] Bello, T.B., Costa, A.G., da Silva, T.R., Paes, J.L. & de Oliveira, M.V. M., *Tomato Quality Based on Colorimetric Characteristics of Digital Images*, Revista Brasileira de Engenharia Agrícola e Ambiental, **24**(8), pp. 567-572, 2020.

[4] Elhariri, E., El-Bendary, N., Fouad, M.M.M., Platoš, J., Hassanien, A.E. & Hussein, A. M. M., *Multi-class SVM Based Classification Approach for Tomato Ripeness,* Advances in Intelligent Systems and Computing, pp. 175-186, 2014.

[5] El Hariri, E., El-Bendary, N., Hassanien, A.E. & Badr, A., *Automated Ripeness Assessment System of Tomatoes Using PCA and SVM Techniques*, Advances in Computational Intelligence and Robotics, pp. 101-130, 2014.

[6] El-Bendary, N., El Hariri, E., Hassanien, A.E. & Badr, A., *Using Machine Learning Techniques for Evaluating Tomato Ripeness*, Expert Systems with Applications, **42**(4), pp. 1892-1905, 2015.

[7] Garcia, M.B., Ambat, S., & Adao, R.T., *Tomayto, Tomahto: A Machine Learning Approach for Tomato Ripening Stage Identification Using Pixel-Based Color Image Classification*, 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), 2019.

[8] Kusuma, A., Setiadi, D.R.I.M. & Putra, M.D.M., *Tomato Maturity Classification using Naive Bayes Algorithm and Histogram Feature Extraction*, Journal of Applied Intelligent System, **3**(1), pp. 39-48, 2018.

[9] Ceh-Varela, E. & Hernandez-Chan, G., *Tomatoes Classifier using Color Histograms*, International Conference on Advances in Engineering Science and Management, 2015.

[10] Opeña, H.G. & Yusiong, J.P.T., *Automated Tomato Maturity Grading Using ABC-Trained Artificial Neural Networks*, Malaysian Journal of Computer Science, **30**(1), pp. 12-26, 2017.

[11] Astrianda, N. & Mohamad, F.S., *Ripeness Identification of Tomato Using Different Color Models Based on Neural Networklevenberg-Marquardt*, World Applied Sciences Journal, **35**, pp. 57-61, 2017.

[12] Kassem, A.E.-W.S., Sabbah, M.A., Aboukarima, A.E.W.M. & Kamel, R.M., *A Study on Color Sorting of Tomatoes Maturity using Machine Vision and Artificial Neural Networks*, NETWORKS. Egyptian Journal of Agricultural Research, **93**(1), pp. 147-161, 2015.

[13] Wan, P., Toudeshki, A., Tan, H. & Ehsani, R., *A Methodology for Fresh Tomato Maturity Detection Using Computer Vision*, Computers and Electronics in Agriculture, **146**, pp. 43-50, 2018.

[14] Kaur, S., Girdhar, A. & Gill, J., *Computer Vision-Based Tomato Grading and Sorting*, Lecture Notes in Networks and Systems, pp. 75-84, 2018.

[15] Indriani, O.R., Kusuma, E.J., Sari, C.A., Rachmawanto, E.H. & Setiadi, D. R.I.M., *Tomatoes Classification Using K-NN Based on GLCM and HSV Color Space*, 2017 International Conference on Innovative and Creative Information Technology (ICITech), 2017.

[16] Pavithra, V., Pounroja, R. & Bama, B.S., *Machine Vision Based Automatic Sorting of Cherry Tomatoes,* 2015 2nd International Conference on Electronics and Communication Systems (ICECS), 2015.

[17] Goel, N. & Sehgal, P., *Fuzzy Classification of Pre-Harvest Tomatoes for Ripeness Estimation – An Approach Based on Automatic Rule Learning Using Decision Tree*, Applied Soft Computing, **36**, pp. 45-56, 2015.

[18] Krizhevsky, A., Sutskever, I. & Hinton, G.E., *Imagenet Classification with Deep Convolutional Neural Networks*, Communications of the ACM, **60**(6), pp. 84-90, 2017.

[19] Das, P. & Yadav, J.P.S., *Automated Tomato Maturity Grading System using CNN*, 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020.

[20] Huang, G.B., Zhu, Q.Y. & Siew, C.K., *Extreme Learning Machine: Theory and Applications*, Neurocomputing, **70**(1-3), pp. 489-501, 2006.

[21] Park, Y. & Yang, H.S., *Convolutional Neural Network Based on an Extreme Learning Machine for Image Classification*, Neurocomputing, **339**, pp. 66-76, 2019.

[22] Khellal, A., Ma, H. & Fei, Q., *Convolutional Neural Network Based on Extreme Learning Machine for Maritime Ships Recognition in Infrared Images*, Sensors, **18**(5), 1490, 2018.

[23] Duan, M., Li, K., Yang, C. & Li, K., *A Hybrid Deep Learning CNN–ELM for Age and Gender Classification*, Neurocomputing, **275**, pp. 448-461, 2018.

[24] Wang, R.J., Li, X. & Ling, C.X., *Pelee: A Real-Time Object Detection System on Mobile Devices*, Advances in Neural Information Processing Systems, **31**, 2018.

[25] Japkowicz, N. & Shah, M., *Evaluating Learning Algorithms: A Classification Perspective*, Cambridge: Cambridge University Press, 2011.

[26] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, Z., Dean, J., Devin, M., Ghemawat, S., Irving, G. & Isard, M., *Tensorflow: A System for Large-Scale Machine Learning*, 12th USENIX conference on Operating Systems Design and Implementation, pp. 265-283, 2016.

[27] Chen, J., Zhou, D., Tang, Y., Yang, Z., Cao, Y. & Gu, Q. *Closing the Generalization Gap of Adaptive Gradient Methods in Training Deep Neural Networks*, Twenty-Ninth International Joint Conference on Artificial Intelligence, pp. 3267-3275, 2020.

[28] Shorten, C. & Khoshgoftaar, T.M., *A Survey on Image Data Augmentation for Deep Learning*, Journal of Big Data, **6**, 60, 2019.