



Tomato Ripeness Identification Using Recurrent Neural Network Algorithm

Dede Hamdani¹, M. Hizbul Wathan²

¹ Universitas Respati Yogyakarta, Indonesia

² Politeknik Manufaktur Negeri Bangka Belitung, Indonesia

Abstract

Tomatoes undergo distinct ripeness stages, typically categorized into ripe, semi-ripe, and unripe phases. Traditional methods for assessing ripeness often face challenges in accuracy due to difficulties in comparing variables and subjective interpretations. This study proposes an innovative approach to classify tomato ripeness using a dataset of 200 tomato images and employs a Recurrent Neural Network (RNN) for precise classification. The experimental results demonstrate that the RNN-based model achieves a 95.0% accuracy rate in identifying ripeness stages, significantly outperforming conventional methods. This high level of accuracy highlights the model's potential to minimize errors and provide reliable assessments of tomato maturity. The proposed method offers a robust and efficient solution for agricultural applications, enabling improved quality control and harvest timing. Future research could explore the integration of additional data sources or advanced machine learning techniques to further enhance the model's performance and applicability across diverse agricultural contexts.

Keywords:

Classification, Tomato, Ripeness, Deep Learning, RNN

This is an open-access article under the [CC BY-SA](#) license



1. Introduction

Watermarking, steganography, and cryptography are all examples of digital picture processing. On the other hand, image processing also can be done for classification, and prediction categorized in image mining. The classification method can be utilized to classify the maturity level of fruit. One fruit that can be classified is tomatoes. Tomatoes, at any given time, have different levels of maturity that can be grouped into specific classes. The maturity of tomatoes is usually used to satisfy different requirements [8].

Several ways have been employed to determine the maturity condition of the fruits, most of which are based on the accuracy of a visual and tactile study. These methods do not guarantee a quality process of fruit classification in optimal conditions due to the possibility of presenting significant errors in the comparison of important variables such as color, size, and shape. Tomato classification by size and maturity degree is the most used. Artificial vision techniques and digital image processing are increasingly useful in the agricultural industry, especially for quality inspection applications, such as fruit classification by color or by some type of feature [6].

The sorting process and class distribution of tomatoes are done manually by humans. However, manual sorting and class division are quite tiring and prone to errors. In addition, differences in perception between workers also can influence the determination of the tomato's ripeness level. Based on these problems, using computer technology as a remedy is a viable option. To be able to classify the ripeness level of tomatoes from an image, a computer needs to recognize the right parameters, which are unique and can be used to

represent the ripeness level of tomatoes [16].

Color is one of the most important quality features of fruits which indicates the degree of doneness, sugar content, acidity, and taste. For example, a deep red hue signifies a higher level of fruit in the fresh fruit market, such as apples and peaches, than a pink hue. In other words, tomatoes are a type of food that has a distinctive color that is widely used as an indicator of maturity [1]. Machine learning can identify ripening phases using physical parameters like color in the case of tomatoes since there is a positive relationship between color and maturity [3]. Because tomato maturity is directly related to its surface color, visual identification is a viable means of determining the tomato's maturity level. The green stage, breakers stage, changing stage, pink stage, pale red stage, and red stage are the six phases of tomato color maturation. Furthermore, the current literature generally picks two or three degrees of maturity (matured, partly matured, and immature phases) [2].

Another source presents approaches based on machine learning algorithms for automatically annotating raw movement trajectories that have been developed. Before developing a classifier like the SVM or Random Forests, these approaches necessitate a feature engineering procedure. Because the feature engineering process is difficult and context-dependent, standard methodologies are not suitable for widespread application. RNN is one of the most important deep learning algorithms, and it has recently gotten a lot of attention in image recognition [10].

To classify the ripeness of freshly made tomatoes, a study uses RNN with LSTM for sentiment classification and recommendations to achieve this goal [5]. Consequently, RNN stands for neural network design, which is capable of extracting this type of data from data sequences. Furthermore, LSTM cell-based RNNs are becoming increasingly popular due to their capacity to learn both in the short and long term [9]. The RNN learns to detect image features over time and extract the features needed for classification from the input. They were created to deal with sequence prediction problems. The RNN remembers the previous input because of the feedback loop in the iterative layer. The LSTM layer of the RNN is famous for its memory, which helps in long-term information storage [4].

In this paper, we propose RNN to classify the maturity level of red tomatoes by computing including diameter RGB value tomato. We establish an effective model to solve the tomato ripeness classification issue. In fruit types classification, we present several crucial contributions to this study, particularly in the determination of tomato ripeness using learning methods as follows:

1. We introduce a new technique for the classification of tomato ripeness in training data sets and develop a viable model. We use multiple images of tomatoes with different ripeness to build a model.
2. We designed a new classification model using the RNN algorithm to classify ripe and raw tomatoes. We built a model using deep learning methods to classify tomato ripeness. We get an accurate result and a low error to prove the model
3. This research produces an accurate classification model that uses deep learning methods to compare ripe and raw data. The proposed model can study future Tomato ripeness classification.

Organization: This is how journals are organized: Part II delves deeper into related research. The characterization of the research problem is described in Section III. Part IV describes the experimental design, including the feature learning approach, data collection, and data preprocessing, while Part V summarizes the findings and provides a complete analysis. Finally, section VI summarizes the findings and identifies some research gaps in classifying maturity levels.

2. Related Works

Fruit classification is a growing research area in computer vision. A study investigated changes in tomatoes at different ripeness using MRI and the potential use of MRI for the classification of tomato ripeness. A set of 5 MR images were collected on processing tomatoes at various stages of maturity. 5 MRI sequences were selected so that information about the proton properties of water, including proton density, T1, T2, and diffusion rate, was encoded in the signal intensity of the MR image. PLS-DA was applied to a total of 48 image features calculated from 5 MR images of 144 tomatoes processed to predict tomato ripeness. The model with 4 latent variables captured 70% of the variation in tomato maturity. The classification accuracy of the PLS-DA model is about 90% for the green, bright red, and red maturity stages. The Coefficient of Importance of Variables in the Projection of 48 image features in the model shows that the diffusion weight image and spin-echo image with a higher T2 weight are the most important for tomato ripeness classification [21].

To make it easier to determine tomato ripeness accurately with conventional techniques, an article proposed the mechanical properties of the fruit under the falling impact test were used to establish a non-destructive method for determining tomato maturity. With linear discriminant analysis, classification performance was enhanced, while stepwise regression analysis reduced the number of indices evaluated. The categorization accuracy is 82.3% when all nine indices are used and 79.2% when the three most dominant indices are used. The results suggest that combining declining impact with linear discriminant analysis yields a potential non-destructive method for determining tomato ripeness [20].

Machine learning is one method for classifying fruits. An article explored feature histogram extraction and the Naive Bayes Classifier. Histogram feature extractions are commonly utilized and influence classification outcomes. When applied to many databases, Naive Bayes provides great accuracy and computational speed, is robust to isolated noise points, and only requires limited training data to predict the parameters needed for classification. 76% accuracy was attained based on the outcomes of the experiment employing 75 training data and 25 testing data [15]. Then, another study proposed machine learning using an SVM classifier and the CIELab color space. 900 photos were gathered from a farm and various image search engines for the modeling and validation experiment in a 5-fold cross-validation technique. Experiments demonstrated that the proposed strategy was successful with 83.39% accuracy in ripeness categorization detection [3].

Then too, another study discussed the GLCM and HSV methods to do texture analysis. The findings of the GLCM and HSV computations can be classed to estimate the maturity level of tomatoes using K-NN, a basic and easy classification algorithm that compares the similarity level of the picture using distance (k). According to research, utilizing 100 data sets with 75 training data and 25 testing data, the greatest accuracy rate of 100% is achieved with a p-value of 9 on GLCM and a membership value (k) of 3 in K-NN. Based on the outcomes of the experiments, they can infer that our proposed method has maximum accuracy [8].

Deep learning is a growing research area in the current years [11][12][13][14]. An article proposed a CNN-ELM classification model for automated tomato maturity grading that combines CNN's automated feature learning capabilities with extreme learning machines' efficiency to conduct rapid and accurate classification even with minimal training data. The suggested CNN-ELM model identified six maturity stages from the test data with a classification accuracy of 96.67% and an F1 score of 96.67%, according to the results [17]. Then, another paper discussed CNN-based classification systems by training and evaluating the model on several augmented datasets and attempting to find the best augmentation strategy for each dataset. The trial findings revealed an average accuracy of 91.9% and a prediction time of fewer than 0.01 seconds. In comparison to other methods,

our solution produced better prediction results in terms of accuracy and time consumption. Furthermore, this is a flexible strategy that may be used in other similar fields [7].

To accelerate classification model performance efficiently, a study contributes an annotated visual dataset of green and reddish tomatoes. Considering robotic platform specifications, only the Single-Shot MultiBox Detector (SSD) and YOLO architectures were considered. The results proved that the system could detect green and reddish tomatoes, even those occluded by leaves. SSD MobileNet v2 had the best performance when compared against SSD Inception v2, SSD ResNet 50, SSD ResNet 101, and YOLOv4 Tiny, reaching an F1-score of 66.15%, a mAP of 51.46%, and an inference time of 16.44 ms with the NVIDIA Turing Architecture platform, an NVIDIA Tesla T4, with 12 GB. YOLOv4 Tiny also had impressive results, mainly concerning inferring times of about 5 ms [18]. Then, there was a paper discussing a computer vision system developed to identify and classify the various stages of maturity in tomatoes using deep learning. Here we have evaluated three neural network architectures, namely AlexNet, GoogleNet, and ResNet-50. Based on the evaluation, it was concluded that ResNet-50 performs better in terms of training time and accuracy. The developed system comprises ResNet-50 integrated with YOLO v2 and is tested in real-time. The accuracy rate for the detection of tomatoes in real-time using the proposed system is 90% [25].

The precise and accurate assessment of the maturity level of tomatoes using machine learning algorithms is still quite challenging to achieve due to these algorithms being reliant on hand-crafted features. Hence, in this paper, we propose a deep learning-based tomato maturity grading system that helps to increase the accuracy and adaptability of maturity grading tasks with less amount of training data. The performance of a proposed system is assessed on the real tomato datasets collected from the open fields using a Nikon D3500 CCD camera. The proposed approach achieved an average maturity classification accuracy of 99.8 %, which seems to be quite promising in comparison to the other state-of-the-art [23].

Another paper presented a method that detects tomato fruits grown in hydroponic greenhouses using the Faster R-CNN (region-based convolutional neural network). They trained the detection model using an open dataset and tested the whole system in real-time in a hydroponic greenhouse. A total of 53 tomato fruits were used to verify the developed system, and the developed system achieved 88.6% detection accuracy when completely obscured fruits not captured by the camera were included. When excluding obscured fruits, the system's accuracy was 90.2%. For the maturity classification, we conducted qualitative evaluations with the assistance of experts [22]. Then a paper where they build a tomato detection model to automatically detect intact green tomatoes regardless of occlusions or fruit growth stage using deep learning approaches. The tomato detection model used a faster region-based convolutional neural network (R-CNN) with Resnet-101 and transferred learned from the Common Objects in Context (COCO) dataset. The detection on the test dataset achieved a high average precision of 87.83% (intersection over union ≥ 0.5) and showed a high accuracy of tomato counting ($R2 = 0.87$) [24].

Based on the paper review, we present a novel model to deal with the classification of tomato maturity levels by proposing a new method based on photo-based tomato datasets to solve this problem. To carry out our experiment, we collected a large dataset, including photos of tomatoes consisting of ripe, undercooked, and unripe classes.

3. Proposed Method

A. Problem Definitions

This study uses the RNN algorithm and tomato dataset to focus on the classification of tomato maturity levels. The purpose of this research is to integrate an RNN layer into ConvNet models. This study proposes several RNNs and presents the computational formulas. The concept of training a model to learn predictions from multiple individual models and the design of such a model. LSTM extends RNN by adding three gates to an RNN neuron: a forget gate f to control whether to forget the current state; an input gate i to indicate if it should read the input; an output gate o to control whether to output the state. These gates enable LSTM to learn long-term dependency in a sequence and make it easier to optimize because these gates help the input signal to effectively propagate through the recurrent hidden states $r(t)$ without affecting the output [18][19].

B. Proposed Method

The RNN algorithm classifies tomato ripeness using a dataset in the form of images. Several things that are different from previous research. Most studies use RNN for sequential and time-series data. Except for rare cases, RNNs are used mostly to generate a sequence of image pixels. Therefore, the researcher has the main idea, which is to integrate the RNN layer into the ConvNet model. The researcher proposes several RNNs and presents a computational formula. The concept of training a model to learn the predictions of several individual models and the design of such a model is also discussed [18].

LSTM also effectively deals with the gradient vanishing/exploding issues that commonly appear during RNN training.

$$\begin{aligned}
 x_t &= \delta(U_r \cdot r(t-1) + U_w w_k(t)) \\
 i_t &= \delta(U_i \cdot r(t-1) + U_{i_w} w_k(t)) \\
 f_t &= \delta(U_f \cdot r(t-1) + U_{f_w} w_k(t)) \\
 o_t &= \delta(U_o \cdot r(t-1) + U_{o_w} w_k(t)) \\
 r(t) &= f_t \odot r(t-1) + i_t \odot x_t
 \end{aligned} \tag{1}$$

Where $\delta(\cdot)$ is an activation function, \odot is the product with gate value, and various W matrices are learned parameters. In our implementation, we employ rectified linear units (ReLU) as the activation function.

The RNN model is employed as a compact yet powerful representation of the label co-occurrence dependency in this space. It takes the embedding of the predicted label at each time step and maintains a hidden state to model the label co-occurrence information. The a priori probability of a label given the previously predicted labels can be computed according to their dot products with the sum of the image and recurrent embeddings. The probability of a prediction path can be obtained as the product of the a-prior probability of each label given the previous labels in the prediction path.

A label k is represented as a one-hot vector $e_k = [0, \dots, 0, 1, 0, \dots, 0]$, which is 1 at the k -th location, and 0 elsewhere. The label embedding can be obtained by multiplying the one-hot vector with a label embedding matrix U_l . The k -th row of U_l is the label embedding of the label k .

$$w_k = U_l \cdot e_k. \tag{2}$$

The dimension of w_k is usually much smaller than the number of labels. The recurrent layer takes the label embedding of the previously predicted label and models the co-occurrence dependencies in its hidden recurrent states by learning non-linear functions:

$$o(t) = h_o(r(t-1), w_k(t)), r(t) = h_r(r(t-1), w_k(t)) \quad (3)$$

Where $r(t)$ and $o(t)$ are the hidden states and outputs of the recurrent layer at the time step t , respectively, $w_k(t)$ is the label embedding of the t -th label in the prediction path, and $h_o(\cdot)$, $h_r(\cdot)$ are the non-linear RNN functions.

The output of the recurrent layer and the image representation is projected into the same low-dimensional space as the label embedding.

$$x_t = h(U_o^x o(t) + U_I^x I) \quad (4)$$

Where U_o^x and U_I^x are the projection matrices for recurrent layer output and image representation, respectively. The number of columns of U_o^x and U_I^x is the same as the label embedding matrix U_I . I is the convolutional neural network image representation. The learned joint embedding effectively characterizes the relevance of images and labels.

Finally, the label scores can be computed by multiplying the transpose of U_I and x_t to compute the distances between x_t and each label embedding.

$$s(t) = U_I^T x_t. \quad (5)$$

The predicted label probability can be computed using SoftMax normalization on the scores [19].

4. Experimental Setup

A. Main Idea

The main goal of this paper is to create a classification model based on diameter size and average RGB value to classify tomato ripeness using the RNN algorithm. RNNs learn to recognize image features over time. They extract the required features from the input for further classification. They are designed to deal with problems with division sequences. Due to the presence of a feedback loop in the repeating layer, the use of a sequence-to-sequence classification method like RNN additionally brings the benefit of being able to classify measurements with a variable observation time which, on average, is notably less than the state-of-the-art. Another advantage of the LSTM-RNN is that it can also be used with measurements that include transitions between classes over time [9].

B. Dataset

In this study, we collect a dataset from Kaggle.com with an amount of 200. We separate the dataset into two parts to create the model: the training dataset to construct or train the model and the testing dataset to evaluate our model's performance or accuracy. In this study, we divide the dataset by 80% data for training and 20% data for a testing dataset. The table shows the details of the dataset distribution table used in this study as follows:

Table 1. Detail of the dataset

Dataset	Sample
Dataset Training	160
Dataset Testing	40
Total	200

C. Pre-Processing Data

In this phase, we conduct preprocessing by looking at the data type of each variable in the dataset and see if there is NA data or blank data. If there is empty data, it is necessary to handle missing values. However, in this study, there is no open data. In each variable, it is filled in for the data type. There are two types of float data, three types of integer data, and one object or string. The next step is to vectorize using the label encoding and feature scaling methods. The label encoding method is used to convert raw data into a vector, while feature scaling is used to normalize data so that the data storage is not too ample. To minimize errors, normalization is carried out on the dataset by changing the actual value to a range interval value [0,1] to get the best predictive value. The normalization technique used is the mix-max scaler. As for the min-max scaler normalization formula is:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (6)$$

Table 2 Normalization

Variable	Information
x'	Normalized value
x	The actual value of the data to be normalized
x_{min}	Actual data's minimum value
x_{max}	Actual data's maximum value

D. Classification Method

To conduct our study, we gather datasets in three categories: ripe, semi-cooked, and raw. After the dataset is collected, the next step is vectorization, which is converting the raw data into vectors using label coding and feature scaling methods. Then, following the preprocessing stage, we compute the acquired features to train the classifier model. The dataset is divided into two parts during the feature extraction process: training and testing. The training dataset is used to develop or train a model that can categorize different citrus varieties, while the test dataset is used to evaluate the performance or accuracy of the model. To achieve the best training model classifier, we changed many parameters to obtain the best accuracy value for training. After that, the model is entered into the RNN algorithm.

5. Result and Analysis

A. Classification Test

In this phase, we calculate the performance of our model by considering several indicators. Fig. 1. shows the movement of model accuracy and model loss.

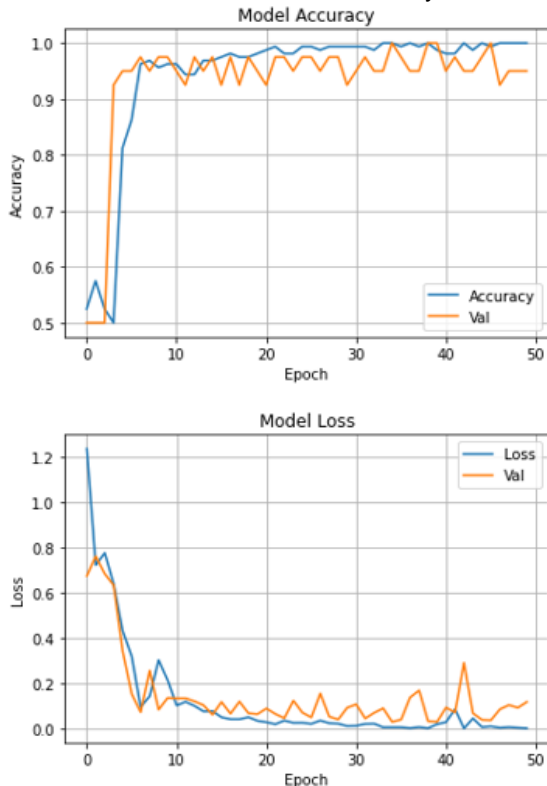


Fig.1 Model accuracy and model loss

In this experiment, we obtained the trade-off between accuracy and performance time by adjusting various hyperparameters to get the best network performance. We set epoch = 50, batch size = 64, and validation split = 0.2 during the training and testing phase. Based on the classification test, our proposed model can classify with an accuracy rate of 95.0%. With an accuracy of 95.0%, the model made is good enough to help classify or distinguish the level of tomato ripeness, as can be seen from the training accuracy and training loss in the image below.

B. Evaluation Metrics

1. Classification Report

Table 3 The result of the classification report

Classification Report	Precision	Recall	F1-Score	Support
Ripe	0.50	0.50	0.52	20
Raw	0.50	0.45	0.47	20
Accuracy	-	-	0.50	40
Macro avg	0.50	0.50	0.50	40
Weighted avg	0.50	0.50	0.50	40

The classification model is used to measure the predictive quality of the classification algorithm. From the results above, we get the accuracy, precision, and recall values of 0.95. then why calculate the f1-score the f1-score is the harmonic mean of precision and recall. The best value of the f1-score is 1.0, and the worst value is 0. Representationally, if the f1-score has a better score, it indicates that our classification model has good precision and recall.

2. Confusion Metrics

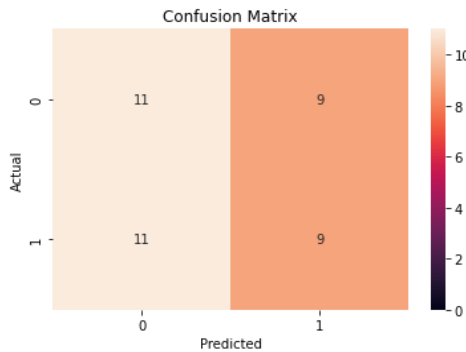


Fig.2 Confusion Matrix of Proposed Model

Fig.2 depicts TP = 11, where the original status is Ripe (class 0). It turns out that our model is successful in declaring that the group is Ripe. TN managed to predict as many as 9, where the original status is Raw (class 1). It turns out that our model is successful in declaring that the group is Raw. Our model succeeded in predicting 11 Ripe and 9 Raw, a total of 20.

6. Conclusion

Traditional fruit classification techniques rely on visual abilities or conventional image processing methods. However, there is drawbacks in identifying a large amount dataset. To solve this problem, we build a classification model using the RNN method to classify tomato ripeness efficiently. In this experiment, we collect large datasets, perform preprocessing, train our model by setting parameters to get the highest accuracy results, and then test the model using new data. The proposed model can perform a trade-off between accuracy and performance time by adjusting various hyperparameters. We tune several hyperparameters to improve the neural network's performance by setting epoch = 50, batch size = 64, and validation split = 0.2. Based on the classification test table, the model can obtain an accuracy rate of 95.0%. Based on the evaluation metrics table, our model can produce TP = 11 and TN = 9. Therefore, the proposed model can be a promising solution to deal with huge fruit classification problems.

Future work can adopt other dynamic algorithms to improve this model using new approaches such as the DBN algorithm, which can be combined with new techniques such as creating new hyperparameters and creating new regulators to train the network. To improve the dynamic learning model, the architecture can tune hyperparameters to produce more efficient accuracy with huge features. Thus, the dynamic model can be a prospective choice for increasing prediction capabilities.

References

- [1] P. Wan, A. Toudeshki, H. Tan, and R. Ehsani, "A methodology for fresh tomato maturity detection using computer vision," *Journal of Computers and Electronics in Agriculture*, 2018.
- [2] J. B. U. Dimatira et al., "Application of fuzzy logic in recognition of tomato fruit maturity in smart farming," in *IEEE Region 10 Conference (TENCON)*, 2016.
- [3] M. B. Garcia, S. Ambat, and R. T. Adao, "Tomayto, tomahto: A machine learning approach for tomato ripening stage identification using pixel-based color image classification," in *IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 2019.
- [4] H. E. David, K. Ramalakshmi, R. Venkatesan, and G. Hemalatha, "Tomato leaf disease detection using hybrid CNN-RNN model," *Journal of Smart Computing and Communication Technology*, 2021.
- [5] A. F. M. Agarap, "Statistical analysis on e-commerce reviews, with sentiment classification using bidirectional recurrent neural network," *Journal of Computation and Language (CL)*, 2018.
- [6] W. D. N. Pacheco and F. R. Jimenez Lopez, "Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms K-NN, MLP, and K-means clustering," in *Journal of Symposium in Image, Signal Processing and Artificial Vision (STSIVA)*, 2019.
- [7] L. Zhang, J. Jia, G. Gui, X. Hao, W. Gao, and M. Wan, "Deep learning based improved classification system for designing tomato harvesting robot," *Journal of Special Section on AI-Driven Big Data Processing, Theory, Methodology, and Applications*, 2018.
- [8] O. R. Indriani, E. J. Kusuma, C. A. Sari, E. H. Rachmawanto, and D. R. I. M. Setiadi, "Tomatoes classification using K-NN based on GLCM and HSV color space," in *International Conference on Innovative and Creative Information Technology (ICITech)*, 2017.
- [9] G. Klarenbeek, R. I. A. Harmanny, and L. Cifola, "Multi-target human gait classification using LSTM recurrent neural networks applied to micro-Doppler," in *European Radar Conference (EURAD)*, 2017.
- [10] H. Liu and I. Lee, "End-to-end trajectory transportation mode classification using Bi-LSTM recurrent neural network," in *12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 2017.
- [11] H. L. Huang, M.-Y. Cheng, and T.-Y. Huang, "A rapid base parameter physical feasibility test algorithm for industrial robot manipulator identification using a recurrent neural network," *IEEE Access*, vol. 11, pp. 145692–145705, 2023.
- [12] A. D. Ronaldo, "Effective soil type classification using convolutional neural network," *Int. J. Informatics Comput.*, 2021.
- [13] M. Fahrurrogi and A. Witanti, "Decision support system design to determine the feasibility of students staying in the dormitory using the simple additive weighting method," *IJIMATIC*, vol. 1, no. 1, pp. 40–52, Nov. 2024.
- [14] H. Purwanto and I. Susilawati, "Web-based expert system chatbot determines the best cheap camera for new users using the Vikor method," *IJIMATIC*, vol. 1, no. 1, pp. 29–39, Nov. 2024.
- [15] A. Kusuma, D. R. I. M. Setiadi, and M. D. M. Putra, "Tomato maturity classification using Naïve Bayes algorithm and histogram feature extraction," *Journal of Applied Intelligent System*, 2018.
- [16] E. E. Lavindi et al., "Neural network based on GLCM, and CIE Lab* color space to classify tomatoes maturity," in *International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2019.
- [17] S. A. Magalhães et al., "Evaluating the single-shot multibox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse," *Journal of Sensors*, 2021.
- [18] J. Wang et al., "CNN-RNN: A unified framework for multi-label image classification," *Journal of Computer Vision Foundation (CVF)*, 2016.
- [19] C.-C. Lien, C. Ay, and C.-H. Ting, "Non-destructive test for assessment of tomato maturity," *Journal of Food Engineering*, 2009.
- [20] A. Kusuma, D. R. I. M. Setiadi, and M. D. M. Putra, "Tomato maturity classification using Naïve Bayes algorithm and histogram feature extraction," *Journal of Applied Intelligent System*, 2018.
- [21] J. Blasco, N. Aleixos, and B. Park, "Development of monitoring robot system for tomato fruits in hydroponic greenhouses," *Journal of Agronomy*, 2021.

- [22] P. Das, J. K. P. Yadav, and L. Singh, "Deep learning-based tomato's ripe and unripe classification system," *International Journal of Software Innovation (IJSI)*, 2022.
- [23] Y. Mu, T.-S. Chen, S. Ninomiya, and W. Guo, "Intact detection of highly occluded immature tomatoes on plants using deep learning techniques," *Journal of Sensors*, 2020.
- [24] N. A. Mulla, S. Ravicandran, P. K. Naik, and B. U. Balappa, "Computer vision system to detect maturity of tomatoes in real using deep learning," *Journal of Advances in Distributed Computing and Machine Learning*, 2022.