International Journal of Sustainable Applied Sciences (IJSAS)
Vol.2, No.5, 2024: 405-416

# CNN Algorithm Approach for Classification of Tomato Fruit Maturity Levels

Taufik Hidayat[1*], Muhammad Fatchan[2], Wahyu Hadikristanto[3]
Universitas Pelita Bangsa
**Corresponding Author:** Taufik Hidayat taufik.hidayat100700@gmail.com

A R T I C L E I N F O

*Keywords:* Classification, Tensorflow, CNN

*Received : 03, April*
*Revised : 17, April*
*Accepted: 22, May*



A B S T R A C T

The categorization of tomato maturity is covered in this study, which has important ramifications for the food sector and agriculture. For training efficiency, the approach uses augmentation with adjustments to rescale picture pixel values and shrink image sizes. According to the experiment's findings, accuracy increased by 93% throughout five training epochs. The training and validation graph indicates steady progress, despite the lack of significance in the improvement. Misclassifications that require correction are found during evaluation utilizing the confusion matrix. The study emphasizes that to enhance agricultural production management, flaws in the model must be filled and accuracy must be increased. The amount and diversity of photos in the dataset should be increased, as should the shooting angles and lighting conditions, and hyperparameters should be adjusted for future model performance optimization.

## INTRODUCTION

Many breakthroughs have evolved in this era of rapid technological advancement to make life easier for humans. The use of technology by humans is also growing, particularly in education, communication, and agriculture. Still, very few technological advancements have benefited the agricultural industry, particularly plantations. Despite this, fruits continue to be a lucrative crop because of their wide range of types and ideal climate, which results in an intriguing assortment of fruits (3.pdf, 2020).

Tomato ripeness classification is an important problem in many agricultural and food industry situations. Based on their color shift, tomatoes can be categorized into six maturity degrees, according to Harllee. Tomatoes begin the ripening process as green, uncooked ones, go through a breaking stage where they turn pink or light crimson, and then eventually turn red (de Luna et al., 2020). Consequently, one of the most useful indicators of tomato ripeness and quality is color. The goal of this maturity level rating is to lower the possibility of rotting tomatoes. Thus, the distance of delivery and the level of freshness of such tomatoes have an impact on the distribution of tomatoes (Riska & Subekti, 2016).

Technological developments in computer vision and machine learning have presented exciting prospects for automating the tomato ripeness classification procedure in recent times. Particularly Convolutional Neural Networks (CNNs) have demonstrated outstanding performance in image classification tasks, such as determining the ripeness of fruit. To put it simply, a CNN is a kind of artificial neural network that incorporates a complex matrix's multiplication operation within its structure. A multilayered perceptron processes the input data after features are extracted using this convolution function, producing output (Muhamad Hafiez et al., 2022).

Previous studies employing neural network algorithms to identify the maturity levels of passion fruit based on color features reported an accuracy rate of 94.44% from 30 fruit samples. A 94% accuracy rate was attained in the study by Novan Wijaya and Anugrah Ridwan on the categorization of apple species using the K-Nearest Neighbors approach. The study employed 800 photos as datasets, 600 for training, and 200 for testing (Widodo et al., 2018).

This study provides a CNN architecture-based TensorFlow-based method for classifying tomato ripeness. Deep learning model development and deployment are made easier with TensorFlow, a well-liked open-source machine learning framework. Convolutional Neural Network (CNN) techniques and the TensorFlow library will be used in this job to apply Deep Learning and assess the achievable training accuracy (Nurfita & Ariyanto, 2018). This study sought to explore the possible use of the CNN algorithm in determining the maturity of tomatoes within this framework. It is hoped that this study can reach a high degree of accuracy in differentiating between different levels of tomato ripeness by utilizing CNN's capability to extract features from photos.

## LITERATURE REVIEW

### Tomato

Lycopersicon esculentum Mill, better known as tomatoes, is a fruit that is commonly grown in tropical and subtropical climates. The need for tomatoes is rising, and tomato agriculture is expanding to meet this demand (Pratama et al., 2019). Fruits with hairy surfaces that were originally spherical and green are called tomatoes. Tomatoes have a rather squishy texture and can turn a dazzling pink, crimson, or yellow when ripe (Muhamad Hafiez et al., 2022).

### Image Processing

Fruit identification by color is now feasible with the aid of computers thanks to the development of information technology. Indirect visual observation is used in this computational technique, where the fruit is photographed with a camera and subsequently processed with computer software (Saputra et al., 2023). creation of image processing techniques to use computer technology to categorize tomatoes according to their level of ripeness. Before the development of image processing technologies, tomato ripeness levels were typically determined by hand by people (3.pdf, 2020).

### Classification

The training step starts the classification process, which then moves on to the testing and classification phases. The system learns to identify data that has passed through earlier phases, including segmentation and feature extraction, during the training phase. The training phase of this investigation started with the creation of a folder containing a dataset of tomato images (Humaira, 2021).

### CNN

Deep Learning technologies such as Convolutional Neural Networks (CNN) have a big impact on picture identification. (Arkadia et al., 2021). CNN is now a sophisticated image classification technology thanks to recent advances. CNNs can comprehend feature structure at critical phases for image classification. With more layers, this method enables CNN to analyze more intricate characteristics, enhancing classification accuracy. For tasks involving object identification and object recognition, CNN is thought to be the best model (Putri, 2020). CNNs are employed in supervised learning techniques for data classification. This is because the supervised learning strategy makes use of both desired variables and training data. This method's objective is to classify data into preexisting categories (Surya et al., 2023).

### Tensorflow

Many programming languages are supported by Google's open-source machine learning library TensorFlow. TensorFlow is primarily used in Transfer Learning to analyze Inception-v3 Models so they may be retrained with fresh data, producing classifications that are both computationally quick and highly accurate. TensorFlow is compatible with several operating systems (Nugroho et al., 2020).

## METHODOLOGY

### Data Collection

The data used in this study consisted of images of ripe and unripe tomatoes. Tomato sampling was carried out outdoors using a smartphone camera. Shoots are carried out from above the tomatoes at a distance of 25 cm from the object. Researchers used tomato samples to identify the level of ripeness. The total sample data in this study consisted of 42 images divided into two maturity classes, namely mature class and raw class. The sample data is grouped into three parts, namely test data, training data, and validation data. The training data consisted of 42 images, with 21 mature class images and 21 raw class images. Meanwhile, the test data consisted of 8 images, with 4 mature grade images and 4 raw grade images. The validation data also consists of 8 images, with the same granularity as the test data.

Figure 1. Sample Data

### Modeling

There are multiple steps in the methodology used in this investigation. Four steps will be completed in total. Data collection for tomato images is the first step, which is then followed by data pre-processing, dataset sharing, data training, validation data, and data testing. The CNN model is the next step, which is followed by evaluation and analysis, as shown in Figure 2 below.

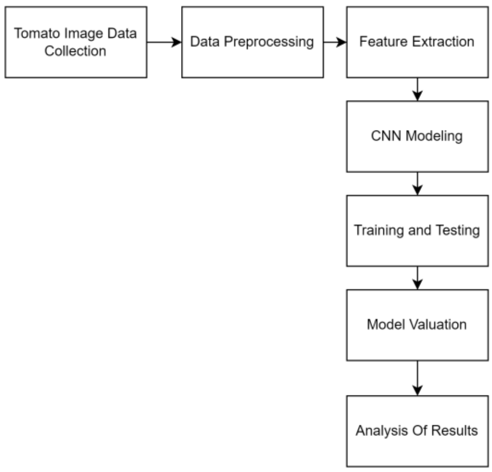


Figure 2. Methods Used

**Pre-processing**

At this stage, we enhance the image using an ImageDataGenerator object from TensorFlow keras. The goal of these augmentations is to lower the likelihood of overfitting and increase diversity in the training set. Among the enhancement strategies used are:

*Augmentation*

In image processing, augmentation is the technique of randomly modifying preexisting images to introduce diversity to training datasets. In our study, we employ augmentation methods by configuring rescale to 1./255. Scale the image's pixel values so that they fall between [0, 1], rotation_range=20: Rotate the picture with width_shift_range=0 up to 20 degrees.2: Apply a height_shift_range=0 horizontal shift to the image, corresponding to 20% of its width. Step 2: Apply a vertical shift to the image such that shear_range=0 and the image height is 20%.2. Apply a 20% angle rotation to the image with zoom_range=0.2: Apply a 20% magnification, horizontal_flip=Indeed: Randomly mirror the image horizontally with fill mode set to "nearest": uses the closest pixel value to fill vacant pixels.

*Resize*

Resize is the process of altering an image's dimensions. The image can be resized or magnified during this process if necessary. To improve the effectiveness of the training process, we utilized resize in our study to change the image's pixel size from 297x398 to 244x244.

**Feature Extraction**

In this stage, we use TensorFlow Hard's ResNet50 function to load a pre-trained ResNet50 model without a top classification layer. Many ImageNet datasets were used to train the model. The input_shape parameter tells the ResNet50 model how big the input image should be. To make sure the model can accept photos of the right size, this step is required.

**Building the Model**

*Stop Training an Existing Layer*

The loaded ResNet50 model's layers are examined one at a time in this step, and each layer's potential for training is set to inactive (False). To prevent any changes to the parameters in the existing layers during the model training process, this step tries to stop the training process on those layers.

*Addition of Classification Layer*

We add a new classification layer to the output produced by the preloaded ResNet50 model after terminating the training process on the current layers. In this stage, the output is first flattened to create a one-dimensional vector. Next, a Dense layer with 512 units and the ReLU activation function is added. Finally, a Dropout layer is applied to lower the chance of overfitting. Dense with three units make up the last output layer, and the softmax activation function is utilized there. The quantity of classes in the dataset determines its value.

### *Merge Model*

The Model object from TensorFlow Keras is used to integrate the preloaded ResNet50 model with the newly added classification layer. ResNet50 provides the input for this merge, while the recently added classification layer produces the output.

### *Determination of Optimizers, Loss Functions, and Evaluation Metrics*

Adam's optimizer with a learning rate of 0.001, a SparseCategoricalCrossentropy loss function, and accuracy evaluation criteria were used for the model compilation.

## CNN Model Training

### *Model Training*

By feeding training data through the train_generator and adjusting the number of steps per epoch for the train_generator's length, the model is fitted using the fit technique. The example model is configured to be trained for 5 epochs, which is the number of iterations that will be conducted throughout the training process, determined by the epochs setting.

### *Model Validation*

The model is assessed during training with validation data that the test_generator provides. The test generator's length is taken into account when determining the number of evaluation steps per period. It is crucial to track model performance at each iteration's stage and look for any instances of overfitting or underfitting during this phase.

### *Training History Store*

The test generator's validation data is used to assess the model during the training phase. To account for the test generator's duration, the number of evaluation steps each epoch is modified. This step is crucial for tracking model performance during each iteration and spotting potential overfitting or underfitting.

## Model Evaluation

### *Model Formation and Training*

The model had already been developed and trained using the training dataset in phases that were not included in the code. The processes in this approach include defining the model's structure, building the model with the help of suitable optimizers and loss functions, and training the model with training data.

*Model evaluation on test data*

The next stage after training a model is to assess its performance using test data, which is data that has never been seen before. This is done to evaluate the trained model's generalization ability to new data.

*Use of the Evaluate Method*

The previously built model is subjected to the evaluation method in the provided code. Test data in the form of a test_generator generator are accepted as inputs by this method, which yields two values: test_acc (accuracy on test data) and test_loss (loss on test data).

**RESULTS**

***Train a Model***

Table 2. CNN Model Training Results using epoch

| Accuracy | Val_accuracy |
|---|---|
| 0.6333 | 0.8889 |
| 0.9333 | 1.0000 |
| 1.0000 | 1.0000 |
| 1.0000 | 1.0000 |
| 1.0000 | 1.0000 |

Show Evaluation Results in Table 2. The results are printed out for the user to see after determining the test data's correctness and loss. The epochs parameter, which we supply, determines the number of iterations (epochs) to be performed during training. After five epochs of training, the model produced an accuracy result ranging from 0.63% to 1.00% and a val_accuracy result ranging from 0.88% to 1.00% in this case.

*Trmining and Validation Accuracy*

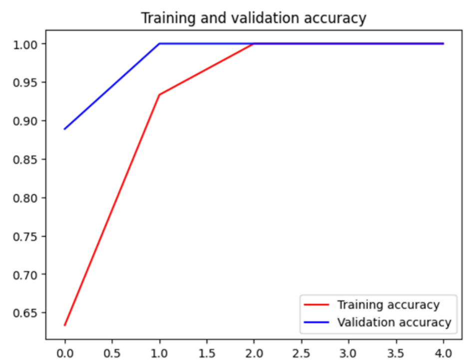


Figure 3. The Graph Above Illustrates the Results of Training Data Processing and Validation Accuracy.

Figure 3 shows graphs that visually illustrate the data from model validation and training. At each iteration (epoch) of the model training process, the graph displays trends in training accuracy and validation accuracy. The training iterations, or epochs, are shown on the x-axis. These graphs help observe how the model's performance changes over time and determine if the model is underfitting or overfitting.

***Training Results Graph***

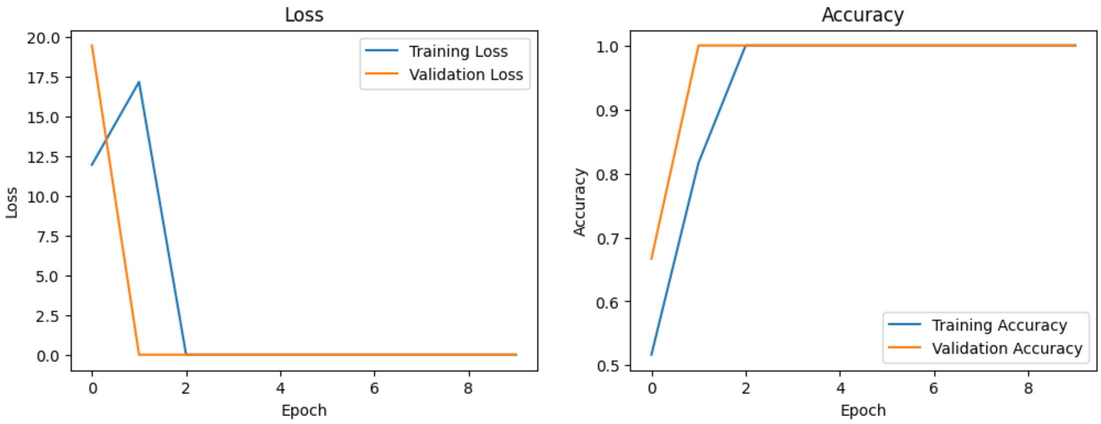


Figure 4. Results of loss value and accuracy

A graph displaying the accuracy values for training and validation data at each epoch is displayed on the right side of the screen, while a graph displaying the loss values for training and validation data at each epoch is displayed on the left (Figure 4).

***Display Precision, Recall, and F1-Score Graphs***

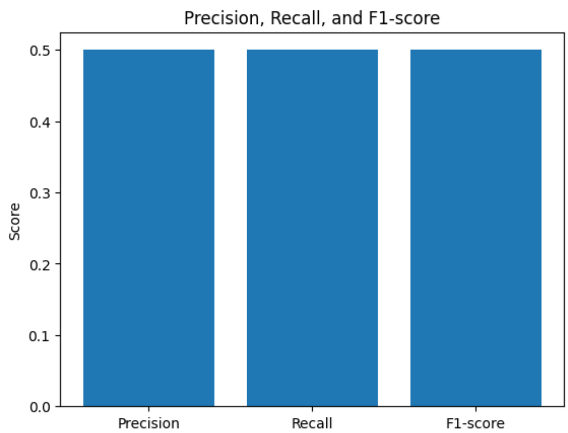


Figure 5. The Figure Shows the Results of Precision, Recall, and F1-Score.

Figure 5 uses the f1_score, recall_score, and precision_score metrics from the matplotlib. pyplot and sklearn.metrics libraries for visualization. The precision, recall, and F1-score are displayed in the graph results. Each measure has a label on the x-axis, and the score value is shown on the y-axis. The outcomes of the model evaluation against precision, recall, and F1-score can be seen in this informative graph.

***Confusion Matrix***

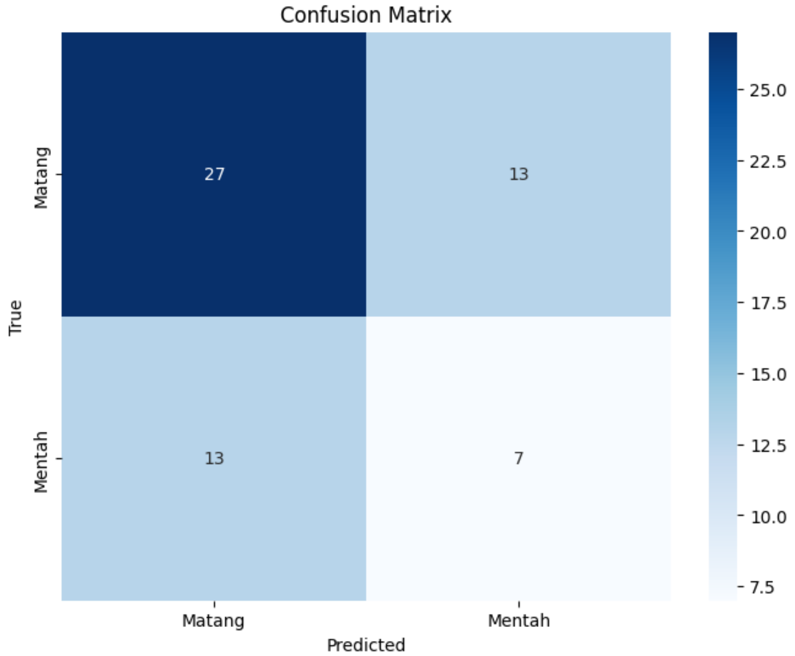


Figure 6. The Figure Illustrates the Results of the Performance Evaluation of the Model Using a Confusion Matrix.

The outcome of assessing the model's performance using the confusion matrix, precision, recall, and F1-score metrics is shown in Figure 6. The model's accuracy in predicting actual classes is shown by the confusion matrix, and its capacity to categorize each class is further elucidated by precision, recall, and F1 scores.

**DISCUSSION**

In many facets of agriculture and food production, the classification of tomato maturity is crucial. Tomatoes can be grown to the ideal maturity level to minimize post-harvest losses, enhance quality control, and streamline the supply chain. This is consistent with the results. (Maulana Alfaruq et al., 2023) that tomatoes (Solanum lycopersicum) are one of the vegetable commodities having a high commercial value and extensive societal use. The quality and longevity of tomato products are significantly influenced by the maturity stage of the tomatoes. Conventional approaches to classifying maturity tend to be manual and subjective, which frequently leads to errors and inefficiencies. The availability of deep learning and machine learning methods offers automatic, impartial, and reliable answers to these issues.

Numerous fields have extensively documented the use of CNN for image classification, showcasing its capacity to extract intricate features from picture input. Using CNN to classify tomato maturity capitalizes on this feature and has the potential to yield more accurate and consistent results than conventional techniques. Based on results from (Appe et al., 2023) Deep Learning techniques employ multilayer architectures to extract different features. Convolutional Neural Networks (CNNs) are Deep Learning techniques that produce impressive results for picture recognition and categorization. But for picture classification, conventional Deep Learning techniques need bigger datasets and more processing time. A lot of scholars suggest using transfer learning techniques to solve this issue.

The results of this investigation show that, despite the constructed CNN model's reasonable performance in determining tomato ripeness, considerable improvement is still possible. In comparison to ripe tomatoes, this model has trouble correctly identifying unripe tomatoes, as evidenced by its lower gains and precision for unripe tomato grades. The quality of the training data, the intricacy of the characteristics that differentiate the phases of ripeness, or the variation in tomato appearance could all be contributing factors to this discrepancy. These results have a practical implication in that, although promising, current models might not be entirely dependable enough to be used in actual agricultural contexts without additional development. Nonetheless, these findings offer a strong basis and illustrate the promise of deep learning in agricultural applications.

## CONCLUSIONS AND RECOMMENDATIONS

With quite good findings, this research has shown the potential of CNN for tomato maturity classification. The study highlights the strengths and weaknesses of the already employed approaches, adding to the body of information regarding the implementation of deep learning techniques in agriculture. Accuracy increases with epochs, according to training and validation graphs, but the gain is not statistically significant. By using the confusion matrix for evaluation, one may visualize the model's performance and see which classification errors still need to be fixed.

To obtain improved accuracy and reliability, future research should concentrate on filling in the gaps that have been found and improving the model even further. In the end, this will lessen post-harvest losses by improving agricultural production management. Increasing the quantity and variety of tomato photos in the dataset can aid in improving the accuracy and learning process of the model. Taking pictures in various lighting and angle configurations can also aid in enhancing the model's generalizability. Additionally, adjusting hyperparameters like learning rate, batch size, and epoch count might aid in further enhancing model performance.

## FURTHER STUDY

With TensorFlow, this study has effectively illustrated how Convolutional Neural Networks (CNN) may be used to classify tomato ripeness. Even if the outcomes are encouraging, there are a few areas that could be improved and the focus of subsequent research, like expanding the dataset, enhancing augmentation methods, and delving deeper into CNN architecture.

## ACKNOWLEDGMENT

We would like to thank everyone who helped us to finish this research project. First and foremost, we would like to express our gratitude to our organization for the support offered in the current facilities. We also appreciate the insightful comments from reviewers and supervisors, which have raised the caliber of our work. We also thank tomato wholesalers and growers for their pertinent information and statistics. Lastly, we would like to express our gratitude to our family and friends for their constant moral support during the research process. Hopefully, the results of our study will significantly influence the development of agricultural technology, especially in the distribution and handling of fruit.

## REFERENCES

3.pdf, file:///C:/Users/Lenovo/OneDrive/JURNAL D. S. (2020). Klasifikasi Kematangan Buah Tomat Dengan Variasi Model Warna Menggunakan Support Vector Machine. *VOCATECH: Vocational Education and Technology Journal*, *1*(2), 45–52. https://doi.org/10.38038/vocatech.v1i2.27

Appe, S. R. N., Arulselvi, G., & Balaji, G. N. (2023). Tomato Ripeness Detection and Classification using VGG-based CNN Models. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(1), 296–302.

Arkadia, A., Ayu Damayanti, S., & Sandya Prasvita, D. (2021). Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*, *2*(2), 158–165. https://conference.upnvj.ac.id/index.php/senamika/article/view/1813

de Luna, R. G., Dadios, E. P., Bandala, A. A., & Vicerra, R. R. P. (2020). Tomato growth stage monitoring for smart farm using deep transfer learning with machine learning-based maturity grading. *Agrivita*, *42*(1), 24–36. https://doi.org/10.17503/agrivita.v42i1.2499

Humaira. (2021). *408418-Klasifikasi-Tingkat-Kualitas-Dan-Kematan-Cc588905*. *02*, 20–26.

Maulana Alfaruq, B., Erwanto, D., & Yanuartanti, I. (2023). Klasifikasi Kematangan Buah Tomat Dengan Metode Support Vector Machine. *Generation Journal*, *7*(3), 64–72. https://doi.org/10.29407/gj.v7i3.21092

Muhamad Hafiez, T., Iskandar, D., Wiranata S.K, A., & Fitri Boangmanalu, R. (2022). Optimasi Klasifikasi Gambar Varietas Jenis Tomat dengan Data Augmentation dan Convolutional Neural Network. *Smart Comp: Jurnalnya Orang Pintar Komputer*, *11*(2), 175–186. https://doi.org/10.30591/smartcomp.v11i2.3524

Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia. *Algor*, *2*(1), 12–21.

Nurfita, R. D., & Ariyanto, G. (2018). Implementasi Deep Learning berbasis Tensorflow untuk Pengenalan Sidik Jari. *Emitor: Jurnal Teknik Elektro*, *18*(1), 22–27. https://doi.org/10.23917/emitor.v18i01.6236

Pratama, R., Fuad, A., & Tempola, F. (2019). Deteksi Kematangan Buah Tomat Berdasarkan Fitur Warna Menggunakan Metode Transformasi Ruang Warna His. *JIKO (Jurnal Informatika Dan Komputer)*, *2*(2), 81–86. https://doi.org/10.33387/jiko.v2i2.1318

Putri, O. N. (2020). Implementasi Metode CNN Dalam Klasifikasi Gambar Jamur Pada Analisis Image Processing (Studi Kasus: Gambar Jamur Dengan Genus Agaricus Dan Amanita). *Skripsi*, 1–80.

Riska, S. Y., & Subekti, P. (2016). Klasifikasi Level Kematangan Buah Tomat Berdasarkan Fitur Warna Menggunakan Multi-Svm. *Jurnal Ilmiah Informatika*, *1*(1), 39–45. https://doi.org/10.35316/jimi.v1i1.442

Saputra, J., Sa, Y., Yoga Pudya Ardhana, V., & Afriansyah, M. (2023). RESOLUSI : Rekayasa Teknik Informatika dan Informasi Klasifikasi Kematangan Buah Alpukat Mentega Menggunakan Metode K-Nearest Neighbor Berdasarkan Warna Kulit Buah. *Rekayasa Teknik Informatika Dan Informasi*, *3*(5), 196–203. https://www.djournals.com/resolusi/article/view/763

Surya, I., Idris, K., & Bode, A. (2023). Klasifikasi Jenis Buah Tomat Menggunakan Covolutional Neural network. *Jurnal Ilmiah Ilmu Komputer*, *2*(2), 83–89.

Widodo, R., Widodo, A. W., & Supriyanto, A. (2018). Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (Citrus reticulata Blanco) untuk Klasifikasi Mutu. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(11), 5769–5776. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/3420