Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2026. Vol. 5. No. 2; e-ISSN: 2808-4519

Classification of Purple Passion Fruit Ripeness Levels Using Convolutional Neural Network (CNN)

Mochammad Gani Alfa Alkhoiri Siregar[1]*, Said Iskandar Al Idrus [2], Hermawan Syahputra[3], Insan Taufik[4], Kana Saputra S[5]

[1]*Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Negeri Medan, Indonesia*
m.alfa223@gmail.com[1*]

Abstract

Passiflora edulis Sims (purple passion fruit) is a fruit that offers numerous health benefits and possesses high economic value. However, the manual assessment of ripeness by traders tends to be subjective and inconsistent, leading to post-harvest losses of up to 50%. This study developed a classification model for determining the ripeness level of purple passion fruit using a Convolutional Neural Network (CNN) and implemented it in a web-based application. The CNN model was designed to classify four ripeness stages (*unripe, half-ripe, ripe, and rotten*) with the addition of a non-passion-fruit class to enhance the system's robustness. The dataset consisted of 2,000 images divided into five classes: four ripeness levels of purple passion fruit (*unripe, half-ripe, ripe, and rotten*) and one non-passion-fruit class as a comparator. All images were in JPG and PNG formats. The CNN architecture comprised four convolutional layers with 16, 32, 64, and 128 filters, respectively. Evaluation of various data-splitting ratios (80:20, 70:30, 60:40) and learning rates (0.001, 0.0001, 0.01) showed that the optimal configuration was achieved at a ratio of 80:20 with a learning rate of 0.001, resulting in a training accuracy of 96.72% and a testing accuracy of 95.76%, with a loss value of 0.1811. Validation using 5-Fold Cross Validation produced an average accuracy of 95.40%. The model was integrated into a web application developed using Flask and JavaScript, deployed on the PythonAnywhere cloud platform, enabling users to upload images and automatically obtain ripeness predictions to assist traders in sorting fruits more quickly and accurately.

Keywords: *Purple passion fruit; Fruit classification; Convolutional Neural Network; Machine learning; Web application*

1. Introduction

Indonesia is an agrarian country with a strong agricultural sector and diverse commodities that have the potential to produce various types of fruits [1]. One of the fruits cultivated in Indonesia is the purple passion fruit (*Passiflora edulis Sims*), which is known for its numerous health benefits and is commonly consumed both in fresh and processed forms [2]. Purple passion fruit contains high levels of vitamin A, vitamin C, beta-carotene, and flavonoids, which provide strong antioxidant effects to prevent LDL oxidation and reduce the risk of atherosclerosis formation [3]. In addition, extracts from the peel of purple passion fruit exhibit effective antibacterial activity against *Staphylococcus* aureus and *Escherichia coli* [4].

Owing to its various benefits, purple passion fruit is widely used as a primary raw material in the juice and beverage processing industry. Thus, consuming purple passion fruit has great potential in preventing oxidative damage and offering significant health benefits [5]. However, despite its many advantages, data from Statistics Indonesia (Badan Pusat Statistik) show that passion fruit production declined from 59,270 tons in 2018 to 53,319 tons in 2020. This decrease indicates issues in the passion fruit supply chain, one of which is suboptimal post-harvest handling. Post-harvest losses in fruits can reach up to 50%, depending on the commodity type, due to factors such as improper handling and inaccurate determination of fruit ripeness [6]. This highlights an urgent need to improve accuracy in classifying the ripeness level of purple passion fruit, as errors in ripeness determination can significantly contribute to post-harvest losses.

Proper post-harvest handling is crucial to maintaining the quality and market value of fruits. During the ripening process, the skin of purple passion fruit changes from green to dark purple, accompanied by alterations in texture and taste [7]. Studies have shown that purple passion fruits harvested at the appropriate ripeness stage yield optimal fruit weight, juice content, total soluble solids, and total acidity [8]. Therefore, an accurate and efficient method is required to classify the ripeness levels of purple passion fruit.

The issue of classifying the ripeness level of purple passion fruit has become highly crucial, particularly for new traders who are just beginning their business in the purple passion fruit trade. New traders generally lack sufficient experience and knowledge to visually determine the ripeness level of purple passion fruit. This limitation often leads to errors in ripeness assessment, resulting in financial losses, either due to inaccurate stock purchases or consumer rejection caused by poor fruit quality. Conventional methods for assessing fruit ripeness typically involve visual inspection and manual testing, which can be subjective and time-consuming [9]. In the digital era, technology-based approaches such as image processing and machine learning provide more accurate and efficient solutions.

The use of image processing technologies such as the Convolutional Neural Network (CNN) in the agricultural field has shown promising results in various studies, particularly in accurately detecting plant pests and diseases [10]. This technology enables a more in-depth analysis of visual features of fruits, such as color, texture, and shape, which can be utilized to determine ripeness levels [11]. Previous studies have demonstrated that CNN-based methods are highly effective for fruit ripeness classification. For instance, a study on pineapple ripeness classification achieved an accuracy of 99.4% on training data and 92.4% on validation data [12]. Moreover, a study that employed CNN for passion fruit ripeness classification achieved an accuracy of 91.52% [13].

CNN is one of the most effective methods in artificial intelligence for pattern recognition and image classification tasks [14]. It has been widely used in various applications, including face recognition and object detection [15]. The ability of CNN to recognize complex visual features makes it an ideal tool for classifying the ripeness levels of purple passion fruit. In this study, the CNN model was trained using an image dataset of purple passion fruit at four ripeness levels: *unripe, half-ripe, ripe, and rotten*.

After the CNN model was successfully developed and tested, the next step was to implement it in the form of a web application. A website is a collection of information that can be accessed online via the Internet, allowing anyone to access it from various locations and at any time [16]. The implementation of the CNN model in this web application is expected to assist purple passion fruit sellers in identifying fruit ripeness levels more accurately and efficiently. With this technology, the sorting process of purple passion fruit can be carried out faster and more precisely, enabling new sellers to reduce financial losses and build consumer trust from the early stages of their business.

Based on this background, this study aims to develop an effective CNN model for classifying the ripeness levels of purple passion fruit with high accuracy and to implement it in a web-based application that is easily accessible and user-friendly for new purple passion fruit sellers.

2. Literature Review

A study developed a passion fruit detection and ripeness classification system using RGB-Depth images [13]. The system utilized a CNN to classify three ripeness levels of passion fruit with an accuracy of 91.52%. Although this research demonstrated the potential of CNN for passion fruit classification, the developed system required a special RGB-Depth camera, which is costly and adds complexity to implementation. Moreover, the study did not integrate the system into an application that could be widely accessed by end users such as sellers or farmers.

Another study implemented CNN for classifying the ripeness and size of pineapples using an Android-based system [12]. This research achieved 99.4% accuracy on training data and 92.4% on validation data using a CNN architecture consisting of five convolutional layers. The developed system was successfully implemented in a mobile application that facilitated user classification activities. However, this study focused on pineapples, which have different visual characteristics compared to purple passion fruit, and it did not employ cross-validation to test the model's reliability across data variations.

A study on the classification of cayenne pepper ripeness levels using a combination of CNN and K-Nearest Neighbor methods has also been conducted [17]. The research evaluated various data split ratios (60:40, 70:30, and 80:20) to determine the optimal configuration. The results showed that the 80:20 ratio provided the best performance with an accuracy of 94.3%. Although this study offered insight into the importance of data ratio selection, the method used combined two algorithms, which increased computational complexity and training time.

The optimization of CNN accuracy for classifying the quality of green apples has been investigated using a CNN architecture with four convolutional layers, achieving an accuracy of 95.8% [18]. The study also explored various hyperparameters to find the optimal configuration. However, this research focused only on quality classification (good/bad) without considering detailed ripeness levels and did not implement the system into a practical application.

A classification system for passion fruit ripeness levels using an Artificial Neural Network based on digital image processing was previously developed by employing RGB color feature extraction, achieving an accuracy of 87.5% [19]. Although this study specifically addressed passion fruit, the use of a conventional Artificial Neural Network with manual feature extraction is limited in capturing complex features compared to CNN, which can automatically extract hierarchical features.

The implementation of CNN in a mobile application for signature identification demonstrated that CNN can perform well on mobile devices and provide fast response times [20]. However, the context of this research differs from fruit classification, and it did not include a discussion on model validation using methods such as K-Fold Cross Validation.

Based on the review of previous studies, several research gaps can be identified. First, studies on purple passion fruit ripeness classification using CNN remain limited, particularly those utilizing simple RGB images without requiring special devices. Second, most studies have not implemented cross-validation methods such as K-Fold Cross Validation to ensure model reliability across data variations. Third, the implementation of the system into a web-based application that can be widely accessed is still rarely conducted, even though accessibility is an important factor for technology adoption among end users such as new fruit sellers.

This study addresses the identified research gaps by developing a CNN model for classifying four ripeness levels of purple passion fruit (unripe, half-ripe, ripe, and rotten) along with an additional non-passion fruit class to enhance system robustness. The model was validated using 5-Fold Cross Validation to ensure reliability and generalization capability. Furthermore, this research implements the model into a web-based application built with the Flask framework and deployed on a cloud platform, allowing new purple passion fruit sellers to access it anytime and anywhere using simple devices such as smartphones. Thus, this study contributes a practical and accessible solution to the problem of purple passion fruit ripeness classification faced by sellers, particularly those who are new and lack sufficient experience.

3. Methodology

3.1. Research Stages

This study employs the Research and Development (R&D) method, which aims to develop and test a CNN-based classification model. The research stages are systematically designed to achieve the goal of developing a purple passion fruit ripeness classification model and implementing it into a web application. The flowchart of the research stages is presented in Fig. 1.

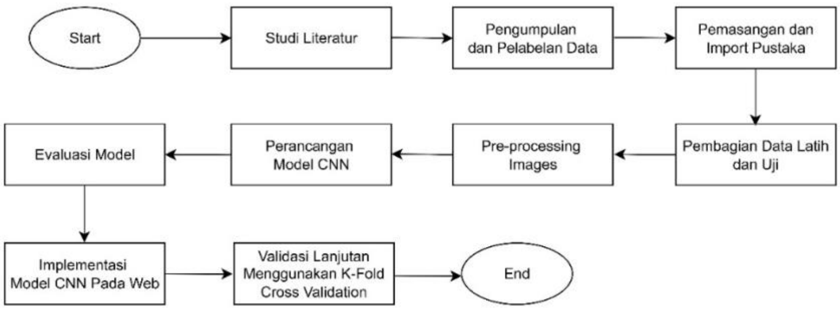

Fig. 1: Flowchart of the research stages

The research process begins with a literature review to explore relevant problems and identify suitable methods. It is then followed by data collection, preprocessing, CNN model development, model evaluation, validation using K-Fold Cross Validation, and finally, the implementation of the model into a web application.

3.2. Data Collection

The data used in this study are primary data obtained directly through the image acquisition process of purple passion fruits at Pasar Raya MMTC and Pasar Besar Medan. Data collection was carried out using the rear camera of a Redmi 12 smartphone with a resolution of 50 megapixels. Images were captured with a white background at a distance of approximately 20–30 cm from the front, back, and side views of the fruit under adequate lighting conditions. The captured images were saved in JPEG (Joint Photographic Experts Group) format.

The dataset consists of 1,600 images of purple passion fruits classified into four ripeness categories: unripe, half-ripe, ripe, and rotten. Each category contains 400 images, consisting of 200 JPG images and 200 PNG images obtained after background removal. To enhance system robustness, an additional non-passion fruit class containing 400 images of other objects was added, sourced from a public dataset available on the Kaggle platform. The total dataset thus comprises 2,000 images across five classes.

The labeling process was conducted by grouping each image into folders according to its ripeness category, based on visual observation and references from purple passion fruit sellers. This organized folder structure facilitated data reading during the model training process.

3.3. Training and Testing Data Split

The dataset was divided using three ratio variations, namely 80:20, 70:30, and 60:40, utilizing the Scikit-learn library. The data splitting was conducted in a stratified manner, meaning that each class maintained a balanced proportion in both the training and testing sets. This approach aimed to ensure that the model learned from a well-represented dataset and that evaluation was performed objectively. The ratio that produced the highest accuracy was selected as the optimal configuration.

3.4. Data Pre-processing

Data pre-processing was carried out to adjust the dataset according to the input requirements of the CNN model. This process consisted of three main stages. First, the images were resized to 224×224 pixels to accelerate training time and align with the standard CNN input size. Second, normalization was performed by converting pixel values from the range of 0–255 to 0–1 by dividing each pixel value by 255. Third, data augmentation was applied to artificially increase the diversity of the training data, including rotation, width and height shifting, zooming, and horizontal flipping.

3.5. CNN Model Design

The CNN model was designed based on the architecture adapted from a previous study on apple classification [18]. The architecture consists of four convolutional layers with 16, 32, 64, and 128 filters, respectively. Each convolutional layer uses a 3×3 kernel followed by a ReLU activation function and a MaxPooling layer with a 2×2 pool size. After the convolutional layers, a Flatten layer is added to convert the output into a one-dimensional vector, followed by a Dense layer with 5 output neurons corresponding to the number of purple passion fruit classes and a Softmax activation function for multi-class classification. The CNN model architecture used in this study is shown in Table 1.

Table 1: CNN Model Architecture

Layer	Kernel	Filter
Conv1	3×3	16
MaxPooling1	2×2	-
Conv2	3×3	32
MaxPooling2	2×2	-
Conv3	3×3	64
MaxPooling3	2×2	-
Conv4	3×3	128
MaxPooling4	2×2	-
Flatten	-	-
Dense	-	5

The model was compiled using the *Categorical Cross-Entropy* loss function, the *Adam optimizer*, and accuracy as the evaluation metric. Several learning rate values were tested, namely 0.001, 0.0001, and 0.01, to determine the optimal configuration. The training process was carried out for 100 epochs.

3.6. Model Evaluation

The model evaluation was carried out by measuring the accuracy and loss values on both the training and testing datasets. Accuracy was calculated using the formula of the number of correct predictions divided by the total number of data samples, while the loss was calculated using the Cross-Entropy Loss function. The combination of data ratio and learning rate that produced the highest accuracy and the lowest loss was selected as the best model configuration.

3.7. Validation Using K-Fold Cross Validation

To ensure the model's reliability and generalization capability toward new data, an additional validation was conducted using the 5-Fold Cross Validation method. The dataset was divided into five subsets of equal size. In each iteration, four subsets were used as training data and one subset as testing data. This process was repeated five times so that each subset served as the testing data once. The accuracy and loss values obtained from each fold were then averaged to determine the overall model performance.

3.8. Model Implementation on the Web

The best-trained model was saved in the .h5 format and implemented as an API using the Flask framework. The backend was built with Python Flask to load the model and handle prediction requests, while the frontend was developed using HTML, CSS, and JavaScript to display the user interface and prediction results. The web application was deployed on the PythonAnywhere cloud platform, allowing users to access it online from various locations and devices.

4. Results & Discussion

4.1. Data Collection and Labeling

The data collection process produced a dataset consisting of 2,000 images divided into five classes. The initial dataset included 800 images of purple passion fruits captured directly from markets with various viewing angles and daylight lighting conditions. Each ripeness class (unripe, half-ripe, ripe, and rotten) contained 200 images in JPG format. To increase data variation, background removal was performed on each image using the Rembg library, generating an additional 200 PNG images for each class. Thus, each ripeness class contained a total of 400 images.

To enhance the system's robustness in distinguishing passion fruits from other objects, a non-passion fruit class containing 400 images of non-passion fruit objects was added, sourced from a public dataset available on the Kaggle platform. The addition of this class aimed to enable the model to reject predictions for non-passion fruit images, thereby improving system reliability in real-world applications. The final dataset distribution consisted of 400 images for each of the five classes (unripe, half-ripe, ripe, rotten, and non-passion fruit), with a total of 2,000 images used for model training and testing.

The labeling process was conducted manually by grouping images into folders according to their ripeness category. The ripeness levels were determined based on visual observation and consultation with experienced fruit sellers at the market. The organized folder structure facilitated data reading using the ImageDataGenerator function from the Keras library.

4.2. Data Splitting and Pre-processing

The dataset was divided into training and testing data using three ratio variations: 80:20, 70:30, and 60:40. In the 80:20 ratio, the training data consisted of 1,600 images and the testing data of 400 images. In the 70:30 ratio, the training data consisted of 1,400 images and the testing data of 600 images. In the 60:40 ratio, the training data consisted of 1,200 images and the testing data of 800 images. The division was carried out in a stratified manner to ensure that each class maintained a balanced proportion in both subsets.

The pre-processing process consisted of three main stages. The first stage was resizing the images to 224×224 pixels to match the standard CNN input size and accelerate training time. The second stage was normalization, in which pixel values were scaled from the range of 0–255 to 0–1 to accelerate model convergence. The third stage was data augmentation applied to the training data to increase variation and reduce overfitting. The augmentation techniques used included rotation up to 20 degrees, horizontal and vertical shifting up to 20%, zooming up to 20%, and horizontal flipping.

4.3. CNN Model Training Results

The CNN model was trained using various combinations of data split ratios and learning rates to determine the optimal configuration. Each combination was trained for 100 epochs using the Adam optimizer and the Categorical Cross-Entropy loss function. The training results from the nine configurations are presented in Table 2.

Table 2: Model Training Results with Various Configurations

Rasio	Learning Rate	Training Accuracy	Training Loss	Testing Accuracy	Testing Loss
80:20	0.0001	90.96%	0.3056	88.76%	0.3237
80:20	0.001	96.72%	0.0903	95.76%	0.1811
80:20	0.01	23.08%	1.5504	23.07%	1.5509
70:30	0.0001	91.53%	0.2425	90.51%	0.3324
70:30	0.001	96.92%	0.0683	94.23%	0.2403
70:30	0.01	23.08%	1.5503	23.07%	1.5508
60:40	0.0001	89.03%	0.2863	87.50%	0.3731
60:40	0.001	96.43%	0.0622	95.28%	0.1910
60:40	0.01	23.09%	1.5502	23.07%	1.5508

Based on Table 2, the configuration with an 80:20 data ratio and a learning rate of 0.001 achieved the best performance, with a training accuracy of 96.72% and a testing accuracy of 95.76%, along with a low testing loss of 0.1811. This configuration demonstrates a good balance between the model's learning ability and its generalization capability, as indicated by a small accuracy difference (less than 1%), suggesting that the model did not experience significant overfitting.

The accuracy value was calculated using the formula for the number of correct predictions divided by the total number of samples. For example, in the best configuration (80:20 ratio with a learning rate of 0.001), out of 400 testing samples, the model correctly predicted 383 samples, resulting in a testing accuracy of (383/400) × 100% = 95.76%. Meanwhile, the loss value was calculated using the Cross-Entropy Loss function, which measures the difference between the model's predicted probability distribution and the actual labels. A loss value of 0.1811 indicates a low prediction error and shows that the model effectively learned the data patterns.

A learning rate of 0.01 for all ratios produced very poor performance, with accuracy around 23% and loss values above 1.55. This occurred because an excessively high learning rate caused the optimization process to overshoot the global minimum, preventing the model from converging and learning effectively. Conversely, a learning rate of 0.0001 achieved moderate accuracy (87–91%) but required longer training time to reach convergence. The learning rate of 0.001 proved to provide the best balance between convergence speed and model accuracy.

The training curves of the best-performing model are shown in Figure 2. The graph shows that both training and testing accuracy increased consistently with the number of epochs, while the loss values decreased. The testing accuracy curve follows a similar pattern to the training accuracy without significant fluctuations, indicating that the model learned stably and did not experience overfitting.

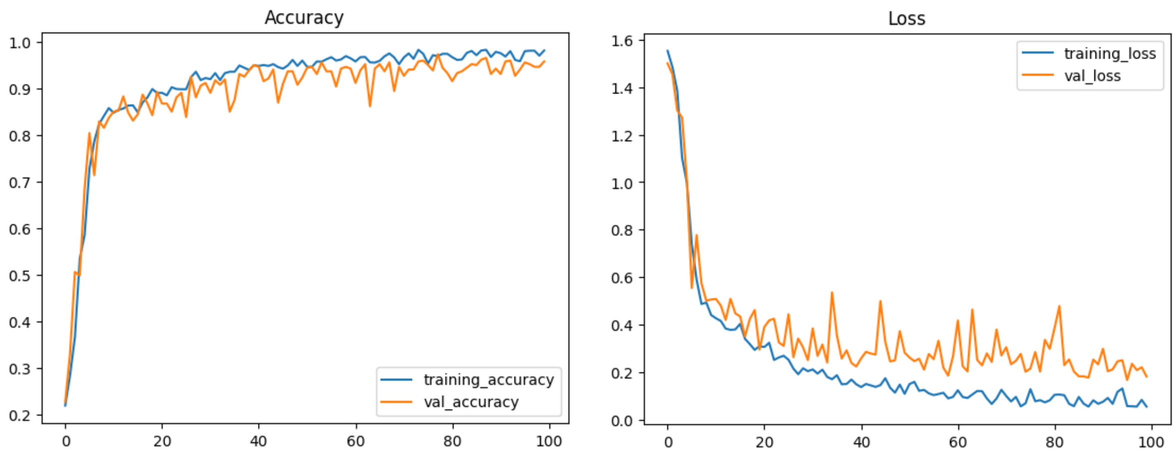

Fig. 2: Accuracy and Loss Curves

4.4. K-Fold Cross Validation Results

To validate the model's reliability against variations in the dataset, an additional evaluation was conducted using the 5-Fold Cross Validation method. The validation results are presented in Table 3.

Table 3: Results of 5-Fold Cross Validation

Fold	Accuracy	Loss
1	96.17%	0.1764
2	97.33%	0.0757
3	96.33%	0.1410
4	95.83%	0.1211
5	91.32%	0.2487
Average	95.40%	0.1526

The K-Fold Cross Validation results show an average accuracy of 95.40% with a relatively small standard deviation, indicating that the model performed stably and consistently across different data splits. Although a slight decrease in accuracy was observed in the fifth fold (91.32%), this difference remains within an acceptable range and may be caused by variations in data distribution within that subset. The relatively low loss values (ranging from 0.07 to 0.24) indicate that the model effectively learned image patterns without experiencing significant overfitting.

A comparison between the evaluation results using the single data split (95.76%) and the 5-Fold Cross Validation (95.40%) demonstrates strong consistency, with a difference of less than 0.5%. This validates that the model possesses high generalization capability and can reliably classify the ripeness levels of purple passion fruits on unseen data.

4.5. Web Application Implementation

The best-trained model was saved in the .h5 format and implemented as a REST API using the Flask framework. The backend was developed with a /predict endpoint that receives image files from users, performs pre-processing (resizing and normalization), and returns the prediction results in JSON format containing the class label and confidence level.

The frontend was developed using HTML, CSS, and JavaScript with a responsive and intuitive user interface. Users can upload purple passion fruit images through an upload button or by using drag-and-drop functionality. Additionally, they can paste an image directly from the clipboard using the Ctrl+V keyboard shortcut. After uploading the image, users can click the "Predict Ripeness" button to obtain the classification results.

The web application was deployed on the PythonAnywhere cloud platform with a public domain accessible from various devices and locations. Deployment on the cloud ensures that the application can be used at any time without requiring special software installation—only a web browser on a smartphone or computer is needed.

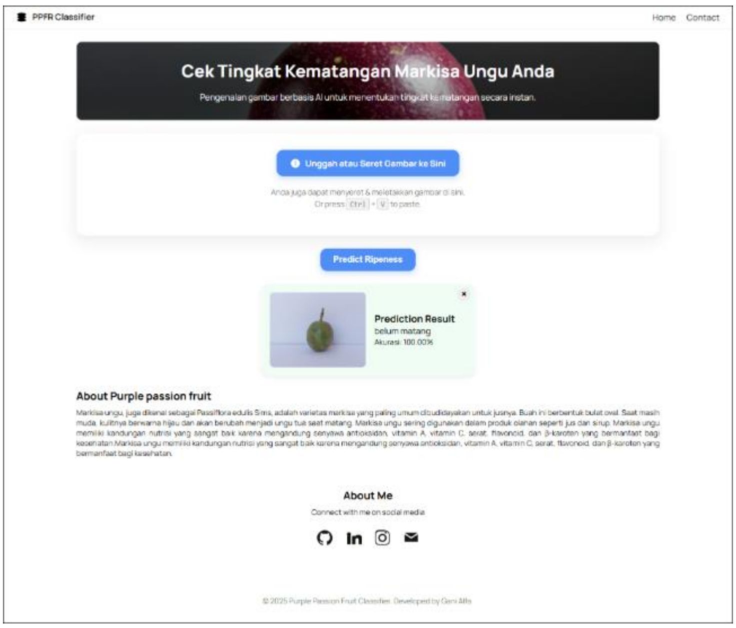

Fig. 3: Web Application Interface

An example of the prediction results from the application shows that the system can accurately classify ripeness levels. For an image of a ripe passion fruit, the system produced the prediction “ripe” with a confidence level of 100.00%. The system was also able to detect non-passion fruit images, demonstrating that the inclusion of the non-passion fruit class successfully improved the system’s robustness.

4.6. System Strengths and Weaknesses

The strengths of the developed system include:

1. High classification accuracy (95.76%) validated using K-Fold Cross Validation.
2. Ability to detect non-passion fruit objects, enhancing system reliability.
3. A web application implementation that is easily accessible without requiring special installation.
4. Antarmuka pengguna yang intuitif dan responsif untuk berbagai perangkat.
5. An intuitive and responsive user interface compatible with various devices.

The identified weaknesses of the system include:

1. The dataset is limited to daylight lighting conditions, which may reduce performance under different lighting environments such as nighttime or artificial lighting.
2. The dataset only includes purple passion fruits from markets in Medan, meaning that variations of fruits from other regions may not be well represented.
3. The system requires an internet connection to access the web application, making it unavailable for offline use.

5. Conclusion and Recommendations

5.1. Conclusion

This study successfully developed a classification model for purple passion fruit ripeness levels using the Convolutional Neural Network (CNN) method and implemented it in a web application. The CNN model was designed with an architecture consisting of four convolutional layers with 16, 32, 64, and 128 filters, adapted from previous fruit classification research. Among various combinations of data split ratios and learning rates tested, the 80:20 ratio with a learning rate of 0.001 produced the best performance, achieving a training accuracy of 96.72%, a testing accuracy of 95.76%, and a loss value of 0.1811.

Additional validation using the 5-Fold Cross Validation method resulted in an average accuracy of 95.40%, confirming that the model possesses high reliability and generalization capability across data variations. The addition of a non-passion fruit class successfully enhanced the system’s robustness in distinguishing passion fruits from other objects. The trained model was integrated into a web-based system using Flask as the backend and JavaScript, HTML, and CSS as the frontend, deployed on the PythonAnywhere cloud platform. The developed web application allows users to upload images and automatically obtain ripeness predictions with a response time of less than 5 seconds, thereby assisting new purple passion fruit sellers in performing fruit sorting more quickly, accurately, and efficiently.

5.2. Recommendations

Based on the research findings and the identified limitations, several recommendations are proposed for future work. First, additional datasets should be collected under more diverse lighting conditions, including nighttime and artificial lighting, to improve the model’s robustness across different environmental conditions. Second, the dataset can be expanded by collecting purple passion fruit images from various regions in Indonesia to enhance the model’s generalization capability toward variations in fruit characteristics from different geographical locations.

Third, further exploration of deeper CNN architectures or the application of transfer learning techniques using pre-trained models such as ResNet, VGG16, or EfficientNet can be conducted to improve classification accuracy. Fourth, the system can be developed into a native mobile application for Android or iOS platforms, enabling offline usage without the need for an internet connection. Fifth, a multiple-object detection feature can be added to improve sorting efficiency when multiple fruits appear in a single frame. Lastly, further research can be carried out to implement this system for other fruits that exhibit similar color changes during ripening, such as mangoes, bananas, or tomatoes.

References

[1] S. Rahayuningsih and A. Nugraha, “Analisis sektor pertanian sebagai sektor unggulan di Indonesia,” *Jurnal Agribisnis Indonesia*, vol. 9, no. 2, pp. 112–121, 2021.
[2] L. Muntafiah and N. Hidayati, “Analisis kandungan kimia dan manfaat buah markisa ungu (Passiflora edulis Sims),” *Jurnal Sains dan Kesehatan*, vol. 2, no. 3, pp. 45–52, 2019.
[3] R. Nugroho and D. Wibowo, “Kandungan flavonoid dan aktivitas antioksidan buah markisa ungu,” *Jurnal Kimia dan Pangan*, vol. 8, no. 1, pp. 25–33, 2020.
[4] M. Rosiana and S. Yuliani, “Aktivitas antibakteri ekstrak kulit buah markisa ungu terhadap Staphylococcus aureus dan Escherichia coli,” *Jurnal Bioteknologi*, vol. 7, no. 1, pp. 59–66, 2021.
[5] N. Lestari and M. Syahrial, “Pemanfaatan markisa ungu dalam industri pangan dan minuman,” *Jurnal Agroindustri*, vol. 12, no. 2, pp. 88–95, 2020.
[6] Badan Pusat Statistik (BPS), *Statistik Produksi Hortikultura Indonesia Tahun 2020*. Jakarta: BPS, 2021.
[7] R. Rukmana, *Budidaya dan Pascapanen Buah Markisa Ungu*. Yogyakarta: Kanisius, 2022.
[8] R. Sembiring and H. Fitriani, “Pengaruh tingkat kematangan terhadap kualitas buah markisa ungu,” *Jurnal Teknologi Pertanian*, vol. 5, no. 3, pp. 101–109, 2019.
[9] D. Wahyuni and R. Firmansyah, “Analisis metode penentuan kematangan buah secara manual dan digital,” *Jurnal Keteknikan Pertanian*, vol. 11, no. 1, pp. 23–30, 2020.

[10] I. A. Putra and M. Pradana, “Penerapan convolutional neural network (CNN) untuk deteksi penyakit tanaman padi,” *Jurnal Teknologi Informasi Pertanian*, vol. 9, no. 2, pp. 77–86, 2021.
[11] N. Hasanah and D. Ramadhani, “Pengenalan pola warna dan tekstur buah menggunakan deep learning CNN,” *Jurnal Sains Komputer*, vol. 6, no. 3, pp. 114–122, 2020.
[12] L. Salamah and A. Supriyadi, “Klasifikasi kematangan dan ukuran buah nanas menggunakan CNN berbasis Android,” *Jurnal Teknologi dan Rekayasa*, vol. 5, no. 1, pp. 14–21, 2023.
[13] X. Tu, Y. Li, and Q. Zhang, “Detection and classification of passion fruit ripeness using RGB-depth images and CNN,” *Computers and Electronics in Agriculture*, vol. 156, pp. 563–572, 2018.
[14] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, pp. 1097–1105, 2012.
[16] M. Nielsen, Neural Networks and Deep Learning. *Determination Press*, 2015.
[17] M. Ulum and E. Prasetyo, “Klasifikasi tingkat kematangan cabai rawit menggunakan CNN dan K-Nearest Neighbor,” *Jurnal Ilmu Komputer*, vol. 12, no. 1, pp. 45–54, 2024.
[18] A. Setya Nugraha and H. Hermawan, “Optimasi akurasi CNN untuk klasifikasi kualitas buah apel hijau,” *Jurnal Teknologi Informasi dan Komputer*, vol. 7, no. 2, pp. 99–107, 2023.
[19] D. Kurniawan and B. Santoso, “Klasifikasi tingkat kematangan buah markisa menggunakan jaringan syaraf tiruan,” *Jurnal Pengolahan Citra Digital*, vol. 6, no. 1, pp. 31–38, 2020.
[20] M. Azhari and F. Wijaya, “Implementasi CNN untuk identifikasi tanda tangan pada aplikasi mobile,” *Jurnal Informatika dan Komputer*, vol. 9, no. 2, pp. 55–63, 2021.