Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 3 Iss. 1 April 2023, pp: 36-42
ISSN(P) 2797-2313 | ISSN(E): 2775-8575

Classification of Date Fruit Types Using CNN Algorithm Based on Type

Klasifikasi Jenis Buah Kurma Menggunakan Algoritma CNN Berdasarkan Jenisnya

M. Fajrun Nadhif[1*], Saruni Dwiasnati[2]

[1,2]Informatics Engineering Study Programme, Faculty of Computer Science,
Universitas Mercu Buana, Indonesia

E-Mail: [1]41519010084@student.mercubuana.ac.id, [2]saruni.dwiasnati@mercubuana.ac.id

Received Mar 13th 2023; Revised Apr 10th 2023; Accepted Apr 30th 2023
Corresponding Author: M. Fajrun Nadhif

Abstract

Date fruits are an important commodity in the agriculture and food industry. However, in the process of sales and distribution to ordinary people, there are often errors in identifying different types of date fruits. Therefore, this research aims to develop an automatic classification system to distinguish the types of date fruits based on their types using the Convolutional Neural Network (CNN) algorithm. The case study was conducted at Hamima Dates date shop. The data used are fruit images with 9 categories and a total of 1658 samples, which are divided into 1496 samples for training data and 162 samples for testing data. The test results show that the CNN algorithm has a high level of accuracy in classifying the type of date fruit, with an accuracy of 96%. In this study, feature analysis was also conducted to determine the contribution of each feature to the classification of date fruit types. The results of this study can be the basis for the development of a more sophisticated date fruit automatic classification system and can be applied to other types of fruits.

Keywords: Classification, Convolutional Neural Network (CNN), Date Fruit, Mobile NetV2

Abstrak

Buah kurma merupakan salah satu komoditas penting dalam dunia pertanian dan industri makanan. Namun, dalam proses penjualan dan pendistribusiannya kepada masyarakat awam, sering kali terjadi kesalahan dalam mengidentifikasi jenis-jenis buah kurma. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi otomatis untuk membedakan jenis buah kurma berdasarkan jenisnya dengan menggunakan algoritma Convolutional Neural Network (CNN). Studi kasus dilakukan di toko kurma Hamima Dates. Data yang digunakan adalah citra buah dengan 9 kategori dan total 1658 sampel, yang terbagi menjadi 1496 sampel untuk data training dan 162 sampel untuk data testing. Hasil pengujian menunjukkan bahwa algoritma CNN memiliki tingkat akurasi yang tinggi dalam mengklasifikasikan jenis buah kurma, dengan tingkat akurasi sebesar 96%. Pada penelitian ini juga dilakukan analisis fitur untuk mengetahui kontribusi masing-masing fitur terhadap klasifikasi jenis buah kurma. Hasil dari penelitian ini dapat menjadi dasar pengembangan sistem klasifikasi otomatis buah kurma yang lebih canggih dan dapat diaplikasikan pada jenis buah lainnya.

Kata kunci: Buah Kurma, Convolutional Neural Network (CNN), Klasifikasi, Mobile NetV2

1. INTRODUCTION

Dates are a fruit that is familiar to the people of Indonesia. In the month of Ramadan, someone widely served dates as takjil to break the fast Muslims. Dates are also one of the important commodities in the agriculture and food industry, In 2017 the worldwide production of dates reached 8 million tonnes and the export value of this commodity reached 1.63 billion US dollars and in 2018 the export of this commodity continues to increase in the years [1], [2]. In Indonesia, dates often face challenges in the sales and distribution process because of the unfamiliarity of Indonesians with introducing this fruit. One problem that often occurs is the error in identifying different dates. The custom of buying dates during the month of Ramadan is a cultural habit for Muslims, where eating dates when breaking the fast is sunnah. Often traders offer the best types of dates but the contents of the dates are other types because of the same level of similarity, expert traders are very experienced in distinguishing types of dates while ordinary people will find it very difficult to distinguish because visually many dates have almost the same visual and difficult to distinguish for some people there. To overcome this problem, this research aims to help develop an automatic classification system using the Convolutional Neural Network (CNN) algorithm to distinguish the type of date fruit based on its type. They conducted this case study at Hamima Dates date shop. The CNN method is one of the deep learning methods that are capable of performing a self-learning process for object recognition, object extraction, and classification and can be applied to high-resolution images that have a nonparametric distribution model [3].

In the journal "Classification of tomato leaf diseases using MobileNet V2" written by Zaki S. and his colleagues, the research shows the use of the Convolutional Neural Network (CNN) method using MobileNet V2 architecture to identify diseases on tomato leaves. The research used a test dataset comprising 4,671 images from the Plant village data set. The results showed that the developed model achieved an accuracy rate of over 90% [4]. This finding shows that the use of MobileNet V2 architecture is quite good at identifying images, especially in tomato leaf disease recognition. The MobileNet V2 architecture has a good balance between accuracy and computational efficiency, making it very suitable for image recognition applications on devices with limited resources, such as mobile devices. Based on these studies, researchers in the current research implemented the CNN algorithm with the MobileNet V2 architecture to classify date fruits based on their type. By looking at the success of MobileNet V2 in recognizing diseases in tomato leaves, it is expected that the architecture will also provide good results in classifying date fruit types.

Although findings from previous studies can provide direction and inspiration, each study has different characteristics and parameters. Therefore, in the current study, researchers must test, validate, and adjust parameters according to the context and data used in date fruit classification. By implementing the CNN algorithm with Mobile Net V2 architecture, it is expected that this research will achieve good results in classifying date fruit types. The results of the research can contribute importantly to the development of applications that can recognize and classify date fruit types with high accuracy, which can be helpful in the agricultural industry, fruit trade, or further research related to date fruits.

The data used in this study comprises date fruit images with 9 different categories and 1658 samples. It divided these data samples into 1496 samples for training data and 162 samples for testing data. By using the CNN algorithm, the test results show that this classification system has a high accuracy rate, reaching 96% in classifying date fruit types. In addition, in this study, a feature analysis was also conducted to understand the contribution of each feature in the classification of date fruit types. The results provide valuable insights into the features that are most important in distinguishing date fruit types. This research has significant implications for the development of automated classification systems for date fruits and can also apply to other types of date fruits. By using the CNN algorithm, this system can help improve efficiency and accuracy in the process of date fruit type identification, reduce errors, and increase customer satisfaction.

In this context, this research will contribute importantly to the development of a more sophisticated automatic classification system for distinguishing date fruit types, which can improve efficiency and reliability in the agricultural and food industries. The results can serve as a solid foundation for the development of more sophisticated automatic classification systems and can apply to other types of fruits.

2. MATERIALS AND RESEARCH METHODS

This research will use image processing to identify patterns in dates to estimate and identify dates based on their type. This research helps people to distinguish the type of date fruit based on its type by utilizing deep learning technology. The step begins with analyzing the data set as date fruit images comprising 9 types of date fruit variants, the total image data used is 1658 date fruit images which will then be carried out the training process using the CNN algorithm with the Mobile Net v2 architecture to produce a model that will be used as the basis for date fruit classification. the reason for using the Mobile Net V2 architecture is because of the computational efficiency of Mobile Net V2, which is specifically designed to maintain computational efficiency with a relatively small number of parameters, and the speed of inference to classify images can be quickly and efficiently. The Mobile Net v2 architecture has good accuracy even though it focuses on computational efficiency.

2.1 Research Methodology

Convolutional Neural Network (CNN) is a type of deep learning algorithm commonly used for image and video recognition tasks. The algorithm was developed based on the way the visual cortex in the brain is structured and how it functions. CNNs comprise multiple layers of interconnected nodes that perform convolution on the input data to extract features. It then passed these features through additional layers to classify the input into different categories. CNNs have proven to be very effective in a variety of image and video recognition tasks, including object detection, face recognition, and other classifications [5]. This research method will begin with collecting date fruit data that will be taken from Kaggel, which will then be preprocessed and image augmentation will be carried out to improve the quality of the results of accuracy in the training process. After the image is processed, the Training process will be carried out using the

Convolutional Neural Network method with the Mobile Net V2 architecture. After performing the data training stage, testing or validation will be carried out to determine the level of accuracy got when carrying out the training process, and then the results will be analyzed as a conclusion of this research.

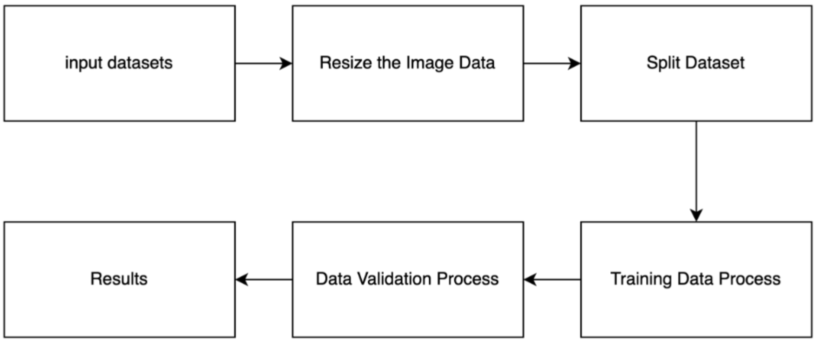


Figure 1. Research Methodology

2.2 Convolutional Neural Network

Convolutional Neural Networks (CNN) is a type of artificial neural network that has become dominant in various computer vision tasks. CNNs are designed to automatically and adaptively learn a spatial hierarchy of features through backpropagation by using several building blocks, such as convolution layers, union layers, and fully connected layers [6]. CNN has three main layers: the convolution layer, the pooling layer, and the fully connected layer. Where the neurons are arranged in 3 dimensions (width, height, color). Using the CNN method is very popular in image detection because this method has achieved near-human accuracy in many visual recognition tasks, making it a popular choice for implementations of various types [7]. In the real world, there are many uses for convolutional neural network techniques. One use of this method in Indonesia is the application of e-tickets where CNN will identify the image data of a person and vehicle Police number Images to determine errors. Then this decision will be sent to the central server for follow-up. Here's how the CNN algorithm works:

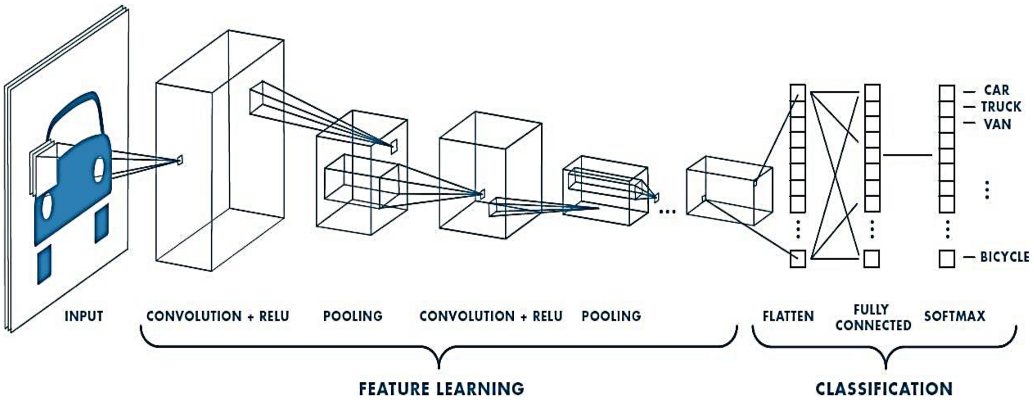


Figure 2. The structure of CNN

In the CNN architecture, there are several main layers that make up this architecture. The layers in the architecture will weigh each weight of the features extracted from the image. When doing the training process the image will enter through the first layer, namely the Convolutional Layer, then will continue to enter the pooling layer, after calculating the weight of each feature, it will continue to enter the fully connected layer to classify the right category, and the result of CNN is the output layer which is the last reference to determine the most relevant class/category from the calculation. Below, I summarize each CNN layer:

1. Convulution Layer: This layer will use a collection of filters or kernels that are used to extract relevant information from the input image, such as texture patterns, corners, and edges, from each image pixel[8]. As shown in figure 2 after the input image.
2. Pooling Layer: The pooling layer is used to downsample the output created by the previous convolution layer by taking each pixel's maximum or average value. The pooling layer will reduce the spatial scale of the input image and increase the model's robustness to changes in input data [9]. We can see the pooling process in Figure 2 after the data enters the convolution layer.

3. Fully Connected Layer: The fully connected layer is the last layer in the CNN architecture. This layer has a very important task, which is to calculate the probability of the specified output. In this layer, each neuron is connected to one another. This layer gets input data from each feature that has entered the pooling layer and will produce predictions of the class/category that have been determined[10]. As shown in figure 2 after the flatten laayer

2.3 Tensorflow

TensorFlow is an open-source software platform used to develop and train machine learning models. Developed by the Google Brain Team, TensorFlow provides a powerful framework for building various types of machine learning models, such as neural networks, deep learning, and other algorithms.TensorFlow uses data representations called "tensors" to perform numerical computations. TensorFlow provides a flexible environment for describing algorithms as a computational graph comprising a series of mathematical operations. This graph can then be executed using an anatomical backend such as CPU, GPU, or TPU (Tensor Processing Unit) [11]. The current use of TensorFlow is quite extensive developers have the freedom to use TensorFlow (TF) to explore new techniques in training and optimization. TF offers a wide range of applications, with a focus on Deep Neural Network (DNN) inference and training. Several Google services have successfully used TF applications in production. Along with that, TensorFlow has been released as an open-source project and is becoming a popular choice in Machine Learning (ML) research [12]. One advantage of TensorFlow is its high scalability, which allows users to train models on one or multiple hardware devices easily. In addition, TensorFlow also offers various APIs that make it easy for users to build, train, and deploy machine learning models..

2.4 Mobile Net V2

MobileNetV2 is a convolutional neural network (CNN) architecture developed by Google. It is specifically designed to meet the limited computational needs of mobile and embedded devices.MobileNetV2 uses several optimization normals to achieve a good trade-off between accuracy and model size. It incorporates concepts such as depthwise separable convolution, linear bottleneck, and inverted residual blocks. Depthwise separable convolution enables computational reduction by separating the convolution process into two stages: depthwise convolution and point-wise convolution. A linear bottleneck is used to reduce feature dimension and computational complexity by using convolution layers with a smaller number of filters. Inverted residual blocks are the basic unit in the MobileNetV2 architecture, which comprises convolution layers, batch normalization, and shortcut connections [13].

The MobileNetV2 architecture performs well in object recognition and image classification tasks, with normally small model sizes and lower computational requirements compared to traditional CNN architectures. This makes MobileNetV2 very suitable for implementing resource-constrained mobile and embedded devices.

2.5 Juypiter Notebook

Jupyter Notebook is an open-source web application that allows users to create and share interactive documents containing code, visualizations, narrative text, and other elements. The application is very popular among data scientists and developers due to its ability to combine code, visualizations, and narrative explanations in a single document that can be run and accessed interactively [14]. Jupyter Notebook supports a variety of programming languages, including Python, R, Julia, and more. In Jupyter Notebook, the code is executed in separate cells, which allows users to run and test the code interactively, as well as see the results in the form of output directly displayed below the code cells. In addition, Jupyter Notebook also supports the creation of interactive visualizations and interactive data processing.

The main advantage of Jupyter Notebook is its flexibility in exploring, analyzing, and sharing the results of experiments or data analysis projects. Jupyter Notebook documents can be exported to various formats, including HTML, PDF, and presentation slides, making it easy to share discoveries and understanding through interactive materials.

2.6 Python

Python is a popular programming language widely used in areas such as data analysis, web development, and machine learning. It is known for its simplicity, readability, and ease of use. Python has a large and active community that contributes to its development and provides support to its users. It also has various libraries and packages that make it a versatile language for different applications. Scikit-Learn is a machine learning package with Python that is open-source and complete [15]. Python is an excellent choice for machine learning and deep learning projects. Its extensive library support and compatibility with various deep learning frameworks make it a popular choice for developing and implementing deep learning models. In addition, Python's integration with other common programming languages and tools, such as C++ and CUDA for GPU acceleration, enables greater performance and flexibility in Deep Learning applications.

3. Results and Analysis

This research will use image-shaped data, where the data comprises 1658 images of dates with 9 different categories. At the time of training, it will divide the data into two parts training data and testing data using a ratio of 80% training data and 20% testing data. This data division is to validate the accuracy of the model created. Here is a picture of the fruit from the dataset.

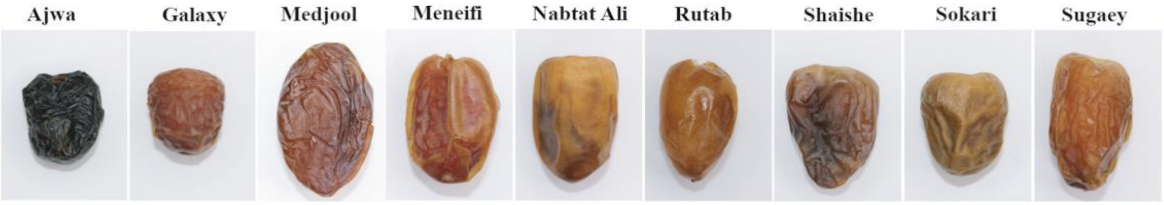
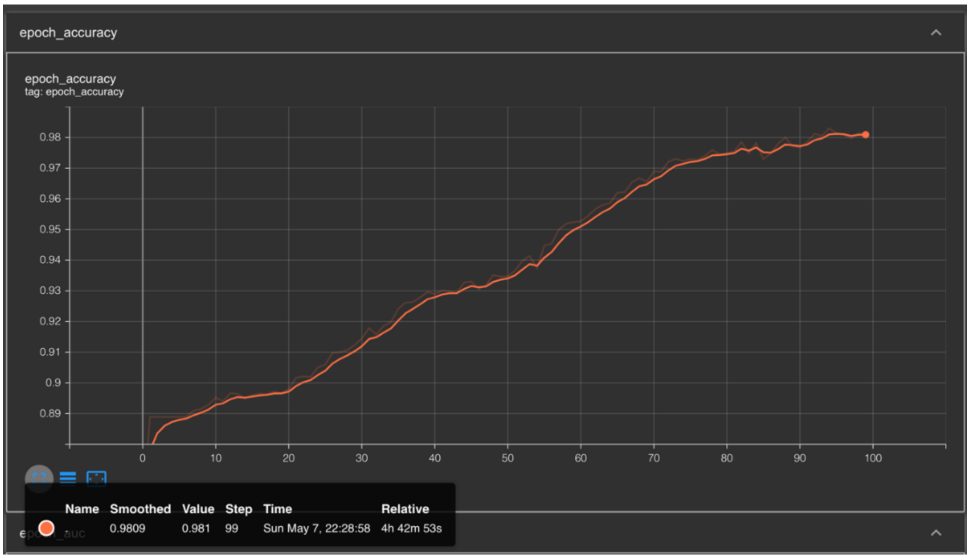


Figure 3. Date Fruit Images in the Dataset

After the dataset undergoes the preprocessing stage, the next step is to enter the data into the training stage (training data). In this stage, it will train the data using the Convolutional Neural Network (CNN) algorithm using the MobileNet V2 architecture. During the training process, it is very important to monitor the movement of data accuracy periodically. To do this, one can use TensorBoard, a visualization tool provided by TensorFlow. TensorBoard makes it possible to monitor model performance metrics, including accuracy, in real-time through an interactive graphical display. By monitoring the movement of accuracy through TensorBoard, it is possible to see how the model evolves during the training process. Whether the accuracy increases as iterations/epochs increase or fluctuate or even decrease. This monitoring provides valuable insight into the training progress and helps in making decisions such as when to stop training the model or whether any further extensions need to be made.



	Name	Smoothed	Value	Step	Time	Relative
●		0.9809	0.981	99	Sun May 7, 22:28:58	4h 42m 53s

Figure 4. Model Accuracy Graph

When training the model with a setting of 100 iterations/epoch and a learning rate of 0.001, we expected that the model does not experience overfitting. However, to confirm whether or not the model is over-fitting, it is necessary to monitor the training loss graph beside the accuracy graph. The training loss graph provides information on how well the model learns during training. Training loss measures how close the model's prediction is to the true value at each iteration/epoch. If the model does not experience overfitting, the training loss decreases as the iterations/epochs increase. However, if the model is overfitting, the training loss graph will show a different pattern. For example, when you see that the training loss continues to decrease significantly as the iterations/epochs increase, while the accuracy of the validation data decreases after reaching a certain peak. This shows that the model starts to "memorize" the training data and does not generalize well to new data. To check whether the model is overfitting or not, it is also necessary to compare the training loss graph with the accuracy graph on the validation data. If the accuracy of the validation data decreases or stagnates as the iterations/epochs increase, while the training loss continues to decrease, this may show over-fitting. Below is an image of the training loss graph.

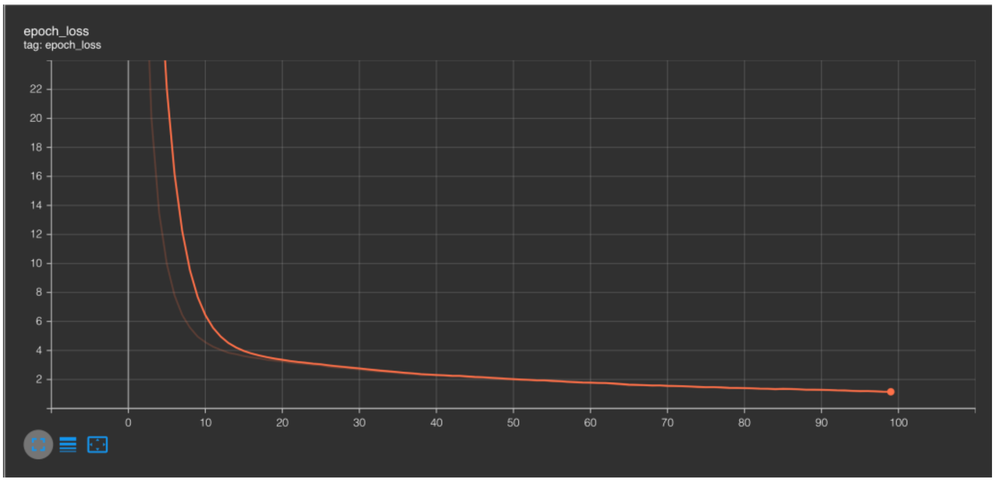


Figure 5. Training Loss Graph

Here, steps that can be taken to overcome overfitting include using regularization techniques (such as dropout or L1/L2 regularization), reducing model complexity, or collecting more training data if possible. The training process above took a relatively long time of 4 hours and 42 minutes. After conducting the training process, it is necessary to validate the model created to measure the accuracy of the model and then analyze it to get a conclusion from the facts. When testing the model the results get a fairly high accuracy of 96%. These results can be seen in the following table.

Table 1. Model Testing Results

K	Precision	Recall	F1-Score	Support
Ajwa	1.00	1.00	1.00	17
Galaxy	1.00	0.95	0.97	19
Medjool	0.93	1.00	0.96	13
Meneifi	1.00	0.96	0.98	23
Nabtat Ali	0.89	1.00	0.94	17
Rutab	1.00	0.93	0.96	14
Shaishe	1.00	0.88	0.94	17
Sokari	0.96	1.00	0.98	26
Sugaey	0.88	0.94	0.91	16
Accuracy			0.96	162
Macro Avg	0.96	0.96	0.96	162
Weighted Avg	0.97	0.96	0.96	162

From the table presented above, it can be concluded that the classification for Ajwa fruit types produces perfect performance with precision, recall, and F1-score values of 1.00. This shows that the model accurately classifies Ajwa fruits on the testing data comprising 17 samples. This indicates that the model accurately classifies Ajwa fruit types on the testing data comprising 17 samples. The results of each class also show good classification performance with high precision, recall, and F1-score values. This indicates that the model has a good ability to classify each type of date fruit individually, including the Ajwa fruit type. In analyzing the whole dataset, it was found that the average precision, recall, and F1 score were around 0.96. This shows that the model has a good ability to classify the various date fruits in the dataset. With values close to 1.0, the model can provide very accurate and consistent predictions. Based on these good results, it can be considered to export the model and implement it in relevant applications. The trained model can perform date fruit classification based on the features. However, it is important to note that before implementing the model in production, it is important to test and validate the model's performance more comprehensively using independent, never-before-seen data. In conclusion, the model trained and tested on this dataset performed well in classifying date fruit types, especially Ajwa fruit types. The consistency in classification results and high performance give confidence that the model is reliable for practical applications to date in fruit recognition and classification.

4. CONCLUSION

The conclusion of this research is that the use of CNN with MobileNet V2 architecture on the date fruit image dataset can provide good classification results with a high level of accuracy, namely getting a 96% success value. The trained model can be implemented in an automatic classification system to distinguish the date fruits, and this can contribute significantly to the development of technology in the agriculture and food industry. The results provide a positive outlook for the development of more sophisticated automatic classification systems in identifying date fruit types and can apply to other types of fruits.

REFERENCES

[1] Tridge, “overview of global date market (2020),” *https://www.tridge.com/ intelligences/dates*, 2022.

[2] R. A. Rahmadani, S. Bulkis, and M. A. Budiman, “POTENSI BUDIDAYA KURMA DI INDONESIA DITINJAU DARI PERSPEKTIF EKONOMIS DAN EKOLOGIS,” 2017.

[3] M. H. Romario, E. Ihsanto, and T. M. Kadarina, “Sistem Hitung dan Klasifikasi Objek dengan Metode Convolutional Neural Network,” *Jurnal Teknologi Elektro*, vol. 11, no. 2, p. 108, Jun. 2020, doi: 10.22441/jte.2020.v11i2.007.

[4] S. Z. M. Zaki, M. Asyraf Zulkifley, M. Mohd Stofa, N. A. M. Kamari, and N. Ayuni Mohamed, “Classification of tomato leaf diseases using MobileNet v2,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 2, p. 290, Jun. 2020, doi: 10.11591/ijai.v9.i2.pp290-296.

[5] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A Convolutional Neural Network for Modelling Sentences,” Apr. 2014.

[6] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.

[7] R. B. Arif, Md. A. B. Siddique, M. M. R. Khan, and M. R. Oishe, “Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Convolutional Neural Network,” Sep. 2018, doi: 10.1109/CEEICT.2018.8628078.

[8] D. Raval and J. Undavia, “A Comprehensive assessment of Convolutional Neural Networks for skin and oral cancer detection using medical images,” *Healthcare Analytics*, p. 100199, May 2023, doi: 10.1016/j.health.2023.100199.

[9] L. F. de J. Silva, O. A. C. Cortes, and J. O. B. Diniz, “A novel ensemble CNN model for COVID-19 classification in computerized tomography scans,” *Results in Control and Optimization*, vol. 11, p. 100215, Jun. 2023, doi: 10.1016/j.rico.2023.100215.

[10] R. K. V, J. M. K, and K. R. V, “Modelling a deep network using CNN and RNN for accident classification,” *Measurement: Sensors*, p. 100794, May 2023, doi: 10.1016/j.measen.2023.100794.

[11] M. Abadi *et al.*, “TensorFlow: A system for large-scale machine learning,” May 2016.

[12] M. Ramchandani *et al.*, “Survey: Tensorflow in Machine Learning,” *J Phys Conf Ser*, vol. 2273, no. 1, p. 012008, May 2022, doi: 10.1088/1742-6596/2273/1/012008.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Jan. 2018.

[14] T. Kluyver *et al.*, “Jupyter Notebooks—a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas - Proceedings of the 20th International Conference on Electronic Publishing, ELPUB 2016*, IOS Press BV, 2016, pp. 87–90. doi: 10.3233/978-1-61499-649-1-87.

[15] J. Hao and T. K. Ho, “Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language,” *Journal of Educational and Behavioral Statistics*, vol. 44, no. 3, pp. 348–361, Jun. 2019, doi: 10.3102/1076998619832248.