ILKOM Jurnal Ilmiah Vol. 17, No. 3, December 2025, pp. 276-291
Accredited 2nd by KEMDIKTISAINTEK No. 10/C/C3/DT.05.00/2025; E-ISSN 2548-7779 | P-ISSN 2087-1716

Research Article **Open Access (CC–BY-SA)**

Detection of Curcuma and Turmeric Differences Utilizing Fuzzy Tsukamoto Android-Based CCN Model

Fajar Rahardika Bahari Putra [a,1]; Muhammad Rizki Setyawan [b,2,*]; Ahmad Ilham [c,3]; Dimas Adi Suseno [d,4]

[a] *Universitas Muhammadiyah Sorong, Southwest Papua, 98416, Indonesia*
[1] *rizki@um-sorong.ac.id;* [2] *fajar_rbp@um-sorong.ac.id;* [3] *ahmadilham@gmail.com;* [4] *dimasadisuseno@gmail.com*
* *Corresponding author*

Article history: Received Month xx, 2021; Revised Month xx, 2021; Accepted Month xx, 2021; Available online Month xx, 2021 (Times 8pt)

Abstract

Turmeric and curcuma are herbs that are often used in medicine and cooking. However, their similar shapes and colours make it difficult for people, especially in Southwest Papua, to distinguish between them directly. According to the Central Statistics Agency (BPS) in 2023, turmeric production reached 18,302 units, far higher than turmeric, which only reached 2,950 units. Based on field interviews in Southwest Papua, more than 60% of respondents had difficulty distinguishing turmeric from turmeric. To address this issue, this research develops an Android-based classification system by integrating the Fuzzy Tsukamoto algorithm with Convolutional Neural Network (CNN) models. Five CNN models VGG16, MobileNetV2, NASNetMobile, EfficientNetB2, and EfficientNetB3 were selected based on their balance between computational efficiency (MobileNetV2, NASNetMobile), depth and proven stability (VGG16), and modern scalable architectures (EfficientNetB2 and B3). Each model was combined with fuzzy logic to enhance classification accuracy. he dataset consisted of 800 images of curcuma and turmeric obtained from Kaggle and field collections. The data were divided into training, validation, and testing sets, and augmented through a series of transformations including rescaling to a range of 0 to 1, rotation up to 40 degrees, horizontal shift of 20%, angular distortion (shear) of 20%, zoom up to 30%, horizontal flipping, and brightness adjustment. Empty areas generated during augmentation were filled using the nearest pixel value with the 'nearest' mode to preserve image integrity. Training was performed using the AdamW optimizer and fine-tuning. Model evaluation employed accuracy, precision, recall, F1-score, and confusion matrix metrics. The results showed that the VGG16 model performed best, achieving 97% accuracy, 98% precision, 97% recall, and 98% F1-score, as confirmed by the classification report and confusion matrix. This model was also the most stable when tested on the Android system, while EfficientNetB2 and B3 produced less satisfactory outcomes. These findings demonstrate that combining CNN and Fuzzy Tsukamoto improves the classification accuracy of images with high visual similarity. The proposed system has the potential to be applied as a direct plant identification tool in the field and can be further extended to classify other visually similar plants.

Keywords: Curcuma, Turmeric, CNN, Fuzzy Tsukamoto, Android Classification

Introduction

Technology is currently developing rapidly, and has transformed artificial information processing into computerized information [1]. Indonesia is a country with a very large area, which is reflected in the number of people, natural resources and cultural diversity [2]. Thanks to its climate and soil fertility, Indonesia can grow a variety of plants, one of which is spice plant [3], [4]. Some spice plants with high production value in Indonesia include Curcuma and turmeric. These plants can grow well and are often referred to as biopharmaca. Curcuma longa and turmeric contain curcumin with health benefits. Curcuma longa and turmeric are rhizomes that contain active compounds. Curcuma longa is used as a source of curcumin through supercritical CO_2 extraction, while turmeric is utilized for its functional properties. Both have potential for food and health applications [5], [6], [7]. Curcuma longa extraction yields 3.2% curcumin and its solubility is increased through micronization. Turmeric contains 3% curcuminoids and produces 19-38 μm starch. NADES and CO_2 methods increased the yield of turmeric extract to 33.35 mg/g with antioxidant activity [8], [9], [10].

Turmeric (Curcuma longa) and Curcuma (Curcuma zanthorrhiza) are rhizome plants that have many benefits in the fields of health, culinary, and pharmaceutical industries, and are widely cultivated in Southwest Papua. However, local communities still struggle to distinguish between the two due to their high visual similarity, including in terms of shape, color, and rhizome texture, often leading to misidentification. BPS data from 2023 shows that turmeric

http://dx.doi.org/10.33096/ilkom.v17i3.2857.276-291

production reached 18,302 units, far higher than Curcuma, which only reached 2,950 units, indicating the importance of accurate classification in supporting agricultural yield management [10]. Previous research by [11] has shown that the transfer learning approach using CNN models such as VGG19 and Inception V3 is capable of accurately classifying images of turmeric and Curcuma, mainly because the image data was taken from a smartphone camera compatible with the Android-based system. However, this approach has not integrated fuzzy logic such as Fuzzy Tsukamoto, which can handle visual uncertainty more flexibly. Meanwhile, a study by [12] proved that combining CNN with an ANFIS-based fuzzy system is effective in detecting plant diseases from leaf images, providing a strong methodological foundation for this study in developing a more adaptive and accurate rhizome classification system.

Then, [13] in their research proposed a Fuzzy Rank-Based Ensemble approach to combine several CNN models in medical image classification (cervical cytology). Although the domain is different, this research is relevant in the context of using fuzzy logic to improve image classification accuracy. This fuzzy ensemble technique can be an inspiration in improving the performance of your model that combines CNN with Tsukamoto fuzzy logic. The research by [14] also has substantial proximity because it focuses on detecting leaf diseases of turmeric plants using CNN models such as TurmericNet and ResNet50. Although the object studied is turmeric leaves and not differences between species, this research supports the implementation side of CNN and deep learning-based plant image processing similar to your research approach. Finally, Juwono et al. (2023) examined the design of an IoT-based vehicle diagnosis system that utilizes the Fuzzy Tsukamoto algorithm as a decision maker. Although the field of application is different, this research provides a strong theoretical foundation on how the Fuzzy Tsukamoto algorithm is used in real-time intelligent systems, which is very relevant if applied in an Android-based classification system as designed in your research.

Deep learning is an artificial intelligence method based on deep neural networks that is effective for image classification in agriculture and medicine [15], [16]. This study uses CNN architectures such as EfficientNetB3, VGG16, MobileNetV2, EfficientNetB2, and NASNetMobile, which were selected for their respective advantages in accuracy, computational efficiency, and generalization capabilities for complex images. EfficientNet and NASNetMobile offer a balance between high accuracy and lightweight model size, VGG16 is known for its stability and ease of implementation, while MobileNetV2 is designed for Android devices with limited resources. Unlike approaches that only evaluate CNNs individually, this study evaluates the effectiveness of integrating CNNs with the Tsukamoto Fuzzy inference system, which adds a fuzzy rule-based decision-making layer to handle visual ambiguity and improve the readability of classification results. By leveraging the unique characteristics of turmeric and curcuma, this approach not only improves accuracy but also produces a more reliable and explainable classification system, especially in cases of blurred class boundaries. The main objective of this research is to identify the optimal combination of methods for visually distinguishing turmeric and curcuma on Android-based systems.

Method

Deep Learning (DL) is a branch of Machine Learning (ML) that has grown rapidly in recent years, driven by the availability of large datasets and advances in computing technology [17], [18]. DL is an artificial intelligence technology that can recognize complex patterns from data automatically, especially from images [19], [20], [21], [22], [23], [27]. Deep learning applications, particularly the (CNN) model, have proven effective in detecting image-based objects accurately and efficiently [28], [29]. His paper uses a qualitative approach with data processing from machine learning classification or deep learning with the tsukamoto fuzzy method and 5 CNN models. The research design is exploratory and predictive, aiming to detect the differences between Curcuma and turmeric [30], [31], [32]. These developments have also contributed to the increasing research on the application of DL and ML in anomaly detection, especially in artificial intelligence-based intrusion detection systems (IDS) [33], [34]. Methodology of this research can be described through the steps shown in **Figure 1** [35], [36], [37].

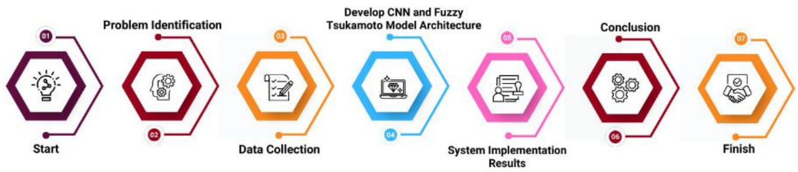

Figure 1. Research Flow [31]

There is an explanation of the research flow in Figure 1 as follows:

1. Problem Identification

   This study begins by identifying problems in distinguishing Curcuma and turmeric.

2. Data Collection

   After the problem is identified, data is collected through literature study and search for datasets of Curcuma and turmeric images. This data is used to train the model.

3. Develop CNN and Fuzzy Tsukamoto Model Architecture

   Develop a hybrid architecture that integrates the Fuzzy Tsukamoto algorithm with five CNN models (EfficientNetB3, VGG16, MobileNetV2, EfficientNetB2, NASNetMobile) for classification. Model development was conducted using Google Collaboratory and Python. with the stages as shown in **Figure 2**.

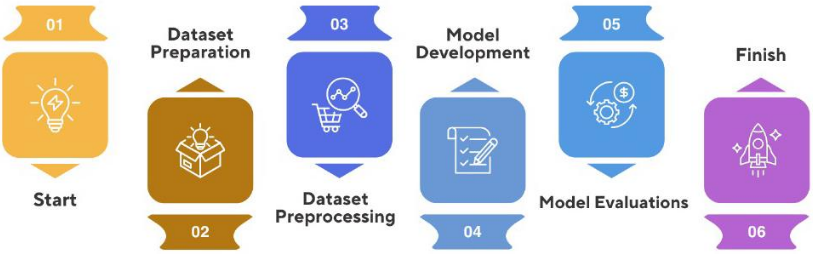

Figure 2. Model development

   An explanation of the Tsukamoto Fuzzy Algorithm Modeling and CNN model structure shown in **Figure 2** is as follows:

   a. Dataset Preparation

      Our research utilizes a dataset consisting of two categories, namely curcuma and turmeric, with 400 data each. The dataset was divided into three parts: for training (Train), validation (Valid), and testing (Test). The data was collected from the Kaggle website and directly from the field.

   b. Dataset Preprocessing

      Afterwards, the dataset was processed by applying augmentation techniques to increase the variation in the training data. The augmentation techniques used include rotation, horizontal and vertical shift, angle change, zoom, horizontal flipping, brightness adjustment, as well as filling in empty areas with the 'nearest' method.

   c. Model Development

      In this phase, CNN model architecture and Tsukamoto Fuzzy Algorithm were developed to recognize the characteristics of Curcuma and turmeric. Each model is trained for 10 epochs using AdamW optimization and a learning rate of 1e-4.

   d. Model Evaluations

      Trained models will be evaluated using confusion matrix and evaluation metrics on test data to assess the best model performance.

4. System Implementation Results

   Implementation results through an android-based system with the results of differences in accuracy or presentation based on 5 models that have been developed to get a suitable model used to detect Curcuma and turmeric.

5. Conclusion

   At this stage, conclusions are drawn up which include a summary of the research results and the performance of the models that have been developed.

Results and Discussion

Preparation and data collection process as well as the characteristics of the dataset used greatly support the performance of the model later [38]. CNN models have brought great changes in the field of computer vision and are widely applied in various tasks, such as image classification and object detection [39], [40]. These models are adopted for image recognition to distinguish between internal and external noise under different conditions [41]. In this study, the model is applied to classify images as well as run the training process [42]. Optimized CNN architecture is used to evaluate its effectiveness in a Deep Learning environment [43]. Datasets used contain images of Curcuma and Turmeric plants. Preparation of Implementation of Fuzzy Tsukamoto Algorithm with CNN Model Dataset. At this stage, the implementation of the Tsukamoto Fuzzy Algorithm with the CNN model is carried out through the following processes.

A. Preparations

Data was collected through the Kaggle site on the internet and also directly in the field to get more relevant ones. An example of the dataset that has been obtained is shown in **Figure 3**

Figure 3. Dataset Gathering

Figure shows an example of a dataset that was successfully collected, consisting of turmeric and Curcuma images with a total of 800 data.

B. Pre-processing Dataset

Collected Curcuma and Turmeric images then go through a pre-processing stage. The dataset is then separated into three parts, namely 70% for training, 20% for validation, and 10% for testing. Details of the data division can be seen in **Table 1**.

Table 1. Dataset Division

Class	*Train*	*Valid*	*Test*	Number of Dataset
Turmeric	560	160	80	800
Curcuma				

A process of image augmentation is then applied to increase the variety and amount of training data without having to do manual data collection, as shown in Figure 4.

```
# === Data Augmentation and Generators ===
data_gen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.3,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Figure 4. Augmentation Process

Figure 4 displays the applied augmentation process including rescaling to a range of 0 to 1, rotation up to 40 degrees, horizontal shift by 20%, angular distortion (shear) by 20%, and zoom up to 30% to increase the variety of the dataset. In addition, augmentation also includes horizontal flipping to flip the image. The empty areas generated during the augmentation process are filled using the nearest pixel value with the ‘nearest’ mode. The result of the augmentation of the dataset that has been performed is shown in **Figure 5**.

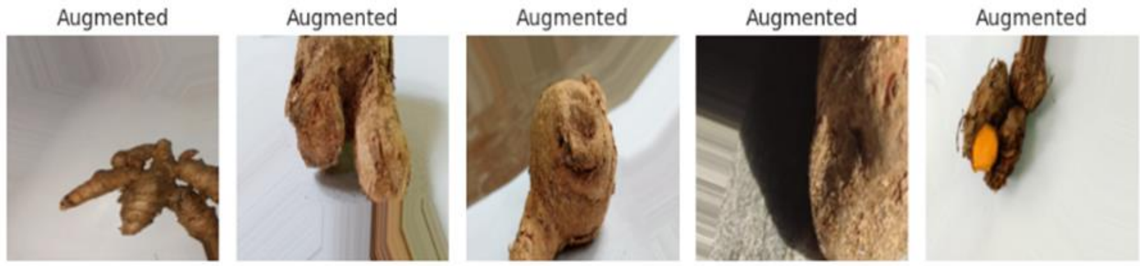

Figure 5. Example of Image Augmentation Result

C. Developing a Model

At this stage, the definition of Fuzzy Tsukamoto and the architecture of the CNN models used, namely VGG16, EfficientNetB3, MobileNetV2, EfficientNetB2, and NASNetMobile, as shown in **Figure 6**.

```
# === FUZZY TSUKAMOTO LAYER ===
def fuzzy_tsukamoto(x):
    turmeric = tf.clip_by_value((x - 0.4) / (1.0 - 0.4), 0.0, 1.0)
    curcuma = tf.clip_by_value((0.6 - x) / (0.6 - 0.2), 0.0, 1.0)
    rule_turmeric = turmeric * 0.8
    rule_curcuma = curcuma * 0.2
    numerator = rule_turmeric + rule_curcuma
    denominator = turmeric + curcuma + 1e-6
    fuzzy_score = numerator / denominator
    fuzzy_score = tf.reduce_mean(fuzzy_score, axis=1, keepdims=True)
    return fuzzy_score

# === CNN + FUZZY MODEL DEFINITION ===
base_models = {
    "VGG16": VGG16,
    "EfficientNetB3": EfficientNetB3,
    "MobileNetV2": MobileNetV2,
    "EfficientNetB2": EfficientNetB2,
    "NASNetMobile": NASNetMobile
}

models = {}
```

Figure 6. Tsukamoto Fuzzy Function implementation and CNN Model Initialization

In **Figure 6**, the fuzzy_tsukamoto() function is used as the fuzzy layer in the Convolutional Neural Network (CNN) model. This function receives input in the form of a tensor with the form (batch_size, feature_dim), which means that each batch contains several data samples, and each sample has a certain feature dimension. then the fuzzification process is carried out In the fuzzification stage, the values in the tensor are converted into fuzzy membership values. The two membership functions used are turmeric and Curcuma. This process is done by clipping the tensor values based on a certain threshold. The smaller the input value, the larger the turmeric membership value, while the larger the input value, the larger the Curcuma membership value. This creates two fuzzy categories that represent two different types of features. After the fuzzification stage, inference based on fuzzy rules is performed. The two rules used are rule_turmeric and rule_ Curcuma. rule_turmeric is calculated by multiplying the membership value of turmeric with a weight of 0.8, while rule_ Curcuma is calculated by multiplying the membership value of Curcuma with a weight of 0.2. These weights determine how important each rule is in decision making. The results of the two rules are then used in the defuzzification stage. Defuzzification is the process of converting fuzzy values into a single value that can be used for decision making. In this case, the final fuzzy_score value is calculated by summing the two rules and dividing it by the total membership of turmeric and Curcuma, plus a factor of 1e-6 to avoid division by zero. This fuzzy_score value is then averaged using reduce_mean, resulting in one representative value of all features. After that, a CNN model is combined with a fuzzy layer. Some CNN architectures that can be used in this fusion are VGG16, EfficientNetB3, MobileNetV2, EfficientNetB2, and NASNetMobile. These models are listed in the base_models dictionary and can be selected according to specific needs. Combining the fuzzy layer with the CNN model aims to improve the interpretability of the classification between turmeric and Curcuma.

Next **Figure 7** shows the process of building a deep learning pipeline with TensorFlow and Keras that combines CNN and fuzzy logic. CNN models such as VGG16, EfficientNetB3, MobileNetV2, EfficientNetB2, and NASNetMobile are used as feature extractors with ImageNet weights, with no top layer. About 80% of the layers are frozen to keep the underlying features intact, while the rest are retrained to match the new dataset. The CNN output is processed through the fuzzy_tsukamoto() function and then merged with the fully connected layer. Dense (512-256-128), BatchNormalization, and Dropout (0.3) layers are used to deepen the understanding of the features and prevent overfitting. The final output is a two-class classification using softmax. The model was compiled with the AdamW optimizer and categorical_crossentropy loss.

```python
for name, BaseModel in base_models.items():
    base_model = BaseModel(weights='imagenet', include_top=False, input_shape=img_shape)

    # Hitung 80% dari total layer untuk di-freeze
    freeze_until = int(len(base_model.layers) * 0.8)

    for layer in base_model.layers[:freeze_until]:
        layer.trainable = False
    for layer in base_model.layers[freeze_until:]:
        layer.trainable = True

    x = Flatten()(base_model.output)
    x_fuzzy = Lambda(fuzzy_tsukamoto, output_shape=lambda input_shape: (input_shape[0], 1), name=f'fuzzy_
tsukamoto_{name}')(x)
    x_combined = tf.keras.layers.Concatenate(axis=1)([x, x_fuzzy])

    x = Dense(512, activation='relu')(x_combined)
    x = BatchNormalization()(x)
    x = Dropout(0.3)(x)
    x = Dense(256, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.3)(x)
    x = Dense(128, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.3)(x)

    out = Dense(2, activation='softmax')(x)

    model = Model(inputs=base_model.input, outputs=out)
    model.compile(optimizer=tf.keras.optimizers.AdamW(learning_rate=5e-5),
                  loss='categorical_crossentropy', metrics=['accuracy'])
    models[name] = model
```

Figure 7. CNN + Fuzzy Tsukamoto Architecture Suite

After training, each architecture was tested and evaluated with the Classification Report showing precision, recall, and f1-sc gingerore, as a basis for determining the best model for turmeric and Curcuma classification.

Table 2. Classification Report for VGG16

VGG16	precision	*recall*	f1-socre	support
Kunyit	1.00	0.96	0.98	50
Temulawak	0.94	1.00	0.97	30
Accuracy			0.97	80
macro avg	0.97	0.98	0.97	80
weighted avg	0.98	0.97	0.98	80

In **Table 2**, VGG16 performed very well in the training process, with an accuracy of 97%. The model was able to distinguish turmeric and curcuma very precisely. Precision for turmeric was 1.00 and for Curcuma was 0.94, indicating a very accurate positive prediction. Recall was also high at 0.96 for turmeric and 1.00 for Curcuma, meaning that almost all of the actual data was successfully recognized. The high F1-score values (0.98 for turmeric and 0.97 for Curcuma) indicate that the model's performance is consistent across all aspects of evaluation. With these results, VGG16 proved to be the most optimal model for this dataset.

Table 3. Classification Report for EfficientNetB3

EfficientNetB3	precision	*recall*	f1-socre	support
Kunyit	0.00	0.00	0.00	50
Temulawak	0.38	1.00	0.55	30
Accuracy			0.38	80
macro avg	0.19	0.50	0.27	80
weighted avg	0.14	0.38	0.20	80

On the other hand, **Table 3** shows that the EfficientNetB3 model performs very poorly and is not recommended for use in classifying this dataset. Although the recall for the turmeric class reaches 1.00, the precision value is only 0.38, indicating many false predictions. This model also completely failed to recognize the turmeric class, as evidenced by precision, recall, and f1-score values that were all 0.00. Its overall accuracy was only 38%, reflecting the model's inability to understand the visual characteristics of turmeric and indicating an imbalance in distinguishing between the two classes.

Table 4. Classification Report for MobileNetV2

MobileNetV2	**precision**	***recall***	**f1-socre**	**support**
Kunyit	1.00	0.92	0.96	50
Temulawak	0.88	1.00	0.94	30
Accuracy			0.95	80
macro avg	0.94	0.96	0.95	80
weighted avg	0.96	0.95	0.95	80

Then, in **Table 4,** the MobileNetV2 model delivered very satisfactory performance with an accuracy of up to 95%. The precision value reached 1.00 for turmeric and 0.88 for curcuma, while the recall was 0.92 for turmeric and 1.00 for curcuma. This shows that the model was able to classify both species proportionally and effectively. The high F1-score values (0.96 for turmeric and 0.94 for curcuma) signify a good balance between the precision and sensitivity of the model. With its lightweight and efficient architecture, MobileNetV2 is a good choice for the classification of these two classes, and is a more practical alternative to more complex models such as VGG16.

Table 5. Classification Report for EfficientNetB2

EfficientNetB2	**precision**	***recall***	**f1-socre**	**support**
Kunyit	0.62	1.00	0.77	50
Temulawak	0.00	0.00	0.00	30
Accuracy			0.62	80
macro avg	0.31	0.50	0.38	80
weighted avg	0.39	0.62	0.48	80

Table 5 shows that the EfficientNetB2 model performs suboptimally with an accuracy of only 62%. Although it can recognize turmeric images very well (recall 1.00, f1-score 0.77), it is completely unable to recognize curcuma images, as evidenced by the precision, recall, and f1-score values that are all zero. This imbalance indicates that the model failed to learn the visual features of curcuma and was too inclined to classify the data as turmeric. This suggests that EfficientNetB2 is not suitable for balanced two-class classification tasks.

Table 6. Classification Report for NASNetMobile

NASNetMobile	**precision**	***recall***	**f1-socre**	**support**
Kunyit	1.00	0.90	0.95	50
Temulawak	0.86	1.00	0.92	30
Accuracy			0.94	80
macro avg	0.93	0.95	0.94	80
weighted avg	0.95	0.94	0.94	80

Table 6 shows that the NASNetMobile model performs well with an accuracy of 94%. Precision for turmeric and curcuma classification was 1.00 and 0.86, respectively, while the recall values were 0.90 for turmeric and 1.00 for curcuma. The F1-score obtained is also quite high, which is 0.95 for turmeric and 0.92 for curcuma. These results indicate that NASNetMobile is able to classify both types of rhizomes well and handle differences between classes effectively. Despite its slightly lower accuracy compared to MobileNetV2 and VGG16, NASNetMobile remains an attractive option as it is lightweight and efficient with almost equivalent performance.

The model used a single neuron with sigmoid activation for binary classification, compiled with AdamW (learning rate 1e-4) and binary crossentropy as loss function. The ReduceLROnPlateau technique was used to decrease the learning rate when the validation loss stagnated. The model was trained for 10 epochs, and the accuracy and loss results are shown in **Figures 8** and **9**.

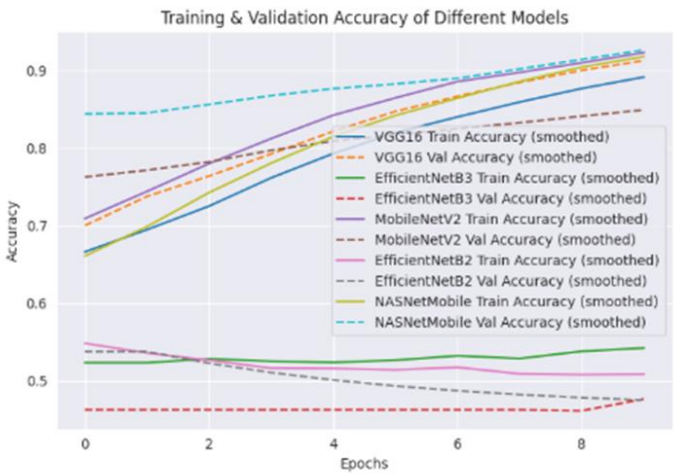

Figure 8. Accuracy Results

Figures 8 shows the training and validation accuracy graphs of the five CNN architectures tested over 10 epochs to assess the effectiveness of image classification. The models used include VGG16, EfficientNetB3, MobileNetV2, EfficientNetB2, and NASNetMobile. Each model is depicted in two curves (train and val) that have been smoothed for easy observation of performance trends. Results show that VGG16 and NASNetMobile have the highest validation accuracy, with VGG16 showing a steady gradual improvement, while NASNetMobile excels from the beginning and is consistently above 85%, signifying its strong generalization ability. Meanwhile, EfficientNetB3 and MobileNetV2 showed improvement, but their accuracy lagged slightly. On the other hand, EfficientNetB2 showed a decline in validation performance, indicating overfitting, which is when the model is only good at recognizing training data but fails on new data

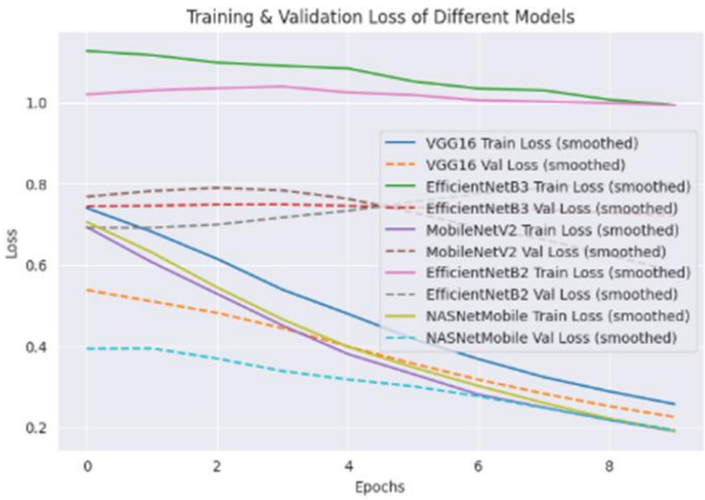

Figure 9. Lose Results

Figure 9 shows the training and validation loss trends of five CNN models-VGG16, EfficientNetB3, MobileNetV2, EfficientNetB2, and NASNetMobile-over 10 epochs. Each model displays two smoothed curves to clarify the trend: one showing the loss during training and the other during validation. This visualization is used to assess the stability of the training process and detect potential overfitting in each model. VGG16 and NASNetMobile showed the best performance with a stable loss decrease in training and validation, indicating effective learning without overfitting. NASNetMobile even maintained a low and stable validation loss throughout training, indicating strong generalization ability. In contrast, EfficientNetB2 experienced overfitting as the training loss decreased sharply, but the validation loss remained high or increased. EfficientNetB3 and MobileNetV2 also did not significantly decrease the validation loss, indicating unbalanced learning. Thus, VGG16 and NASNetMobile are the best models for classification of this dataset as they maintain a balance between accuracy and loss.

D. *Model evaluations*

Testing the models is done with test data using confusion matrix and evaluation metrics. Test results on VGG16, ResNet, EfficientNetB3, MobileNetV2, and NASNetMobile architectures show variations in performance, where each model has its own advantages and disadvantages in classifying test data.

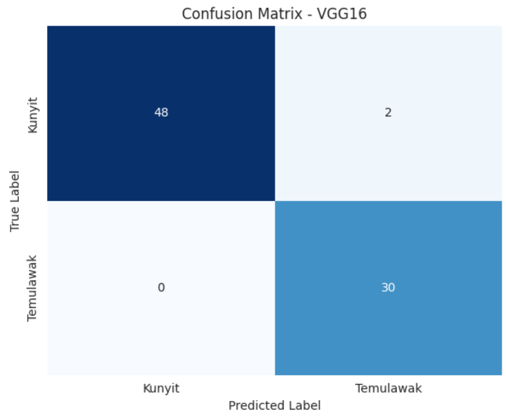

Figure 10. Confusion Matrix VGG16

Figure 10 shows the Confusion matrix results of the VGG16 model. The VGG16 model shows quite good results. From a total of 50 Turmeric samples, 48 turmeric samples were recognized with 2 samples that failed or were misclassified as turmeric. Meanwhile, for the curcuma category, out of a total of 30, all of them were successfully recognized and classified correctly without errors.

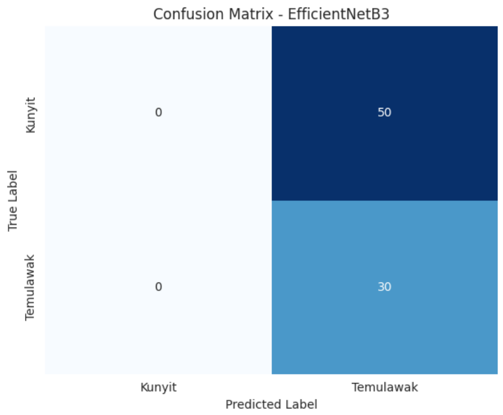

Figure 11. Confusion MatrixEfficietNetB3

Figure 11 displays the Confusion matrix results of the EfficientNetB3 model, providing a contrasting picture. Although the model was able to recognize all curcuma samples with perfect accuracy, the classification of turmeric samples out of 50 turmeric samples, none of them were correctly classified to the correct class and were erroneously classified into the curcuma category.

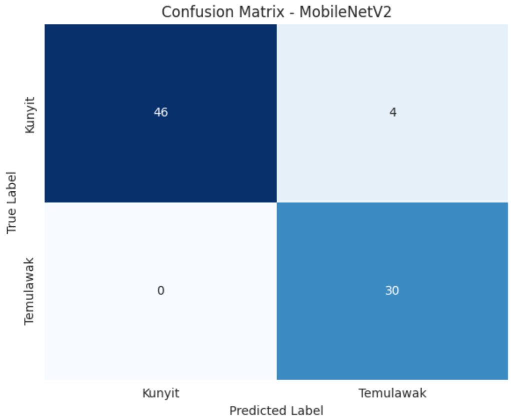

Figure 12. Confusion Matrix MobileNetV2

Figure 12 illustrates confusion matrix results for the MobileNetV2 model. The MobileNetV2 model showed solid performance in recognizing the curcuma class, where all 30 samples were correctly classified. However, the model experienced some errors in the classification of Turmeric samples, with only 46 out of a total of 50 samples being accurately recognized, while 4 samples were misclassified as curcuma.

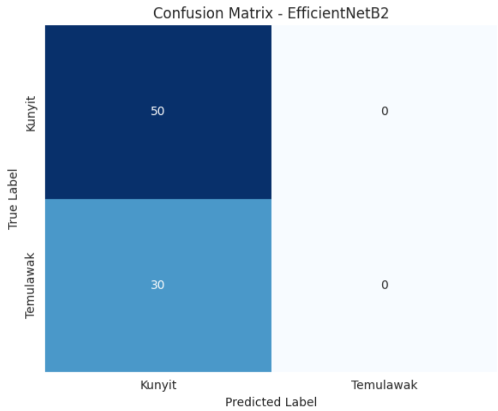

Figure 13. Confusion Matrix EfficietNetB2

Figure 13 shows confusion matrix results of the model on the EfficientNetB2 model. EfficientNetB2 model shows a different classification pattern. Out of 50 Turmeric samples were recognized perfectly without any errors, but this model did not succeed in classifying curcuma samples correctly. 30 curcuma samples were misrecognized as Turmeric.

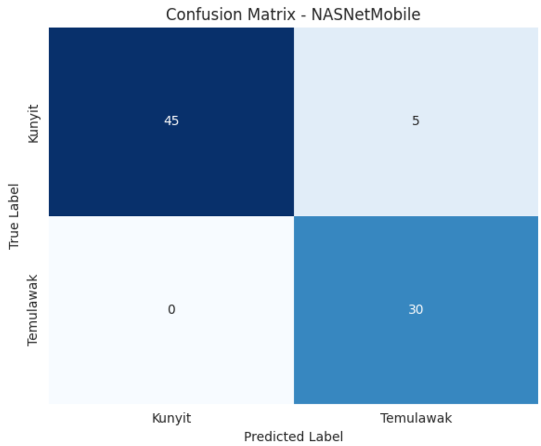

Figure 14. Confusion Matrix NASNetMobile

Figure 14 presents confusion matrix results from the model. NASNetMobile. This model shows a fairly balanced classification result. Out of a total of 50 Turmeric samples, 48 samples were correctly classified, while 5 samples were misrecognized as curcuma. In contrast, all of the 30 curcuma samples were correctly classified without any misclassification.

After obtaining the confusion matrix results, the next step is to measure the model with an evaluation matrix to evaluate the model's performance in more detail. The evaluation matrix includes accuracy, precision, recall, and F1-score, which are useful in assessing how well the model performs classification on new data. The evaluation matrix results are shown in **Table 7**.

Table 7. Classification Report

Model	Accuracy	*Presisi*	*Recall*	F1-score
VGG16 + FT	**0.97**	**0.98**	**0.97**	**0.97**
EfficientNetB3 + FT	0.38	0.31	0.62	0.38
MobileNetV2 + FT	0.95	0.91	0.91	0.91
EfficientNetB2 + FT	0.62	0.31	0.62	0.38
NASNetMobil + FT	0.94	0.92	0.93	0.92

Based on model performance evaluation results, VGG16 with fine-tuning techniques showed the most superior performance compared to other models in turmeric and Curcuma image classification. This model achieved an accuracy rate of 97%, the highest among all tested architectures. In addition, the resulting precision was also very high, at 1.00 for the turmeric class and 0.94 for curcuma, indicating that the model's prediction of both classes was very accurate and had minimal errors. In terms of recall, the model was able to recognize all curcuma data perfectly (1.00), as well as detect turmeric well (0.96), indicating a wide and balanced classification coverage. The F1-score was also consistently high at 0.98 for turmeric and 0.97 for curcuma, indicating a balance between the model's ability to identify positive data and avoid misclassification. When compared to other models such as MobileNetV2 and NASNetMobile that also performed quite well, VGG16 remained more stable and superior across all evaluation metrics. Meanwhile, the EfficientNetB2 and B3 models showed poor results and were not able to classify the data well. Thus, it can be concluded that VGG16 is the best model choice for this two-class classification task, as it has high accuracy, stable performance across classes, and consistent and balanced evaluation results.

E. System Implementation Results

In this section, the system will be used to detect using 5 models that have been converted to tflite format and will compare the accuracy presentation results. The following is a display of the results of turmeric and Curcuma detection based on the models that have been imported into the Android-based system.

Table 8. Detection Result Through System

No	Data Image	Model Detection Results				
		VGG16 + Fuzzy Tsukamoto	***EfficientNetB3 + Fuzzy Tsukamoto***	***MobileNetV2 + Fuzzy Tsukamoto***	***EfficientNetB2 + Fuzzy Tsukamoto***	***NASNetMobil + Fuzzy Tsukamoto***
1	*Turmeric* 	Turmeric (97.44%)	Curcuma (75.66%)	Turmeric (82.89%)	Turmeric (56.65%)	Curcuma (72.35%)
2	*Turmeric* 	Turmeric (70.29%)	Curcuma (77.32%)	Curcuma (91.97%)	Turmeric (56.88%)	Curcuma (99.51%)

No	Data Image	Model Detection Results				
		VGG16 + Fuzzy Tsukamoto	***EfficientNetB3 + Fuzzy Tsukamoto***	***MobileNetV2 + Fuzzy Tsukamoto***	***EfficientNetB2 + Fuzzy Tsukamoto***	***NASNetMobil + Fuzzy Tsukamoto***
3	*Turmeric* 	Curcuma (65.20%)	Curcuma (75.76%)	Turmeric (58.76%)	Turmeric (59.42%)	Curcuma (98.74%)
4	*Turmeric* 	Turmeric (64.13%)	Curcuma (78.03%)	Turmeric (64.61%)	Turmeric (57.77%)	Turmeric (56.53%)
5	*Turmeric* 	Turmeric (85.65%)	Curcuma (75.39%)	Curcuma (67.58%)	Turmeric (59.50%)	Curcuma (59.29%)
6	*Curcuma* 	Turmeric (99.53%)	Curcuma (76.19%)	Turmeric (99.57%)	Turmeric (58.01%)	Turmeric (84,14%)
7	*Curcuma* 	Curcuma (59.82%)	Curcuma (77.03%)	Turmeric (61.39%)	Turmeric (58.58%)	Turmeric (62,91%)
8	*Curcuma* 	Curcuma (96,87%)	Curcuma (76.83%)	Curcuma (100.00%)	Turmeric (58.24%)	Curcuma (99,87%)
9	*Curcuma* 	Curcuma (51,28%)	Curcuma (75.79%)	Turmeric (96.69%)	Turmeric (58.90%)	Curcuma (51.28%)

No	Data Image	Model Detection Results				
		VGG16 + Fuzzy Tsukamoto	***EfficientNetB3 + Fuzzy Tsukamoto***	***MobileNetV2 + Fuzzy Tsukamoto***	***EfficientNetB2 + Fuzzy Tsukamoto***	***NASNetMobil + Fuzzy Tsukamoto***
10	*Curcuma* 	Curcuma (73.41%)	Curcuma (77.45%)	Turmeric (80.40%)	Turmeric (56.73%)	Turmeric (79.02%)

Based on the detection test results on 10 image samples using five different CNN models integrated with the Fuzzy Tsukamoto method on an Android-based system, the VGG16 model showed the most superior performance. The VGG16 + Fuzzy Tsukamoto model achieved the highest accuracy rate of 80%, correctly identifying 8 out of 10 test images. Its performance significantly exceeded that of the other models. In comparison, the EfficientNetB3 and EfficientNetB2 models only achieved an accuracy of 50%, followed by MobileNetV2 with 40%, and NASNetMobile, which showed the lowest accuracy of 30%. Therefore, it can be concluded that VGG16 is the most suitable CNN model to be applied in the Fuzzy Tsukamoto method in an Android-based identification system compared to the other four models. The following are two images of the detection results using the VGG16 model in **Figure 15** and **Figure 16**

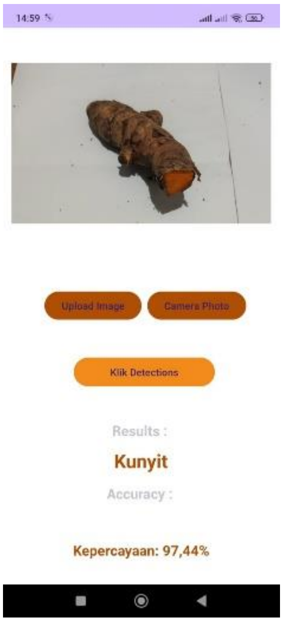

Figure 15. Turmeric detection

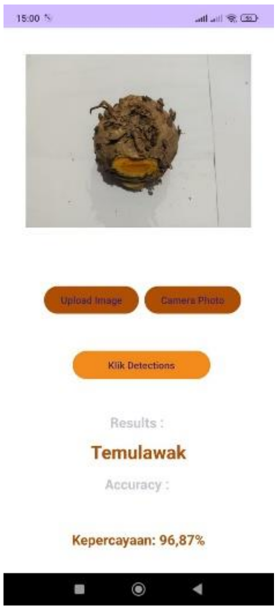

Figure 16. Curcuma detection

Conclusion

Based on the implementation results, this study demonstrates that the integration of the CNN algorithm with Fuzzy Tsukamoto logic in an Android-based system can effectively classify turmeric and curcuma images, despite their high visual similarity. Among the five CNN architectures tested, the VGG16 model consistently delivered the most superior performance across all evaluation stages, achieving the highest accuracy rate of 97% in the initial testing phase. This superiority was further confirmed during the system implementation testing with 10 new image data, as presented in Table 3. The reasoning for VGG16 being the best model is reinforced by these results, where it correctly classified 8 out of 10 images, reaching an accuracy of 80%. This performance significantly surpassed the other models, as EfficientNetB3 and EfficientNetB2 only achieved 50% accuracy, followed by MobileNetV2 at 40%, and NASNetMobile with the lowest accuracy of 30%. Therefore, it can be concluded that VGG16 is the best model choice for this classification task due to its superior accuracy, stability, and the most consistent and balanced evaluation results among all tested models. For future development, it is recommended to expand the dataset's variety, including images captured under different environmental conditions, and to apply similar methods to other spice crops to enhance the system's adaptability and utility in supporting digital agriculture.

Acknowledgement

The authors would like to express their sincere gratitude to all team members for their active participation and collaboration in preparing this paper. It is hoped that the findings of this research will provide valuable insights and benefits to future readers.

References

[1] F. R. B. Putra, A. Fadlil, dan R. Umar, "Analisis Metode Forward Chaining Pada Sistem Pakar Diagnosa Penyakit Hewan Sapi Berbasis Android," *J. Sains Komput. Inform. (J-SAKTI*, vol. 5, no. 2, hal. 1034–1044, 2021. doi: http://dx.doi.org/10.30645/j-sakti.v5i2.398

[2] F. R. B. Putra, M. R. Setyawan, N. Rendra Soekarta, dan L. J. Fakhri, "Implementasi Deep Learning Menggunakan CNN Untuk Klasifikasi Tingkat Kematangan Buah Jeruk Berbasis Android," *J. MAHAJANA Inf.*, vol. 9, no. 2, hal. 85–94, 2024. doi: https://doi.org/10.51544/jurnalmi.v9i2.5462

[3] D. A. Widiyastuti, A. Ajizah, L. Nurtamara, N. Huda, dan M. Afdal, "Pemberdayaan Ekonomi Skala Rumah Tangga Melalui Pembuatan Jamu Bubuk Rempah Temulawak, Jahe, Kunyit, dan Sereh," *J. Pengabdi. ILUNG (Inovasi Lahan Basah Unggul)*, vol. 3, no. 2, hal. 369, 2023, doi: 10.20527/ilung.v3i2.10298.

[4] A. Z. Pramesthi, F. Purnamasari, dan I. Setiawan, "Eksplorasi Keanekaragaman Hayati Tanaman Konsumsi di Pekarangan Desa Sumber RW 07 Surakarta Integrasi Fungsi Pangan dan Pengobatan," vol. 4, hal. 7865–7880, 2024, doi: https://doi.org/10.31004/innovative.v4i6.

[5] P. T. J. Sutarsi, D. Wiradiestia, A. Altway, S. Winardi, Wahyudiono, dan S. Machmudah, "Extraction Process Optimization of Curcumin from Curcuma xanthorrhiza Roxb. with Supercritical Carbon Dioxide Using Ethanol as a Cosolvent," *ACS Omega*, vol. 9, no. 1, hal. 1251–1264, 2024, doi: 10.1021/acsomega.3c07497.

[6] S. Argel-Pérez *dkk.*, "Characterization of a Novel Starch Isolated from the Rhizome of Colombian Turmeric (Curcuma longa L.) Cultivars," *Foods*, vol. 13, no. 1, 2024, doi: 10.3390/foods13010007.

[7] S. Machmudah *dkk.*, "Enhancement of Curcuma xanthorrhiza Roxb Phytochemical Dissolution via Micronization Using a Supercritical Antisolvent Technique," *ACS Omega*, vol. 7, no. 7, hal. 6345–6353, 2022, doi: 10.1021/acsomega.1c06911.

[8] A. Stasiłowicz-Krzemień *dkk.*, "Natural Deep Eutectic Solvents Combined with Supercritical Carbon Dioxide for the Extraction of Curcuminoids from Turmeric," *Pharmaceuticals*, vol. 17, no. 12, 2024, doi: 10.3390/ph17121596.

[9] E. Rahmat, J. Lee, dan Y. Kang, "Javanese Turmeric (Curcuma xanthorrhiza Roxb.): Ethnobotany, Phytochemistry, Biotechnology, and Pharmacological Activities," *Evidence-based Complement. Altern. Med.*, vol. 2021, 2021, doi: 10.1155/2021/9960813.

[10] S. K. Kardos, Sofiano, Destrianto Mursalin, Sofian Kardos, Ibrahim Anas Duwila, - Muh. Hasbiollah, Haedar Ardi Aqsha, - Listar Tandi Roma, Putu Krishnanda Supriyatna, - Rahmadin Muttaqin, Citra Rizky Handayani, Citra Rizky Handayani, Harianto Palanu, Noviria E., "Data BPS Provinsi Papua Barat Daya," *https://satudata.papuabaratdayaprov.go.id/*, vol. 1, no. 1, hal. 1689–1699, 2024, [Daring]. Tersedia pada: https://satudata.papuabaratdayaprov.go.id/files/output/0babbfa68d8841e53090be090d05e565_1730796069.pdf

[11] A. Pratondo, E. Elfahmi, dan A. Novianty, "Classification of Curcuma longa and Curcuma zanthorrhiza using transfer learning," *PeerJ Comput. Sci.*, vol. 8, hal. 1–15, 2022, doi: 10.7717/peerj-cs.1168.

[12] T. H. Kim *dkk.*, "ANFIS Fuzzy convolutional neural network model for leaf disease detection," *Front. Plant Sci.*, vol. 15, no. November, hal. 1–14, 2024, doi: 10.3389/fpls.2024.1465960.

[13] A. Manna, R. Kundu, D. Kaplun, A. Sinitca, dan R. Sarkar, "A fuzzy rank-based ensemble of CNN models for classification of cervical cytology," *Scientific Reports*, vol. 11, no. 1. 2021. doi: 10.1038/s41598-021-93783-8.

[14] P. Pawaskar, H. K. Yogish, B. Pakruddin, dan Y. Deepa, "A Novel and Efficient Deep Learning Models for Assessing AI ' s Impact on Disease Diagnosis in Agriculture," vol. I, hal. 1–9, 2025, doi: 10.18805/IJARe.A-

6388.Submitted.

[15] R. Selvaraj dan M. S. G. Devasena, "A novel attention based vision transformer optimized with hybrid optimization algorithm for turmeric leaf disease detection," *Sci. Rep.*, vol. 15, no. 1, hal. 1–25, 2025, doi: 10.1038/s41598-025-02185-7.

[16] A. Altameem, C. Mahanty, R. C. Poonia, A. K. J. Saudagar, dan R. Kumar, "Breast Cancer Detection in Mammography Images Using Deep Convolutional Neural Networks and Fuzzy Ensemble Modeling Techniques," *Diagnostics*, vol. 12, no. 8, 2022, doi: 10.3390/diagnostics12081812.

[17] N. L. Marpaung, R. J. H. Butar Butar, dan S. Hutabarat, "Implementasi Deep learning untuk Identifikasi Daun Tanaman Obat Menggunakan Metode Transfer learning," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 3, hal. 348, 2023, doi: https://doi.org/10.30591/jpit.v8i2.5217.

[18] V. Devisurya, R. Devi Priya, dan N. Anitha, "Early detection of major diseases in turmeric plant using improved deep learning algorithm," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 70, no. 2, hal. 1–10, 2022, doi: 10.24425/bpasts.2022.140689.

[19] J. J. Albaqami *dkk.*, "Chemical Composition and Biological Activities of the Leaf Essential Oils of Curcuma longa, Curcuma aromatica and Curcuma angustifolia," *Antibiotics*, vol. 11, no. 11. 2022. doi: 10.3390/antibiotics11111547.

[20] R. G. P. Panjaitan, Titin, dan Y. G. S. Yuliana, "Histopathological and Immunohistochemical Features of the Liver of Mice by Giving Temulawak Rhizome Extract (Curcuma xanthorrhiza Roxb.)," *Toxicol. Int.*, vol. 32, no. 1, hal. 25–34, 2025, doi: 10.18311/ti/2025/v32i1/36525.

[21] E. J. Iweala *dkk.*, "Curcuma longa (Turmeric): Ethnomedicinal uses, phytochemistry, pharmacological activities and toxicity profiles—A review," *Pharmacol. Res. - Mod. Chinese Med.*, vol. 6, no. December 2022, hal. 100222, 2023, doi: 10.1016/j.prmcm.2023.100222.

[22] B. Kandel dan S. Mukhopadhyay, "Phytochemicaland pharmacological properties of Curcuma longa Linn," *Int. J. Health Sci. (Qassim).*, vol. 6, no. June, hal. 3501–3510, 2022, doi: 10.53730/ijhs.v6ns6.10185.

[23] J. Lim *dkk.*, "Phytochemical properties and functional characteristics of wild turmeric (Curcuma aromatica) fermented with Rhizopus oligosporus," *Food Chem. X*, vol. 13, no. November 2021, 2022, doi: 10.1016/j.fochx.2021.100198.

[24] A. Lawi, A. A. Dermawan, D. E. Kurniawan, I. M. Reza, F. E. Gulo, dan Z. A. Hasibuan, "Exploring the Landscape of Deep Learning Techniques for IoT Data: A Systematic Literature Review," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 15, no. 2, hal. 578–588, 2025, doi: 10.18517/ijaseit.15.2.20226.

[25] A. Al-Khulaqi, N. Palanichamy, S. C. Haw, dan S. C. Raja, "Evaluating Machine Learning and Deep Learning Algorithms for Predictive Maintenance of Hydraulic Systems," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 15, no. 1, hal. 52–59, 2025, doi: 10.18517/ijaseit.15.1.12407.

[26] Pontiselly, M. Ahsan, dan M. H. Lee, "Deep Learning and Statistical Process Control Approach to Improve Quality in Immigration Services," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 14, no. 5, hal. 1564–1573, 2024, doi: 10.18517/ijaseit.14.5.19978.

[27] H. Rumapea, M. Zarlis, S. Efendy, dan P. Sihombing, "Improving Convective Cloud Classification with Deep Learning: The CC-Unet Model," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 14, no. 1, hal. 28–36, 2024, doi: 10.18517/ijaseit.14.1.18658. doi: 10.18517/ijaseit.14.1.18658

[28] K. D. K. Wardhani, S. Kasim, A. Erianda, dan R. Hassan, "Deep Learning-based Method in Multimodal Data for Diabetic Retinopathy Detection," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 14, no. 5, hal. 1602–1608, 2024, doi: 10.18517/ijaseit.14.5.11677.

[29] L. Wei, Y. Luo, L. Xu, Q. Zhang, Q. Cai, dan M. Shen, "Deep convolutional neural network for rice density prescription map at ripening stage using unmanned aerial vehicle-based remotely sensed images," *Remote Sens.*, vol. 14, no. 1, 2022, doi: 10.3390/rs14010046.

[30] R. B. Fajar Putra, M. Surahmanto, dan Haris, "Implementation of Machine Learning Classification of Obesity Weight using Dicision Tree," *Int. J. Inf. Syst. Technol.*, vol. 8, no. 158, hal. 110–116, 2024, doi: https://doi.org/10.30645/ijistech.v8i2.354.

[31] F. R. B. Putra, A. Fadlil, dan R. Umar, "Application of Forward Chaining Method, Certainty Factor, and Bayes Theorem for Cattle Disease," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 14, no. 1, hal. 365–374, 2024, doi: 10.18517/ijaseit.14.1.18912.

[32] M. Rizki, F. Rahardika, B. Putra, dan L. Jupriadi, "Comparison of Accuracy Level of Certainty Factor Method and Bayes Theorem on Cattle Disease," *Ilk. J. Ilm.*, vol. 16, no. 3, hal. 343–355, 2024, doi: https://doi.org/10.33096/ilkom.v16i3.1943.343-355.

[33] M. Sajid, T. S. M. Kaleem Razzaq Malik, Ahmad Almogren, A. H. Khan, J. Tanveer, dan A. U. Rehman, "Enhancing intrusion detection : a hybrid machine and deep learning approach," *J. Cloud Comput.*, vol. 13, no. 123, hal. 1–24, 2024, doi: https://doi.org/10.1186/s13677-024-00685-x.

[34] J. Jaafari, S. Douzi, K. Douzi, dan B. Hssina, "Towards more efficient CNN - based surgical tools classification using transfer learning," *J. Big Data*, vol. 8, no. 115, hal. 1–17, 2021, doi: https://doi.org/10.1186/s40537-021-00509-8.

[35] L. Alzubaidi *dkk.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, hal. 53–74, 2021, doi: https://doi.org/10.1186/s40537-021-00444-8.

[36] Jalu Nusantoro, Faldo Fajri Afrinanto, Wana Salam Labibah, Zamah Sari, dan Yufis Azhar, "Detection of Covid-19 on X-Ray Image of Human Chest Using CNN and Transfer Learning," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 3, hal. 430–441, 2022, doi: 10.29207/resti.v6i3.4118.

[37] R. A. Tilasefana dan R. E. Putra, "Penerapan Metode Deep Learning Menggunakan Algoritma CNN Dengan Arsitektur VGG NET Untuk Pengenalan Cuaca," *J. Informatics Comput. Sci.*, vol. 05, no. 1, hal. 48–57, 2023. doi: https://doi.org/10.26740/jinacs.v5n01.p48-57

[38] N. Abdelhady, T. H. A. Soliman, dan M. F. Farghally, "Stacked - CNN - BiLSTM - COVID : an effective stacked ensemble deep learning framework for sentiment analysis of Arabic COVID - 19 tweets," *J. Cloud Comput.*, vol. 13, no. 85, hal. 1–21, 2024, doi: https://doi.org/10.1186/s13677-024-00644-6.

[39] N. T. Diba, N. Akter, S. A. H. Chowdhury, dan J. E. Giti, "BMI Prediction from Handwritten English Char acters Using a Convolutional Neural Network," *arXiv Access. Forum*, hal. 1–7, 2024. doi: arXiv:2409.0258 4v1

[40] A. Pandey, M. S. Teja, P. Sahare, V. Kamble, dan M. Parate, "Skin cancer classification using non - local means denoising and sparse dictionary learning based CNN," *J. Electr. Syst. Inf. Technol.*, vol. 11, no. 36, hal. 1–23, 2024, doi: https://doi.org/10.1186/s43067-024-00162-0.

[41] Y. Liang, Y. Ren, J. Yu, dan W. Zha, "Current trajectory image - based protection algorithm for transmission lines connected to MMC - HVDC stations using CA - CNN," *Prot. Control Mod. Power Syst.*, vol. 8, no. 6, hal. 1–15, 2023, doi: https://doi.org/10.1186/s41601-023-00280-3.

[42] A. Sarhan, R. A. Rahem, B. Darwish, A. A. Attia, A. Sneed, dan S. Hatem, "Egyptian car plate recognition based on YOLOv8 , Easy - OCR , and CNN," *J. Electr. Syst. Inf. Technol.*, vol. 11, no. 31, hal. 1–17, 2024, doi: https://doi.org/10.1186/s43067-024-00156-y.

[43] S. Montazeri, H. Jumakhan, S. Afrasiabian, dan A. Mirzaeinia, "Gradient Reduction Convolutional Neural Network Policy for Financial Deep Reinforcement Learning," *arXiv Access. Forum*, hal. 1–8. doi: https://arxiv.org/abs/2408.11859