Infotmatics Management, Engineering, and Information System Journal, Vol. 4, No. 1, June 2026, 8-21
E ISSN : 3025-5384, DOI : https://doi.org/10.56447/imeisj

Classification of Formaldehyde- and Non-Formalin-Containing Chicken Meat Using a Convolutional Neural Network (CNN) with the Mobilenetv2 Architecture in an Android Application

Fauzan Sidik Azis[1], Heri Wahyudi[2], Haris Supriatna[3], Eko Retnadi[4]
STMIK Mardira Indonesia, Bandung [1,2,3,4]
Email: fauzansidikazis13@gmail.com[1], herywahyudi@stmik-mi.ac.id[2], haris.supriatna@stmik-mi.ac.id[3], eko.retnadi@stmik-mi.ac.id[4]

Abstract

Formalin is a toxic substance frequently misapplied to preserve chicken meat; however, it is difficult to identify visually. Conventional laboratory testing is time-consuming, expensive, and unworkable for traditional markets. This study establishes a classification method for identifying chicken flesh adulterated with formalin, using a Convolutional Neural Network (CNN) based on the MobileNetV2 architecture, implemented in an Android application. The image dataset is sourced from primary and secondary sources, then preprocessed and enhanced to increase variability.

The CNN model employs transfer learning and fine-tuning, achieving 98% accuracy and an F1-score exceeding 96% on the test dataset. The trained model is transformed into a TensorFlow Lite (.tflite) file of 2.93 MB, enabling effective offline operation on Android devices. The Android application is constructed with Flutter, allowing users to capture photos or choose images from the gallery for real-time classification. Results are presented with the labels "Formalin" or "Non-Formalin," accompanied by confidence levels.

This system is engineered for user-friendliness and eliminates the need for laboratory equipment, providing precise, consistent categorization outcomes through an intuitive interface. The study also delineates aspects influencing model efficacy, including dataset quality, augmentation methods, and a bifurcated training approach. The designed method aims to serve as a realistic, cost-effective, and autonomous option for consumers and suppliers to identify formalin contamination, while concurrently enhancing knowledge of food safety. Future advancements will include expanding the dataset, field testing, extending to additional meat varieties, and deployment on iOS and web platforms.

Keywords : *: Image Classification,CNN,MobileNetV2*

INTRODUCTION

Chicken meat serves as a primary source of animal protein and is extensively consumed in Indonesia owing to its economical cost and high accessibility. With rising demand, the use of formalin to prolong meat's shelf life has become increasingly prevalent. Formalin detection is typically performed through laboratory testing, which is labor-intensive and cannot be performed autonomously. This necessitates a technology-driven solution that is rapid, precise, and user-friendly for the public.

Numerous studies have examined this matter, including the classification of chicken meat using color space via the KNN algorithm, achieving 84–86% accuracy, and the implementation of CNNs for real-time object identification. Nevertheless, research using the MobileNetV2 architecture for the real-time categorization of formalin- and non-formalin-treated chicken meat via an Android application remains scarce. (Destalia et al., 2023; Sahin et al., 2022) This project aims to develop a MobileNetV2-based Convolutional Neural Network (CNN) model for rapid, precise classification of formalin-treated and non-formalin-treated chicken meat.

Formalin chicken flesh denotes poultry that has been unlawfully preserved with formaldehyde (formalin). This approach is

harmful to human health and violates food safety regulations (Khoiruddin & Tena, 2024). Non-formalin chicken meat features short fibers, a soft consistency, and extensive nutritional benefits, making digestion easier. The natural texture contrasts with that of formalin-treated chicken, which is more robust and abrasive due to chemical modifications. As a result, people prefer non-formalin beef for its enhanced quality and safety (Hestiningsih et al., 2023).

Image classification involves categorizing images into specific classes based on visual characteristics, such as color, texture, and shape. This approach is essential in image analysis as it enables systems to independently recognize and differentiate objects (Velarati et al., 2024). Digital image processing aims to enhance image quality, facilitating interpretation by both people and machines. This method involves examining, improving, and modifying images to produce enhanced visual data for identification and classification (Kusumaningtyas et al., 2024).

A Convolutional Neural Network (CNN) is a deep learning methodology that replicates the functions of biological neural networks. CNN uses a feedforward architecture for classification and backpropagation for learning, enabling autonomous weight updates. The CNN architecture consists of two main phases: the feature extraction layer, which includes convolution and downsampling, and the fully connected layer, which includes flattening, hidden layers, and an output layer for producing final predictions (Banoth & Murthy, 2024).

MobileNetV2 is a convolutional neural network architecture designed to optimize accuracy and processing efficiency, rendering it appropriate for real-time applications on devices with limited specs. This architecture employs depthwise separable convolution, which differentiates spatial feature extraction using depthwise convolution and amalgamates inter-channel information via pointwise convolution (Nazuli et al., 2025).

Android is a Linux-based operating system for mobile devices that includes the operating system, middleware, and various applications, aimed at enabling developers to create applications tailored to user needs (Fahrudin & Illah, 2023). TensorFlow Lite is a TensorFlow framework engineered to deploy complex models on Android, optimized for compactness and performance on mobile devices (Hussain et al., 2022).

This project seeks to offer pragmatic solutions for food safety and advance image processing technologies within the health and food business sectors. This research aims to identify and classify chicken meat treated with formalin from untreated meat using the MobileNetV2 architecture.

METHOD

This study employs a descriptive, quantitative methodology to categorize formalin- and non-formalin-treated chicken meat using a Convolutional Neural Network (CNN) with the MobileNetV2 architecture. The research phases commence with firsthand observations of the circulation practices of formalin-treated chicken meat, hence underscoring the necessity for a detection system. Interviews are conducted with manufacturers and consumers to gather data on formalin use, detection issues, and user specifications for the application under development. A literature review analyzes

several sources, including journals, books, and prior research on formalin identification, convolutional neural networks (CNNs), MobileNetV2, and the creation of Android applications. The dataset comprises 688 photos of chicken meat, including 344 photographs of formalin-treated chicken from a previous study by Muhammad Fikri Ruslan (2024) and 344 images of non-formalin chicken captured by the author using a smartphone camera. All photos are annotated manually prior to the training procedure. The CNN model, employing the MobileNetV2 architecture, is trained on the processed dataset in Google Colab, with assessment metrics including accuracy, precision, recall, F1-score, and a confusion matrix. The trained model is ultimately converted to TensorFlow Lite format and included in a Flutter-based Android application. Functional testing verifies that the system can categorize chicken meat photos in real time in accordance with the research objectives.

System Development Methodology

This study used a Rapid Application Development (RAD) methodology to develop a formalin-detection application for poultry flesh. This method is selected for its capacity to expedite the iterative system development process. The phases of system development are categorized as follows:

1. Requirements Planning: The identification and analysis of issues are performed to ascertain the functional requirements of the formalin and non-formalin chicken meat categorization application. The data is cleansed, standardized, and transformed into a format suitable for algorithmic processing. Missing values and extraneous data are eliminated to uphold model integrity.
2. System Design: The system design serves as a blueprint for developing the Android-based mobile application to fulfill user requirements.
3. Development: The implementation phase converts the design into code until a viable application is generated.
4. Implementation and Testing: The system undergoes Black Box Testing to verify that all functionalities perform accurately in accordance with the study objectives. The examination of the operational system and the design of the forthcoming system is depicted by a flowchart as follows:

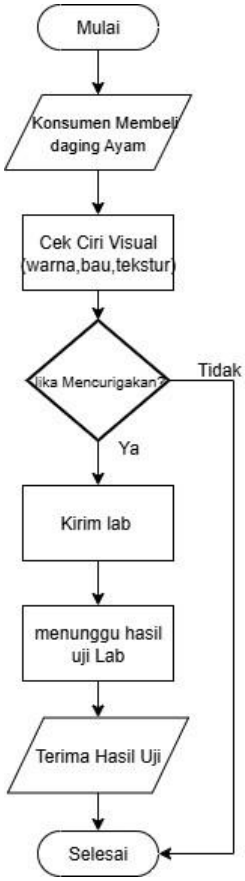

Figure 1. Flowchart of the running system

Figure 1 illustrates the procedure for identifying formalin in chicken meat in markets, which continues to depend on manual inspections of visual characteristics (color, odor, texture) and laboratory analyses when concerns emerge. Interviews with 15 vendors indicate that the majority acknowledge the traits of formalin-treated chicken; however, a minority remain uninformed, despite each vendor serving 10–20 customers daily. Mr. Arienal from the Health Department asserts that routine inspections continue to be performed; however, this approach is considered sluggish, costly, and unworkable, underscoring the need for a more expedient and precise alternative system.

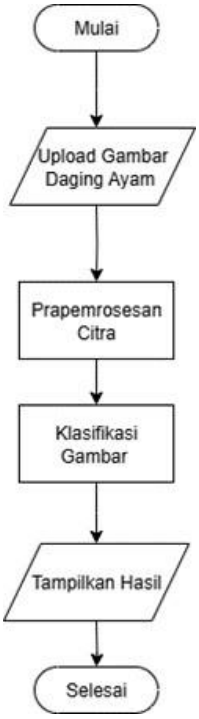

Figure 2. Proposed System Flowchart

Figure 2 depicts the workflow of the proposed system for detecting formalin levels in chicken meat via an Android application. The procedure commences with the user submitting an image, followed by the system preprocessing and classifying using the MobileNetV2 CNN model, ultimately presenting the detection results to the user in real time.

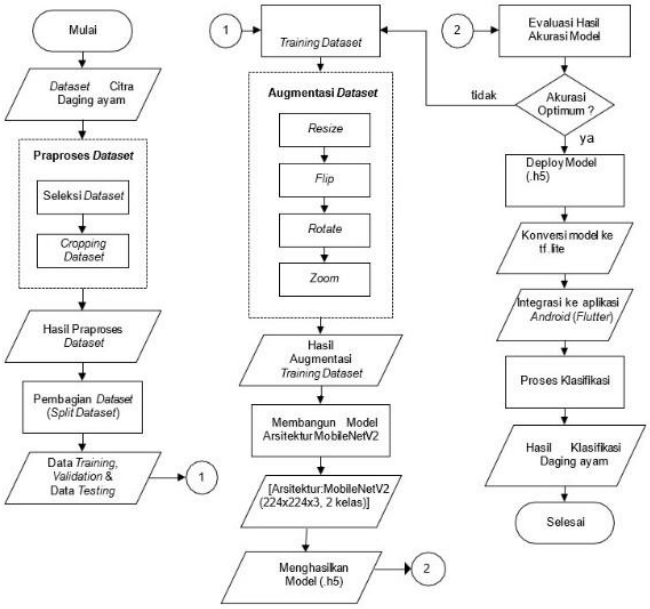

Figure 3. MobileNetV2 Architecture Modeling Flowchart

Figure 3 outlines the phases involved in developing an image classification system for detecting formalin in chicken meat. The procedure commences with the compilation of a dataset comprising 688 photos: 344 of formalin-treated chickens and 344 of non-formalin-treated chickens. The data is subsequently subjected to preprocessing (selection and cropping) to guarantee image quality, along with augmentation (resize, flip, rotate, zoom) to improve variability and avert overfitting. The dataset is partitioned into three segments: 481 photos for training, 103 for validation, and 104 for testing. The model is built on the MobileNetV2 architecture, chosen for its efficiency, lightweight design, and optimization for mobile platforms. Training occurs in two stages: transfer learning and fine-tuning, yielding optimal accuracy and F1-score. Upon evaluation demonstrating optimal performance, the model is preserved in .h5 format and subsequently translated to TensorFlow Lite (.tflite) for incorporation into the Android application (Flutter). In the concluding phase, the application uses this model for real-time image classification, returning predictions as "Formalin" or "Non-Formalin" with confidence levels.

RESULT AND DISCUSSION

This study evaluates the efficacy of the Convolutional Neural Network (CNN) architecture, using MobileNetV2, for classifying photos of formalin- and non-formalin-treated chicken meat. The training procedure consists of two phases: transfer learning and fine-tuning, aimed at attaining a more appropriate feature representation. Model performance evaluation uses metrics such as accuracy, precision, recall, F1-score, and a confusion matrix to assess classification performance across classes. The trained model is subsequently transformed into TensorFlow Lite format and incorporated into a Flutter-based Android application. Real-time testing indicates that the system can swiftly and accurately detect formalin, rendering it a viable alternative for enhancing food safety monitoring within the community.

1. Transfer Learning Stage

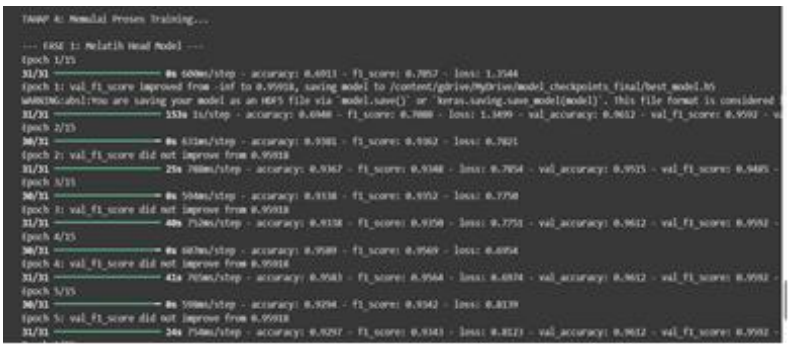

Figure 4. Making a head model

Figure 4 shows that the initial training step emphasizes the classification layer (head classifier) while keeping MobileNetV2 weights frozen. The objective is for the classification layer to adjust to the features derived from the base model. Training outcomes demonstrate that

validation accuracy (val_accuracy) reaches roughly 95–96%, while the F1-score remains consistently above 0.93.

2. Fine-tuning Process

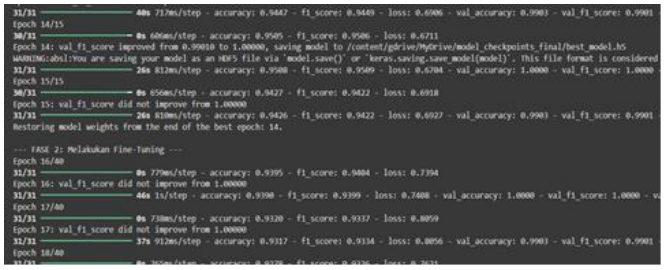

Figure 5. Performing Fine-Tunning

Figure 5 illustrates the residual distribution. Upon stabilization of the head classifier, fine-tuning is executed by unfreezing select layers of MobileNetV2, thereby permitting incremental updates to the weights. This method yields a substantial improvement in model performance, evidenced by attaining val_accuracy = 100% and val_f1_score = 1.000 by the 14th epoch, indicating negligible errors across most of the test data.

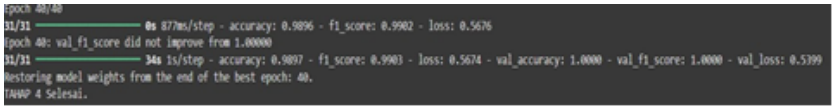

Figure 6. Results of Model Training Implementation

In the concluding phase, the model retains the optimal weights according to validation performance. The training outcomes demonstrate that training accuracy attains 98–99%, whilst validation accuracy remains at 100%, with a consistently flawless F1-score. This indicates that the MobileNetV2 model well accommodates the utilized dataset.

3. Model Evaluation and Analysis Process

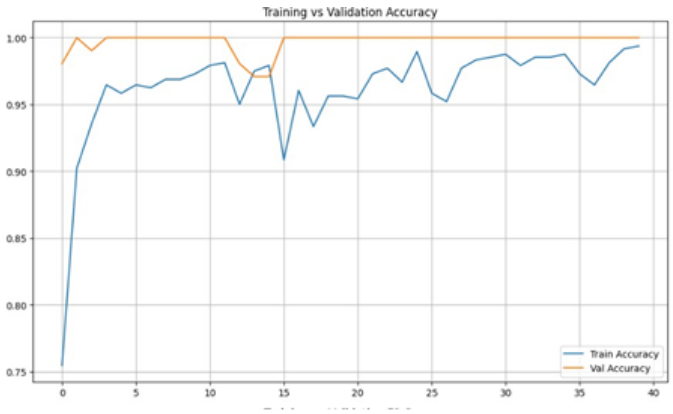

Figure 7. Training and Validation Accuracy Curves

Figure 7 illustrates the accuracy curves for both the training and validation sets. Validation accuracy improves swiftly from the beginning and stabilizes near 1.00, while training accuracy climbs steadily with slight variations over multiple epochs. This indicates that the model generalizes effectively.

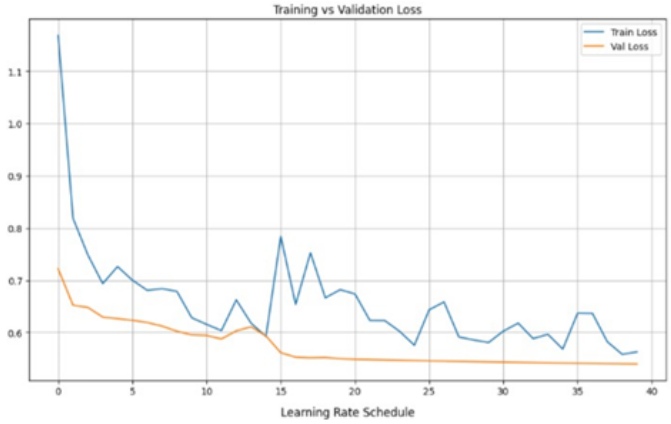

Figure 8. Validation Training Loss Curve

Figure 8 shows the validation loss, which steadily declines to a low, stable value. This indicates that the training process has converged with an exceptionally low error rate.

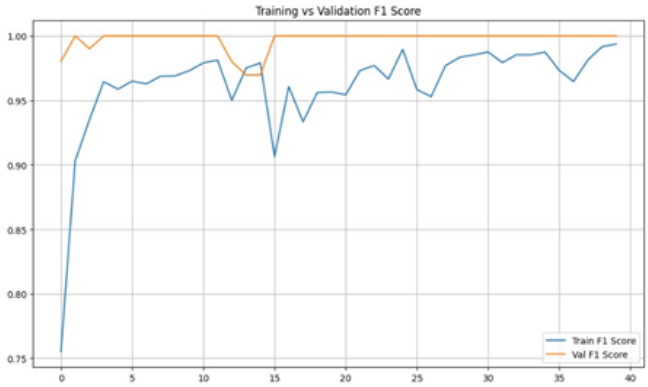

Figure 9. F1-Score Curve of Training and Validation

Figure 9 illustrates the F1-score curve. The validation F1-score remains consistently near 1.00, while the training data exhibits steady enhancement. This requirement affirms that the model is both precise and balanced in its recall.

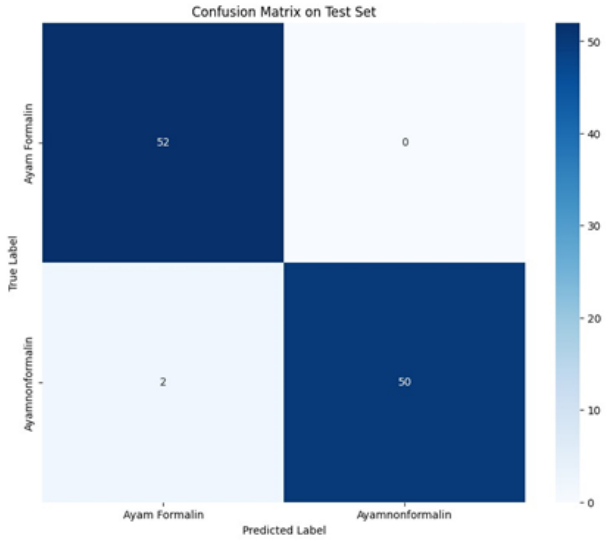

Figure 10. Confusion Matrix

Figure 10 illustrates the confusion matrix derived from the model assessment. Of the 104 test data points, the model committed only 2 classification errors, with the remaining 102 accurately identified. This outcome indicates that the model can accurately differentiate between formalin and non-formalin chicken meat with exceptional precision.

```
Classification Report on Test Set:
                precision    recall  f1-score   support

  Ayam Formalin       0.96      1.00      0.98        52
Ayamnonformalin       1.00      0.96      0.98        52

       accuracy                           0.98       104
      macro avg       0.98      0.98      0.98       104
   weighted avg       0.98      0.98      0.98       104

TAHAP 5 Selesai.
```

Figure 11. Classification Report CNN MobileNetV2 Model

Figure 11 illustrates the categorization report, detailing the precision, recall, and F1-score for each category. All evaluation metrics demonstrate exceptional performance, with values exceeding 96%, while the overall accuracy attains 98%. This outcome indicates that the model can reliably categorize photos of both formalin- and non-formalin-treated chicken meat with a low error rate.

4. Interface Implementation

Figure 12. Splash Screen Page

Figure 12 illustrates the design of the application's splash screen, featuring the logo and the application name. This view is displayed briefly upon the application's initial launch before shifting to the main page. The design employs a straightforward layout that embodies the application's character.

Figure 13. Education Page

Figure 13 illustrates the execution of the education page within the application. This site provides details on the attributes of formalin-treated chicken meat, helping users understand the context. This feature is included in the application's main menu, serving as an educational resource and facilitating detection.

Figure 14. Upload Page

Figure 14 illustrates the image upload interface within the application. This site provides two input methods: taking a photo with the camera or selecting an image from the gallery. This feature initiates the real-time formalin classification process, subsequently processed by the MobileNetV2 model.

Figure 15. Results Display Page

Figure 15 illustrates the classification results page. The system displays the anticipated category (formalin or non-formalin) concurrently.

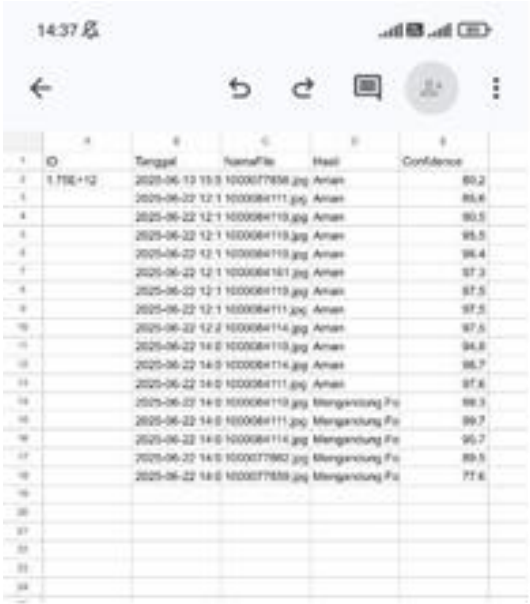

Figure 16. CSV File Contents

Figure 16 illustrates the contents of the CSV file, with columns for classification ID, date, image filename, detection result, and the model's confidence level in the classification. Data is stored in Comma-Separated Values (CSV) format, a straightforward, lightweight, and readily available local format. The CSV file documents the classification outcomes for chicken

flesh, including the model's predictions (Formalin or Safe) and the confidence score, enabling users to view the classification history directly on their devices.

CONCLUSION

The research findings demonstrate that the MobileNetV2 architecture effectively categorized formalin and non-formalin chicken flesh, achieving an accuracy of 98%, precision of 96%, recall of 100%, and an F1-score of 98%. The model demonstrated stability and effective generalization through transfer learning and fine-tuning. The incorporation of TensorFlow Lite into the Android application enables real-time formalin identification with a model size of only 2.93 MB, making the system lightweight, efficient, and readily available to the public without the need for additional equipment.

Several recommendations can serve as references for subsequent research initiatives. These include expanding the dataset by increasing sample size and varying image conditions to enhance the model's robustness in real-world scenarios. Moreover, creating detection systems for additional food products would expand the application's range in food safety. Integrating with cloud-based storage for automatic result archiving and real-time monitoring would be advantageous. Furthermore, exploring other CNN architectures could optimize accuracy and efficiency on mobile devices, while developing cross-platform capabilities and adding image validation features would ensure accurate classification of input images.

REFERENCES

Banoth, R. K., & Murthy, B. V. R. (2024). Soil Image Classification Using Transfer Learning Approach: MobileNetV2 with CNN. SN Computer Science, 5(1), 199. https://doi.org/10.1007/s42979-023-02500-x

Destalia, G., Aditya, M. A., Nurmalasari, R. R., Zaki Hamidi, E. A., Sar'an, M., & Sutisna, D. (2023). Classification of Electronic Components Using MobileNetV2 Architecture-Based Convolutional Neural Network. 2023 9th International Conference on Wireless and Telematics (ICWT), 1–5. https://doi.org/10.1109/ICWT58823.2023.10335269

Fahrudin, T. M., & Illah, I. Z. A. (2023). SkinMate: Mobile-Based Application for Detecting Multi-Class Skin Diseases Classification Using Pre-Trained MobileNetV2 on CNN Architecture. 2023 IEEE 9th Information Technology International Seminar (ITIS), 1–6. https://doi.org/10.1109/ITIS59651.2023.10420370

Hestiningsih, I., Thohari, A. N. A., Kurnianingsih, -, & Kamarudin, N. D. (2023). Mobile Skin Disease Classification using MobileNetV2 and NASNetMobile. International Journal on Advanced Science, Engineering and Information Technology, 13(4), 1472–1479. https://doi.org/10.18517/ijaseit.13.4.18290

Hussain, D., Ismail, M., Hussain, I., Alroobaea, R., Hussain, S., & Ullah, S. S. (2022).

Face Mask Detection Using Deep Convolutional Neural Network and MobileNetV2-Based Transfer Learning. Wireless Communications and Mobile Computing, 2022(1). https://doi.org/10.1155/2022/1536318

Khoiruddin, M., & Tena, S. (2024). Fruit and Vegetable Classification using Convolutional Neural Network with MobileNetV2. Journal of Applied Research In Computer Science and Information Systems, 2(2), 203–210. https://doi.org/10.61098/jarcis.v2i2.197

Kusumaningtyas, E. M., Ramadijanti, N., & Khoiru Rijal, I. H. (2024). Convolutional Neural Network Implementation with MobileNetV2 Architecture for Indonesian Herbal Plants Classification in Mobile App. 2024 International Electronics Symposium (IES), 521–527. https://doi.org/10.1109/IES63037.2024.10665862

Nazuli, M. F., Fachrurrozi, M., Rizqie, M. Q., Abdiansah, A., & Ikhsan, M. (2025). A Image Classification of Poisonous Plants Using the MobileNetV2 Convolutional Neural Network Model Method. MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer, 24(2), 371–380. https://doi.org/10.30812/matrik.v24i2.4284

Sahin, V. H., Oztel, I., & Yolcu Oztel, G. (2022). Human Monkeypox Classification from Skin Lesion Images with Deep Pre-trained Network using Mobile Application. Journal of Medical Systems, 46(11), 79. https://doi.org/10.1007/s10916-022-01863-7

Velarati, K., Sari, C. A., & Rachmawanto, E. H. (2024). A Comparison of Convolutional Neural Network (CNN) and Transfer Learning MobileNetV2 Performance on Spices Images Classification. Journal of Applied Informatics and Computing, 8(2), 413–420. https://doi.org/10.30871/jaic.v8i2.8622