Published online on the journal's webpage: **http://journals.telkomuniversity.ac.id/index.php/dinda**

Journal of Dinda

Data Science, Information Technology, and Data Analytics

Vol. 6 No. 1 (2026) 1 - 8 E-ISSN: 2809-8064

Waste Classification Dashboard Using TensorFlow and Convolutional Neural Network (CNN) for Sustainable Waste Management

Aldi Daim Fauzan[1*], Muhamad Rafif Al Hafizh[2], Woro Isti Rahayu[3], Noviana Riza[4]

[1234] Program Studi Sains Data, Universitas Logistik dan Bisnis Internasional

[1*] aldidaimfauzan15@gmail.com [2]rafif.alhafizh37@gmail.com [3]woroisi@ulbi.ac.id [4] novianariza@ulbi.ac.id

Abstract

Effective waste management remains a major challenge in maintaining environmental cleanliness and public health. Accurate and consistent waste classification plays a crucial role in supporting recycling processes, reducing pollution, and improving disposal efficiency. This study develops a waste classification dashboard based on Convolutional Neural Networks (CNN) implemented with TensorFlow and optimized using TensorFlow Lite (TFLite) for lightweight and cross-platform deployment. The model is integrated into a Flask-based web application that allows users to upload waste images and receive instant classification results. A fine-tuned ResNet50 architecture was trained on the "Recyclable and Household Waste Classification" dataset containing 30 diverse categories of recyclable and household waste. After training and fine-tuning, the model achieved 91% training accuracy, while the real-world deployment accuracy reached 67.33%, showing a notable performance gap caused by domain differences between standardized datasets and locally captured waste images. Unlike previous studies that focused primarily on binary or small-scale classification, this research introduces a scalable 30-class CNN model optimized for real-time and web-based deployment. The proposed dashboard emphasizes both computational efficiency and user accessibility, enabling effective waste recognition on standard devices without the need for high-end hardware. Furthermore, it incorporates a user feedback mechanism for continuous improvement and adaptive learning. Overall, the study highlights how deep learning can advance automated waste identification and classification, enhance environmental awareness, and support the implementation of smart and sustainable waste management systems.

Keywords: *waste classification, cnn, tensorflow, tflite, flask, deep learning*

1. Introduction

Waste management has become a crucial global issue that requires serious attention [1]. Improperly managed waste can cause environmental and public health problems. One important effort in waste management is waste separation based on its types, such as plastic, paper, metal, glass, clothing, and others, which can improve recycling and disposal efficiency.

In Indonesia, public awareness of waste separation remains low in several regions [2]. With a large population, the high volume of daily waste poses a significant challenge. As a solution, Artificial Intelligence, particularly Convolutional Neural Networks (CNN), can be utilized to automatically detect and classify waste categories to support more effective and sustainable waste management. CNN is a deep learning algorithm capable of extracting features from image data and reducing its dimensions without losing key characteristics [3].

This study develops a waste classification dashboard using TensorFlow and CNN to assist users in uploading waste images and receiving category predictions such as plastic, paper, metal, glass, and clothing. The model is trained using TensorFlow, while the web-based dashboard ensures accessibility for individuals and institutions [4].

Similar research was conducted by Dwiatmoko et al. (2024), titled *Classification of Organic and Non-Organic Waste Images Using CNN* [5]. Their system achieved 99% accuracy, proving that CNN can be an effective solution for supporting better waste management.

Received: 09-09-2025 | Revised: 24-11-2025 | Published: 24-02-2026

Previous studies on waste classification have mostly focused on small-scale or binary problems, such as differentiating between organic and non-organic waste. Although these models often achieved high accuracy in controlled laboratory settings, they generally lack scalability and adaptability when applied to real-world conditions. Most of them do not support multi-class recognition and have limited feasibility for web-based or lightweight deployment, which restricts their practical use in community or institutional environments.

To overcome these limitations, this study proposes a 30-class CNN-based waste classification model integrated into a Flask web dashboard and optimized through TensorFlow Lite (TFLite) for real-time and lightweight deployment. This approach allows users to perform waste image recognition efficiently on standard computing devices. The main contributions of this study include: (1) achieving a balance between high training accuracy and real-world deployment, (2) providing a scalable architecture suitable for diverse waste categories, and (3) offering a user-interactive feedback system to improve model adaptability over time. These contributions highlight the novelty of this research in bridging the gap between deep learning performance and practical waste management applications.

The proposed dashboard is expected to serve as a practical tool to help communities separate waste more efficiently, raise environmental awareness, and support sustainable waste management. Furthermore, it holds potential for integration with IoT (Internet of Things) technologies for future smart waste management systems.

2. Research Methods

2.1 *Literature Review*

This research employs the Python programming language. Python is an interpreted language used to execute program code [6]. It was created by Guido van Rossum in 1990 and was designed to be easy to use and efficient, suitable for both beginners and professionals in programming.

The algorithm applied in this study is the Convolutional Neural Network (CNN), a type of artificial neural network specifically designed to process image and video data. CNN is one of the most widely used neural network architectures for image-based data [7].

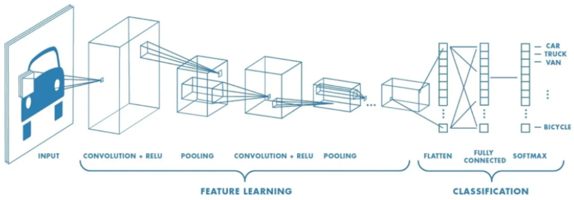

Figure 1. CNN Workflow

CNN consists of several main layers that allow the model to learn features from visual data efficiently. These layers include the Convolutional Layer, which extracts features from the input image; the Pooling Layer, which reduces feature dimensions and produces an output depending on the pooling type; the Rectified Linear Unit (ReLU), which accelerates training by mapping negative values to zero [8]; the Flatten Layer, which transforms 2D matrices into linear feature representations for the next layers [9]; the Fully Connected (Dense) Layer, which demonstrates the information processed by previous layers [10]; and the Softmax Layer, which converts the final output into probabilities representing the likelihood of each class.

This study also utilizes the TensorFlow framework to implement machine learning algorithms and execute tasks using object information for classification and recognition [7]. TensorFlow provides a flexible interface for model development across devices ranging from CPU to GPU, and even mobile platforms. Its main features include computational graphs, deep learning support, distributed computing, integration with Keras, tensor-based data processing, and deployment across platforms.

Additionally, ResNet50 is employed as a CNN architecture with 50 layers. ResNet (Residual Network) is widely known in computer vision tasks because it addresses the degradation problem in deep networks by applying residual or shortcut connections.

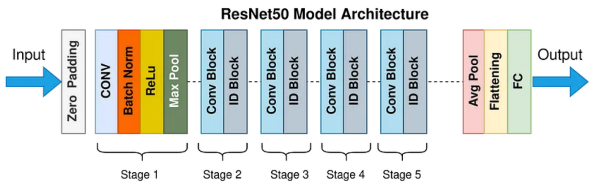

Figure 2. Resnet Achitecture

At the initial stage, input images are processed through convolutional layers of specific sizes. The extracted features are then normalized using batch normalization before passing to activation layers [11]. In stages 2–5, feature extraction and residual blocks enable deeper feature learning. Finally, the fully connected and output layers combine previous features into a final representation, classifying them into the specified categories.

For web integration, this study applies Flask, a Python-based micro web framework. Flask provides essential libraries and functionalities to build web applications without coding everything from scratch [12].
The Pillow library is also used for image manipulation. Pillow supports various operations such as resizing, rotating, changing colors, adjusting contrast, and more. Its primary use includes image archiving and batch processing [13].

2.2 Research Workflow

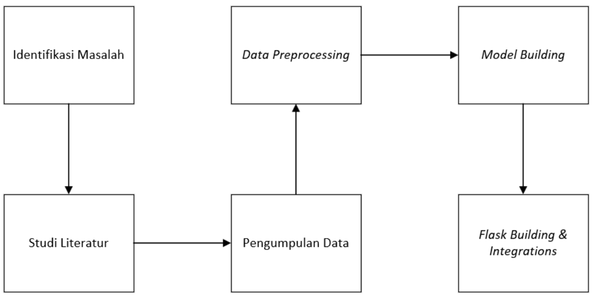

Figure 3. Research Workflow

The research workflow ensures structured and systematic execution to obtain data and achieve research objectives. This study begins with a literature review, a set of activities involving the collection of references, reading, note-taking, and managing research materials [14]. The dataset is sourced from Kaggle, titled *"Recyclable and Household Waste Classification"*, consisting of 30 folders with various waste categories.

2.2.1 Data Preprocessing

Data preprocessing is a technique to transform raw data into a useful and efficient format [15]. In this study, the preprocessing stages include data loading, uploading the dataset and initializing it for model use; preprocessing and exploration, scaling pixel values to standard ranges and analyzing class distribution and image patterns through visualizations; and data augmentation, applying techniques such as rotation, flipping, and zooming to overcome dataset limitations in size and diversity.

Table 1. Dataset Composition (5 Samples)

Category	Images (Train)	Images (Validation)	Total
aerosol_cans	88	22	110
cardboard_boxes	72	18	90
coffe_grounds	81	18	90
tea_bag	88	22	110
plastic_water_bottle	89	22	111

Table 1 represent a portion of the dataset distribution used in this study, showing five representative waste categories. Each category follows and 80:20 split ratio for training and validation, respectively. The complete dataset consists of 30 categories with approximately 15,000 images sourced form the "Recyclable and Household Wate Classification" dataset on Kaggle. The selected samples demonstrate that each class maintains a relatively balanced number of images, ensuring that the model can generalize effectively accrose diverse waste types.

2.2.2 Model Building

The next step is model building, the process of constructing a machine learning or deep learning model from scratch until it is ready for prediction or classification tasks.

It includes model initialization, creating the CNN architecture or applying transfer learning with pre-trained models such as ResNet50, where this model has been previously trained on the ImageNet dataset and is known for its strong performance in image classification tasks; model definition, adding CNN components including convolutional layers for spatial feature extraction, pooling layers to reduce dimensionality and prevent overfitting, activation layer for non-linearity, batch normalization for training acceleration and stability, and dropout layers as a regularization technique to prevent overfitting; and model compilation and training, selecting a suitable multi-class loss function, optimizer (e.g., Adam or SGD), and evaluation metrics, after which training is performed using the dataset over several epochs.

Table 2. Model Hyperparameter Configuration

Parameter	Description
Base Model	ResNet50 (177 layers)
Input Size	244 x 244 pixels
Batch Size	32
Optimizer	Adam
Learning Rate	0.001
Loss Function	Categorical Cross-Entropy
Epochs	5 (initial training) + 5 (fine-tuning)
Validation Split	20%
Regularization	Dropout (0.5)
Data Augmentation	Rotation, Flipping, Zooming, Contrast Adjusment

The hyperparameter configuration was determined through several preliminary trials to obtain the optimal balance accuracy and computational efficiency. The

Adam optimizer with a learning rate of 0.001 ensured stable convergence, while a batch size of 32 provided efficient GPU utilization. Dropout regularization with a rate of 0.5 was applied to minimize overfitting.

A validation split of 20% was used to evaluate model generalization, and the selected data augmentation techniques improved model robustness against variations in object orientation, background, and lighting.

Furthermore, fine-tuning involved adjusting hyperparameters and unfreezing some pre-trained layers to improve accuracy, as fine-tuning allowed the model to adjust learned features to the specific visual characteristics of the waste images. Finally, model prediction was conducted once optimized, where the model was used for predicting new, unseen data, input images were resized to 244x244 pixels and passed through the model to generate class probabilities via softmax activation, and the final predicted label corresponds to the class with the highest probability.

3. Flask Building & Integration

For web deployment, Flask is used to integrate the trained model. The process includes initializing the Flask app, setting up the programming environment and dependencies; loading the TensorFlow Lite model, importing the converted .tflite model for lightweight and efficient prediction; defining routes, creating endpoints to handle user requests and URL access in the application; handling requests, managing GET and POST requests, displaying pages, and receiving uploaded images; processing data and performing computation, preprocessing uploaded images and running predictions with the model; rendering templates, generating HTML pages with prediction results; and client display result, returning classification results (label and confidence score) to the user in real-time.

In the context of university environment, the developed waste classification dashboard can serve as a practical educational and sustainability tool, where it can be implemented in campus recycling stations or environmental laboratories to help students and staff automatically identify and categorize waste types before disposal. Moreover, the system can be integrated into university sustainability programs, promoting digital literacy and environmental awareness through the use of AI-based technologies, which aligns with the goal of many academic institutions to create smart and sustainable campuses, where data-driven tools support waste management, research, and community engagement.

4. Results and Discussion

The development of the waste image classification model began with data loading, where the dataset was retrieved from the local directory and split into training and validation subsets. This was followed by data preprocessing to prepare the data for model training. Data exploration was carried out to analyze class distributions and image characteristics, then data augmentation techniques such as flipping, rotation, zooming, and contrast adjustments were applied to enrich dataset diversity.

The model was developed using ResNet50 as the base architecture, modified with additional layers including Global Average Pooling, Dropout, and Fully Connected layers, resulting in a total of 177 layers. The model was compiled using the Adam optimizer and categorical cross-entropy loss function.

Initial training was conducted for 5 epochs, producing an accuracy of 70%. To improve performance, fine-tuning was applied to selected ResNet50 layers, followed by an additional 5 epochs of training. The fine-tuned model achieved a significantly improved accuracy of 91%, with consistently decreasing validation loss, indicating strong generalization without signs of overfitting or underfitting.

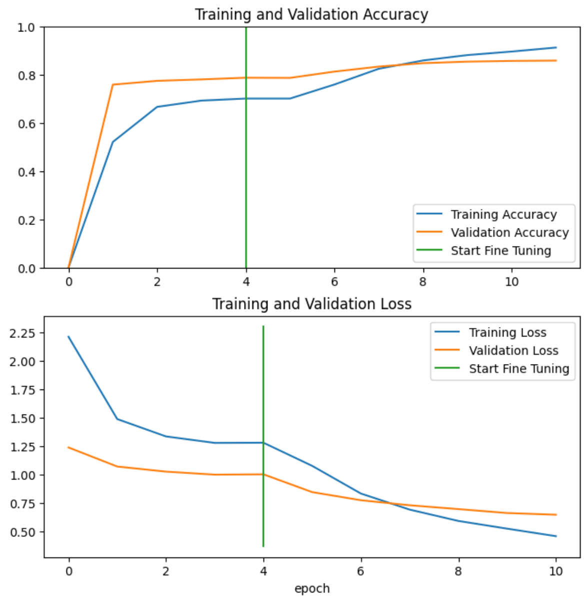

Figure 4. Training and Validation Results

Figure 5. Model Prediction

To further evaluate model performance, a comparison between the baseline CNN and the fine-tuned ResNet50 model was conducted. The comparison focused on training accuracy, validation accuracy, and overall generalization capability.

Table 3. Baseline CNN and ResNet50 Comparation

Model	Epochs	Training
Baseline CNN	10	91.31%
ResNet50	10	91.00%

In this study, model comparison was limited to two architectures – a baseline CNN and the fine-tuned ResNet50 – to maintain experimental focus and computational efficiency. While other architectures such as VCG16 or MobileNet are commonly used for image classification, they were not included in this stage because the main objective was to evaluate the performance improvement achieved through deeper residual learning rather than to perform an extensive model benchmarking. The chosen comparison effectively demonstrates how residual connections in ResNet50 enhance feature extraction and generalization, leading to higher validation accuracy and better robustness for complex waste categories.

```
Classification Report:
                            precision    recall  f1-score   support

              aerosol_cans       0.91      0.92      0.91        87
        aluminum_food_cans       0.61      0.47      0.53       110
        aluminum_soda_cans       0.88      0.95      0.91        94
           cardboard_boxes       0.62      0.68      0.65       100
       cardboard_packaging       0.63      0.54      0.58        90
                  clothing       0.94      0.77      0.85        96
            coffee_grounds       0.98      0.98      0.98       101
disposable_plastic_cutlery       0.98      0.97      0.97        92
                 eggshells       0.92      0.93      0.92       105
                food_waste       0.90      0.93      0.92        89
     glass_beverage_bottles       0.78      0.93      0.85       100
  glass_cosmetic_containers       0.94      0.94      0.94        97
           glass_food_jars       0.93      0.89      0.91        97
                 magazines       0.87      0.92      0.89       106
                 newspaper       0.88      0.87      0.88       118
              office_paper       0.77      0.88      0.82        97
                paper_cups       0.91      0.72      0.80        93
           plastic_cup_lids       0.91      0.86      0.89        96
  plastic_detergent_bottles       0.99      0.87      0.93        95
    plastic_food_containers       0.87      0.93      0.90       100
      plastic_shopping_bags       0.80      0.92      0.86       100
        plastic_soda_bottles       0.87      0.75      0.81       100
             plastic_straws       0.95      0.97      0.96       116
         plastic_trash_bags       0.88      0.90      0.89        99
      plastic_water_bottles       0.91      0.80      0.85       111
                      shoes       0.92      0.96      0.94       106
            steel_food_cans       0.52      0.69      0.60        93
            styrofoam_cups       0.88      0.92      0.90       101
  styrofoam_food_containers       0.93      0.94      0.94       101
                  tea_bags       0.86      0.82      0.84       110

                  accuracy                           0.85      3000
                 macro avg       0.86      0.85      0.85      3000
              weighted avg       0.86      0.85      0.85      3000
```

Figure 6. Classification Report

The classification report in Figure 6 shows that most waste categories achieved precision and recall values above 0.80, indicating strong model consistency across diverse classes. However, certain categories such as “aluminum_food_cans” and “cardboard_packaging” exhibited lower performance compared to other classes.

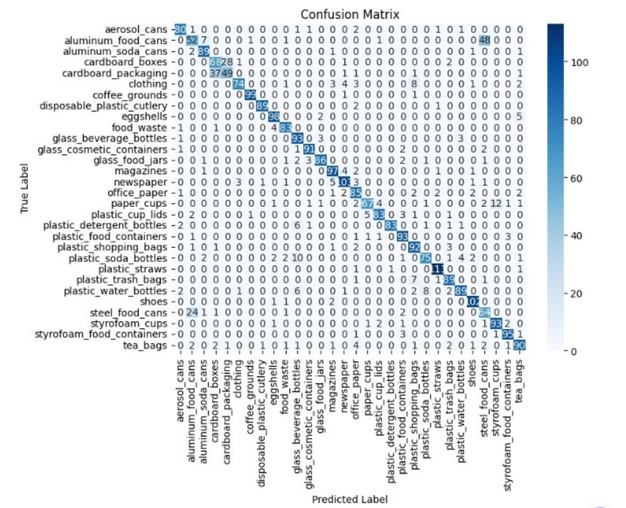

Figure 7. Confusion Matrix

Misclassifications in these two categories were mainly caused by visual and texture similarities with other materials. For example, aluminum food cans often being confused with "steel_food_cans" due to similar reflective metallic surfaces, while "cardboard_packaging" tends to resemble "cardboard_boxes" in both color and shape. The confusion matrix (Figure 7) confirms these overlaps, showing several cross-predictions between these visually similar waste types.

These findings suggest that even though the model performs well overall, class-specific ambiguity remains a challenge in waste classification problems. Improving the dataset with additional localized samples and targeted augmentation for metallic and cardboard textures is expected to enhance model robustness in future work.

After training, the model was converted into TensorFlow Lite (.tflite) format for integration into a Flask-based web application. This format was chosen for its lightweight and efficient execution, particularly on edge devices or web applications. The Flask application was developed with several key features, including image upload, waste classification, and result presentation with class labels, confidence levels, and waste management recommendations.

Classification results are displayed in real-time on the web interface and stored in a database for documentation and model improvement. The system automatically categorizes predictions as *accurate* if the confidence level ≥ 75%, saving them into a new dataset according to their class. If the confidence level < 75%, the system labels them as *inaccurate* and requests user feedback. The web application also displays the model summary and training history stored in the database, allowing for effective monitoring, analysis, and model enhancement without retraining from scratch.

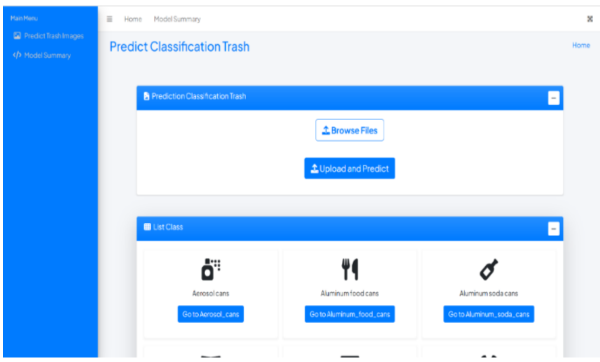

Figure 8. Homepage

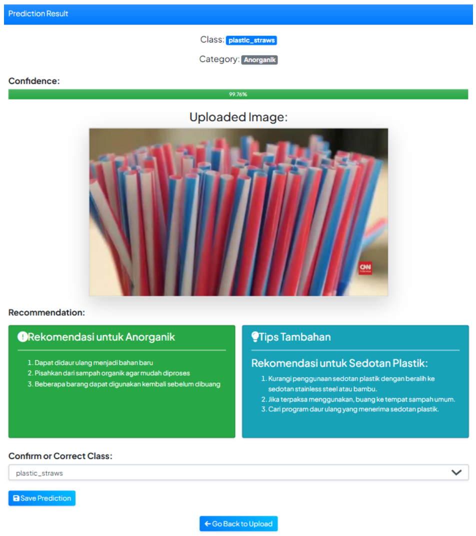

Figure 9. Real-Time Classification Results

Although the training and validation accuracy reached 91%, the model's real-world performance during deployment dropped to an average detection success rate of 67.33%. This performance degradation primarily results from a domain gap between the training dataset and the real-word test images. The training data used in this study originated from the *Recyclable and Household Waste Classification* dataset on Kaggle, which mostly contains foreign (non-Indonesian) waste samples with standardized backgrounds and lighting conditions. In contrast, the testing phase utilized locally sourced waste images collected through the web interface, representing Indonesian waste with more diverse shapes, textures, and environmental contexts.

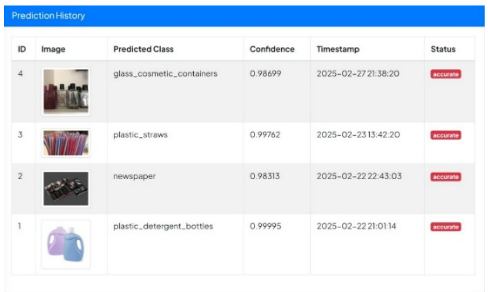

Figure 10. Model Summary

Each class was tested five times using real-world samples, and the model's success rate was computed as the ratio of correct predictions to total trials per class. This discrepancy indicated that the model, while accurate in controlled conditions, is sensitive to domain-specific variations such as packaging style, color tone, and labeling language differences on local products.

These findings suggest that the system is effective for image-based waste classification but still has room for improvement, particularly for classes with lower accuracy. Enhancements such as increasing the amount of training data, applying more targeted augmentation techniques, and retraining deeper layers could further improve performance. Moreover, integrating waste management recommendations through the web application is expected to raise user awareness and participation in promoting smarter and more environmentally friendly waste management.

5. Conclusion

This study developed a web-based dashboard for waste classification using a ResNet50 model with 177 layers, integrated into a Flask application via TensorFlow Lite. The model achieved 91% accuracy after fine-tuning and an average classification accuracy of 67.33% across 30 classes. Despite some misclassifications in visually similar categories, the system performed reliably in most cases and demonstrated the potential of deep learning for automated waste classification.

The result highlights the practicality of integrating deep learning models with web applications for real-time waste recognition. The system not only enables efficient waste identification but also supports environmental education and awareness through user interaction and data feedback collection. This demonstrates that AI-based tools can contribute significantly to sustainable waste management practices.

The limitations of this study lie in the amount of data and the system's performance in complex or visually similar classes. Future research will focus on expanding the dataset with mode diverse local samples and integrating the model with a sensor-based IoT waste monitoring system to support smart waste management policies and campus sustainability initiatives.

6. Acknowledgments

The authors would like to express their sincere gratitude to Universitas Logistik dan Bisnis Internasional for the facilities, support, and academic environment that enabled the completion of this research. We also extend our appreciation to Kaggle and the contributor Alistair King for providing the *Recyclable and Household Waste Classification* dataset, which was essential for the development and evaluation of our model.

7. References

[1] K. Kurnia, A. Sholihah, D. B. Hariyanto, and M. Pd, "KAJIAN TENTANG PENGELOLAAN SAMPAH DI INDONESIA," vol. 3, no. 3, Jun. 2020, Accessed: Feb. 06, 2025. [Online]. Available: https://ejournal.unesa.ac.id/index.php/swara-bhumi/article/view/35038

[2] C. F. Sari, L. Salsabila, K. T. Lodan, and T. Dompak, "Challenges of Waste Growth through Collaborative City Governance in Batam City," *Dialogue : Jurnal Ilmu Administrasi Publik*, vol. 6, no. 2, pp. 761–773, Dec. 2024, doi: 10.14710/dialogue.v6i2.23809.

[3] N. D. Girsang, "Literature Study of Convolutional Neural Network Algorithm for Batik Classification," vol. 1, no. 1, 2021, doi: 10.47709/briliance.v1i1.1069.

[4] Ahmad Dani Eka Putra and Henny Dwi Bhakti, "Implementasi Website Bank Sampah pada Kelurahan Pekelingan," *Switch : Jurnal Sains dan Teknologi Informasi*, vol. 2, no. 4, pp. 12–22, Aug. 2024, doi: 10.62951/switch.v2i4.177.

[5] F. Dwiatmoko, D. Utami, N. A. Sivi, U. Nahdlatul, and U. Lampung, "Klasifikasi Citra Sampah Organik dan Non Organik Menggunakan Algoritma CNN (Convolutional Neural Network)."

[6] S. Rahman *et al.*, *PYTHON : DASAR DAN PEMROGRAMAN BERORIENTASI OBJEK TAHTA MEDIA GROUP*. Tahta Media Group, 2023. Accessed: Feb. 06, 2025. [Online]. Available: https://tahtamedia.co.id/index.php/issj/article/view/344

[7] K. Azmi, S. Defit, and U. Putra Indonesia YPTK Padang Jl Raya Lubuk Begalung-Padang-Sumatera Barat, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," vol. 16, no. 1, p. 2023.

[8] I. Husna, M. Ulum, A. Saputro, Haryanto, D. Laksono, and 6Dian Purnamasari, "Rancang Bangun Sistem Deteksi Dan Perhitungan Jumlah Orang Menggunakan Metode Convolutional Neural Network (CNN)," *Seminar Nasional Fortei Regional 7*, vol. 5, no. 1, pp. 3–3, Dec. 2022.

[9] D. R. R. Putra and R. A. Saputra, "IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK MENDETEKSI PENGGUNAAN MASKER PADA GAMBAR," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 11, no. 3, Aug. 2023, doi: 10.23960/jitet.v11i3.3286.

[10] I. Maulana, N. Khairunisa, and R. Mufidah, "DETEKSI BENTUK WAJAH MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN)," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 6, pp. 3348–3355, Jan. 2024, doi: 10.36040/jati.v7i6.8171.

[11] I. Office, R. Ariffudin, and B. Hidayatulail, "Klasifikasi Penyakit Pneumonia menggunakan Metode Convolutional Neural Network (CNN)," *Techné : Jurnal Ilmiah Elektroteknika*, vol. 23, no. 2, pp. 4–4, Nov. 2024, doi: 10.31358/techne.v23i2.491.

[12] D. J. Evan and P. O. N. Saian, "IMPLEMENTASI PYTHON FRAMEWORK FLASK PADA MODUL TRANSFER OUT TOKO DI PT XYZ," *JIPI (Jurnal Ilmiah Penelitian*

dan Pembelajaran Informatika), vol. 8, no. 4, pp. 1121–1131, Nov. 2023, doi: 10.29100/jipi.v8i4.4020.

[13] J. A. Clark, "Pillow (PIL Fork) Documentation," Feb 04, 2025. Accessed: Feb. 04, 2025. [Online]. Available: https://readthedocs.org/projects/pillow/downloads/pdf/latest/

[14] Z. Mestika, *Metode penelitian kepustakaan*, vol. 1. Jakarta: Yayasan Pustaka Obor Indonesia, 2008.

[15] I. M. Hamdani, N. Nurhidayat, A. Karman, N. F. Adhalia H, and A. H. Julyaningsih, "Edukasi dan Pelatihan Data Science dan Data Preprocessing," *Intisari: Jurnal Inovasi Pengabdian Masyarakat*, vol. 2, no. 1, pp. 19–26, Jun. 2024, doi: 10.58227/intisari.v2i1.125.