Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.5, October 2025, pp. 2278~2287
e-ISSN: 2548-6861

Mobile-Based Multi-Output Animal Taxonomy Classification Using CNN with Edge and Cloud Deployment

Jason Patrick[1], Shumaya Resty Ramadhani[2]*
[1,2] Teknik Informatika, Politeknik Caltex Riau
jason21timahasiswa@pcr.ac.id[1], shumaya@pcr.ac.id [2]*

Article Info

Article history:

Received 2025-08-13
Revised 2025-08-27
Accepted 2025-09-03

Keyword:

Animal Taxonomy Classification,
Android,
CNN,
Multi-Output Multi-class Classification,
Deep Learning.

ABSTRACT

Distinguishing animals that appear visually similar but belong to different species or taxonomic groups, such as Eurasian and house sparrows, koi and common carp, or leopard cat and domestic cat, remains challenging and hinders biodiversity education. This study develops a Convolutional Neural Network (CNN)-based multi-output, multi-class taxonomy classification system capable of identifying seven animal species across five taxonomic levels (class, order, family, genus, species), producing 35 possible outputs. The dataset comprised 6,998 images from public sources. Among various configurations, the best-performing model (D3-M2), trained using the High Dataset with 256×256 input size, 0.2 dropout, and four hidden layers, achieved 90.15% average accuracy, the highest F1-score at the family level (98.11%), and 95.99% at the species level. Slightly lower species-level performance was due to high visual similarity among particular species. Edge AI deployment offered faster inference (0.17s) and offline capability, making it ideal for field use. Real-world testing under bright and low light at 30, 60, and 100 cm showed higher accuracy (64.8%) than low light (57.1%), with the most stable performance at 60 cm. However, limitations include an imbalanced dataset and limited environmental variation affecting species-level accuracy. Future work will focus on expanding dataset diversity and employing advanced architectures to improve fine-grained classification. This system offers a practical tool for biodiversity education and species identification, particularly in field environments where rapid, offline, and accurate classification is essential.

I. Introduction

At the high school level, biology curricula aim to familiarize students with the rich diversity of the animal kingdom and fundamental taxonomic principles. In practical learning, however, distinguishing between visually similar species common in local environments can be a significant challenge for students, often due to a lack of detailed taxonomic expertise. Examples include the house sparrow (Passer domesticus) and Eurasian tree sparrow (Passer montanus), koi (Cyprinus rubrofuscus) and common carp (Cyprinus carpio), as well as domestic cat (Felis catus) and leopard cat (Prionailurus bengalensis). In Indonesia, such animals are often misidentified, leading to ecological awareness, conservation efforts, and species management misunderstandings [2], [17], [18], [19]. This study was validated in collaboration with a biology teacher from SMA Dharma Loka to ensure that the outputs comply with scientific taxonomic standards and remain relevant for educational use.

This issue highlights the need for tools to identify and classify such animals accurately, enabling the public to better understand their characteristics and taxonomic classification. With the advancement of artificial intelligence (AI) technology, Convolutional Neural Networks (CNNs) have been widely applied in image classification, including animal identification. CNNs have demonstrated strong capabilities in recognizing complex visual patterns with high accuracy and efficiency, making them suitable for tackling image-based object classification problems.

Several prior studies have shown the effectiveness of CNNs for animal classification. Nazirman (2023) developed a CNN-based animal learning application capable of identifying ten animal species with an accuracy of 75.42%. Ndun (2020) applied CNNs to identify 255 bird species, though the validation accuracy remained at 61.26%. Nugraha et al. (2022) used the Inception ResNet-V2 architecture to detect 325 bird species, achieving an accuracy of 97.78%. Antonio and Hartati (2023) demonstrated that data augmentation could improve venomous snake classification accuracy to 71.09%, while Andriani et al. (2023) applied CNNs to identify five dog breeds with an accuracy of 72%. These results indicate that CNNs are a relevant and promising approach for developing animal classification systems capable of providing comprehensive and informative taxonomic outputs.

CNN-based animal classification models have been implemented in various digital media, such as websites and Android-based mobile applications, due to their high accessibility and ability to reach a broad range of users, including students, researchers, and the general public. Android-based mobile applications, in particular, offer interactive and practical tools for on-site species identification. Thus, integrating CNN models with mobile applications could improve public awareness of biodiversity conservation through scientific animal recognition.

However, deploying CNN-based models on mobile devices presents processing efficiency and latency challenges. [34] compared AI models implemented locally on devices (Edge AI) with those hosted on cloud servers (Cloud AI). The study found that Edge AI excelled in response time and device autonomy, while Cloud AI offered greater computational capacity and model maintenance flexibility.

Based on these findings, this research aims to develop a CNN-based animal classification system that identifies seven animal species across five taxonomic levels: class, order, family, genus, and species. The system is implemented with cloud and edge approaches, with performance compared in terms of accuracy and efficiency. The proposed system is expected to be accurate, informative, efficient, and user-friendly for direct public use via mobile devices.

II. METHOD

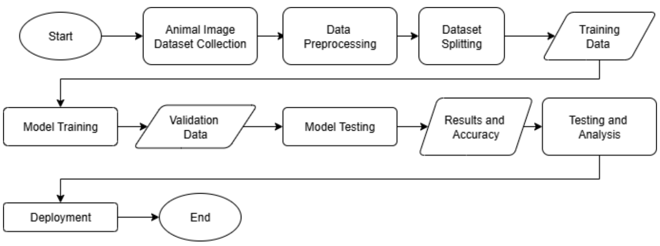

Figure 1 Deep Learning Flowchart

A. Deep Learning Workflow

Deep Learning Workflow in this research consists of several stages: dataset collection from public sources, preprocessing (including cleaning, resizing, normalization, and augmentation), dataset splitting into training and validation sets, CNN model design using the Keras Functional API, model training with Adam optimizer and categorical cross-entropy loss, model evaluation using confusion matrix and performance metrics, optimization if necessary, and finally deployment in both Edge AI and Cloud AI environments.

B. Hardware Specifications

All model training was conducted locally on a laptop with the following specifications:

- Processor: AMD Ryzen 5 4600H
- Memory: 24 GB DDR4 RAM
- Storage: 1 TB Samsung SATA EVO 870 SSD
- GPU: NVIDIA GeForce GTX 1650 with 4 GB VRAM

The system was equipped with CUDA Toolkit 11.2 and the compatible cuDNN library to optimize deep learning training. Training was performed using TensorFlow version 2.10, the last release providing full GPU support with CUDA Toolkit 11.2. The use of GPU accelerated the computation process, particularly for convolutional operations and large-batch processing.

C. Dataset and Preprocessing

The dataset consists of images from seven animal species: *Passer domesticus*, *Passer montanus*, *Columba livia*, *Cyprinus rubrofuscus*, *Cyprinus carpio*, *Felis catus*, and *Prionailurus bengalensis*. Images were collected from publicly available datasets on **Kaggle** and **Roboflow**, including bird, cat, and fish image datasets. Table 1 summarizes the dataset sources and image counts. In total, approximately 6,998 images were compiled.

TABLE 1
DATASET CATEGORIZATION

No	Dataset Source	Species	Images
1	https://universe.roboflow.com/wiings/pigeon-mg46t	*Columba livia*	370
2	https://universe.roboflow.com/spirosmakris/pigeons-visup	*Columba livia*	390
3	https://universe.roboflow.com/yo-mlvgz/yolo-pnjth	*Passer domesticus*	40
4	https://universe.roboflow.com/csirceeri-vvkjd/sparrow-pcyzx	*Passer domesticus*	126
5	https://www.kaggle.com/datasets/elvinrustam/a-dataset-of-bird-species	*Passer domesticus*	210
6	https://images.cv/dataset/house-sparrow-image-classification-dataset	*Passer domesticus*	131
7	https://universe.roboflow.com/agriguard-plsxo/eurasian-tree-sparrow-detection-zz1ho	*Passer montanus*	950
8	https://universe.roboflow.com/project-ozodb/where-s-the-leopard-cat	*Prionailurus bengalensis*	45
9	https://universe.roboflow.com/chanwoo/e-species_detect_project	*Prionailurus bengalensis*	137
10	https://universe.roboflow.com/lgps-twnj8/cat-psuy6	*Prionailurus bengalensis*	13

11	https://www.kaggle.com/competitions/leopardcat-ReID/data?select=train	*Prionailurus bengalensis*	490
10	https://www.kaggle.com/datasets/alessiocorrado99/animals10	*Felis catus*	1668
11	https://www.kaggle.com/datasets/farizp/dataset-images-koi	*Cyprinus Carpio*	1700
12	https://universe.roboflow.com/autofis/fish_detection-6dalm	*Cyprinus rubrofuscus*	113
13	https://universe.roboflow.com/college-rhnpn/mtp-dataset-w5da1	*Cyprinus rubrofuscus*	49
14	https://universe.roboflow.com/fish-finder/fish-finder-p1ljc	*Cyprinus rubrofuscus*	64
15	https://universe.roboflow.com/cselabfu/fish-lvore	*Cyprinus rubrofuscus*	502

Three dataset volume categories were prepared to represent different resource scenarios. All collected images from multiple public sources (Table 1) were combined into a single dataset and then balanced using undersampling and oversampling through data augmentation. Specifically:

- Low Dataset: the smallest number of images per class, balanced through undersampling.
- Average Dataset: moderate-sized dataset, balanced using a combination of undersampling and augmentation.
- High Dataset: the largest dataset volume, obtained primarily through oversampling with augmentation.

D. Data Preprocessing

The preprocessing pipeline was designed to prepare the dataset for multi-output CNN training by addressing class imbalance and ensuring input consistency. Data augmentation was applied using the Albumentations library to enhance variability and reduce overfitting. The augmentation operations included: random rotation up to 30° with a probability of 0.5, horizontal flipping with a probability of 0.5, random brightness and contrast adjustment with a probability of 0.3, and a combined transformation of shifting (up to 5%), scaling (up to 5%), and rotating (up to 20°) with a probability of 0.5. Finally, the augmented images were converted into tensors using ToTensorV2().

For the Average Dataset, augmentation balanced the number of samples for each species to 848 images, while in the High Dataset, augmentation increased the samples per species to 1,646 images. All taxonomic labels (class, order, family, genus, species) were one-hot encoded for multi-output classification. Images were resized to 256×256 or 384×384 pixels, and pixel values were normalized to the range [0,1]. The dataset was split into 80% training and 20% validation with stratification to preserve class distribution.

E. Dataset Splitting

Each dataset category was split into training and validation sets using an 80:20 ratio, with stratification performed at the species level to maintain class balance. The split was also group-aware to prevent data leakage from similar or duplicate images across sets. Table 2 shows the dataset configurations used in this study.

TABLE 2
DATASET CATEGORIZATION

Data ID	Input Size	Volume Level
D1	256×256	Low Dataset
D2	256×256	Average Dataset
D3	256×256	High Dataset
D4	384×384	Low Dataset
D5	384×384	Average Dataset
D6	384×384	High Dataset

F. Model Training

The architecture was implemented using the Keras Functional API, which provides greater flexibility than the Sequential API, particularly for building multi-output models. This design was selected to accommodate the hierarchical structure of taxonomy classification, where the network must simultaneously predict five levels: class, order, family, genus, and species. The model employed three to four convolutional blocks, each consisting of a Conv2D layer with ReLU activation and MaxPooling2D. The convolutional layers progressively extracted spatial and semantic features, starting with 32 filters and increasing to 128 filters to capture higher-level representations. After the convolutional blocks, a Global Average Pooling (GAP) layer was applied to reduce dimensionality while preserving spatial information. The extracted features were then passed through a dense layer of 512 neurons with ReLU activation, followed by a Dropout layer. Depending on the model configuration, the dropout rate was set to 0.2, 0.5, or an adaptive dropout strategy to reduce overfitting.

The variations in hidden layers and dropout strategies for all models (M1–M6) are summarized in Table 3, which illustrates the experimental design used to evaluate performance across different architecture configurations.

TABLE 3
MODEL CONFIGURATION

Data ID	Configuration
D1	0.2 Dropout, 3 Hidden Layer (32,64,128)
D2	0.2 Dropout, 4 Hidden Layer (32,64,128,128)
D3	0.5 Dropout, 3 Hidden Layer (32,64,128)
D4	0.5 Dropout, 4 Hidden Layer (32,64,128,128)
D5	Adaptive Dropout, 3 Hidden Layer (32,64,128)
D6	Adaptive Dropout, 4 Hidden Layer (32,64,128,128)

Finally, the architecture branched into five independent softmax output layers, each corresponding to a taxonomy level: class, order, family, genus, and species. The Adam optimizer was used for model training with categorical cross-entropy loss, and each dataset configuration was trained for 50 epochs. This setup ensured the model optimized classification performance across all taxonomic levels simultaneously.

G. Model Evaluation

The trained model was evaluated on the validation set using accuracy, precision, recall (sensitivity), and F1-score metrics. Confusion matrices were generated for each taxonomic level to analyze classification performance.

H. Deployment

The final model was deployed in two approaches:

- Edge AI: The trained model was converted into TensorFlow Lite format for offline inference within an Android application.
- Cloud AI: The model was integrated into a FastAPI-based web service for online inference.

The overall deployment architecture is illustrated in Figure 2, showing both Edge AI and Cloud AI inference pipelines integrated into the Android application.

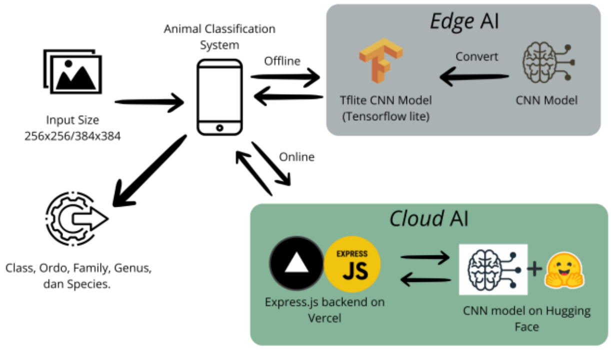

Figure 2 System Architecture

I. Testing

1. Confusion matrix and metrics

Model performance was evaluated using confusion matrices and four primary metrics: accuracy, precision, recall, and F1-score. These metrics measure the model's ability to classify images correctly and consistently across all taxonomic levels.

2. Expert Validation

The system was validated by a biology teacher from Dharma Loka Senior High School (SMA Dharma Loka), a private senior high school in Pekanbaru under the Panca Dharma Educational Foundation. Established in 2008, the school follows the national curriculum and is committed to fostering academic excellence and character development in its students. This validation ensured that the classification outputs adhered to scientific taxonomic conventions, followed correct scientific name formatting, and were relevant for educational purposes. The expert also assessed usability, interface design, and educational suitability, providing suggestions for improving classification accuracy under partial image visibility and refining taxonomy display standards.

3. Real-World Condition Testing

A robustness test evaluated classification accuracy under varying lighting and image capture distances (30 cm, 60 cm, 100 cm). Two lighting conditions were tested: bright and low light. This test aimed to simulate non-ideal field conditions and assess model stability in real-world usage.

4. Application Performance Testing

Performance metrics included latency (s), RAM usage (MB), and CPU usage (%), measured using Android Profiler, Logcat, and in-app logging. Tests were performed on an Android smartphone (≥4 GB RAM) with identical images and conditions for all methods.

III. Results and Discussion

A. Dataset

The dataset comprised images from seven animal species: *Felis catus* (1,646), *Cyprinus rubrofuscus* (1,577), *Passer montanus* (754), *Cyprinus carpio* (695), *Prionailurus bengalensis* (455), *Passer domesticus* (442), and *Columba livia* (367), collected from public sources such as Kaggle and Roboflow, then manually curated to ensure image clarity. The original dataset contained 6988 images, with an imbalanced distribution ranging from 367 to 1,646 images per species. To evaluate the effect of dataset volume on model performance, three dataset categories were prepared:

- Low Dataset: 367 images per species (2,569 total images)
- Average Dataset: 848 images per species (5,936 total images)
- High Dataset: 1,646 images per species (11,522 total images)

Each category was prepared in two input sizes, 256×256 and 384×384 pixels. A detailed comparison of the dataset size before and after augmentation for each taxonomic level is presented in Table 4. This comparison illustrates how augmentation was applied to balance the Average and High dataset categories, particularly for underrepresented classes, while the Low category used undersampling to match the target size.

TABLE 4
Comparison of Dataset Size Before and After Augmentation

Level	Category	Before Augmentation	Avg Dataset	High Dataset
Kelas	Aves	1563	2544	4938
	Actinopterygii	2272	1696	3292
	Mammalia	2101	1696	3292
Ordo	Passeriformes	1196	1696	3292
	Cypriniformes	2272	1696	3292
	Carnivora	2101	1696	3292
	Columbiformes	367	848	1646

Famili	Passeridae	1196	1696	3292
	Cyprinidae	2272	1696	3292
	Felidae	2101	1696	3292
	Columbidae	367	848	1646
Genus	Passer	1196	1696	3292
	Cyprinus	2101	1696	3292
	Felis	1646	848	1646
	Prionailurus	455	848	1646
	Columba	367	848	1646
Spesie s	Passer montanus	754	848	1646
	Felis catus	1646	848	1646
	Cyprinus rubrofuscus	1577	848	1646
	Cyprinus carpio	695	848	1646
	Prionailurus bengalensis	455	848	1646
	Passer domesticus	442	848	1646
	Columba Livia	367	848	1646
	Passer montanus	754	848	1646

B. Data Preprocessing

The preprocessing stage, including image normalization and augmentation (rotation, flipping, and zooming), contributed significantly to improving the model's generalization capability. This can be observed from the lower training and validation accuracy gap compared to models trained without augmentation, indicating reduced overfitting. Furthermore, resizing images to a uniform input size ensured consistent feature extraction across the dataset, enhancing model prediction stability during testing. These preprocessing strategies, therefore, played an essential role in achieving reliable classification performance across all taxonomic levels.

C. Model Training

The architecture was implemented using the Keras Functional API, which provides flexibility in building multi-output models compared to the Sequential API. This design was chosen to support the hierarchical nature of taxonomy classification, where multiple outputs (class, order, family, genus, species) must be predicted simultaneously. The network consisted of various Conv2D layers for feature extraction, followed by GlobalAveragePooling2D, fully connected Dense layers, and a Dropout layer for regularization. Five output branches were implemented, each corresponding to one taxonomic level: class, order, family, genus, and species. Six model configurations (M1-M6) were tested, varying the number of hidden layers (3 or 4) and the dropout type (0.2, 0.5, or adaptive). Adaptive dropout was implemented by linearly increasing the dropout rate from 0.1 to 0.5 over training steps to enhance regularization as learning progressed. Each model was trained on six dataset categories (D1-D6), resulting in 36 experimental runs (see Table 2 for model configurations).

Training was conducted for up to 50 epochs with a batch size of 32, using the Adam optimizer (learning rate = 0.001) and categorical cross-entropy loss for all outputs. Early stopping was applied with a patience of 10 and restore_best_weights=True to prevent overfitting. The evaluation considered the average validation accuracy and loss across all five taxonomy levels. Table 5 summarizes the average validation accuracy (%) for all datasets and model configuration combinations, while Table 6 shows the corresponding average validation loss (%).

TABLE 5
AVERAGE ACCURACY VALIDATION TABLE

Data ID	M1	M2	M3	M4	M5	M6
D1	86.56 %	86.37 %	86.13 %	87.62 %	84.10 %	87.38 %
D2	90.03 %	92.48 %	90.71 %	93.77 %	90.39 %	93.02 %
D3	93.59 %	**97.61 %**	95.53 %	97.42 %	95.50 %	96.62 %
D4	78.71 %	85.04 %	81.87 %	84.73 %	80.97 %	82.11 %
D5	89.48 %	93.28 %	89.11 %	92.57 %	88.95 %	93.56 %
D6	93.77 %	96.62 %	93.49 %	96.89 %	93.94 %	96.42 %

TABLE 6
AVERAGE LOSS VALIDATION TABLE

Data ID	M1	M2	M3	M4	M5	M6
D1	40.20 %	38.43 %	41.40 %	36.20 %	45.47 %	36.73 %
D2	31.70 %	27.36 %	27.52 %	22.79 %	28.63 %	21.74 %
D3	19.46 %	**8.57 %**	14.46 %	8.95 %	12.47 %	11.54 %
D4	55.62 %	46.25 %	50.07 %	41.34 %	48.51 %	53.56 %
D5	31.15 %	22.01 %	29.42 %	23.03 %	33.05 %	24.19 %
D6	19.45 %	13.87 %	19.17 %	9.26 %	17.36 %	9.48 %

Description (applies to Tables 3 and 4):
Dataset IDs (D1–D6):
- D1: 256×256 input, Low Dataset
- D2: 256×256 input, Average Dataset
- D3: 256×256 input, High Dataset
- D4: 384×384 input, Low Dataset
- D5: 384×384 input, Average Dataset
- D6: 384×384 input, High Dataset

Model IDs (M1-M6):
- M1: 0.2 Dropout, 3 Hidden Layers (32, 64, 128)
- M2: 0.2 Dropout, 4 Hidden Layers (32, 64, 128, 128)
- M3: 0.5 Dropout, 3 Hidden Layers (32, 64, 128)
- M4: 0.5 Dropout, 4 Hidden Layers (32, 64, 128, 128)
- M5: Adaptive Dropout, 3 Hidden Layers (32, 64, 128)
- M6: Adaptive Dropout, 4 Hidden Layers (32, 64, 128, 128)

From these results, D3-M2 (high dataset volume, 256×256 input, four hidden layers, 0.2 dropout) achieved the highest average validation accuracy (97.61%) and the lowest average loss (8.57%) among all configurations. This model was therefore selected as the final architecture to be integrated into the Android application. Stability testing through five retraining runs confirmed consistent performance with minimal variation, indicating robustness in the training process.

D. *Testing and Evaluation*

Model evaluation was conducted to assess the classification performance at each taxonomic level and to compare the deployment efficiency of different inference methods. The evaluation process consisted of four stages:

1. Confusion Matrix Analysis

The best-performing model (D3-M2) was evaluated using a validation dataset of 2634 images (329 images per category) to assess its classification performance across five taxonomy levels: class, order, family, genus, and species. The evaluation employed confusion matrices, where a deeper blue hue along the diagonal indicates more correct predictions. Figures 3-7 present the confusion matrices for each taxonomy level evaluated using the best-performing model (D3-M2) on the validation set. The diagonal cells represent correct predictions, with darker orange shades indicating higher counts.

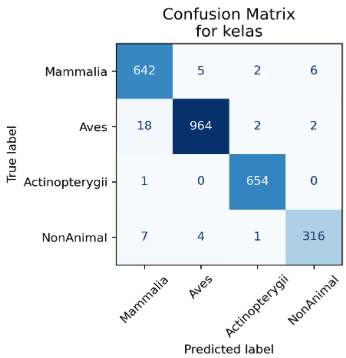

Figure 3: Confusion Matrix, Each Taxonomy Level Class

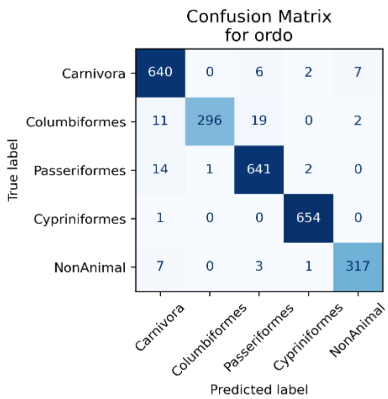

Figure 4: Confusion Matrix, Each Taxonomy Level Order

Figure 3 Class: The model achieved high accuracy for Mammalia, Aves, and Actinopterygii, with only minor misclassifications, such as a few Aves samples misclassified as Mammalia or Actinopterygii. Non-animal images were rarely confused with animal classes.

Figure 4 Order: Passeriformes and Carnivora were predicted accurately. Columbiformes occasionally overlapped with Passeriformes and NonAnimal, while Cypriniformes showed minor misclassifications with other orders.

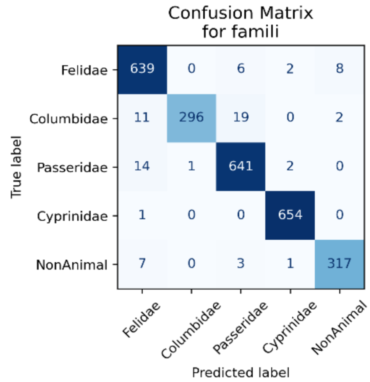

Figure 5: Confusion Matrix, Each Taxonomy Level Family

Figure 5: Passeridae and Felidae were classified correctly. Columbidae had more misclassifications, often confused with NonAnimal, likely due to background or overlapping visual features.

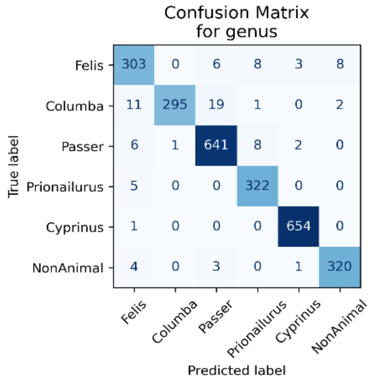

Figure 6: Confusion Matrix, Each Taxonomy Level, Genus

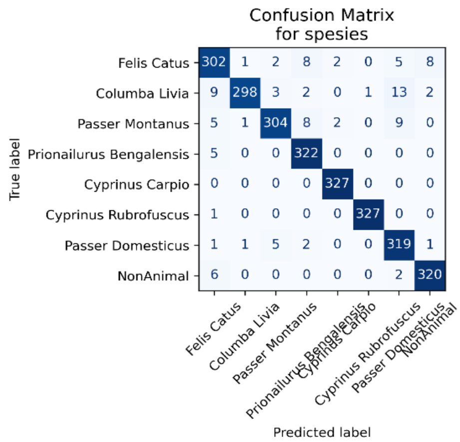

Figure 7: Confusion Matrix, Each Taxonomy Level Species

Figure 6 Genus: Most Felis and Passer samples were correctly identified. Misclassifications mainly occurred between Prionailurus and Felis, reflecting similar fur patterns and coloration.

Figure 7 Species: Species-level accuracy was the lowest. Due to subtle visual differences, notable confusion occurred between Passer domesticus vs. Passer montanus and Cyprinus rubrofuscus vs. Cyprinus carpio.

Broader taxonomic levels, such as class and order, achieved higher classification accuracy due to more distinct visual features. In contrast, finer levels, especially species, suffered lower performance because of subtle inter-class differences and shared morphological characteristics. Table 7 provides a quantitative overview of model performance, including average accuracy, precision, recall, and F1-score for the best-performing model (D3-M2).

TABLE 7
AVERAGE EVALUATION METRICS PER TAXONOMIC LEVEL

Level	Accuracy	Precision	Recall	F1-Score
Kelas	99.16%	98.09%	98.14%	98.11%
Ordo	98.84%	97.41%	96.37%	96.84%
Famili	98.80%	97.29%	96.28%	96.74%
Genus	98.87%	96.39%	95.94%	96.11%
Spesies	99.00%	96.07%	96.00%	95.99%
Avg	**98.93%**	**97.05%**	**96.55%**	**96.76%**

The model achieved high performance across all taxonomic levels, with an overall mean accuracy of 98.93%, precision of 97.05%, recall of 96.55%, and F1-score of 96.76% (Table 7). The highest accuracy was observed at the class level (99.16%) and order level (98.84%), while species-level classification also remained strong (99.00% accuracy), despite the inherent challenge of distinguishing visually similar species.

The strong performance can be attributed to several factors. First, the dataset included augmented images in the previously collected public dataset, providing diverse visual patterns. Second, the model architecture (CNN) effectively extracted hierarchical features at each taxonomic level. Third, the ample and high-quality training data allowed the model to learn distinguishing characteristics for each class, further improving generalization and yielding consistently high accuracy, precision, recall, and F1-score across all taxonomic levels.

Misclassifications at the species level were primarily due to visual similarity between particular species. For example, Felis catus and Prionailurus bengalensis have similar body shapes; Eurasian sparrows and house sparrows are nearly identical; rock pigeons resemble both sparrow species; and koi and common carp are very similar in appearance. Additionally, occasional misclassifications occurred between birds and cats, often due to similar poses and fur or feather coloration, making species-level differentiation more challenging.

2. Expert Validation

A biology teacher from SMA Dharma Loka, Ms. Martina Sihombing, S.Si, validated the animal classification system through an in-person interview on July 23, 2025. The purpose was to ensure that the application's output, particularly the taxonomy classification from class to species level, aligns with the standard biological classification system and to evaluate the application's feasibility as a learning tool. Key validation points and feedback were as follows:

- Application Output: The displayed taxonomy hierarchy (class, order, family, genus, species) was confirmed to follow standard biological systematics. The classification results for all seven animal species were validated. The expert emphasized the correct formatting of scientific names: genus capitalized, species lowercase, and italicized or underlined.
- Ease of Use: The application was considered user-friendly and intuitive. The help page was deemed informative and sufficient to guide users.
- Suggestions: The expert suggested enabling recognition even when only part of the animal is visible (e.g., head only) and ensuring consistent formatting of scientific names according to academic standards.

This validation confirmed that the Animal Classification System application meets scientific and practical requirements for animal classification and is suitable for broader educational use.

3. Real-World Condition Testing

As described in Section Method, Real-world testing evaluates the model's consistency and accuracy under different environmental conditions. The experiments varied two parameters: capture distance (30 cm, 60 cm, 100 cm) and lighting (bright and low light). Each combination was tested five times for the seven animal categories in the training scope.

TABLE 8
RESULTS OF BRIGHT LIGHT CONDITION ANALYSIS

Spesies	30cm	60cm	100cm	Avg
Felis catus	100 %	40 %	80 %	73.3%
Prionailurus bengalensis	0 %	80 %	80 %	53.3%
Columba Livia	100 %	60 %	0 %	53.3%
Passer montanus	40 %	20 %	0 %	20.0%
Passer domesticus	80 %	80 %	40 %	66.7%
Cyprinus carpio	60 %	100 %	100 %	86.7%
Cyprinus rubrofuscus	100 %	100 %	100 %	100.0%
Avg	**68.6%**	**68.6%**	**57.1%**	**64.8%**

Bright Light Results: Table 8 presents the accuracy for each distance. Average accuracy was highest at 30 cm and 60 cm (68.6%), dropping to 57.1% at 100 cm. Koi fish achieved perfect accuracy (100%) at all distances due to its distinct coloration. At the same time, Eurasian tree sparrows had the lowest accuracy due to visual similarity with other bird species and limited training data.

Low Light Results: Table 9 shows that accuracy peaked at 60 cm (74.3%), with reduced accuracy at 30 cm and 100 cm (48.6%). Cats achieved perfect accuracy across all distances, while Eurasian tree sparrows could not be identified in any low-light condition. Koi fish remained robust due to high-contrast colors.

TABLE 9
RESULTS OF LOW LIGHT CONDITION ANALYSIS

Spesies	30cm	60cm	100cm	Avg
Felis catus	100 %	100 %	100 %	100.0%
Prionailurus bengalensis	0 %	100 %	40 %	46.7%
Columba Livia	100 %	40 %	0 %	46.7%
Passer montanus	0 %	0 %	0 %	0.0%
Passer domesticus	20 %	80 %	40 %	46.7%
Cyprinus carpio	20 %	100 %	100 %	73.3%
Cyprinus rubrofuscus	100 %	100 %	60 %	86.7%
Avg	**48.6%**	**74.3%**	**48.6%**	**57.1%**

Across both lighting conditions, bright light yielded higher overall accuracy (64.8%) compared to low light (57.1%). Performance was most stable at 60 cm, and species with distinctive visual features (e.g., koi fish) were more resilient to environmental variations.

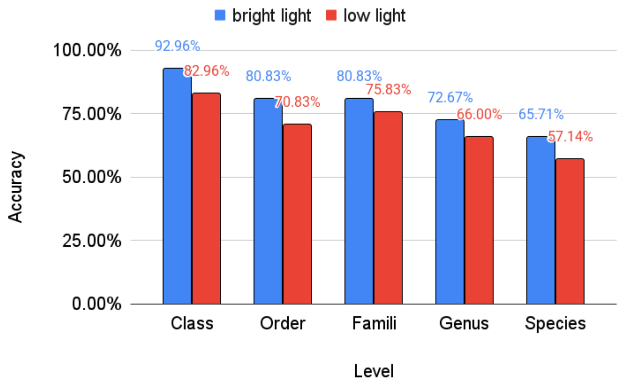

Figure 8: Each Level Taxonomy Real World Condition Testing

Figure 8 shows classification accuracy at different taxonomic levels under bright and low light. Accuracy is highest at the Class level (92.96% bright, 82.96% low) and lowest at Species (65.71% bright, 57.14% low). The gap between lighting conditions grows at finer levels, indicating sensitivity to the environment. Lower species accuracy is due to visual similarities (e.g., *Passer montanus* vs. *Passer domesticus*), limited data (*Columba livia*), and misclassification of species like *Cyprinus carpio* and *Prionailurus bengalensis* as *Felis catus* because of morphological resemblance and varied *Felis catus* data. Similar poses and plumage also confuse.

4. Application Performance Testing

To assess system efficiency, three deployment methods were compared:

- Edge AI: TFLite inference on-device
- Cloud AI (FastAPI): Hosted on Hugging Face Spaces
- Cloud AI (Gradio + Vercel): Frontend in Gradio, backend in Express.js (Vercel)

TABLE 10
RESULTS OF APPLICATION PERFORMANCE TESTING

Deploy Type	No. Experiment	Avg latensi (s)	Avg CPU (%)	Avg RAM (GB)
Edge AI	10	0.17 s	13.79 %	1.33 GB
Cloud AI	10	1.74 s	4.13 %	1.12 GB
Cloud AI HF+Vercel	10	1.90 s	4.00 %	1.11 GB

Results (Table 10) show that Edge AI achieved the lowest average latency (0.17 s) due to local processing without network delay. CPU usage was highest for Edge AI (13.79%), while Cloud AI methods had significantly lower CPU usage (~4%) as computation occurred on the server. RAM usage was similar across all methods. Based on these findings, Edge AI was selected as the optimal deployment method for its real-time response (< 0.2s, meeting human-instant interaction criteria) and offline capability, making it suitable for use in remote or low-connectivity environments.

IV. CONCLUSION

This research successfully developed an Android-based animal taxonomy classification system using a Convolutional Neural Network (CNN) with a multi-output multi-class classification approach. The system can identify seven animal species across five taxonomic levels, class, order, family, genus, and species, with an average accuracy of 90.15%. The best-performing model (D3-M2), trained on the high-volume dataset (256×256 input size, four hidden layers, 0.2 dropout), achieved the highest validation accuracy (97.61%) and lowest validation loss (8.57%), demonstrating strong and consistent performance during stability testing. Evaluation results revealed that broader taxonomic levels (class and order) achieved higher and more balanced precision and recall, while finer-grained levels, such as species, presented more classification challenges due to subtle inter-class differences. The lowest F1-score (57.83%) occurred at the species level, indicating the need for further model optimization and additional diverse training data for visually similar categories.

Real-world condition testing showed that the system performs optimally under bright lighting and a capture distance of 60 cm, with particular species (e.g., koi fish) proving more robust to environmental variations. Performance testing across deployment methods confirmed that Edge AI offers the fastest inference time (0.17 seconds) and offline capability, making it the most suitable for real-time field use compared to Cloud AI approaches. The application has been validated by a biology expert, confirming its alignment with scientific taxonomic standards and its usability as an educational tool. This positions the system as an effective means of improving public knowledge and awareness regarding biodiversity conservation, especially among students and the general public.

In conclusion, this study demonstrates that when optimized and deployed via Edge AI, CNN-based multi-output multi-class classification can provide accurate, efficient, and practical animal taxonomy identification on mobile devices. Expanding the dataset to include a more balanced and diverse range of species with greater variation in environmental conditions, such as different angles, backgrounds, and lighting intensities, is recommended for future work. Incorporating more complex model architectures like ResNet or object detection frameworks like YOLO could enhance classification accuracy, particularly for fine-grained species-level recognition. Moreover, continued validation in diverse real-world environments will be essential to improve the model's generalization and practical utility.

REFERENCES

[1] R. R. Andriani, R. Sitorus, S. A. P. Zai, and Y. S. Pasaribu, “Penggunaan Algoritma CNN untuk Mengidentifikasi Jenis Anjing Menggunakan Metode Supervised Learning,” *Mutiara: Jurnal Penelitian dan Karya Ilmiah*, vol. 1, no. -, pp. 394–402, 2023. [Online]. Available: https://doi.org/10.59059/mutiara.v1i6.741

[2] Antara News, “Masyarakat Sering Keliru Mengidentifikasi Satwa Liar di Sekitar Pemukiman,” *Antaranews.com*, 2022.

[3] K. Antonio and E. Hartati, “Klasifikasi Spesies Ular Menggunakan Metode Convolutional Neural Network,” *Jurnal Ilmu Komputer dan Informatika*, vol. 3, no. -, pp. 357–363, 2023.

[4] AWS Amazon, "What is Python?" *AWS Amazon*. [Online]. Available: https://aws.amazon.com/id/what-is/python/

[5] S. K. Card, T. P. Moran, and A. Newell, *The Psychology of Human-Computer Interaction*, 1st ed. Boca Raton, FL, USA: CRC Press, 1983.

[6] Codecademy, "Deep Learning Workflow," *Codecademy*. [Online]. Available: https://www.codecademy.com/article/deep-learning-workflow

[7] Dicoding Intern, “Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya,” *Dicoding*, May 31, 2023. [Online]. Available: https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/

[8] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith, "Fine-tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping," *arXiv*, 2020. [Online]. Available: http://arxiv.org/pdf/2002.06305

[9] R. A. Fitriansyah and Saparudin, “Penerapan Ensemble Stacking untuk Klasifikasi Multi Kelas,” *Prosiding Annual Research Seminar*, vol. 2, pp. 240–243, 2016. [Online]. Available: https://www.neliti.com/publications/171917/penerapan-ensemble-stacking-untuk-klasifikasi-multi-kelas

[10] GeeksforGeeks, "Understanding the Confusion Matrix in Machine Learning," *GeeksforGeeks*. [Online]. Available: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/#what-is-a-confusion-matrix

[11] Hugging Face, "Spaces," *Hugging Face*. [Online]. Available: https://huggingface.co/docs/hub/spaces

[12] IBM, "Apa itu Convolutional Neural Network?"*IBM*. [Online]. Available: https://www.ibm.com/id-id/topics/convolutional-neural-networks

[13] IBM, "What is Edge AI?"*IBM*. [Online]. Available: https://www.ibm.com/think/topics/edge-ai

[14] *Ilmu Pengetahuan Alam*, Pusat Kurikulum dan Perbukuan; Badan Penelitian dan Pengembangan dan Perbukuan; Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, 2021. [Online]. Available: https://static.buku.kemdikbud.go.id/content/pdf/bukuteks/kurikulum21/IPA-BS-KLS%20VII.pdf

[15] S. Jain, "Convolutional Neural Network (CNN) in Machine Learning," *GeeksforGeeks*, Mar. 13, 2024. [Online]. Available: https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/

[16] A. Joly, L. Wehenkel, and P. Geurts, "Gradient Tree Boosting with Random Output Projections for Multi-label Classification and Multi-output Regression," *arXiv*, pp. 1–40, 2019. [Online]. Available: https://arxiv.org/pdf/1905.07558

[17] KKP RI, “Mengenal Perbedaan Ikan Mas dan Ikan Koi,” *Kementerian Kelautan dan Perikanan Republik Indonesia*, 2021.

[18] Kompas, “Perbedaan Burung Gereja Lokal dan Eurasia,” *Kompas.com*, 2020.

[19] Mongabay Indonesia, “Mengenal Kucing Kuwuk, Satwa Liar yang Sering Disangka Kucing Rumahan,” *Mongabay.co.id*, 2018.

[20] P. Moreno-Muñoz, A. Artés-Rodríguez, and M. A. Álvarez, "Heterogeneous Multi-output Gaussian Process Prediction," in *Proc. 32nd Conf. Neural Information Processing Systems (NeurIPS)*, 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/165a59f7cf3b5c4396ba65953d679f17-Paper.pdf

[21] M. Mosbach, M. Andriushchenko, and D. Klakow, "On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2021. [Online]. Available: https://openreview.net/forum?id=nzpLWnVAyah

[22] Nazirman, “Laporan Proyek Akhir Aplikasi Interaktif Pembelajaran Pengenalan Hewan dengan Penerapan Convolutional Neural Network (CNN) (Studi Kasus: TK Al Husniyah),” *Politeknik Caltex Riau*, 2023.

[23] R. I. Ndun, “Mendeteksi Jenis Burung Berdasarkan Gambar Menggunakan Deep Learning,” *Universitas Dinamika*, 2020.

[24] P. Nugraha, A. Komarudin, and E. Ramadhan, “Deteksi Objek dan Jenis Burung Menggunakan Convolutional Neural Network dengan Arsitektur Inception ResNet-V2,” *INFOTECH Journal*, vol. 8, no. -, pp. 47–55, 2022. [Online]. Available: https://doi.org/10.31949/infotech.v8i2.2889

[25] D. P. Pamungkas and M. F. Amrulloh, “Analisis Hasil Klasifikasi Penyakit Daun Bawang Merah Menggunakan CNN Arsitektur Exception,” *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, vol. 10, 2025.

[26] I. Y. Pangestu and S. R. Ramadhani, “Perancangan Sistem Deteksi Penyakit Kulit pada Kucing Menggunakan Deep Learning Berbasis Android,” *TEKNIKA*, vol. 12, pp. 173–182, 2023. [Online]. Available: https://doi.org/10.34148/teknika.v12i3.673

[27] J. W. G. Putra, *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*, 1.4 ed., 2020. [Online]. Available: https://wiragotama.github.io/resources/ebook/intro-to-ml-secured.pdf

[28] Rina, “Memahami Confusion Matrix: Accuracy, Precision, Recall, Specificity, dan F1-score untuk Evaluasi Model Klasifikasi,” *Medium*, Jun. 12, 2023. [Online]. Available: https://esairina.medium.com/memahami-confusion-matrix-accuracy-precision-recall-specificity-dan-f1-score-610d4f0db7cf

[29] K. Sampigethaya, M. Li, and R. Poovendran, "Real-time Computing Systems and Applications for Intelligent Transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. -, pp. 2744–2755, 2017. [Online]. Available: https://doi.org/10.1109/TITS.2017.2654242

[30] L. Sari, “Klasifikasi Multikelas dengan Menggunakan Metode Multicategory Kernel Distance Weighted Discrimination pada Data Keuangan Sektor Publik di Indonesia,” 2018. [Online]. Available:

https://repository.unpad.ac.id/bitstreams/01b164fe-299b-4c1b-b94f-55873f009d63/download

[31] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, pp. 1–48, 2019. [Online]. Available: http://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0#citeas

[32] Vercel, "Get Started with Vercel," *Vercel*, Dec. 18, 2024. [Online]. Available: https://vercel.com/docs/getting-started-with-vercel

[33] S. R. Widjajanti, *Seribu Pena Biologi SMA untuk Kelas X Rangkuman Materi Contoh Soal dan Pembahasan Soal-soal Evaluasi*. Jakarta, Indonesia: Erlangga, 2005.

[34] D. C. Youvan, "AI at the Edge vs. AI in the Cloud: A Comparative Analysis of High-end and Edge AI Systems," *ResearchGate*, Jun. 13, 2024. [Online]. Available: https://www.researchgate.net/publication/381402818_AI_at_the_Edge_vs_AI_in_the_Cloud_A_Comparative_Analysis_of_High-End_and_Edge_AI_Systems

[35] M. S. Aryanta, C. A. Sari, and E. H. Rachmawanto, "A Banana Disease Detection Using MobileNetV2 Model Based on Adam Optimizer," Journal of Applied Informatics and Computing (JAIC), vol. 9, no. 4, pp. 1207–1218, 2025. [Online]. Available: https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/10183

[36] A. Angdresey, A. F. Ramadhan, and S. Hadiyoso, "Detection of Pests and Plant Diseases Using EfficientNet-B0 Model," Journal of Applied Informatics and Computing (JAIC), vol. 9, no. 4, pp. 1137–1145, 2025. [Online]. Available: https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/9870

[37] M. Fauzan, D. S. Rahardjo, and R. Andriani, "Identification of Rice Plant Diseases Using Convolutional Neural Networks," Journal of Applied Informatics and Computing (JAIC), vol. 9, no. 4, pp. 1770–1777, 2025. [Online]. Available: https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/9373