

**IJECBE**

International Journal of Electrical, Computer and Biomedical Engineering

*IJECBE* (2025), 3, 2, 433–457  
Received (22 April 2025) / Revised (22 May 2025)  
Accepted (4 June 2025) / Published (30 June 2025)  
<https://doi.org/10.62146/ijecbe.v3i2.114>  
<https://ijecbe.ui.ac.id>  
ISSN 3026-5258

## RESEARCH ARTICLE

# Development Of Detection Model For Skin Diseases In Pets Using Image Processing And Deep Learning Techniques

Salma Dewi Taufiqoh\* and Prima Dewi Purnamasari

Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia

\*Corresponding author. Email: [salma.dete@gmail.com](mailto:salma.dete@gmail.com)

### Abstract

Early detection of skin diseases in pets is essential but often hindered by the cost and complexity of clinical diagnosis. This study introduces a deep learning–based system for identifying three common pet skin diseases—Ringworm, Scabies, and Ear mite—using images captured with mobile phone cameras. The system integrates classical image preprocessing techniques, including Contrast Limited Adaptive Histogram Equalization (CLAHE) and HueSaturation–Value (HSV) segmentation, with a custom convolutional neural network (CNN) designed for disease-specific classification tasks. Two separate models were developed: a multi-class CNN model for classifying Ringworm, Scabies, and Undetected conditions, which achieved a test accuracy of 83% and an average accuracy of 76.9% across five-fold cross-validation; and a binary CNN model for classifying Ear mite versus Undetected, which achieved 100% accuracy on an unseen test set and an average accuracy of 86.8% across five-fold cross-validation. Compared to transfer learning models such as ResNet-50 and VGG16, the proposed CNN models demonstrated superior performance under limited-data conditions (72 images total), emphasizing the advantage of domain-specific model design and preprocessing. These findings suggest that disease-adapted CNN architectures, combined with targeted preprocessing, can support accurate and accessible veterinary screening using mobile devices. Future work will focus on expanding the dataset and deploying the model in a real-time mobile diagnostic application.

**Keywords:** Pets, Deep Learning, Image processing, Convolutional Neural Network, Skin Disease Detection, Classification, Accuracy, F1-score, Precision, Recall

## 1. Introduction

Pets, especially dogs and cats, play a vital role in human life—not only as companions but also as family members. However, they can also act as vectors for zoonotic diseases, with more than 70 human infections linked to dogs alone [1]. Among the most common health issues affecting pets are skin diseases, which may arise from allergies, infections, parasites, or autoimmune disorders [2]. Some conditions, such as ringworm, are zoonotic and can be transmitted to humans [3]. Therefore, early detection and appropriate treatment are essential to prevent transmission and improve pet welfare.

Diagnosing skin diseases in pets often requires specialized veterinary expertise and incurs significant time and cost. Deep learning, particularly convolutional neural networks (CNNs), has demonstrated strong potential in automating medical image classification tasks, including dermatological diagnosis [4][5]. Models such as InceptionV3, MobileNetV2, and DenseNet-201 have shown promising results in this domain [5]. However, many prior studies rely on high-end imaging equipment, such as multispectral cameras, which limits practical deployment due to accessibility constraints.

This study addresses that gap by proposing an image-based CNN model tailored for classifying pet skin diseases using images captured from mobile phone cameras. The focus is on three commonly encountered and clinically distinguishable conditions—Ringworm, Scabies, and Earmite—selected in consultation with veterinarians. By combining conventional image preprocessing techniques with a lightweight CNN architecture, the model aims to deliver a costeffective, accessible, and scalable diagnostic tool for mobile-based veterinary screening.

Although the selected skin disease categories may appear visually distinguishable, realworld mobile photography introduces challenges such as variation in lighting, focus, background, and lesion presentation. These conditions complicate the classification task beyond what is typically assumed in controlled datasets. In traditional machine learning approaches, effective performance often depends on manual feature engineering and domain-specific preprocessing, which may limit adaptability and robustness across diverse inputs.

In contrast, convolutional neural networks (CNNs) offer an end-to-end learning framework capable of automatically extracting discriminative features from raw image data. The use of a lightweight CNN in this study balances performance and efficiency, making it suitable for deployment on mobile devices where reliability under varied conditions is essential. Furthermore, model performance was validated not only through conventional train/test splits but also via five-fold cross-validation to evaluate robustness under small-sample conditions. Thus, rather than complexity of the disease itself, it is the variability of real-world inputs and the goal of practical, field-ready deployment that justifies the use of deep learning in this context.

## 2. Related Work

Recent advances in deep learning and image processing have significantly improved the detection of skin diseases in pets. Hwang *et al.* [5] developed a classification model using multispectral and normal images to identify bacterial dermatosis, fungal infections, and allergic dermatitis in dogs. They implemented four CNN architec-

tures—InceptionNet, ResNet, DenseNet, and MobileNet—and proposed a consensus model that combined the best classifiers from both image types. This ensemble achieved an average accuracy of 87.6%, outperforming single-model approaches.

Thoutam et al. [6] extended this work by evaluating InceptionV3 and MobileNetV2 on the same dataset. Their results highlighted the importance of architecture selection, with InceptionV3 achieving 98% accuracy and a 0.996 F1-score. Meanwhile, Sengupta et al. [7] explored classical image processing, optimizing edge detection with Ant Colony Optimization (ACO) and combining it with threshold-based segmentation. ACO-Canny emerged as the most effective for lesion detection, showing the continued relevance of non-deep learning approaches.

In a more application-driven context, Dissanayaka et al. [36] introduced the "HAPPY PET" mobile app, combining SVM-based image classification, deep learning symptom verification, behavioral analysis, and NLP-based treatment recommendations. Their integrated system achieved up to 98.79% accuracy in specific modules, demonstrating a multi-modal approach to pet diagnostics.

While these studies contribute valuable insights, several limitations remain. Most prior work focuses on broad disease categories and relies on high-end imaging tools, such as multispectral cameras, which are impractical for everyday use. Additionally, many models are either deep learning-based or image processing-based, rather than combining both.

This study addresses those gaps by targeting three specific and common pet skin diseases—Ringworm, Scabies, and Earmite—using images captured via mobile phones. It integrates classical preprocessing techniques (CLAHE, Gaussian filtering, HSV segmentation) with a custom CNN model, enabling disease-focused classification in realistic conditions. Despite the use of a small dataset (72 images), the model achieves high performance through optimal hyperparameter tuning and data augmentation, demonstrating the viability of CNNs in constrained-data scenarios.

### 3. Methodology

This study proposes a system to classify three common pet skin diseases—Ringworm, Scabies, and Earmite—based on images captured using smartphone cameras. These diseases present with visually distinct symptoms that can be identified through careful inspection of skin texture, color, and lesion shape. Example images from each disease class are shown in Figure 1.



**Figure 1.** Example pet skin disease images: (a) Ringworm, (b) Scabies c) Earmite

The system integrates classical image preprocessing techniques with a custom convolutional neural network (CNN) to support accurate and accessible disease detection under real-world constraints. The complete workflow, from image acquisition to classification, is illustrated in Figure 2 (System Block Diagram). The methodological pipeline comprises dataset collection, image preprocessing, data augmentation, CNN architecture design, and model evaluation.

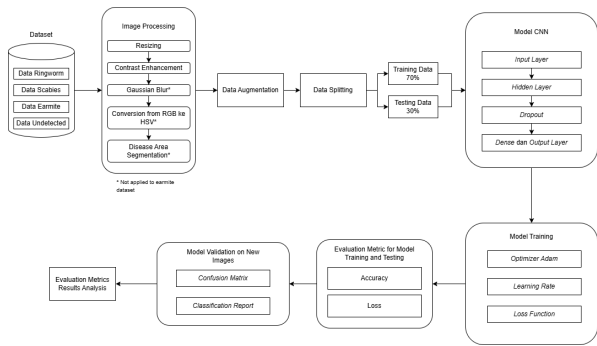


Figure 2. System Block Diagram

3.1 Dataset Collection and Labeling

The dataset used in this study comprised a total of 72 images, evenly distributed across four classes: Ringworm, Scabies, Earmite, and Undetected. The images were obtained from a combination of publicly available sources on the internet and veterinary clinical contributions.

For the Ringworm class, the majority of images were collected from open-access veterinary and pet health websites, such as Colorado Animal Rescue, Purina Australia, The Spruce Pets, Cat World, and VOKRA [8] [9] [10] [11][12]. These sources provided clear examples of typical lesion patterns in cats, which were useful for training purposes. Due to the manual nature of image collection over time, not all URLs were archived, but all publicly sourced images were carefully reviewed for clinical relevance and label accuracy by a certified veterinarian.

For the Scabies class, several images were obtained from veterinary articles and case reports [13][14], while the remainder—along with all Earmite and Undetected class images—were provided directly by veterinary clinics with appropriate permissions.

The curated dataset reflects realistic visual conditions typical of mobile veterinary screening and has been thoroughly verified for clinical accuracy. While this initial dataset size is limited, it serves as a strong foundation for model development.

3.2 Image Processing

Image preprocessing was applied to enhance the quality and consistency of the dataset before feeding it into the CNN. Given that the images originated from heterogeneous sources, standardization was essential to reduce noise, improve contrast, and highlight diseased regions. The steps are described below.

### 3.2.1 Image Resizing

All input images were resized to  $224 \times 224$  pixels to ensure consistency across the dataset and compatibility with the CNN input layer. This resolution was selected to balance detail preservation with computational efficiency.

### 3.2.2 Contrast Enhancement (CLAHE)

To improve local contrast and highlight lesion features, Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied. This method enhances the visibility of subtle skin texture differences, especially under uneven lighting conditions. In this study, CLAHE was implemented using a clip limit of 2.0 and a tile grid size of (8, 8).

### 3.2.3 Filtering (Gaussian Blur)

To suppress high-frequency noise and minor texture inconsistencies, a Gaussian blur was applied to the contrast-enhanced images. This smoothing operation helps the CNN focus on dominant features by reducing irrelevant visual details. The filter used had a kernel size of (5, 5) and a standard deviation of 0.

### 3.2.4 Color Conversion (RGB to HSV)

The RGB color space was converted to HSV (Hue, Saturation, Value) to facilitate more effective color-based segmentation. HSV separates color information (hue) from intensity (value), making it easier to isolate skin tones and lesion regions under varying lighting conditions.

### 3.2.5 Disease Area Segmentation

Disease area segmentation was performed using HSV-based thresholding. Specific hue, saturation, and value ranges were defined to isolate infected skin areas while filtering out healthy regions and background noise. This step allows the CNN to concentrate on relevant areas during training, enhancing classification focus and reducing false feature learning.

## 3.3 Data Augmentation

To improve the model's generalization capability given the limited dataset, data augmentation techniques were applied to the training set using the ImageDataGenerator class from Keras. The augmentation process included several transformations to introduce variability in orientation, scale, and position.

All training images were rescaled to normalize pixel values between 0 and 1. Random rotations up to 20 degrees were used to simulate different viewing angles. Horizontal and vertical shifts were applied within a 20% range to mimic positional variance. Shearing and zooming, both up to 20%, were introduced to distort and scale the images, enhancing robustness to geometric changes. Horizontal flipping was used to simulate left-right symmetry in skin conditions.

The final augmented dataset maintained a 70/30 split between training and testing via the built-in validation\_split parameter. These augmentation techniques enabled the CNN to learn more robust features and reduce overfitting during training.

### 3.4 Data Partitioning

The original dataset of 72 images was divided into training, testing, and validation subsets. Using the ImageDataGenerator's validation split functionality, 70% of the images (50 samples) were allocated for training and 30% (22 samples) for testing. In addition, six previously unseen and unprocessed images were reserved as an independent validation set to simulate real-world deployment conditions.

### 3.5 CNN Architecture Design

The classification model use a custom convolutional neural network (CNN) designed to extract hierarchical features from pet skin images and map them to corresponding disease labels. The architecture consisted of three main blocks: feature extraction, dense classification, and task-specific output layers. The overall structure is illustrated in Figure 3

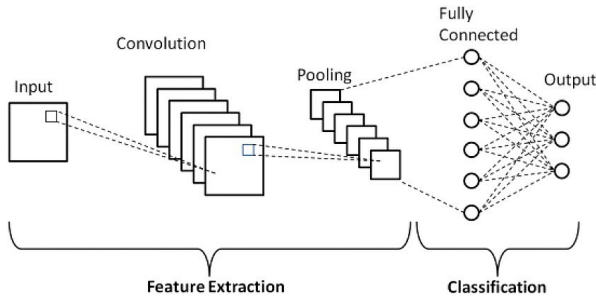


Figure 3. CNN Architecture

The feature extraction stage comprised three convolutional layers with increasing filter sizes (32, 64, and 128) and  $3 \times 3$  kernels. The ReLU activation function used in each convolutional layer is defined in Equation 1

$$f(x) = \max(0, x) \quad (1)$$

And then feature maps were downsampled using a  $2 \times 2$  max pooling as defined in Equation 2

$$P_{ij} = \max_{m,n \in [0,1]} Z_{2i+m, 2j+n} \quad (2)$$

These layers enabled the network to progressively capture texture, edge, and shape-related patterns associated with skin lesions.

The resulting feature maps were flattened and passed to a fully connected layer with 512 units and ReLU activation. To improve generalization and reduce overfitting, a dropout layer with a rate of 0.5 was added between the dense and output layers.

The output layer was adjusted based on the classification task:

- For multi-class classification (Ringworm, Scabies, Undetected), the model uses a dense output layer with three neurons and Softmax activation as defined in Equation 3

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (3)$$

This maps the output logits to a probability distribution over  $K$  classes.

- For binary classification (Earmite vs Undetected), the model employs a single neuron with Sigmoid activation as defined in Equation 4

$$\sigma(z_i) = \frac{1}{1 + e^{-x}} \quad (4)$$

Which produces a single probability score between 0 and 1, indicating the likelihood of the positive class.

This architecture was selected to balance computational efficiency and feature discrimination, making it well-suited for limited-data scenarios and potential deployment on resource-constrained devices.

### 3.6 Model Adaptation for Region-Specific Disease Detection

To improve classification accuracy and interpretability, the model design was adapted to account for anatomical distinctions between skin diseases. Rather than training a single model to classify all conditions, two separate models were developed based on disease location and visual characteristics.

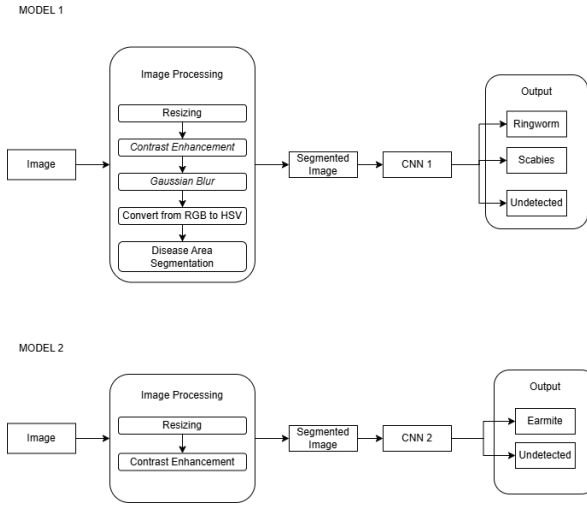
Model 1 addresses multi-class classification of Ringworm, Scabies, and Undetected cases. These conditions can appear on various parts of a pet's body (e.g., face, limbs, back) and often exhibit diverse lesion patterns. To handle this variability, Model 1 incorporates a more advanced preprocessing pipeline, including HSV-based segmentation, to isolate disease-relevant regions and reduce background interference.

Model 2 focuses exclusively on Earmite detection, which is typically confined to the ear canal and exhibits more localized, consistent patterns. Given its anatomical specificity and relatively uniform presentation, a simplified preprocessing approach—contrast enhancement via CLAHE without segmentation—was sufficient. The model then performs binary classification between Earmite and Undetected classes.

This two-model strategy enables each network to specialize in learning features tailored to its disease scope and anatomical context. It reduces task complexity, enhances detection robustness, and aligns model behavior with real-world clinical patterns. The overall workflow for both models, including preprocessing variations and classification objectives, is illustrated in Figure 4.

### 3.7 Training and Optimization

Both CNN models were trained using the Adam optimizer, selected for its adaptive learning rate and efficient convergence properties. The learning rate was set to 0.001, with a batch size of 8, and training proceeded for a maximum of 15 epochs. Early stopping was applied based on validation accuracy to prevent overfitting, particularly given the limited dataset size.



**Figure 4.** Block Diagram of Model 1 and Model 2

Loss functions were chosen according to the classification objective. For multi-class classification (Model 1), categorical crossentropy was used as the loss function, as shown in Equation 5.

$$L_{categorical} = - \sum_{i=1}^c y_i \log(\hat{y}_i) \quad (5)$$

where  $C$  is the number of classes,  $y_i$  the true label, and  $\hat{y}_i$  the predicted probability. For binary classification in Model 2, the optimization was performed using binary crossentropy, as defined in Equation 6.

$$L_{categorical} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (6)$$

These loss functions guided the backpropagation process to minimize prediction errors, allowing both models to effectively update their weights in accordance with their respective classification tasks.

### 3.8 Evaluation Metrics

To evaluate the model's performance, both training-phase metrics (accuracy and loss) and post-training validation metrics (confusion matrix and classification report) were used. These metrics collectively provided a comprehensive view of the model's ability to learn from the data and generalize to unseen inputs.

#### 3.8.1 Accuracy and Loss Monitoring

Model performance was assessed using a combination of training-phase and post-training evaluation metrics to ensure both learning effectiveness and generalization capability.



During training, accuracy and loss were monitored per epoch for both training and validation sets. Accuracy measured the proportion of correct predictions, while loss provided insight into the confidence and convergence of the model. A widening gap between training and validation accuracy or loss was used as an indicator of potential overfitting.

Post-training evaluation was conducted using a confusion matrix and the associated precision, recall, and F1-score for each class. These metrics are particularly important in medical and veterinary domains, where misclassification of positive cases (e.g., undetected diseases) can have greater implications than overall accuracy alone.

- Precision quantifies the proportion of true positives among all positive predictions.

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

- Recall measures the model's ability to detect actual positive cases.

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

- F1-score, the harmonic mean of precision and recall, balances both metrics and is especially useful in imbalanced data scenarios.

$$F1 \text{ Score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (9)$$

A classification report summarizing these metrics was generated for each class to provide a more detailed understanding of model performance, particularly for underrepresented conditions such as Scabies or Earmite. These per-class evaluations helped identify model bias and ensure clinical fairness in predictions.

## 4. Experimental Results and Analysis

This section presents the experimental evaluation of the proposed pet skin disease detection system. The results are structured to highlight key components including training environment, model configurations, and evaluation outcomes.

### 4.1 Experimental Setup

All experiments were conducted on a personal laptop without GPU acceleration. The system was equipped with an Intel Core i7-1065G7 processor and 8 GB of RAM, running on Windows 11. Development was carried out in Python 3.10.13, with an Anaconda virtual environment used to manage packages and dependencies. Model development and training were implemented using TensorFlow 2.16.1 and Keras 3.3.3. Image preprocessing was performed using OpenCV 4.9.0.80, and Scikit-learn 1.5.0 was used for evaluation metrics and data handling.

Training was performed using a batch size of 8 and a maximum of 15 epochs, with early stopping applied based on validation performance to prevent overfitting. The learning rate was empirically set to 0.001 using the Adam optimizer.

Given the compact architecture and small dataset, training remained computationally efficient without GPU support. To improve clarity and reproducibility, the complete training configuration is summarized in Table 1.

**Table 1.** Summary of model training configuration

Component	Value
Programming Language	Python 3.10.13
Libraries / Frameworks	<ul style="list-style-type: none"><li>• TensorFlow 2.16.1</li><li>• Keras 3.3.3</li><li>• OpenCV 4.9.0.80</li><li>• Scikit-learn 1.5.0</li></ul>
Input Image Size	224 × 224 pixels
Optimizer	Adam and RMSprop
Seed	42
Learning Rate	0.0001, 0.001, and 0.01
Batch Size	8
Maximum Epochs	15 (with early stopping)
Loss Function	Categorical Crossentropy (Model 1) Binary Crossentropy (Model 2)
Dropout Rate	0.3 and 0.5
Data Augmentation	Rotation, Shift, Shear, Zoom, Flip, Rescale (via ImageDataGenerator)
Preprocessing	CLAHE (Model 1 & 2) HSV Segmentation (Model 1 only)

**4.2 Experimental Scenarios**

To comprehensively evaluate the system, two experimental tracks were conducted, each aligned with one of the proposed models.

- Model 1 focused on multi-class classification (Ringworm, Scabies, Undetected), evaluating segmentation techniques, hyperparameters, and data augmentation.
- Model 2 focused on binary classification (Earmite vs. Undetected), using simpler preprocessing due to its consistent visual pattern.

Both models were also compared with transfer learning baselines (ResNet-50 and VGG16) using the same dataset. Performance was assessed using accuracy, precision, recall, and F1-score across training, test, and validation sets. Details and diagrams for each model are presented in Sections 4.3 and 4.4.

**4.3 Model 1: Multi-Class Disease Classification**

Model 1 was developed to classify skin images into three categories: Ringworm, Scabies, and Undetected. Due to the visual diversity of lesions across body regions, the

model incorporated HSV-based segmentation to isolate affected areas, along with data augmentation and hyperparameter tuning. A series of experiments were conducted to optimize preprocessing and model configuration. The block diagram for Model 1 is shown in Figure 5.

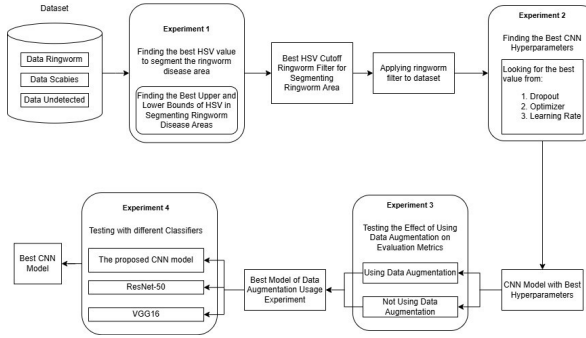


Figure 5. Block Diagram of Experimental Scenario Model 1

Throughout the experiments, four key aspects were explored: the optimal range for HSV segmentation, tuning of hyperparameters such as dropout rate and learning rate, the impact of data augmentation, and comparison against well-established transfer learning models.

#### 4.3.1 Experiment 1 HSV Range Tuning

This experiment aimed to identify the optimal HSV threshold range for lesion segmentation, particularly for Ringworm, which presents with distinct circular patterns. Multiple combinations of hue, saturation, and value ranges were tested to enhance lesion visibility while minimizing background interference. The best performance was achieved with the following HSV range:

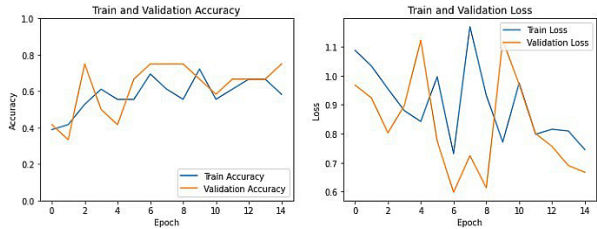
- Hue: 0–10 and 170–175
- Saturation: 50–210
- Value: 50–255.

This configuration produced the most consistent and visually accurate segmentation across images in the Ringworm class, especially in terms of highlighting circular lesion patterns with strong contrast and minimal background interference. A complete list of tested HSV values and visual outcomes is provided in Appendix A: Table 2.

#### 4.3.2 Experiment 2 Hyperparameter Optimization

Once the segmentation process was finalized, various combinations of hyperparameters were tested to determine the optimal configuration for training the CNN. Among the parameters examined were dropout rates (0.3 and 0.5), optimizer types (Adam and RMSprop), and learning rates (0.01, 0.001, and 0.0001). The best results were consistently achieved when using the Adam optimizer with a learning rate of 0.001 and a dropout rate of 0.5.

This configuration led to a test accuracy of 83%, an F1-score of 82%, a precision of 89%, and a recall of 83%. These metrics indicate that the model was both accurate and balanced in detecting the three skin conditions. Detailed experimental results for all hyperparameter combinations, including loss values and confusion matrices, are provided in Appendix A Tables 3–4 and Figures 19–22.



**Figure 6.** Training and Validation Accuracy and Loss for Experiment 2 using Best Hyperparameter

To better understand the model’s learning behavior, Figure 6 illustrates the training and validation accuracy and loss curves across 15 epochs. As shown, the model’s validation accuracy fluctuates but remains mostly aligned with training accuracy throughout the training process,

suggesting stable generalization. Although the curves show some non-monotonic trends—especially between epochs 5 to 10—the model avoids severe divergence, and both accuracy curves converge toward similar levels by the final epochs.

In terms of loss, the validation loss exhibits a decreasing trend overall, with occasional spikes. However, these fluctuations are mirrored in the training loss, indicating that the model is learning from the data rather than overfitting to the training set. The lack of a large performance gap between training and validation metrics suggests that the model maintained relatively good generalization throughout the experiment.

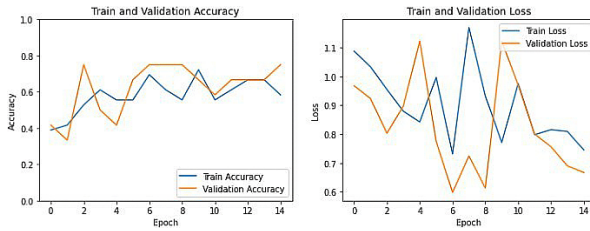
**4.3.3 Experiment 3 Impact of Data Augmentation**

This experiment evaluated the effect of data augmentation on the model’s generalization capability. Two training scenarios were compared on Model 1 performance: one with augmentation enabled and one without, each applied to both filtered and non-filtered images.

Without augmentation, the model achieved a test accuracy of 67% and an F1-score of 60%, with lower recall on underrepresented classes such as Scabies. After applying augmentation, performance improved substantially. The model reached 83% test accuracy, with an F1-score of 82%, and significantly better class-wise recall.

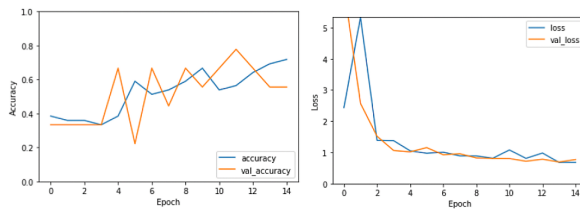
The most notable gain was observed in the Scabies class, where the model previously failed to generalize but successfully predicted more samples correctly after augmentation. This demonstrates that data augmentation was especially beneficial for underrepresented and visually variable categories. Below are the comparisons and explanations for each training scenario tested.

Figure 7 shows the learning curve for the best-performing model: trained with augmentation on filtered images. Both training and validation accuracy remained relatively high and aligned, while the validation loss decreased steadily. This reflects



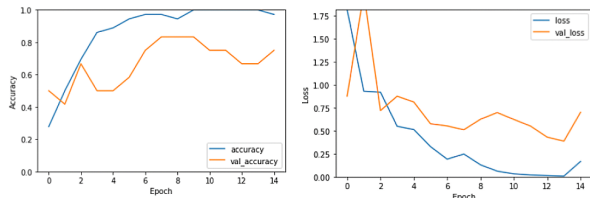
**Figure 7.** Training and validation accuracy and loss curves in Experiment 3, using filtered images with data augmentation.

good convergence and generalization, indicating that the combination of filtering and augmentation stabilized training and reduced overfitting.



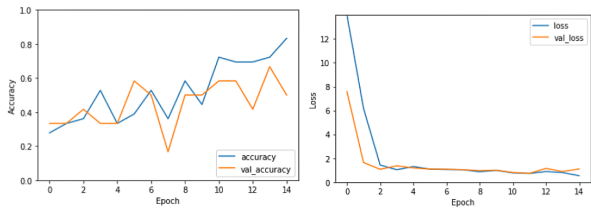
**Figure 8.** Training and validation accuracy and loss curves in Experiment 3, using non-filtered images with data augmentation

Figure 8 presents the result for non-filtered images with augmentation. The accuracy plot demonstrates more fluctuation compared to the filtered version, but overall, validation accuracy still trends upward. Meanwhile, training and validation loss converge smoothly after the initial few epochs, suggesting that augmentation helped mitigate the noise introduced by unfiltered backgrounds.



**Figure 9.** Training and validation accuracy and loss curves in Experiment 3, using filtered images without data augmentation

Figure 9 illustrates the model trained on filtered images without augmentation. Although training accuracy reached nearly 100%, validation accuracy plateaued and the gap between training and validation loss widened in later epochs. These patterns indicate that overfitting occurred due to lack of augmented variation, even though filtering helped reduce some visual noise.



**Figure 10.** Training and validation accuracy and loss curves in Experiment 3, using non-filtered images without data augmentation

Figure 10 depicts the weakest scenario: non-filtered images without augmentation. The curves exhibit significant fluctuation, especially in validation accuracy, and the training appears unstable. This result suggests that without either filtering or augmentation, the model failed to generalize effectively—likely due to inconsistent lighting, backgrounds, and lesion presentation in raw mobile images.

In conclusion, this experiment highlights that data augmentation plays a critical role in improving generalization and training stability, particularly in data-constrained settings. When combined with preprocessing (filtering), augmentation provides the strongest improvement. Supporting metrics are included in Appendix A: Table 5 and Figure 22–24.

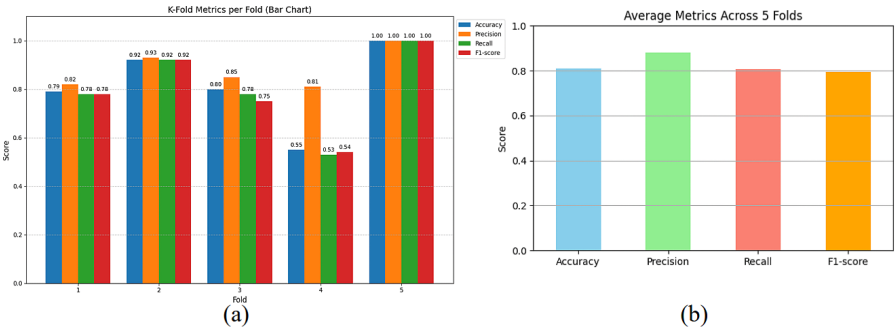
4.3.4 Cross-Validation for Model Robustness

To assess the generalization ability of the proposed CNN under limited data conditions, a 5-fold stratified cross-validation was conducted. In this evaluation, the dataset was partitioned into five subsets, where each fold used approximately 70% of the data for training and 30% for testing. This approach ensures that the model’s performance is not overly dependent on a particular train-test split and provides a more reliable estimate of real-world robustness.

The average performance across five folds was strong, with an accuracy of 81.1%, macro precision of 87.9%, recall of 80.6%, and F1-score of 79.4%. These metrics are presented in Figure 4.7, which includes (a) a fold-wise comparison of each performance metric and (b) the average values across all folds.

As shown in Figure 11 (a), the model maintained consistently high precision across most folds, while the recall and F1-score experienced moderate fluctuations, particularly in Fold 4, which exhibited the lowest performance. This drop is likely due to the small dataset, where a few hard-to-classify samples can disproportionately affect fold-specific outcomes. Despite this, the model achieved perfect scores in Fold 5 and strong results in the remaining folds, demonstrating its overall robustness.

Figure 11 (b) highlights the strength of the proposed architecture, where average precision and F1-score remained above 79%, indicating that the model not only predicts correctly but also balances sensitivity and specificity across all classes.



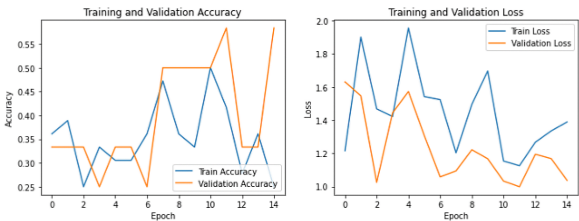
**Figure 11.** Cross-validation results of the proposed CNN: (a) Performance per fold; (b) Average metrics across five folds

These findings confirm that the CNN, supported by the proposed preprocessing pipeline, generalizes well to unseen data. Cross-validation thus reinforces the model’s potential to perform reliably in real-world deployments, addressing concerns of overfitting associated with holdout validation.

4.3.5 Experiment 4 Comparison with Transfer Learning Models

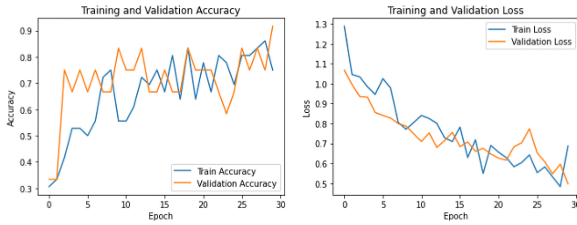
Finally, to assess performance in a low-data setting, Model 1 was compared with two widely used transfer learning architectures: ResNet-50 and VGG16, both pre-trained on ImageNet and fine-tuned on the same dataset.

Despite their architectural depth and popularity, both models underperformed compared to the custom CNN. As shown earlier in Figure 4.3, the proposed model achieved stable convergence and strong generalization. In contrast, ResNet-50 and VGG16 struggled to adapt. On the validation set, both models reached only 33% accuracy, with significantly lower F1- scores and inconsistent predictions—particularly for underrepresented and visually variable classes such as Scabies.



**Figure 12.** Training and validation accuracy and loss curves in Experiment 4 using ResNet-50

Figure 12 shows the training and validation curves for ResNet-50, where both accuracy and loss fluctuate across epochs. The validation accuracy peaks inconsistently, and the loss fails to converge smoothly, indicating unstable learning and poor generalization.



**Figure 13.** Training and validation accuracy and loss curves in Experiment 4 using VGG16

Figure 13 presents the results for VGG16, which displayed smoother convergence compared to ResNet. However, the validation accuracy continued to oscillate and did not show clear improvement over time, despite the decline in training and validation loss. This suggests limited effectiveness in extracting relevant features from the dataset, even with fine-tuning.

The inferior performance of these pre-trained models can be attributed to two main factors:

1. The limited dataset size, which makes it challenging to fine-tune large models with a high number of parameters without overfitting; and
2. The discrepancy between the general-purpose features learned from ImageNet and the specialized visual patterns associated with pet skin lesions.

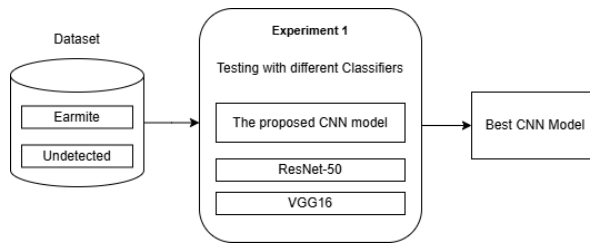
These findings, along with the comparative metrics in Appendix A: Figure27, reinforce the advantage of using a lightweight CNN model tailored for the task. In low-resource, domain- specific settings, such custom models can outperform transfer learning approaches by achieving better generalization with fewer parameters and task-appropriate feature learning.

#### **4.4 Model 2: Binary Classification for Earmite Detection**

Model 2 was developed specifically to detect Earmite infections, which are typically localized to the ear canal and characterized by consistent visual features such as dark, crust-like textures. Due to this anatomical confinement, a simpler preprocessing pipeline was used, applying only contrast enhancement (CLAHE) without segmentation.

The model employed the same CNN architecture described in Section 3.5, comprising three convolutional layers (32, 64, and 128 filters), max pooling layers, a fully connected dense layer with 512 units, and a dropout layer with a rate of 0.5. The output layer was adapted for binary classification by using a single sigmoid-activated neuron. To ensure reproducibility and fair comparison across experiments, a fixed random seed of 42 was used during model initialization and training. The full processing and classification pipeline for Model 2 is illustrated in Figure 14.

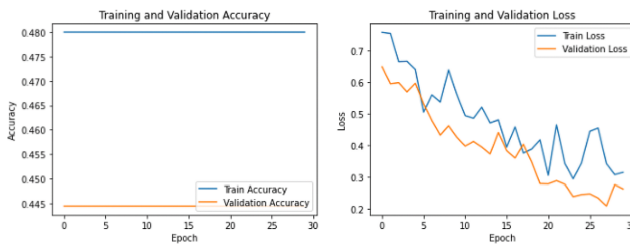




**Figure 14.** Block Diagram of Experimental Scenario Model 2

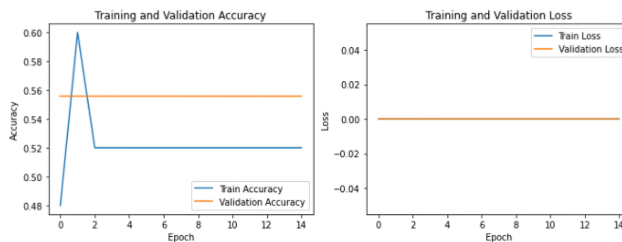
Trained to distinguish between “Earmite” and “Undetected” conditions, the model achieved 100% accuracy, precision, recall, and F1-score on both the test set and a separate, unseen validation set. This consistent performance indicates that the model was able to generalize effectively despite the limited dataset.

For comparison, transfer learning models ResNet-50 and VGG16 were fine-tuned on the same binary task. The following figures illustrate the training and validation behavior of each model.



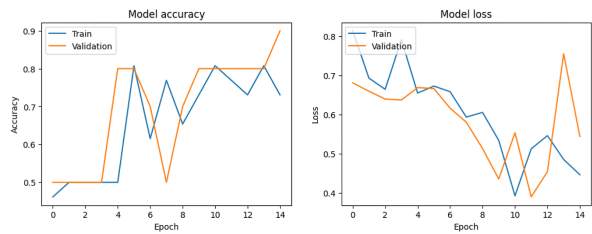
**Figure 15.** Training and validation accuracy and loss curves in Model 2 Experiment using VGG16

Figure 15 presents the learning curves for VGG16. Despite gradual loss reduction, both training and validation accuracy remained flat near 45%, indicating the model failed to learn meaningful discriminative features for earmite detection. This under-performance is likely due to the model’s reliance on generalized ImageNet features that do not transfer well to specialized, texture-specific tasks like this one.



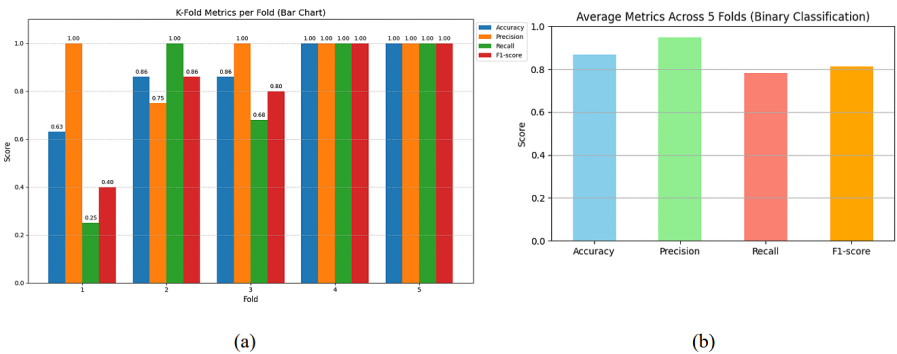
**Figure 16.** Training and validation accuracy and loss curves in Model 2 Experiment using Resnet

Figure 16 shows the performance of ResNet-50, which similarly failed to converge. Accuracy plateaued early around 53–55%, and loss values remained nearly constant, suggesting issues with gradient flow or over-parameterization relative to the small dataset.



**Figure 17.** Training and validation accuracy and loss curves in Model 2 Experiment using Proposed model

Figure 17 displays the training and validation learning curves of the proposed model over 15 epochs. While some fluctuation was observed—especially in the middle epochs—both training and validation accuracy generally increased and converged, with final validation accuracy reaching 80%. The loss curves similarly declined over time, and the relatively small gap between training and validation loss suggests that the model maintained a healthy balance between learning and generalization.



**Figure 18.** Cross-validation results of the proposed CNN: (a) Performance per fold; (b) Average metrics across five folds

To further validate the model’s robustness, a 5-fold stratified cross-validation was conducted. The results are shown in Figure 18 (a) and summarized in 18 (b). The proposed model achieved an average accuracy of 86.8%, precision of 95.0%, recall of 78.3%, and F1-score of 81.1% across the five folds. Notably, in two folds, the model achieved perfect classification performance (100% across all metrics), indicating consistent learning across different subsets of the dataset. The relatively high recall and F1-score suggest that the model can reliably detect earmite features with minimal false negatives.

These results confirm that, for domain-specific binary classification tasks with limited data, a custom lightweight CNN not only outperforms deeper pre-trained networks such as VGG16 and ResNet-50 but also generalizes reliably when trained on anatomically consistent input samples. Detailed performance metrics and per-fold analysis are provided in Appendix B: Table B1 and Figures B1–B2.

## 5. Conclusion

1. This study proposed a CNN-based image classification system for detecting three common pet skin diseases—Ringworm, Scabies, and Earmite—using images captured via smartphone cameras.
2. Two separate models were developed:
  - 2.1. Model 1 (multi-class): Classified Ringworm, Scabies, and Undetected conditions, achieving a test accuracy of 83% through HSV segmentation and data augmentation, with an average accuracy of 76.9% across five-fold cross-validation.
  - 2.2. Model 2 (binary): Model 2 (binary): Distinguished Earmite from Undetected cases, achieving 100% accuracy on an unseen test set and strong performance with an average cross-validation accuracy of 86.8%, using a simplified preprocessing pipeline.
3. Experimental results showed that the custom CNN models outperformed transfer learning baselines (ResNet-50 and VGG16) in terms of accuracy and robustness, especially under limited-data conditions.
4. The approach demonstrates the effectiveness of combining domain-specific preprocessing with task-focused CNN architectures for small-scale, real-world veterinary applications.
5. Looking forward, future work will focus on expanding the dataset to include more diverse samples and additional disease classes. Integration into a real-time mobile application, complete with image capture and automatic diagnosis features, is also envisioned. Further clinical validation, particularly through collaboration with veterinarians and field practitioners, will be crucial for assessing the system's reliability in operational settings.
6. Future directions may also include alternative evaluation strategies such as k-fold cross-validation on multiclass models, and the exploration of lightweight pre-trained models (e.g., MobileNet-V3, EfficientNet-Lite) as potential baselines to assess trade-offs in generalization and efficiency.



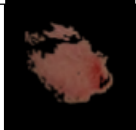
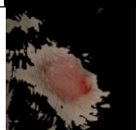
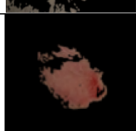
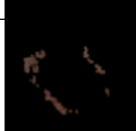
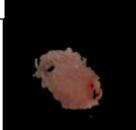
## References

- [1] E. S. A. Melman. "Mycology". In: *Skin diseases of dogs and cats: a guide for pet owners and professionals*. Potomac, Md.: DermaPet, 1994, pp. 73–82.
- [2] P. Sue. *Manual of Skin Diseases of the Dog and Cat*. 2nd. Chichester, West Sussex, United Kingdom: Blackwell, 2008.
- [3] R. G. H. McKeever and P. J. McKeever. *Colour handbook of skin diseases of the dog and cat: a problem-oriented approach to diagnosis and management*. London: Manson, 1998.

- [4] S. W. J. Liu and C. T. X. Zhang. "Towards improving diagnosis of skin diseases by combining deep neural network and human knowledge". In: *BMC Medical Informatics and Decision Making* 18,59 (2018).
- [5] S. S. Hwang, P. J. Hwang, and et al. "Classification of dog skin diseases using Deep Learning with images captured from multispectral imaging device". In: *Molecular Cellular Toxicology* 18 (2022), pp. 299–309.
- [6] N. Thoutam et al. "Detection and Classification of Dog Skin Disease using Deep Learning". In: *International Journal of Scientific Research in Engineering and Management (IJSREM)* 07.03 (2023).
- [7] Sudhriti Sengupta and N. M. M. M. "Improved skin lesions detection using color space and artificial intelligence techniques". In: *Journal of Dermatological Treatment* 31.5 (2020), pp. 511–518.
- [8] Colorado Animal Rescue. *Ringworm in Kittens*. Accessed: 2024-06-06. 2023. URL: <https://www.coloradoanimalrescue.org/ringworm-in-kittens/>.
- [9] Purina Australia. *What is Ringworm in Cats?* Accessed: 2024-06-06. 2023. URL: <https://www.purina.com.au/what-is-ringworm-in-cats.html>.
- [10] M. L. DiLonardo. *Ringworm in Cats: What It Looks Like & How to Treat It*. Accessed: 2024-06-06. The Spruce Pets. 2023. URL: <https://www.thesprucepets.com/ringworm-in-cats4175211>.
- [11] Cat-World. *Ringworm in Cats: Symptoms, Causes, and Treatment*. Accessed: 2024-06-06. Cat-World Veterinary Advice & Information. 2023. URL: <https://cat-world.com/ringworm-in-cats/>.
- [12] Vancouver Orphan Kitten Rescue Association (VOKRA). *There Are No Worms in Ringworm*. Accessed: 2024-06-06. Jan. 23, 2022. URL: <https://www.vokra.ca/news/2022/1/23/there-are-no-worms-in-ringworm>.
- [13] B. U. Palgunadi, K. K. G. Wangge, and L. D. K. Wardhani. "Handling of Scabies in Domestic Cat at Q-one Petklinik Surabaya". In: *Journal of Applied Veterinary Science and Technology* 2.2 (Oct. 2021), pp. 50–53.
- [14] Hello Sehat. *Scabies pada Kucing: Penyebab, Gejala, dan Cara Mengobatinya*. Accessed: 2024-06-06. HelloSehat.com. 2024. URL: <https://hellosehat.com/sehat/informasikesehatan/scabies-kucing/>.

Appendix 1. Supporting Results for Model 1 (Multi-Class Classification)

Table 2. Results of Experiment 1 – HSV Range Tuning

No	Lower Bound 1	Upper Bound 1	Lower Bound 2	Upper Bound 2	Segmentation Result
1	(170, 70, 125)	(180, 150, 200)	-	-	
2	(0, 60, 200)	(15, 69, 250)	(170, 70, 125)	(180, 150, 200)	
3	(0, 100, 100)	(10, 255, 255)	-	-	
4	(0, 50, 50)	(175, 250, 250)	-	-	
5	(0, 100, 100)	(10, 255, 255)	(0, 150, 100)	(175, 225, 250)	
6	(0, 50, 50)	(10, 150, 100)	(170, 205, 250)	(175, 255, 255)	
7	(0, 50, 50)	(10, 205, 255)	(170, 50, 50)	(175, 210, 255)	

**Table 3.** Results of Experiment 2 – Filtered Data (Hyperparameter Optimization)

dropout	Optimizer	Learning_ rate	Loss	val_loss	Accuracy	test_Accu racy	Validation Accuracy
0.3	Adam	0.01	0.4514	0.4617	0.5833	0.58	0.67
0.3	Adam	0.001	0.3838	0.3955	0.8333	0.83	0.67
0.3	Adam	0.0001	0.5161	0.5282	0.5	0.5	0.17
0.5	Adam	0.01	0.4915	0.498	0.6667	0.67	0.67
0.5	Adam	0.001	0.7454	0.6671	0.5833	0.75	0.83
0.5	Adam	0.0001	0.5375	0.5378	0.4167	0.42	0.67
0.3	RMSprop	0.01	0.6376	0.6376	0.3333	0.33	0.33
0.3	RMSprop	0.001	0.4546	0.4373	0.6667	0.67	0.67
0.3	RMSprop	0.0001	0.4618	0.4164	0.8333	0.83	0.5
0.5	RMSprop	0.01	0.4942	0.483	0.75	0.75	0.67
0.5	RMSprop	0.001	0.4146	0.3753	0.75	0.75	0.5
0.5	RMSprop	0.0001	0.5211	0.5524	0.75	0.75	0.67

**Table 4.** Results of Experiment 2 – Unfiltered Data (Hyperparameter Optimization)

dropout	Optimizer	Learning_ rate	loss	val_loss	Accuracy	test_Accu racy	Validation Accuracy
0.3	Adam	0.01	0.6387	0.6387	0.33	0.33	0.33
0.3	Adam	0.001	0.5392	0.5301	0.67	0.67	0.33
0.3	Adam	0.0001	0.5834	0.5845	0.75	0.75	0.17
0.5	Adam	0.01	0.6422	0.6422	0.33	0.33	0.33
0.5	Adam	0.001	0.5654	0.5543	0.5	0.5	0.5
0.5	Adam	0.0001	0.6279	0.6271	0.33	0.33	0.33
0.3	RMSprop	0.01	0.6216	0.6313	0.5	0.5	0.33
0.3	RMSprop	0.001	0.6834	0.6878	0.33	0.33	0.33
0.3	RMSprop	0.0001	0.5573	0.5604	0.75	0.75	0.5
0.5	RMSprop	0.01	0.6565	0.6608	0.4167	0.42	0.33
0.5	RMSprop	0.001	0.6144	0.6062	0.5833	0.58	0.17
0.5	RMSprop	0.0001	0.7152	0.7121	0.33	0.33	0.33

**Table 5.** Accuracy Comparison in Experiment 3 (With vs. Without Augmentation)

Metric	The developed CNN model	ResNet-50	VGG16
Train Accuracy	0.5833	0.25	0.75
Test Accuracy	0.75	0.58	0.83
Validation Accuracy	0.83	0.33	0.33
Macro Avg Precision	0.89	0.11	0.13
Macro Avg Recall	0.83	0.33	0.33
Macro Avg F1-score	0.82	0.17	0.19

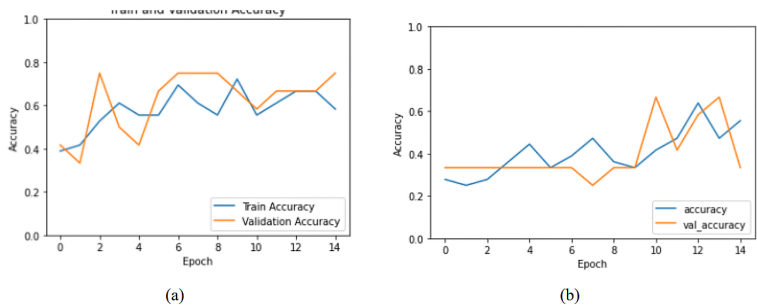


Figure 19. Training and Validation Accuracy/Loss in Experiment 2: (a) Filtered Data, (b) Unfiltered Data

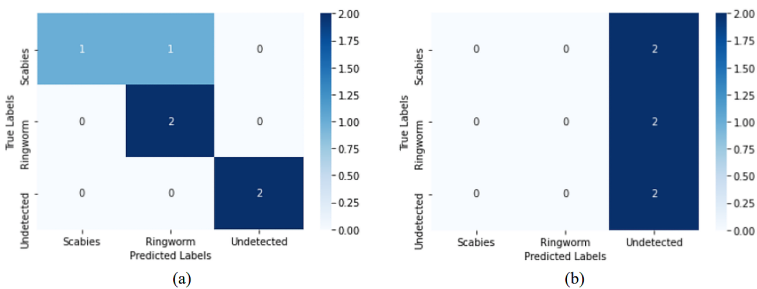


Figure 20. Confusion Matrix for Experiment 2: (a) Filtered Data, (b) Unfiltered Data

Classification Report:					Classification Report:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Scabies	0.67	1.00	0.80	2	Scabies	0.00	0.00	0.00	2
Ringworm	1.00	0.50	0.67	2	Ringworm	0.00	0.00	0.00	2
Undetected	1.00	1.00	1.00	2	Undetected	0.33	1.00	0.50	2
accuracy			0.83	6	accuracy			0.33	6
macro avg	0.89	0.83	0.82	6	macro avg	0.11	0.33	0.17	6
weighted avg	0.89	0.83	0.82	6	weighted avg	0.11	0.33	0.17	6

Figure 21. Classification Report for Experiment 2: (a) Filtered Data, (b) Unfiltered Data

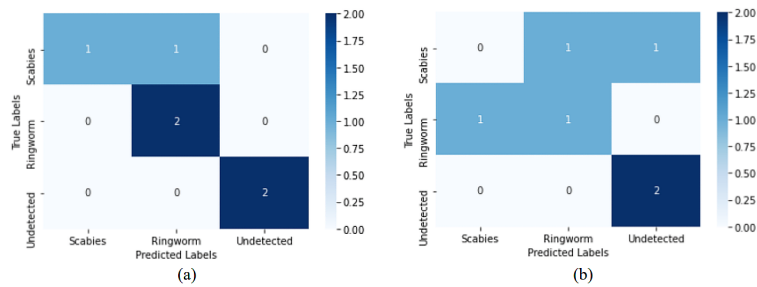
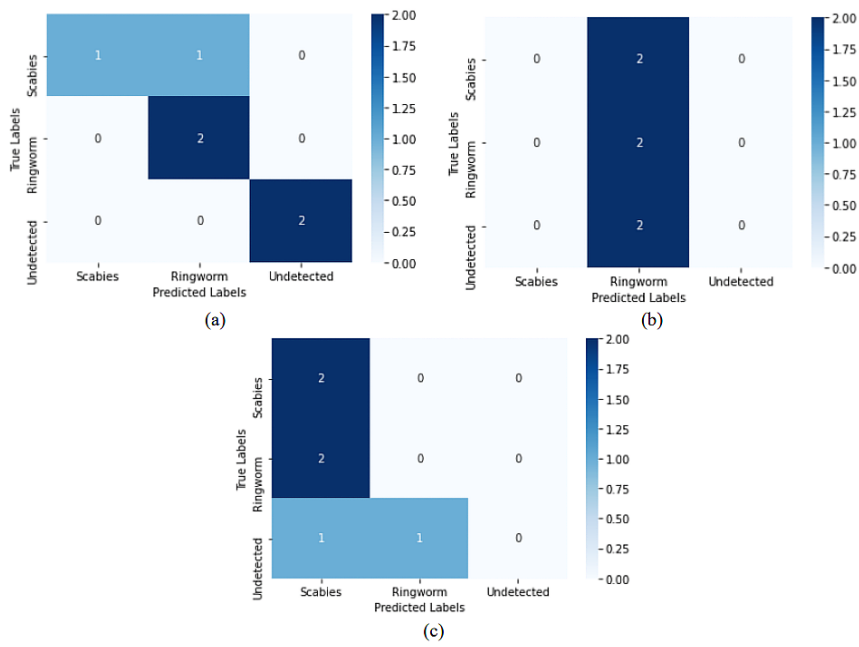


Figure 22. Confusion Matrix with and without Data Augmentation: (a) With Augmentation, (b) Without Augmentation

Classification Report:					Classification Report without augmentation:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Scabies	0.67	1.00	0.80	2	Scabies	0.50	0.50	0.50	2
Ringworm	1.00	0.50	0.67	2	Ringworm	0.00	0.00	0.00	2
Undetected	1.00	1.00	1.00	2	Undetected	0.67	1.00	0.80	2
accuracy			0.83	6	accuracy			0.50	6
macro avg	0.89	0.83	0.82	6	macro avg	0.39	0.50	0.43	6
weighted avg	0.89	0.83	0.82	6	weighted avg	0.39	0.50	0.43	6

(a) (b)

**Figure 23.** Classification Report with and without Data Augmentation: (a) With Augmentation, (b) Without Augmentation



**Figure 24.** Confusion Matrix Comparison – Experiment 4: (a) Custom CNN, (b) ResNet-50, (c) VGG16

Classification Report:					Classification Report with ResNet-50:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Scabies	0.67	1.00	0.80	2	Scabies	0.33	1.00	0.50	2
Ringworm	1.00	0.50	0.67	2	Ringworm	0.00	0.00	0.00	2
Undetected	1.00	1.00	1.00	2	Undetected	0.00	0.00	0.00	2
accuracy			0.83	6	accuracy			0.33	6
macro avg	0.89	0.83	0.82	6	macro avg	0.11	0.33	0.17	6
weighted avg	0.89	0.83	0.82	6	weighted avg	0.11	0.33	0.17	6

(a) (b)

Classification Report with VGG16:				
	precision	recall	f1-score	support
Scabies	0.00	0.00	0.00	2
Ringworm	0.40	1.00	0.57	2
Undetected	0.00	0.00	0.00	2
accuracy			0.33	6
macro avg	0.13	0.33	0.19	6
weighted avg	0.13	0.33	0.19	6

(c)

**Figure 25.** Classification Report Comparison – Experiment 4: (a) Custom CNN, (b) ResNet-50, (c) VGG16



Appendix 2. Supporting Results for Model 2 (Binary Classification)

Table 6. Accuracy Metrics for Model 2 Experiments

Metric	The developed CNN model	ResNet-50	VGG16
Train Accuracy	0.56	0.52	0.48
Test Accuracy	0.5556	0.55	0.45
Validation Accuracy	1.00	0.5	0.5
Macro Avg Precision	1.00	0.25	0.25
Macro Avg Recall	1.00	0.5	0.5
Macro Avg F1-score	1.00	0.33	0.33

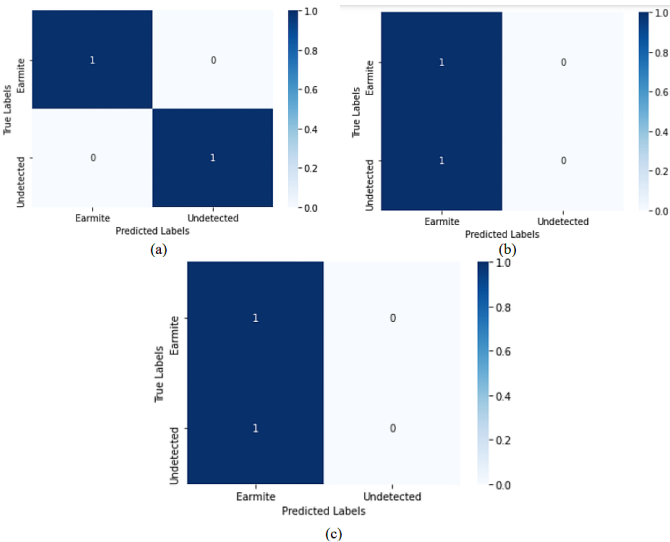


Figure 26. Confusion Matrix Comparison – Model 2:(a) Custom CNN, (b) ResNet-50, (c) VGG16

Classification Report:					Classification Report with ResNet-50:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Earmite	1.00	1.00	1.00	1	Earmite	0.50	1.00	0.67	1
Undetected	1.00	1.00	1.00	1	Undetected	0.00	0.00	0.00	1
accuracy			1.00	2	accuracy			0.50	2
macro avg	1.00	1.00	1.00	2	macro avg	0.25	0.50	0.33	2
weighted avg	1.00	1.00	1.00	2	weighted avg	0.25	0.50	0.33	2

(a)

Classification Report with VGG16:					Classification Report with ResNet-50:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Earmite	0.50	1.00	0.67	1	Earmite	0.50	1.00	0.67	1
Undetected	0.00	0.00	0.00	1	Undetected	0.00	0.00	0.00	1
accuracy			0.50	2	accuracy			0.50	2
macro avg	0.25	0.50	0.33	2	macro avg	0.25	0.50	0.33	2
weighted avg	0.25	0.50	0.33	2	weighted avg	0.25	0.50	0.33	2

(b)

(c)

Figure 27. Classification Report Comparison – Model 2: (a) Custom CNN, (b) ResNet-50, (c) VGG16