Scalable Dog Identification Using Nose Prints and CNNs for Urban Welfare

Apurba Koirala[1]

Vellore Institute of Technology, Vellore, India

Abstract

This paper presents a scalable dog identification system utilizing nasal print biometrics and deep learning to address the challenges posed by stray and abandoned dogs. Traditional methods of identification, such as microchipping, are invasive, expensive, and often inaccessible in resource-limited regions, making them less viable for widespread use in such contexts.The proposed system utilizes pre-trained convolutional neural networks (CNNs), including ResNet50, MobileNetV2, and EfficientNetB0, to extract distinctive nasal features from dog images. These features are processed through a Metric Learning framework to generate embeddings in a discriminative feature space, enabling identification via cosine similarity. The system's performance was evaluated using a dataset of 1,165 images, revealing that ResNet50 achieves the highest accuracy at 82%, though it requires significant computational resources, which limits its practical application in resource-constrained environments. On the other hand, MobileNetV2 strikes a balance between

*Corresponding author

Email address: apurbakoirala33@gmail.com (Apurba Koirala)

accuracy and computational efficiency, achieving 71% accuracy while being well-suited for mobile device deployment. EfficientNetB0 provides a moderate level of accuracy at 78%, with slightly higher resource demands than MobileNetV2. Graphical analyses of these models further highlight the trade-offs between computational efficiency and accuracy, emphasizing the practical advantages of MobileNetV2 for deployment in real-world scenarios. This non-invasive and cost-effective approach holds significant potential for improving the welfare of stray dogs, reducing abandonment rates, and enabling efficient tracking and management of dogs in urban environments.
Keywords: Convolutional Neural Networks (CNNs), ResNet50, MobileNetV2, EfficientNetB0, Cosine similarity, Feature extraction

1. Introduction

The increasing number of stray dogs in urban environments poses significant challenges globally, with an estimated 200 million stray dogs worldwide [22]. In Kathmandu, the stray dog population is estimated to range between 2,000 and 20,000 [23], creating challenges for pet owners, animal welfare organizations, local authorities, and communities. Overcrowded shelters, limited resources, and the inability to efficiently track or identify these animals exacerbate the issue, making it difficult to provide care, manage populations, or reunite lost dogs with their owners. Furthermore, abandoned pet dogs, left to survive in harsh conditions, face poor health, unsanitary environments, and even death.

Traditional identification methods, such as microchipping, tags, or collars, are either invasive, expensive, or prone to failure in the harsh conditions stray dogs face. While microchipping is a reliable method, it is often unaffordable and inaccessible in resource-limited regions, and many stray dogs remain unchipped [16]. These challenges necessitate a non-invasive, cost-effective, and scalable solution, particularly for regions with limited resources.

This paper proposes a novel deep learning-based dog identification system using unique nasal print biometrics. Studies have shown that dog nose prints are as unique as human fingerprints [17]. Leveraging computer vision and Siamese neural networks, this system matches dog nose prints to a stored database. It employs lightweight, mobile-friendly CNN architectures such as MobileNetV2 [18], EfficientNet [19], and ResNet50 [8] for feature extraction and matching.

MobileNetV2 and EfficientNet, with their computational efficiency, are ideal for mobile deployment, enabling real-time, on-the-go identification of stray dogs. These models are particularly suited for resource-constrained areas like Kathmandu, where infrastructure limitations prevail. ResNet50, despite its computational demands, offers superior feature extraction capabilities for high-accuracy scenarios. The system is trained on a manually curated dataset of dog nose images collected across Kathmandu, including shelters, homes, and pet shops.

The goal of this research is to develop a scalable, non-invasive system to improve stray dog management and welfare, reduce abandonment rates, and increase efficiency. This system can assist in reuniting lost or abandoned dogs with their owners or finding them new homes, enhancing the overall quality of life for stray animals.

The rest of this paper is organized as follows: Section 2 reviews existing literature on animal and dog biometrics and tracking systems. Section 3 describes the proposed identification method. Section 4 discusses the datasets, experimental setup, and results analysis. Section 5 presents the results, and Section 6 concludes the paper and highlights future directions.

2. Related Works

Biometric identification of animals, particularly using deep learning techniques, has become a crucial area of research, with a specific focus on nose prints as a unique identifier for dogs. Several studies have proposed various methods leveraging deep learning and biometric features for accurate identification.

Shen et al. (2022) proposed a competitive approach for dog nose-print re-identification using convolutional neural networks (CNNs). Their method focused on utilizing CNNs to extract high-dimensional feature vectors, emphasizing the robustness and scalability of deep learning models for animal identification tasks in large-scale datasets [1]. Cho and Kim (2021) developed a canine biometric identification system using electrocardiogram (ECG) signals and CNN-LSTM neural networks. Though focused on ECG data, their work underscores the broader application of CNNs for biometric identification in animals, highlighting the versatility of deep learning frameworks [2].

Bae et al. (2020) introduced a deep neural network-based system for dog nose-print identification, where they successfully captured unique nose patterns and demonstrated that deep learning models, such as CNNs, can efficiently extract features necessary for accurate identification [3]. Chakraborty et al. (2020) conducted a comprehensive survey of visual animal biometrics, providing an overview of the challenges and methodologies in this domain. Their research emphasized the role of advanced feature extraction techniques, such as those utilized by CNN models, in identifying animals based on phenotypic appearance [4].

Kumar et al. (2020) investigated the muzzle of pigs as a biometric for breed identification, applying pattern recognition and image processing techniques. Their study demonstrates the cross-species applicability of biometric identification and offers valuable insights for dog identification through nose prints [5]. Kühl and Burghardt (2013) discussed the quantification and detection of phenotypic appearance in animals, further contributing to the understanding of how visual traits can be utilized for biometric identification. Their work laid the foundation for using phenotypic features like nose prints for identifying dogs [24].

Coldea (1994) established the use of nose prints as a method for dog identification, demonstrating that a dog's nose print is as unique as a human fingerprint. This early work serves as a foundation for the later development of automated identification systems based on nose-print patterns [7]. He et al. (2016) introduced the ResNet architecture, which revolutionized image recognition by addressing the vanishing gradient problem in deep networks. ResNet has since been widely applied in various fields, including animal biometrics, for extracting robust features from visual data [8].

Toshev and Szegedy (2014) presented DeepPose, a deep learning-based approach for human pose estimation, which showcased the potential of CNNs in accurately identifying and tracking individuals through visual data. Though this work focused on human subjects, it is relevant to the challenges of animal identification using visual biometrics [9]. Deb et al. (2018) explored face recognition for primates in the wild, further demonstrating the efficacy of CNNs in biometric identification in uncontrolled environments. This work parallels the challenges of identifying street dogs using biometric features like nose prints [25].

Duyck et al. (2015) developed the Sloop pattern retrieval engine for individual animal identification, showcasing the importance of pattern recognition in accurately identifying animals in the wild. Their approach supports the concept of using distinct visual features for identification, relevant to the use of nose prints in dogs [10]. Noviyanto and Arymurthy (2013) developed a method for identifying beef cattle based on muzzle patterns using a refinement technique in the SIFT method. Their research illustrates the potential for applying feature extraction techniques to identify animals based on facial patterns, which is directly applicable to dog nose-print recognition [11].

Hansen et al. (2018) extended biometric identification to pigs using face recognition with CNNs. Their success in identifying pigs based on facial features reinforces the effectiveness of deep learning models for animal biometric identification across species [12]. A variety of animal biometrics methods have been explored across species. Ku¨hl and Burghardt (2013) review various phenotypic-based identification techniques that have demonstrated promising results in tracking wildlife and domestic animals through biometric markers like skin patterns and body structures [24]. Similarly, Bae et al. (2023) proposed a deep neural network-based approach to identify individual dogs through their nose prints, a method inspired by the uniqueness of canine nose prints [3].

Recent advancements in animal identification technologies have paved the way for novel solutions to address the challenges of tracking and managing stray dog populations. Gamble et al. (2018) highlight the importance of accurate identification systems in managing the growing issue of stray dogs, underlining the need for scalable and efficient methods [13]. Traditional animal identification approaches such as visual markers and RFID tags have shown limitations, pushing researchers to explore biometric techniques that leverage phenotypic features, as reviewed by Awad (2016), especially in cattle tracking and identification [15].

Limitations of Existing Methods

The limitations of the existing methods for dog identification are:

- **Limited Image Quality:** Not all photos are taken in high resolution because of the wide range of camera quality that residents of cities like

Kathmandu have access to. This can affect the model's accuracy when detecting minute features in dog nose patterns.

- **Device Compatibility:** Device-specific variations in the system's performance are possible, particularly on low-end smartphones that can have trouble processing high-dimensional characteristics taken from photos. This may restrict the system's use for a large audience.

- **Network Dependency for Heavy Models:** Network connectivity is frequently necessary for the deployment of complicated models on mobile devices in order to offload computation to cloud servers, which isn't always available or dependable in all parts of Kathmandu.

- **Processing Power Constraints:** The usability of the system in real-time situations is impacted by the considerable computing resources that current high-accuracy models frequently demand, which may not be possible on typical mobile devices without optimization.

3. Proposed Method

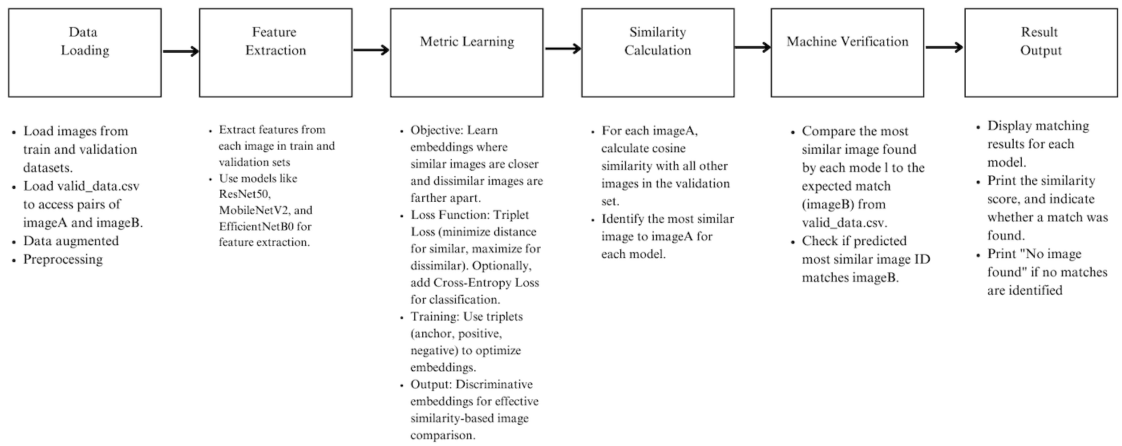

Figure 1 Displays the pipeline for the proposed model. Each section is explained as follows:

3.1. Data Loading

- **Dataset Information:** The dataset is divided into two subsets: a training set and a validation set. Each subset contains images of dog noses, with each dog having two or three images taken from different angles and under various lighting conditions. These images are captured from a diverse set of dogs across different locations, including shelters, homes, and pet shops. The multiple images per dog are intended to capture variations in the dog's nose print under different real-world conditions, ensuring robustness and generalizability for the identification system. The goal is to train the model to accurately match these variations and identify dogs with high precision.

- **Metadata Loading:** A `valid.csv` file is used to load metadata associated with the image pairs. Each entry in this file corresponds to a pair of images, `imageA` and `imageB`, where both images belong to the same dog. The `valid.csv` file provides crucial information such as the image paths and a label indicating whether the pair represents the same dog (i.e., a ground truth for verifying model predictions). This metadata is essential for training the model by providing labeled examples of positive pairs (images of the same dog) and negative pairs (images of different dogs), which allows the Siamese network to learn to distinguish between them.

- **Data Augmentation:** To further enhance the model's performance and generalize well to real-world scenarios, data augmentation techniques are applied. These include random rotations, flips, and brightness adjustments to simulate the variations in lighting, angles, and orientations that may be encountered during deployment. By artificially expanding the dataset in this manner, the model is exposed to a broader range of conditions, helping improve its ability to handle unseen images during inference.

- **Preprocessing:** The images are resized to a uniform dimension, ensuring consistency across the dataset. Additionally, normalization is performed on the pixel values to bring them within a specific range, typically between 0 and 1, for faster convergence during model training. The images are also converted to grayscale to emphasize the unique texture patterns in the nose prints and reduce computational complexity.

- **Data Split:** The dataset is split into a training set and a validation set, with the training set used to train the model and the validation set used to assess its generalization ability. 80% (400 dogs) are used for training, and 20% (100 dogs) are used for testing.

3.2. Feature Extraction

- **Feature Extraction Models:** To effectively capture the unique characteristics of each dog nose image, we use pre-trained deep learning models for feature extraction. These models, which have been trained on large, diverse datasets such as ImageNet, are capable of extracting high-level semantic features from images. The three primary models utilized in this study are ResNet50, MobileNetV2, and EfficientNetB0. These models were selected due to their proven success in various image recognition tasks and their ability to generalize well to different domains, including biological recognition tasks like dog identification.

- **ResNet50:** ResNet50 is a 50-layer deep convolutional neural network designed to address the challenges of training very deep networks, such as the vanishing gradient problem. The key innovation in ResNet50 is the introduction of residual connections, where the output of a set of layers is added to the input, forming a shortcut connection. This enables the network to preserve gradients during backpropagation, allowing effective training of deeper models.

The architecture of ResNet50 consists of five stages:

- An initial convolutional layer with batch normalization, ReLU activation, and max pooling for feature extraction and dimensionality reduction.
- Stages 2 to 5, each comprising multiple residual blocks. Each residual block contains convolutional layers with varying filter sizes (1×1, 3×3, and 1×1), followed by batch normalization and ReLU activation. The blocks are categorized into:
  * **Conv Blocks:** Include a convolutional shortcut to match dimensions when necessary.
  * **Identity Blocks:** Maintain direct shortcut connections when input and output dimensions match.
- The network concludes with an average pooling layer, flattening, and a fully connected layer to produce the final output.

ResNet50 is widely used for feature extraction, capable of learning low-level features (e.g., edges and textures) and high-level abstractions (e.g., structures and patterns), making it highly effective for tasks like dog nose identification. Figure 2. below illustrates the architecture and its key components.

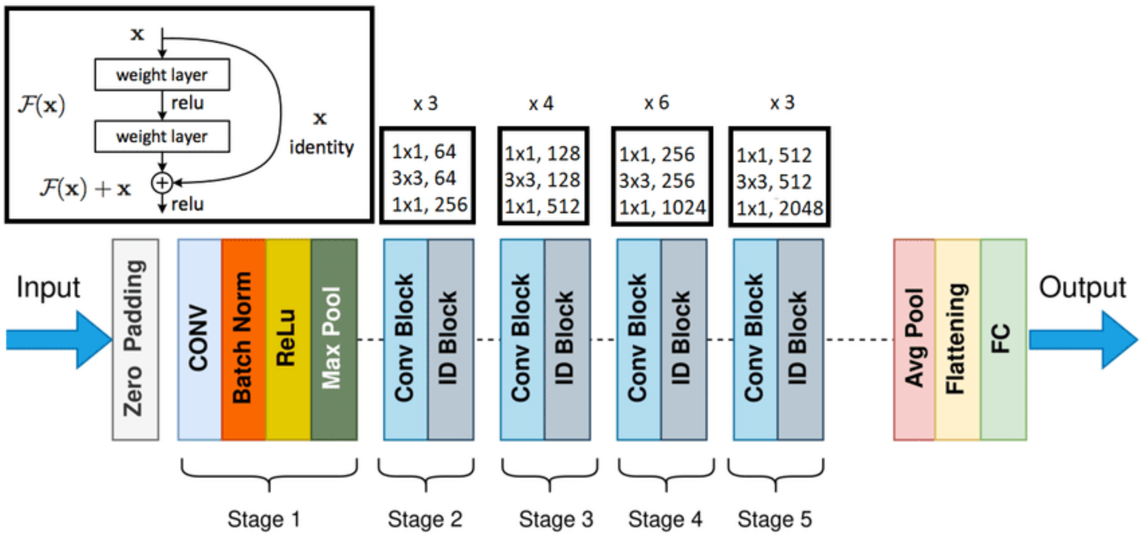

Figure 2: Structure of the ResNet50 model. [27]

- **MobileNetV2:** MobileNetV2 is a lightweight and efficient deep neural network architecture designed for mobile and embedded systems. It introduces inverted residuals and linear bottlenecks to improve performance while reducing computational cost. The key innovation lies in the use of depthwise separable convolutions, which significantly reduce the number of parameters compared to traditional convolutions, making the model both fast and resource-efficient.

  The architecture of MobileNetV2 is structured as follows:

  - Bottleneck Blocks: The core building blocks include:
    - A 1×1 convolution with ReLU activation to project features to higher dimensions.

    * A depthwise convolution ($Dwise$) that performs spatial filtering independently for each channel, followed by ReLU activation.
    * Another 1×1 convolution (linear) to project features back to lower dimensions.
  - Shortcut Connections: If the input and output dimensions are the same, a residual (shortcut) connection is added to enhance gradient flow and model efficiency.
  - Customized Head: The final layers include:
    * Global average pooling to reduce feature maps to a single vector.
    * A fully connected layer (Dense) with ReLU activation.
    * Dropout ($p = 0.5$) to prevent overfitting.
    * A final Dense layer with softmax activation for classification.

  MobileNetV2 is particularly suited for tasks requiring real-time processing, such as dog nose identification on mobile devices, due to its efficiency and speed. Figure 3. below illustrates the MobileNetV2 architecture, including its bottleneck blocks and customized classification head.

- **EfficientNetB0:** EfficientNetB0 is the foundational model of the EfficientNet family, developed using a compound scaling method to balance network depth, width, and resolution effectively. This compound

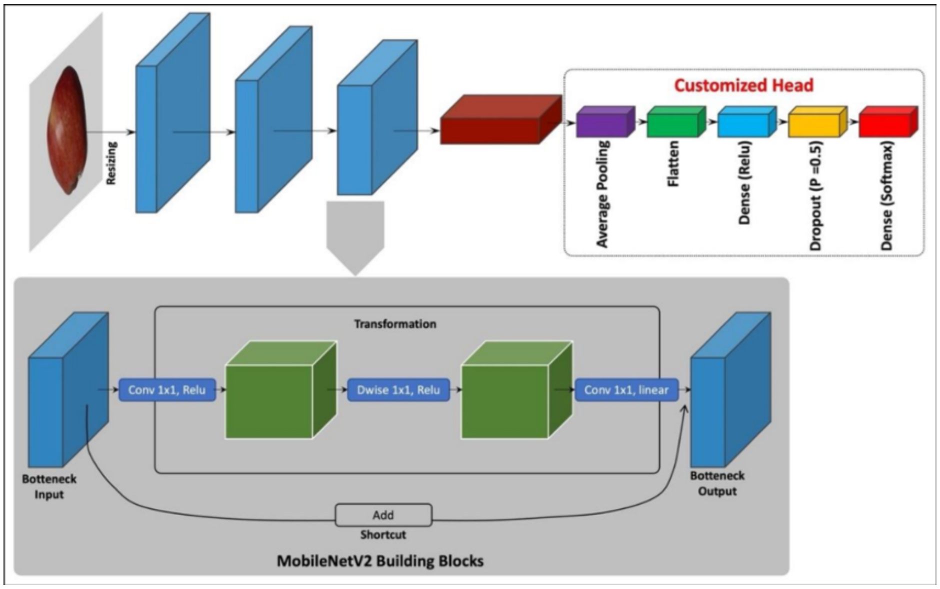

Figure 3: Example of an image processed by the MobileNetV2 model [26].

scaling uniformly scales all three dimensions, enabling the model to achieve higher accuracy with significantly reduced computational cost compared to traditional architectures.

The architecture of EfficientNetB0 is structured as follows:

– Initial Layers: A standard 3×3 convolution layer followed by batch normalization and ReLU activation to extract basic features from the input image.

– Mobile Inverted Bottleneck Convolution (MBConv) Blocks: The core of EfficientNetB0 consists of stacked MBConv layers, which include:

  * Depthwise Convolutions: Perform spatial filtering for each

channel independently.

    * Pointwise Convolutions: Combine information across channels while maintaining computational efficiency.

    * Squeeze-and-Excitation (SE) Mechanism: Enhances the model's ability to focus on the most informative features by recalibrating channel weights.

  – Scaling Blocks: The architecture transitions between different block types (e.g., 3×3, 5×5, and 7×7) and increases feature map sizes or reduces spatial dimensions as needed.

  – Final Layers: Global average pooling, followed by a fully connected layer and softmax activation for classification.

EfficientNetB0 achieves a remarkable balance between accuracy and efficiency, making it well-suited for recognizing fine-grained patterns like dog nose prints. Figure 4. below illustrates the EfficientNetB0 architecture, showcasing its sequential MBConv blocks and scaling transitions.

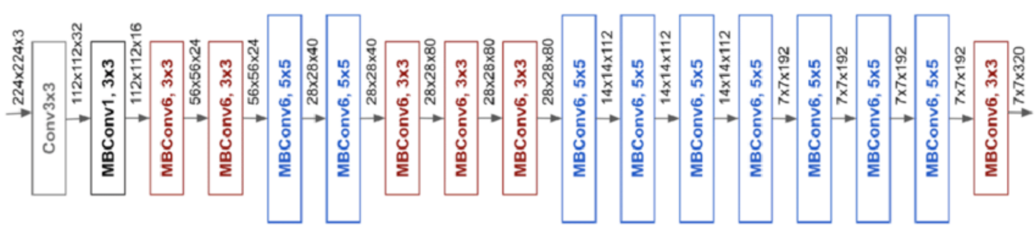

Figure 4: Example of an image processed by the EfficientNetB0 model. [28]

- **Feature Vector Generation:** Each of the pre-trained models processes the dog nose images to generate a feature vector. These feature vectors are compact numerical representations of the image, capturing its unique attributes and high-level patterns such as texture, shape, and other discriminative features. The feature vectors are extracted from the layers of the models just before the final classification layer (i.e., using a global average pooling operation). The resulting vectors are then used for further analysis, such as comparing image similarities and performing classification tasks. These feature vectors provide a high-dimensional representation of the images that allows the system to distinguish between different dogs effectively.

- **Summarizing model parameters:** The given table 1 summarizes the model parameters.

Model	**Depth (Layers)**	**FLOPS (GFLOPs)**	**Parameters (Million)**	**Model Size (MB)**	**Key Feature**
ResNet50	50	4.1	25.6	98	High accuracy
MobileNetV2	88	0.3	3.4	14	Lightweight, mobile-friendly
EfficientNetB0	53	0.4	5.3	20	Balance of efficiency and accuracy

Table 1: Summary of Model Parameters

3.3. Metric Learning

- **Objective of Metric Learning:** Metric learning aims to learn a distance function that measures the similarity between two images in a meaningful way. In the context of dog nose recognition, the objective is to embed images into a feature space such that images of the same dog are close to each other, while images from different dogs are farther apart. This learning task is typically achieved using a Siamese Network or a Triplet Network, which is trained using a special loss function that encourages the model to minimize the distance between similar images and maximize the distance between dissimilar images.

- **Triplet Loss:** The primary loss function used in this study is triplet loss, which operates on triplets of images: an anchor image (imageA), a positive image (imageB), and a negative image (imageC). The anchor and positive images come from the same dog, while the negative image comes from a different dog. The triplet loss function encourages the model to learn embeddings such that:

$$\text{Loss} = \max(0, d(\mathbf{f}(A), \mathbf{f}(B)) - d(\mathbf{f}(A), \mathbf{f}(C)) + \alpha)$$

  where $d(\mathbf{f}(A), \mathbf{f}(B))$ is the distance between the anchor and positive image embeddings, $d(\mathbf{f}(A), \mathbf{f}(C))$ is the distance between the anchor and negative image embeddings, and α is a margin that ensures a sufficient gap between positive and negative pairs. This loss function encourages the network to pull similar images closer and push dissimilar images further apart in the feature space.

- **Embedding Space Optimization:** During training, the model is optimized to minimize the triplet loss, refining the feature space to better distinguish between different dog nose prints. The network learns to map images into a space where the geometric distance corresponds to the semantic similarity. To improve the learning process, hard negative mining is employed. Hard negative mining selects negative examples that are most challenging for the model to classify correctly, providing more informative gradients that accelerate the model's learning and improve the discriminative power of the embedding space.

- **Integration with Pretrained Models:** The ResNet50, MobileNetV2, and EfficientNetB0 models are used as the backbone networks for feature extraction. These pre-trained models are fine-tuned with the triplet loss function to generate embeddings that are highly discriminative for dog nose identification. The embeddings from these models are passed through the triplet loss network, which optimizes the model to increase the separability of dog nose prints in the learned feature space.

- **Effectiveness of Metric Learning:** The use of metric learning with triplet loss improves the robustness of the model in differentiating between dog noses, even in the presence of variations in lighting, angle, or noise. By optimizing the model's embedding space, it learns to recognize finer-grained details and structural features of each dog's nose, ultimately enhancing the accuracy of matching dog noses based on their unique prints.

3.4. Cosine Similarity

Cosine similarity is a metric used to measure the cosine of the angle between two non-zero vectors in an inner product space. It is used to determine how similar two images are based on their feature vectors. For each image pair in the validation set, cosine similarity is calculated between the feature vector of the first image (imageA) and the feature vectors of all other images (imageB). The formula for cosine similarity is given by:

$$\text{Cosine Similarity}(A, B) = \frac{\mathbf{f}(A) \cdot \mathbf{f}(B)}{\|\mathbf{f}(A)\|\|\mathbf{f}(B)\|}$$

Where:

- $\mathbf{f}(A)$ and $\mathbf{f}(B)$ are the feature vectors of images A and B, respectively.
- $\cdot$ denotes the dot product between the two feature vectors.
- $\|\mathbf{f}(A)\|$ and $\|\mathbf{f}(B)\|$ represent the Euclidean norms (magnitudes) of the vectors.

The cosine similarity metric ranges from -1 to 1, where a value closer to 1 indicates that the two images are highly similar, and a value closer to 0 indicates that the images are dissimilar. In the context of dog nose identification, a higher cosine similarity indicates that two images are likely to belong to the same dog, while a lower similarity suggests they are from different dogs.

For each imageA, the image with the highest cosine similarity score is identified as the most similar image and is predicted to be the match. This process is repeated for all images in the validation set.

3.5. Machine Verification

- Predicted Match Comparison: The most similar image found for each imageA (predicted match) is compared to the ground truth match (imageB) from the valid.csv file.

- Match Verification: The predicted match's ID is checked to determine if it corresponds to the ID of imageB. This verification step assesses the accuracy of the feature extraction and similarity calculation steps in identifying the correct matching image.

3.6. Result Output

- Display of Results: For each image pair, the matching results are displayed. If a match is found, the similarity score is printed alongside an indication of success.

- Handling No Match Scenarios: If no matches are identified, a message stating "No image found" is printed.

The algorithm for the entire process is as follows:

Algorithm 1 Dog Nose Recognition Algorithm

procedure DogNoseRecognition
  Input: Images of dog noses, Metadata file `valid.csv`, Pretrained models (ResNet50, MobileNetV2, EfficientNetB0)
  Output: Predicted matches and similarity scores for image pairs
  Initialize pretrained models: ResNet50, MobileNetV2, EfficientNetB0
  Initialize margin α for triplet loss
  Initialize empty list to store feature vectors
  for each image in the dataset **do**
    Resize image to 224×224
    Preprocess image (normalize pixel values)
    Extract feature vector using ResNet50, MobileNetV2, EfficientNetB0
    Store feature vectors for each image
  end for
  for each triplet (Anchor, Positive, Negative) in the dataset **do**
    Compute triplet loss:

$$L = \max(d(A, P) - d(A, N) + \alpha, 0)$$

    Update the model weights to minimize the triplet loss using hard negative mining
  end for
  for each image pair (ImageA, ImageB) **do**
    Compute cosine similarity between feature vectors:

$$\text{Cosine Similarity} = \frac{\mathbf{f}_A \cdot \mathbf{f}_B}{\|\mathbf{f}_A\|\|\mathbf{f}_B\|}$$

    Store similarity score for each image pair
  end for
  for each image pair (ImageA, ImageB) **do**
    Compare the similarity score with predefined threshold
    if similarity score $\geq$ threshold **then**
      Output "Match"
    else
      Output "No match"
    end if
  end for
  Return the predicted matches and similarity scores
end procedure

4. Datasets and Experimental results analysis

The study's dataset was meticulously collected from a variety of sources in Kathmandu, including the city's streets, different dog shelters, private residences, and pet stores. By guaranteeing a broad range of dog breeds, ages, and looks, this method improves the dataset's diversity and robustness, both of which are essential for trustworthy model training and validation. A total of 1,165 photos, of around 500 distinct canines, were gathered. Every picture highlights the delicate, recognizable patterns that are crucial for precise identification by capturing different aspects of the dog's nose. To facilitate model training and evaluation, the dataset was divided into two parts: Training Set: Consisting of 1,049 images, this subset was used to train the model in recognizing and distinguishing between the individual features of different dog noses, as displayed by Table 2. Validation Set: Consisting of 116 images, this subset was reserved for evaluating the model's accuracy and generalization capability as displayed by Table 3.

Table 2: Training Dataset Details

Attribute	Description	Values	Count
Total Images	Number of images in the training dataset	-	1,049
Unique Dogs	Number of unique dogs represented	-	443
Image Resolution	Resolution of images	224 x 224	-
Label Distribution	Proportion of labels across the dataset	Balanced	-

Table 3: Validation Dataset Details

Attribute	Description	Values	Count
Total Images	Number of images in the validation dataset	-	116
Unique Dogs	Number of unique dogs represented	-	57
Image Resolution	Resolution of images	224 x 224	-
Label Distribution	Proportion of labels across the dataset	Balanced	-

Experimental setup

The experiments were conducted on a MacBook Pro powered by the Apple M3 Pro chip, featuring a 11-core CPU, 14-core GPU, and a 16-core Neural Engine, offering enhanced computational power and parallel processing capabilities. The code was executed in a Python environment, version 3.12.7, taking advantage of the chip's high-performance architecture for optimized processing and execution.

- **Image pre-processing**: All images were resized to a standard resolution of 224x224 pixels to ensure consistency across the dataset. Image pre-processing functions specific to each model (ResNet50, MobileNetV2, EfficientNetB0) were applied to prepare the images in accordance with the model's requirements.

- **Feature Extraction**: Three pre-trained models from TensorFlow's Keras library—ResNet50, MobileNetV2, and EfficientNetB0—were utilized to extract high-dimensional feature vectors from each image. Each model was initialized with weights pre-trained on ImageNet and set to exclude the final fully connected layers (`include_top=False`) to use the extracted features for similarity calculations. The features were flattened into 1-dimensional vectors for further processing.

- **Metric Learning**: The aim of metric learning in this study was to learn a feature space that enhances the model's ability to distinguish between similar and dissimilar dog nose prints. This was achieved using a triplet loss function, which forces the network to minimize the distance between similar images (anchor-positive pairs) and maximize the distance between dissimilar images (anchor-negative pairs). The triplet loss function is given by:

$$L_{\text{triplet}} = \max(d(f(x_{\text{anchor}}), f(x_{\text{positive}})) - d(f(x_{\text{anchor}}), f(x_{\text{negative}})) + \alpha, 0)$$

  where:

  - $f(x)$ represents the feature extraction function, mapping an image x to a high-dimensional embedding.
  - $d(\cdot, \cdot)$ denotes the Euclidean distance between two feature vectors.
  - x_{anchor} is the anchor image, x_{positive} is a similar image (from the same dog), and x_{negative} is a dissimilar image (from a different dog).
  - α is a margin parameter that ensures the negative pair is sufficiently distant from the anchor image.

The objective of this loss function is to ensure that the feature vectors of similar images are close together, while the feature vectors of dissimilar images are far apart in the embedding space. This approach is particularly effective in the context of image matching tasks, where the goal is to associate images that depict the same object (dog nose) while distinguishing them from other objects. Additionally, hard negative mining was applied during training, which involves selecting negative samples that are most challenging for the model to classify, thus providing more informative gradients during training.

- **Similarity Calculation**: The main metric for evaluating feature vector similarity was cosine similarity. To find the most comparable match, the similarity score between the characteristics of a chosen validation image and every image in the training dataset was calculated for each model. As seen in Figure 5, each model's maximum cosine similarity score and matching picture ID were noted.

  In this context, Boundry represents the threshold between similarity and dissimilarity for the feature vectors. The line, as illustrated in Figure 2, distinguishes between vectors that are considered similar (within a certain angle or threshold) and those that are not. For instance, feature vectors x_1 and x_2, with angles θ_1 and θ_2 relative to the reference vector, lie close to or within the , suggesting higher similarity. In contrast, vector x_3, which exceeds the , represents a lower similarity due to the larger angle. This boundry helps in defining the acceptable range for cosine similarity scores when matching images.

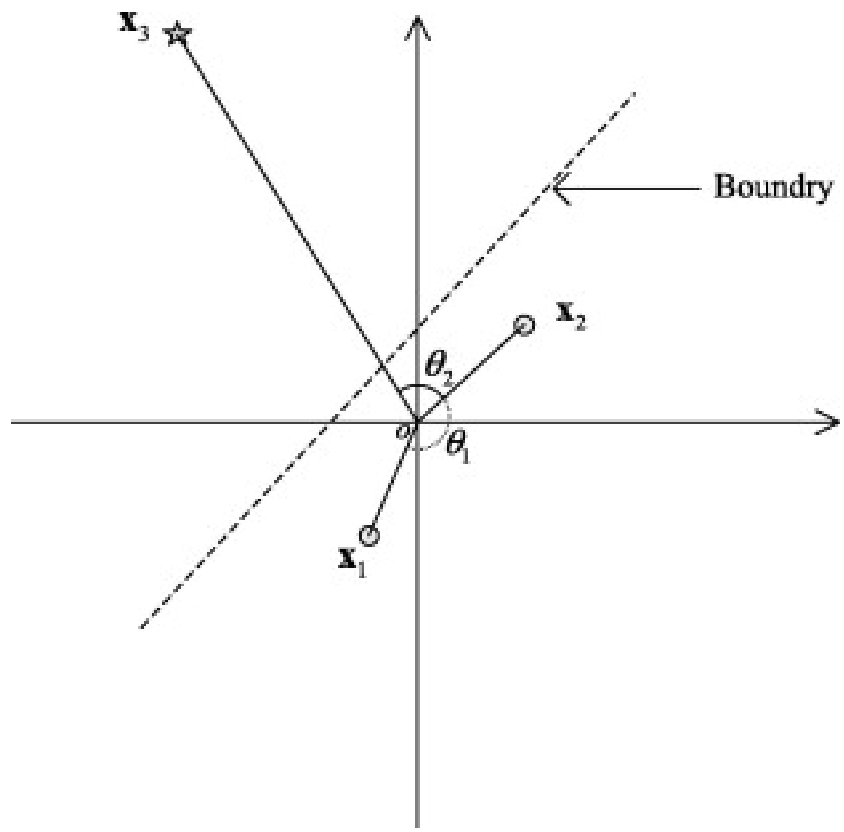

Figure 5 Cosine Similarity (Xia et al. 2015).

Evaluation and Figure Analysis

The Figure 6 illustrates the cosine similarity heatmap between 200 feature vectors extracted from images using ResNet50. Each feature vector corresponds to a unique image, and the similarity scores are displayed as a color gradient ranging from dark blue (low similarity) to

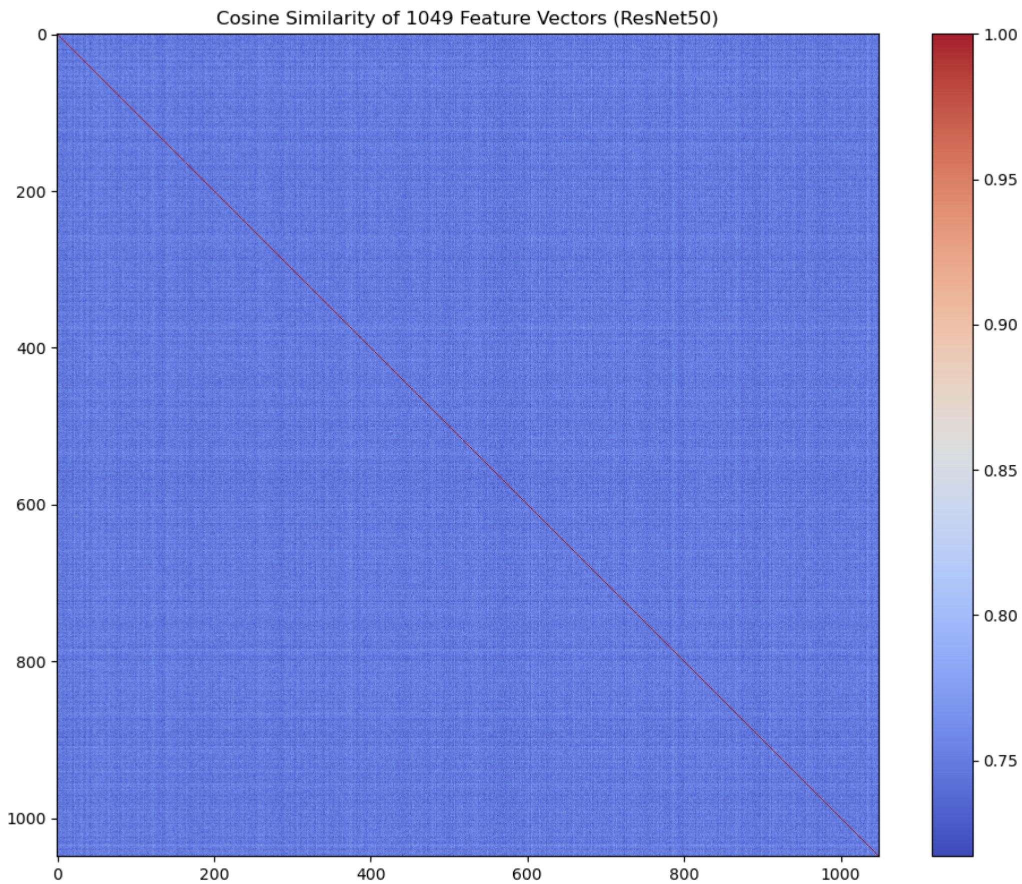

Figure 6: Cosine similarity heatmap of 200 feature vectors extracted using ResNet50. Each entry represents the similarity score between two images, ranging from 0.75 (least similar) to 1.00 (most similar).

red (high similarity).

The diagonal (in red) represents the self-similarity of each image, with a score of 1.00, as every image is identical to itself. The off-diagonal elements indicate the similarity between different image pairs. Darker blue shades represent lower similarity, suggesting effective differentiation between images, while lighter shades signify higher similarity, potentially indicating overlap or misclassification risk.

For model evaluation, the most similar image (highest cosine similarity) was identified for each query and compared with the expected matching image from the validation set. The model's ability to accurately recognize individual dogs based on their unique nose patterns was assessed by reporting the cosine similarity scores and the image IDs of the closest matches. This visualization provides insights into the model's effectiveness in distinguishing individual dogs and highlights areas for potential improvement.

5. Results

In this section, we evaluate the performance of three deep learning models—ResNet50, MobileNetV2, and EfficientNetB0—based on their accuracy in identifying individual dog noses. The accuracy results for each model are presented in Figure 7.

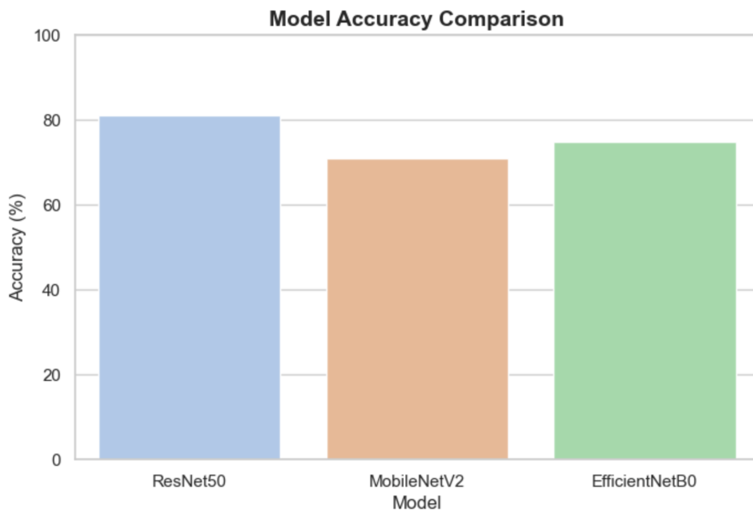

Figure 7: Model Accuracy Comparison: The accuracy (%) of each model in identifying individual dog noses is displayed.

The accuracy values for each model are as follows:

- **MobileNetV2**: Recorded an accuracy of approximately 71%. While slightly lower than the other models, MobileNetV2's lightweight architecture and efficiency make it ideal for deployment in mobile applications, particularly in resource-constrained and densely populated urban environments. Its compact design enables fast computations, making it a practical choice for real-time dog identification in crowded cities with limited computational resources.

- **ResNet50**: Achieved an accuracy of around 82%. Although it performs slightly better in terms of accuracy, ResNet50's deeper architecture requires more computational power, making it less suited for mobile and low-resource applications.

- **EfficientNetB0**: Reached an accuracy of approximately 78%. EfficientNetB0 provides a good balance between efficiency and accuracy; however, it is still more resource-intensive compared to MobileNetV2.

To put these results in context, we compared our model accuracy with those reported in a similar study by Shen et al. 2022. Table 4 presents a comparison of model accuracies from both studies. Although the models in Shen et al. 2022. achieved higher accuracies overall, MobileNetV2 stands out in our study as an optimal choice for building deployable applications in densely populated and resource-limited cities, due to its lightweight design.

Table 4: Model Accuracy Comparison, comparing with models trained by Shen et al. 2022 and other experiments.

Model	**Accuracy (%) - Our Experiment**	**Accuracy (%) - Shen et al.**
MobileNetV2	71	-
EfficientNetB0	78	84
ResNet50	82	86
Swin-transformer-base	-	89.8
ConvNet-base	-	90.3

Table 5 compares results obtained by this study to other, existing results whereas Table 6 compares the inference time for each image.

Experiment	**Accuracy**
Experiment conducted in this study	71.0%
Kang et al.	87.2%
Jarraya et al.	86.2%
Fikri et al.	82.0%
Himel et al.	84.3%

Table 5: Comparative analysis of results

Model	Inference Time (per image)
ResNet50	6 ms
MobileNetV2	3 ms
EfficientNetB0	5 ms

Table 6: Inference times for different models used in the proposed system.

Overall, MobileNetV2 showed a distinct edge overall. Even while it might not have had the best accuracy, its small size and effective design made it perfect for practical use, particularly in busy areas where access to powerful hardware is scarce. This makes it possible to create useful, scalable dog identification apps that can be used on mobile devices, increasing usability and accessibility.

6. Conclusion and Future Scope

In the context of dog nose identification, this study assessed the performance of three deep learning models: ResNet50, MobileNetV2, and EfficientNetB0 when passed through the proposed metric learning system. Higher accuracy levels were attained by ResNet50 and EfficientNetB0, but MobileNetV2 stood out as a viable option because of its efficient and lightweight design, which makes it especially well-suited for mobile deployment. With an accuracy of roughly 71%, MobileNetV2 shows that deep learning may be used to develop scalable and useful identification systems even in contexts with restricted resources, such crowded cities with little processing power. The potential of leveraging MobileNetV2 as a basis for creating deployable, real-time dog recognition applications is highlighted by this study.

Although MobileNetV2 strikes a balance between accuracy and efficiency, there is room to increase its precision even further to make it even more useful in practical applications. Future research could focus on the following methods to enhance MobileNetV2's performance through additional image pre-processing techniques:

- **Advanced Image Pre-processing Techniques**: Applying more sophisticated pre-processing techniques, such as histogram equalization and contrast adjustment, could improve the visibility of unique nose patterns, making it easier for the model to learn fine details in the images.

- **Model Architecture Optimization**: Experimenting with different neural network architectures or integrating attention mechanisms could enhance the model's ability to focus on the most distinctive regions of the nose patterns, potentially improving accuracy in identifying individual dogs.

Further studies could investigate the use of transfer learning with MobileNetV2, enhancing its feature extraction capabilities by fine-tuning it on a sizable dataset of dog photos with annotated nose prints, in addition to pre-processing techniques. The performance of MobileNetV2 in dog nose identification tasks could be further optimized by improving the training methods and the quality of the data, possibly reaching accuracies that are more comparable to more sophisticated models while maintaining the benefits of lightweight technology. Because of this, it would be a perfect fit for real-world uses in low-resource and mobile canine identification systems.

References

[1] F. Shen, Z. Wang, Z. Wang, X. Fu, J. Chen, X. Du, and J. Tang, "A Competitive Method for Dog Nose-print Re-identification," *arXiv preprint arXiv:2205.15934*, 2022.

[2] M. K. Cho and T. S. Kim, "Canine Biometric Identification Using ECG Signals and CNN-LSTM Neural Networks," 2021.

[3] H. B. Bae and others, "Dog Nose-Print Identification Using Deep Neural Networks," 2020.

[4] S. Chakraborty and others, "Visual Animal Biometrics: Survey," 2020.

[5] S. Kumar and others, "Investigation on the Muzzle of a Pig as a Biometric for Breed Identification," 2020.

[6] H. S. Kühl and T. Burghardt, "Animal Biometrics: Quantifying and Detecting Phenotypic Appearance," 2013.

[7] N. Coldea, "Nose Prints as a Method of Identification in Dogs," 1994.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[9] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[10] J. Duyck and others, ”Sloop: A Pattern Retrieval Engine for Individual Animal Identification,” 2015.

[11] A. Noviyanto and A. M. Arymurthy, ”Beef Cattle Identification Based on Muzzle Pattern Using a Matching Refinement Technique in the SIFT Method,” 2013.

[12] M. F. Hansen and others, ”Towards On-Farm Pig Face Recognition Using Convolutional Neural Networks,” 2018.

[13] L. Gamble and others, ”The Problem of Stray Dogs,” *Revue scientifique et technique (International Office of Epizootics)*, vol. 37, no. 2, pp. 543–550, 2018.

[14] H. S. Kühl and T. Burghardt, ”Animal Biometrics: Quantifying and Detecting Phenotypic Appearance,” *Trends in Ecology Evolution*, vol. 28, no. 7, pp. 432–441, 2013.

[15] A. I. Awad, ”From Classical Methods to Animal Biometrics: A Review on Cattle Identification and Tracking,” *Computers and Electronics in Agriculture*, vol. 123, pp. 423–435, 2016.

[16] E. Lancaster, J. Rand, S. Collecott, and M. Paterson, ”Problems Associated with the Microchip Data of Stray Dogs and Cats Entering RSPCA Queensland Shelters,” *Animals*, vol. 5, pp. 332–348, 2015. doi:10.3390/ani5020332.

[17] H. I. Choi, M.-Y. Kim, H.-Y. Yoon, and others, "Study on the Viability of Canine Nose Pattern as a Unique Biometric Marker," *Animals*, vol. 11, no. 3372, 2021. doi:10.3390/ani11123372.

[18] M. Sandler and others, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[19] M. Tan, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *arXiv preprint arXiv:1905.11946*, 2019.

[20] P. Xia, L. Zhang, and F. Li, "Learning Similarity with Cosine Similarity Ensemble," *Information Sciences*, vol. 307, pp. 39–52, 2015.

[21] F. Shen and others, "A Competitive Method for Dog Nose-print Re-identification," *arXiv preprint arXiv:2205.15934*, 2022.

[22] Four Paws. (2024). Stray animals – pets without a home. Retrieved November 17, 2024, from https://www.four-paws.org/campaigns-topics/topics/help-for-stray-animals/stray-animals-pets-without-a-home

[23] NCBI. (2024). Stray dog population management. Retrieved November 17, 2024, from https://pmc.ncbi.nlm.nih.gov/articles/PMC10570566/

[24] Kühl HS, Burghardt T. Animal biometrics: quantifying and detecting phenotypic appearance. Trends Ecol Evol. 2013 Jul;28(7):432-41. doi: 10.1016/j.tree.2013.02.013. Epub 2013 Mar 25. PMID: 23537688.

[25] Deb, D., Wiper, S., Russo, A., Gong, S., Shi, Y., Tymoszek, C., Jain, A. (2018). Face recognition: Primates in the wild. arXiv:1804.08790.

[26] Gulzar, Yonis. "Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique." Sustainability, vol. 15, no. 3, 2023, p. 1906. doi:10.3390/su15031906.

[27] Gomes, Rahul, et al. "A Comprehensive Review of Machine Learning Used to Combat COVID-19." Diagnostics, vol. 12, no. 8, 2022, p. 1853. doi:10.3390/diagnostics12081853.

[28] Montalbo, Francis Jesmar, and Alvin S. Alon D.Eng. "Empirical Analysis of a Fine-Tuned Deep Convolutional Model in Classifying and Detecting Malaria Parasites from Blood Smears." KSII Transactions on Internet and Information Systems, vol. 15, no. 1, 2021, pp. 147-165. doi:10.3837/tiis.2021.01.009.

[29] Kang, Myungjin. "Okeydoggy: A Mobile Application That Helps Understand Calming Signals of Dog Using MOBILENETV2—Focusing on "Play Bowing" and "Licking Nose". Advances in Artificial Intelligence and Machine Learning. 2024; 4 (2): 130." Age, 2024,

[30] Jarraya, Islem, et al. "A new convolutional neural network based on a sparse convolutional layer for animal face detection." Multimedia Tools and Applications 82.1 (2023): 91-124.

[31] Fikri, Ahmad, and Aniati Murni. ”New Generation Indonesian Endemic Cattle Classification: MobileNetV2 and ResNet50.” Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI) 9.4 (2023): 941-950.

[32] Himel, Galib Muhammad Shahriar, and Md Masudul Islam. ”Benchmark Analysis of Various Pre-trained Deep Learning Models on ASSIRA Cats and Dogs Dataset.” arXiv preprint arXiv:2401.04666 (2024).