Online ISSN: 3104-9338
Print ISSN: 3104-932X

Open-set Goat Face Recognition with MobileFaceNet Adaptation for Livestock Farming

Chenlin Li*, Zhibo Liu

Henan Polytechnic University

Received: October 10, 2025

Revised: October 13, 2025

Accepted: October 16, 2025

Published online: October 16, 2025

To appear in: *International Journal of Advanced AI Applications,* Vol. 1, No. 7 (November 2025)

* Corresponding Author: Chenlin Li (212309020033@home.hpu.edu.cn)

Abstract. In the context of intelligent livestock farming, precise individual animal identification has become a critical requirement. To address the challenges posed by high facial similarity among black goats and the need for model retraining in incremental learning scenarios, this paper introduces GoatFaceNet, a lightweight goat face recognition model based on MobileFaceNet. GoatFaceNet incorporates MixConv to enhance its feature extraction capability. Furthermore, the CurricularFace loss function is employed to improve inter-class separability and intra-class compactness, thereby increasing the model's robustness in open-set recognition tasks. A comprehensive goat face dataset was constructed to support the evaluation. Experimental results show that GoatFaceNet achieves an accuracy of 94.2% on the test set. Additional evaluations involving 3,200 goat face pairs confirm the model's superior open-set discrimination performance, validating its practical applicability and deployment potential in real-world farming environments.

Keywords: Black Goat, Lightweight Model, High Similarity Recognition;

1. Introduction

Goat farming needs reliable individual identification for health monitoring, end-to-end traceability, and daily herd management. Manual checks, branding, and RFID ear tags are common and quick to apply. Ear tags may cause ear infections and can fall off or break in practice [1]. Iris scan is expensive and difficult to deploy at scale. Goat face recognition is non-contact and low-cost. It is attracting growing interest for contact-free identification on farms.

Early face recognition used geometric features. It needed manual feature engineering and did

not generalize well. Corkery et al. [2] studied sheep face recognition with independent component analysis on whole-face images, then evaluated a pre-trained classifier. Deep learning changed the field. Convolutional neural networks became standard for hierarchical feature extraction in human face recognition. DeepFace [3] reported high accuracy on the LFW dataset. Practical systems need fast inference and low compute. This pushed research toward lightweight models such as MobileNet [4,5] and ShuffleNet [6,7]. Face-oriented variants include MobileFaceNet [8] and ShuffleFaceNet [9]. Margin losses such as ArcFace [10] enlarge inter-class separation. These designs run on embedded and edge devices, supporting transfer learning for animal identification.

In animal face recognition, Zhang et al. [11] integrated an ECCSA spatial-attention module into MobileFaceNet for sheep-face recognition, achieving 96.73% accuracy. Xu et al. [12] proposed a two-stage lightweight cattle-face recognition framework, using ArcFace loss for robust feature extraction. In closed-set scenarios, ArcFace improved classification accuracy by 11% compared to Softmax. Billah et al. [13] developed an automated recognition framework for dairy goats and attained 93% accuracy under closed-set conditions without extensive data preprocessing. Zhang et al. [14,15] explored sheep face recognition from multiple perspectives, proposing a multi-view fusion method to enhance robustness to pose variation and a Siamese-based approach to improve generalization under limited training samples. Li et al. [16] proposed an optimized lightweight network structure that balances recognition accuracy and model complexity, making it suitable for deployment in resource-constrained farm environments. Porting human face recognition models to animal face recognition often yields strong performance on closed-set training. In real farm scenarios, however, short breeding cycles together with selling cause herd fluctuations; when the population changes incrementally, the classifier requires retraining; misclassification risk also rises under atypical husbandry conditions. Hence, open-set recognition becomes necessary. Sheep face recognition further suffers from class imbalance, limited data acquisition, and small dataset size. This study focuses on black goats with uniformly dark coats, weakly distinctive facial cues, and high inter-individual similarity; existing models struggle to learn discriminative features effectively [17], posing a challenge to overall accuracy.

In summary, aiming to improve goat-face recognition accuracy in open-set scenarios, this paper enhances the MobileFaceNet architecture to achieve higher precision without significantly increasing parameter count. A specialized loss function is employed to enlarge the decision boundaries between classes. The main contributions of this research are as follows:

1. Two datasets were constructed to evaluate the performance of facial recognition methods on Black Goats under both closed-set and open-set conditions;

2. Design a new Mix block to strengthen feature representation capabilities, thus reducing misclassification rates in open-set scenarios;

3. Introduce a practical two-stage goat-face recognition method, enabling the recognition of new individuals immediately after their registration, without necessitating retraining of the entire model.

2. Dataset

The goat face data were collected on a farm in Liuyang, Hunan Province. To ensure diversity, 126 Black Goats were randomly selected for video recording under various weather conditions. During recording, a handheld camera tracked each goat's frontal face while operators varied angles using a mobile device; camera resolution was 1920×1080 pixels, yielding 126 video sequences. Because consecutive frames are highly redundant, directly using them may cause overfitting. Each video stream was first processed with OpenCV to extract one frame every 15 frames. Extracted frames were filtered using SSIM; blurred or otherwise invalid images were manually removed, resulting in 3,167 raw images. These images are divided into training and testing sets in a ratio of 8:2.

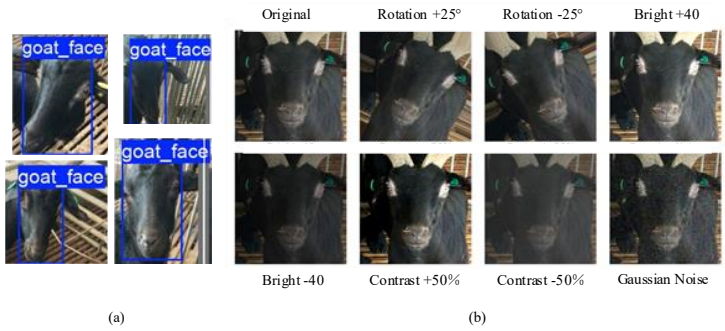

Figure 1. (a) Detection results. (b) Data augmentation examples.

Both data collection and application were single-target scenes. Goat face detection was performed using YOLOv8s; some detection results are shown in Figure 1(a). The detected goat face images were cropped and resized to 112×112 pixels. A random data augmentation strategy is adopted, including random horizontal flipping, rotation, changing brightness and contrast, adding salt and pepper noise, and Gaussian noise, as shown in Figure 1(b). Each goat's image count was augmented to 50 to ensure balanced distribution. In total, 6,300 goat-face images

were obtained, forming the GF train set. The number of goats and images included in the dataset are shown in Table 1.

Additionally, we collected facial images from 84 black goats, obtaining 537 samples in total. We created 1,600 positive pairs and 1,600 negative pairs, forming 3,200 pairs with a balanced ratio of positive and negative samples. This paired dataset, named GF-open, was used specifically to evaluate the generalization capability of the model under open-set recognition conditions.

Table 1. Summary of the datasets used in this study.

Dataset	Goats	Images
GF train set	126	6300
GF test set	126	633
GF-open set	84	537

3. Method

3.1. GoatFaceNet

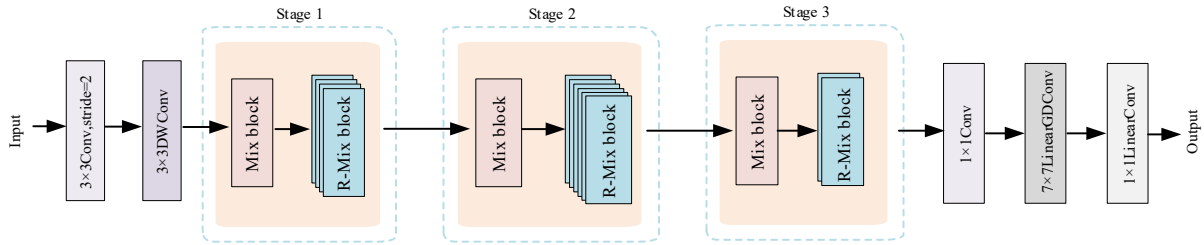

Figure 2. Architecture of GoatFaceNet.

This study is based on MobileFaceNet, a lightweight face-recognition method that uses the bottleneck block from MobileNetV2 to enhance nonlinear expressivity. To improve robustness and generalization for goat face recognition, we designed GoatFaceNet, which retains the inverted residual structure of MobileNetV2. The overall architecture is shown in Figure 2. Initially, GoatFaceNet extracts edge features from input images. Subsequently, depthwise separable convolutions are applied to reduce computational complexity while preserving crucial feature information. The subsequent three stages each consist of multiple Mix blocks, designed for hierarchical feature extraction and refinement. When the dimensions of inputs and outputs match, R-Mix blocks with residual connections are utilized to mitigate gradient vanishing issues. Following feature extraction, a linear global depthwise convolution (LinearGDConv) replaces global average pooling to generate feature representations. Finally, a linear 1×1 convolution compresses the resulting high-dimensional features into a compact

128-dimensional embedding vector, representing each goat face.

3.2. Mix Block

Depthwise convolution is widely used in lightweight network architectures to significantly reduce computational complexity by independently performing convolutions on each channel. However, employing kernels of a single size limits the ability of the model to capture both fine local details and broader global structures. To enhance multi-scale spatial representation, we introduce the MixConv [18], inspired by multi-scale convolutions in the Inception [19] architecture, as illustrated in Figure 3. To minimize computational redundancy, MixConv avoids using multiple parallel branches. Instead, it combines kernels of different sizes (e.g., 3×3, 5×5, and 7×7) within a single depthwise convolution operation. Specifically, feature maps with c channels are divided evenly into g groups, each containing c/g channels. Each group is convolved separately using a kernel of a designated size, with weights denoted as $w^{(t)}$, where $t \in [1, g]$ indexes the group number. The outputs from all groups are then concatenated along the channel dimension to form a unified feature map $y \in R^{h \times w \times c}$. This approach enables multi-scale feature extraction with reduced computational cost, thereby improving recognition performance in black goat face feature extraction tasks.

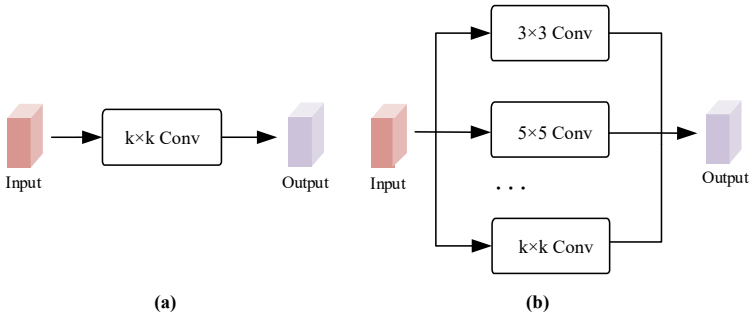

Figure 3. (a) Vanilla Conv; (b) Mix Conv.

Previous recognition tasks [20,21] have demonstrated that MixConv can improve model performance. Based on these findings, we design a novel Mix block module (see Figure 4) to exploit multi-scale receptive fields further. Input channels are evenly divided into groups, each group undergoes depthwise separable convolutions with kernel sizes of 3×3, 5×5, 7×7, as well as a parallel 1×1 convolution. Outputs from these convolutions are concatenated along the channel dimension, effectively balancing multi-scale receptive fields without substantially increasing parameter count. To maintain consistent spatial dimensions after downsampling, appropriate padding is applied for each kernel branch. The Swish activation function is used for richer feature representation, and an SE module is embedded to adaptively recalibrate channel

responses [22,23]. Finally, a channel shuffle operation is included to disrupt fixed group boundaries, promoting better interaction and representation among feature channels.

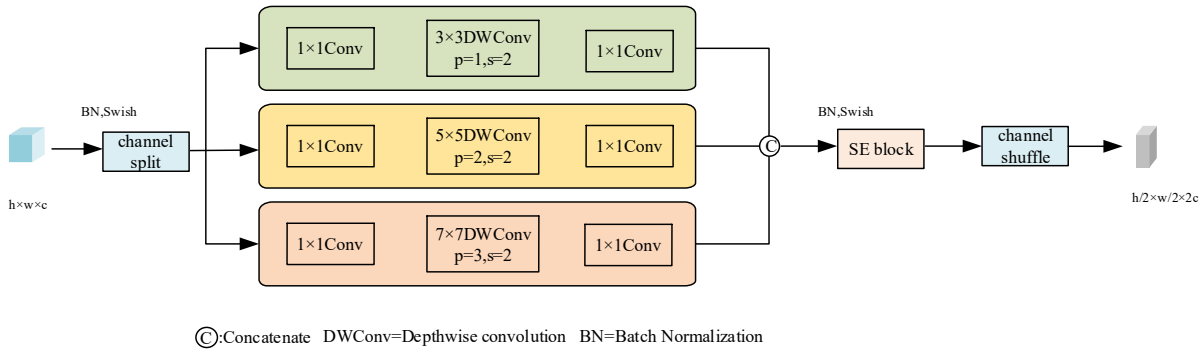

Figure 4. Mix block

3.3. R-Mix Block

When the input and output channel dimensions are equal, we add a residual connection to create the R-Mix Block, as shown in Figure 5. This residual connection involves element-wise addition between the input feature and the output of the multi-scale convolutions, thereby preventing information loss during feature transformation. This mechanism mitigates gradient vanishing and network degradation in deeper layers. Consistent with the inverted residual structure, nonlinear activations are applied only in intermediate layers, preserving a linear mapping between input and output. This strategy effectively balances representational power and training stability without significantly increasing structural complexity.

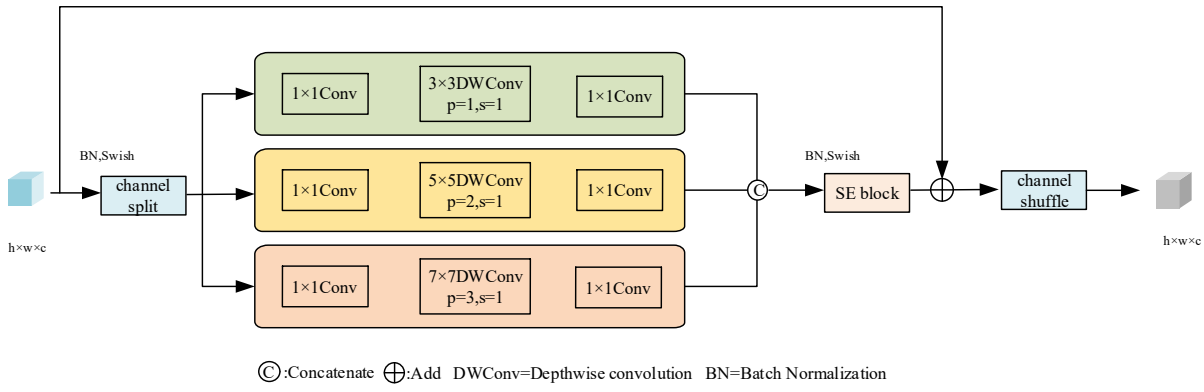

Figure 5. R-Mix block with residual connection.

3.4. Loss Function

In goat face recognition, the ideal case is that the embedding features learned by the model exhibit good separability in the vector space. Although the Softmax loss function is commonly used for classification tasks, it has inherent limitations in minimizing intra-class distances and expanding inter-class margins. To address these limitations and enhance the discriminability of

embedding features, researchers have developed various angular-margin–based loss functions, which can generally be described under a unified Softmax loss framework:

$$L = -\mathrm{G}(\mathrm{p}(x_i))\log\frac{e^{s(cos\theta_{y_i})}}{e^{s(cos\theta_{y_i})} + \sum_{j=1,j\neq y_i}^{n} e^{s(cos\theta_j)}} \tag{1}$$

where x_i denotes the embedding feature of the i-th sample, y_i is its corresponding class label, and $\mathrm{p}(x_i)$ refers to the predicted probability of assigning x_i to class y_i. $G(\cdot)$ is an indicator function that equals 1 in the case of ArcFace. The weight vector $\mathrm{W}_j \in \mathrm{R}^d$ denotes the j-th column of the weight matrix $\mathrm{W} \in \mathrm{R}^d$. $\mathrm{b}_j \in \mathrm{R}$ represents the bias term. The total number of classes is denoted as n.

Within this unified framework, margin-based loss functions differ primarily in how they adjust cosine similarities for positive and negative classes. ArcFace introduces a margin directly into the positive-class adjustment function, defined as $T(cos\theta_{y_i}) = \cos(\theta_{y_i} + m)$, bringing same-class embeddings closer together. However, ArcFace treats all samples equally and does not consider variations in sample difficulty. During training, some negative-class samples exhibit very high cosine similarity. These hard samples are easy to misclassify. MV-Arc-Softmax [24] addresses this issue by introducing a fixed coefficient t to amplify the negative-class cosine similarities of these hard samples, thereby increasing their loss weights. However, due to the fixed value of t, this method struggles to converge effectively early in training and cannot dynamically shift its focus to new hard samples later in training.

This study adopts the CurricularFace [25] loss function, which dynamically adjusts the parameter t throughout training, allowing the model to progressively shift focus from easy to more difficult samples. Initially, the training emphasizes easy samples with smaller values of t^k. As training progresses, t^k gradually increases, shifting the model's focus to harder samples, thus enhancing feature discrimination. While the positive-class adjustment remains consistent with ArcFace, the negative-class adjustment function is defined as follows:

$$N(t, cos\theta_j) = \begin{cases} cos\theta_j, & T(cos\theta_{y_i}) - cos\theta_j \geq 0 \\ cos\theta_j(t + cos\theta_j), & T(cos\theta_{y_i}) - cos\theta_j < 0 \end{cases} \tag{2}$$

where t^k is the curriculum parameter at iteration k. In the early stages of training, t^k is relatively small and approaches 0, resulting in $t^k + cos\theta_j < 1$. This suppresses the weight of hard samples and emphasizes easy ones. As training progresses, t^k gradually increases, making $t^k + cos\theta_j > 1$, which shifts the model's focus toward hard samples and enhances

discriminative ability. An exponential moving average (EMA) is used to estimate t^k where r^k denotes the mean cosine similarity of positive samples in iteration k, and α is the smoothing factor (default set to 0.99):

$$t^k = \alpha r^k + (1-\alpha)t^{k-1} \tag{3}$$

the complete loss function of CurricularFace is defined as follows:

$$L = -\log \frac{e^{s(cos\theta_{y_i}+m)}}{e^{s(cos\theta_{y_i}+m)} + \sum_{j=1,j\neq y_i}^{n} e^{sN(t^k,cos\theta_j)}} \tag{4}$$

This method compresses the intra-class variations that introduce noisy features while enlarging inter-class distances, thereby improving the discriminative capacity of the embedding features. It enables the model to maintain a low false acceptance rate (FAR) even when encountering unregistered identities.

4. Experiments and Analysis

4.1. Experimental Settings and Evaluation Metrics

The experiments in this study were conducted using a Windows 10 operating system, NVIDIA GeForce RTX 4060 GPU, and 32 GB of memory. PyTorch 2.3.1 was used as the deep learning framework, running under Python 3.8. The stochastic gradient descent (SGD) optimizer was employed with an initial learning rate of 0.01, momentum of 0.9, and weight decay set at 0.0001. The model was trained for 50 epochs, and the learning rate was adjusted using a cosine annealing schedule.

Evaluation metrics for the goat face recognition model included accuracy, precision, recall, F1-score, and model parameters. Accuracy refers to the proportion of correctly classified samples relative to all samples. Precision measures the proportion of true positives among all samples predicted as positive, while recall measures the proportion of actual positive samples correctly identified by the model. The F1-score, calculated as the harmonic mean of precision and recall, provides a balanced evaluation of model performance. The number of parameters indicates the model's complexity, with fewer parameters generally indicating better suitability for resource-constrained deployments. The following formulas depend on the values of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$precision = \frac{TP}{TP + FP} \tag{6}$$

$$recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{8}$$

4.2. Two-stage recognition method

To achieve accurate automatic recognition of individual goats, we propose a two-stage recognition method. Firstly, goat face regions are annotated using LabelImg, and a YOLOv8s model is trained to detect and crop goat faces. This YOLOv8s detector is later employed in the final application. In the training stage, the known categories from the GF train set are used. Subsequently, inference is performed on images in the GF test set to evaluate closed-set recognition performance. Additionally, the GF-open dataset simulates an open-set recognition scenario to test the model's generalization capability. The entire workflow is illustrated in Figure 6. By minimizing computational load during recognition, this method ensures suitability for deployment on edge devices with limited resources.

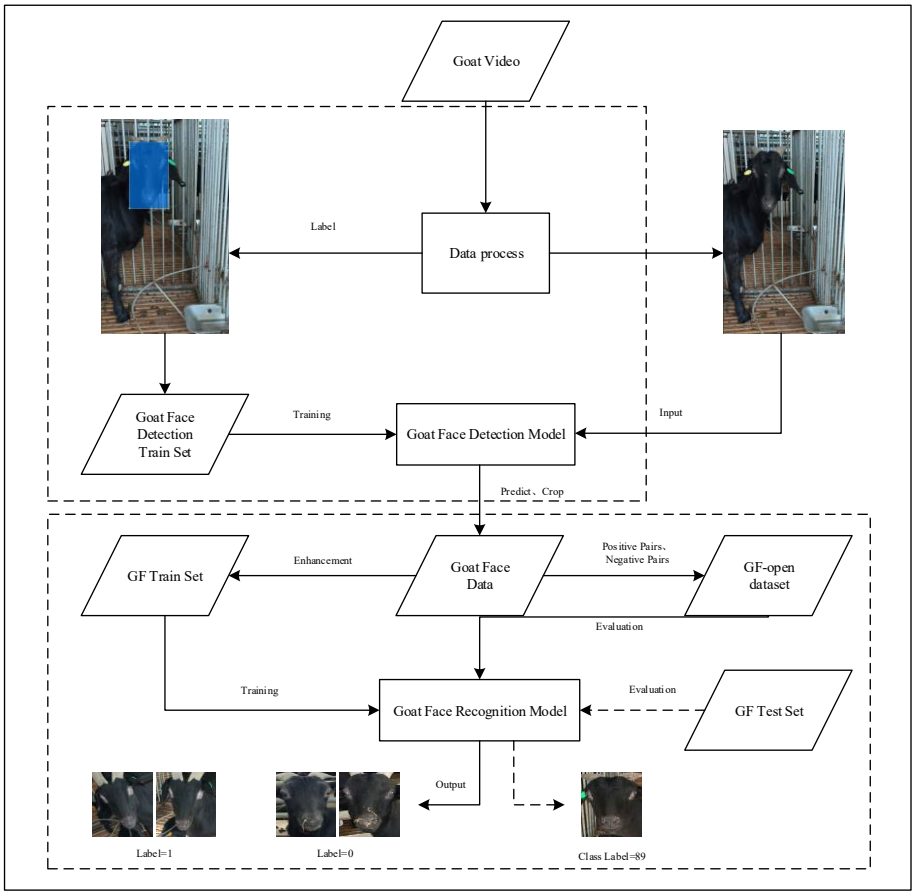

Figure 6. Two-stage recognition method.

4.3. Comparison with Baseline Models

Based on transfer learning principles, the officially released pre-trained weights were used for model initialization during training. Trends of training accuracy and loss are displayed in Figure 7. To evaluate the effectiveness of our proposed GoatFaceNet architecture, comparisons were made with mainstream lightweight models, including MobileFaceNet, ShuffleNetV2, MobileNetV2, and VarGFaceNet [26]. All models were trained using the CurricularFace loss function with hyperparameters s=64.0, m=0.5, smoothing factor $\alpha = 0.99$, and an initial curriculum parameter t_0=0. Predictions were made by selecting the class with the highest predicted probability.

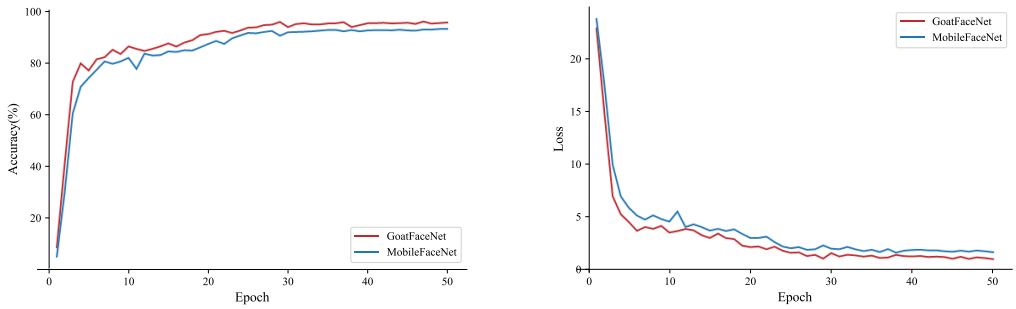

Figure 7. Training accuracy curves and loss curves.

The final metrics for each model on the test set, shown in Table 2, quantify their classification performance. The MobileViT-S [27] model performed poorly in the test, with an accuracy rate of 89.7%. Compared with MobileFaceNet, the proposed GoatFaceNet achieved a 1.9% improvement in accuracy without a significant increase in the number of model parameters. When compared to classical lightweight models such as MobileNetV2 and ShuffleNetV2, GoatFaceNet showed more than a 5% increase in accuracy while maintaining a similar parameter count. Additionally, GoatFaceNet achieved the highest F1-score, further confirming its superior discriminative ability and efficiency. By explicitly handling difficult samples through the CurricularFace loss function, the model optimizes boundary samples more effectively during training, thus achieving improved accuracy during testing.

Table 2. Comparison of results between GoatFaceNet and other models.

Model	Accuracy	Precision	Recall	F1-score	Parameters/M
MobileNetV2	89.1	90.9	89.9	90.40	3.5
ShuffleNetV2	88.7	90.7	89.5	90.10	2.4
MobileFaceNet	92.3	94.0	93.7	93.85	*1.0*
VarGFaceNet	93.9	94.6	93.4	94.00	5.0
MobileViT-S	89.7	91.5	89.3	90.39	5.6
ResNet-50	88.9	90.4	89.6	89.95	25.6
GoatFaceNet	*94.2*	94.8	93.6	*94.20*	1.1

4.4. Effect of Different Loss Functions

This section also compares the performance of different loss functions in the goat face recognition task, specifically ArcFace, MV-Arc-Softmax, and CurricularFace. Furthermore, the generalization capability of the proposed method was evaluated using a public dataset [13], which contains facial images of 10 dairy goats. The original method proposed by the dataset authors achieved an accuracy of 93% without additional preprocessing.

On the GF test set, MobileFaceNet achieved an accuracy of 90.3% with ArcFace, which increased to 92.0% with MV-Arc-Softmax, and further improved to 92.3% when using CurricularFace. These results suggest that applying stronger margin constraints can significantly enhance the discriminative capacity of learned features by placing greater emphasis on correctly classifying hard samples.

Using ArcFace, GoatFaceNet outperformed baseline by 3.2% on the GF test set and by 3.4% on the public dataset. When combining GoatFaceNet with CurricularFace, accuracy reached 94.2% on the test set, achieving the highest accuracy among all evaluated combinations. On the public dataset, accuracy improved to 93.3%. Although this represents only a modest 0.3% gain over the original method proposed by the dataset authors, it remains meaningful given the limited number of goats included.

These outcomes demonstrate that the enhanced multi-scale structure and optimized feature extraction method synergistically reinforce the benefits of CurricularFace, thus significantly improving the model's generalization and accuracy.

Table 3. Accuracy (%) of different loss functions on different datasets.

Method	GF test set	Dataset [13]
MobileFaceNet+ArcFace	90.3	86.6
MobileFaceNet+MV-Arc-Softmax	92.0	89.7
MobileFaceNet+ CurricularFace	92.3	91.3
GoatFaceNet +ArcFace	93.5	90.0
GoatFaceNet + MV-Arc-Softmax	93.7	91.2
GoatFaceNet+ CurricularFace(ours)	94.2	93.3

4.5. Performance under Simulated Open-set Scenarios

To evaluate the model's generalization capabilities in incremental open-set recognition scenarios, additional experiments were conducted on the GF-open dataset, which includes 84 goats and comprises 3,200 image pairs for verification. Among these pairs, half (1,600) were positive pairs (images of the same goat labeled as 1), and half (1,600) were negative pairs (images of different goats labeled as 0).

In this experiment, normalized feature vectors were extracted from each image pair, and cosine similarity was computed between the vectors. A classification threshold, determined through grid search on the test set, was then applied to distinguish whether the two images represented the same goat or different individuals.

Results presented in Table 4 clearly illustrate that CurricularFace offers greater generalization capability compared to ArcFace. Specifically, adding CurricularFace increased the accuracy of MobileFaceNet from 73.3% to 79.4%, an improvement of 6.1%. GoatFaceNet with ArcFace achieved an accuracy of 78.9%, while combining GoatFaceNet with CurricularFace achieved the best performance at 84.5%. Overall, the structural improvements and adoption of CurricularFace effectively enhance the model's open-set recognition performance.

Table 4. Accuracy (%) of GF-open set.

Method	Accuracy
MobileFaceNet+ArcFace	73.3
MobileFaceNet+ CurricularFace	79.4
GoatFaceNet +ArcFace	78.9
GoatFaceNet+ CurricularFace	84.5

4.6. Attention Visualization

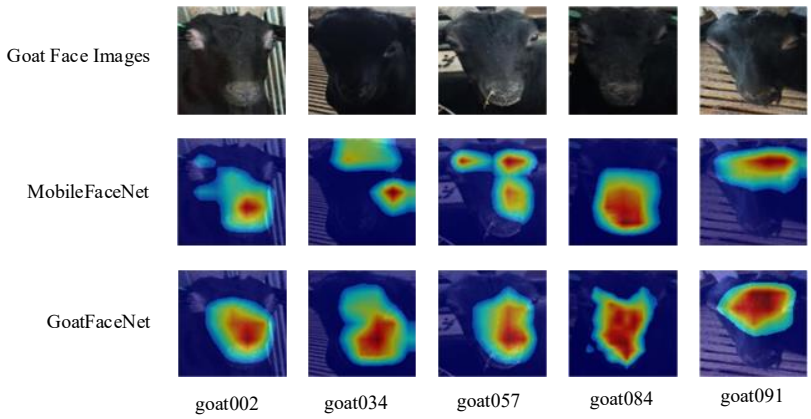

Figure 8. Heat maps of some goat faces.

While closed-set recognition can be framed as a classification task, open-set recognition fundamentally constitutes a metric learning challenge, as the model must robustly handle incremental unknown-class samples. This challenge is further intensified by the high similarity among black goat facial appearances, which increases the difficulty of accurate identification. This study employed Grad-CAM to illustrate the attention distributions of MobileFaceNet and GoatFaceNet during recognition tasks. Figure 8 shows the visualization results, where red

indicates higher activation and blue indicates lower activation for the predicted class. MobileFaceNet's attention is scattered across the image, whereas GoatFaceNet demonstrates a more focused activation pattern, indicating stronger feature-selection capabilities.

5. Conclusion

This study employs YOLOv8s for goat face detection and cropping before the recognition stage, significantly reducing computational requirements and making the entire recognition process suitable for resource-constrained edge devices. To address challenges associated with open-set recognition and the dynamic number of black goats, we improved MobileFaceNet by introducing mixed depthwise convolution and developing a novel Mix block. This new block achieves enhanced multi-scale feature extraction without significantly increasing model complexity. Additionally, adopting the CurricularFace loss function notably improves the model's capability to discriminate hard samples in open-set scenarios. The effectiveness of these improvements has been validated through additional open-set simulation experiments, and their applicability in non-contact identification in real-world farm settings has been confirmed.

The GF train set used in this study is relatively small. To enhance data diversity, we applied data augmentation techniques; however, these methods may introduce risks of overfitting and constrain the model's generalization capability. Despite this limitation, our experimental results demonstrate the effectiveness of the proposed approach. Nevertheless, we fully recognize that performance is inherently constrained by the scale and diversity of the training data. Zhang et al. [15] introduced a twin network-based recognition method that effectively improves model performance under low-data regimes by learning pairwise sample similarities. Motivated by this, we plan to collect larger-scale, high-quality datasets with natural distribution characteristics in future work and further investigate advanced strategies, such as few-shot learning, to mitigate data scarcity and strengthen model generalization.

References

[1] Ait-Saidi A, Caja G, Salama AAK, Carné S. Implementing electronic identification for performance recording in sheep: I. Manual versus semiautomatic and automatic recording systems in dairy and meat farms. J Dairy Sci. 2014;97:7505-7514.

[2] Corkery GP, Gonzales-Barron UA, Butler F, et al. A preliminary investigation on face recognition as a biometric identifier of sheep. Trans ASABE. 2007;50(1):313-320.

[3] Taigman Y, Yang M, Ranzato M, Wolf L. DeepFace: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014; Columbus, OH, USA. p. 1701-1708. doi:10.1109/CVPR.2014.220.

[4] Howard AG. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 2017.
[5] Sandler M, Howard A, Zhu M, et al. MobileNetV2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 4510-4520.
[6] Zhang X, Zhou X, Lin M, et al. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2018. p. 6848-6856.
[7] Ma N, Zhang X, Zheng HT, et al. ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 116-131.
[8] Chen S, Liu Y, Gao X, et al. MobileFaceNets: Efficient CNNs for accurate real-time face verification on mobile devices. In: Chinese Conference on Biometric Recognition (CCBR). Cham: Springer; 2018. p. 428-438.
[9] Martindez-Diaz Y, Luevano LS, Mendez-Vazquez H, et al. ShuffleFaceNet: A lightweight face architecture for efficient and highly accurate face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops; 2019.
[10] Deng J, Guo J, Xue N, et al. ArcFace: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 4690-4699.
[11] Zhang H.-M., Zhou L.-X., Li Y.-H., et al. Sheep Face Recognition Method Based on Improved MobileFaceNet. Transactions of the Chinese Society for Agricultural Machinery. 2022;53(05):267-274. (in Chinese)
[12] Xu B, Wang W, Guo L, et al. CattleFaceNet: A cattle face identification approach based on RetinaFace and ArcFace loss. Computers and Electronics in Agriculture. 2022;193:106675.
[13] Billah M, Wang X, Yu J, et al. Real-time goat face recognition using convolutional neural network. Computers and Electronics in Agriculture. 2022;194:106730.
[14] Zhang X, Xuan C, Ma Y, et al. An efficient method for multi-view sheep face recognition. Eng Appl Artif Intell. 2024;134:108697.
[15] Zhang X, Xuan C, Ma Y, et al. High-similarity sheep face recognition method based on a Siamese network with fewer training samples. Comput Electron Agric. 2024;225:109295.
[16] Li X, Zhang Y, Li S. Rethinking lightweight sheep face recognition via network latency-accuracy tradeoff. Comput Electron Agric. 2024;227:109662.
[17] Zhang X, Xuan C, Ma Y, et al. Lightweight model-based sheep face recognition via face image recording channel. J Anim Sci. 2024;102:skae066.
[18] Tan M, Le QV. MixConv: Mixed depthwise convolutional kernels. arXiv preprint arXiv:1907.09595. 2019.
[19] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 2818-2826.
[20] Zhang X, Wu X, Xiao Z, et al. Mixed-decomposed convolutional network: A lightweight yet efficient convolutional neural network for ocular disease recognition. CAAI Trans Intell Technol. 2024;9(2):319-332.
[21] Boutros F, Damer N, Fang M, et al. MixFaceNets: Extremely efficient face recognition networks. In: 2021 IEEE International Joint Conference on Biometrics (IJCB); 2021. p. 1-8.
[22] Ramachandran P, Zoph B, Le QV. Searching for activation functions. arXiv:1710.05941. 2017.
[23] Howard A, Sandler M, Chen B, et al. Searching for MobileNetV3. In: Proceedings of the

IEEE/CVF International Conference on Computer Vision (ICCV); 2019; Seoul, Republic of Korea. p. 1314-1324.
[24] Wang X, Zhang S, Wang S, et al. Mis-classified vector guided softmax loss for face recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2020;34(7):12241-12248.
[25] Huang Y, Wang Y, Tai Y, et al. CurricularFace: Adaptive curriculum learning loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 5901-5910.
[26] Yan M, Zhao M, Xu Z, et al. VarGFaceNet: An efficient variable group convolutional neural network for lightweight face recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops; 2019.
[27] Mehta S, Rastegari M. MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer. arXiv preprint arXiv:2110.02178. 2021.
[28] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 770-778.