agriculture

Article

When Mobilenetv2 Meets Transformer: A Balanced Sheep Face Recognition Model

Xiaopeng Li, Jinzhi Du, Jialin Yang and Shuqin Li *

College of Information Engineering, Northwest A&F University, Xianyang 712100, China; li_xiaopeng@nwafu.edu.cn (X.L.); 2021056013@nwafu.edu.cn (J.D.); swan1861@163.com (J.Y.)
* Correspondence: lsq_cie@nwsuaf.edu.cn

Abstract: Sheep face recognition models deployed on edge devices require a good trade-off between model size and accuracy, but the existing recognition models cannot do so. To solve the above problems, this paper combines Mobilenetv2 with Vision Transformer to propose a balanced sheep face recognition model called MobileViTFace. MobileViTFace enhances the model's ability to extract fine-grained features and suppress the interference of background information through Transformer to distinguish different sheep faces more effectively. Thus, it can distinguish different sheep faces more effectively. The recognition accuracy of 96.94% is obtained on a self-built dataset containing 5490 sheep face photos of 105 sheep, which is a 9.79% improvement compared with MobilenetV2, with only a small increase in Params (the number of parameters) and FLOPs (floating-point operations). Compared to models such as Swin-small, which currently performs SOTA, Params and FLOPs are reduced by nearly ten times, whereas recognition accuracy is only 0.64% lower. Deploying MobileViTFace on the Jetson Nano-based edge computing platform, real-time and accurate recognition results are obtained, which has implications for practical production.

Keywords: sheep face recognition; deep learning; vision transformer; Mobilenetv2; precision agriculture; Jetson Nano platform

check for updates

Citation: Li, X.; Du, J.; Yang, J.; Li, S. When Mobilenetv2 Meets Transformer: A Balanced Sheep Face Recognition Model. *Agriculture* **2022**, *12*, 1126. https://doi.org/10.3390/agriculture12081126

Academic Editor: Claudia Arcidiacono

Received: 2 July 2022
Accepted: 26 July 2022
Published: 29 July 2022

1. Introduction

The automatic identification of individual livestock has become an integral part of the livestock sector. It can capture various types of information that can be interpreted into various reports and provide farmers with critical information to manage their farms.

Traditional methods of livestock identification include branding, tattooing, and ear tagging [1]. However, these methods can cause animal stress and injury and require frequent maintenance and cleaning. In addition, they require manual registration, which is both time-consuming and prone to error [2]. Radio frequency identification (RFID) is widely used in individual livestock identification [3]. However, it is more costly than traditional methods and is susceptible to interference. Some studies have used retina, iris, and nasal prints combined with traditional machine learning methods for individual livestock identification [4,5]. Still, these methods require manual feature design and do not satisfy the need for individual livestock identification in unconstrained environments.

In recent years, face recognition has matured thanks to the development of convolutional neural networks (CNNs) and is an active area of research. Inspired by face recognition, researchers have investigated the effectiveness of CNNs in identifying livestock using various biometric features [6–9]. Since faces contain many important features such as eyes, noses, and other biometric features, they possess a great scope for development. Unlike human faces, livestock faces have distracting factors such as hair and texture changes, and it is impossible for livestock to consciously hold their faces steady in front of the camera for longer periods [10–12]. Suboptimal face images lead to poor recognition by deep learning models. Thus, livestock face recognition research has been limited [13]. Moreover, these

research works either have recognition accuracy that does not meet the requirements of real production or have large model parameters and high FLOPs that cannot be deployed on resource-constrained edge devices to effectively facilitate the application of sheep face recognition in practice. Some of the recently published works are shown in Table 1. As can be seen, the overall identification accuracy is still not high enough, and they do not report the number of parameters and FLOPs of the proposed models.

Table 1. Some livestock identification studies.

Reference	Dataset	Method	Accuracy (%)
[14]	81 sheep	CNN	95.00%
[15]	28 pigs	CNN	96.80%
[16]	over 5000 images of 547 sheep	CNN	85.00%
[17]	3278 pictures of goats	CNN	96.40%
[18]	945 images of cow faces	CNN	91.67%
[13]	1553 images of 10 pigs	CNN	96.70%
[19]	2364 images of pigs	CNN	83.00%
[20]	2318 images of 90 cows	CNN	91.30%

To reduce the number of parameters in the model, this paper chose Mobilenetv2 as the feature extraction network. Mobilenetv2 is a lightweight deep learning model proposed by [21], which not only uses depthwise separable convolution to reduce the FLOPs compared with Mobilenetv1 but also introduces the inverse residual structure to improve the feature extraction ability of the network. However, the spatial localization of the convolutional structure limits its overall modeling of sheep faces, which leads to poor recognition when sheep face images are occluded and when faced with factors such as lighting changes [22].

The Transformer architecture is a standard paradigm for natural language processing [23] and has recently caused a stir in computer vision [24–27]. Transformer is complementary to convolutional structures. The Transformer architecture has a global perceptual field that allows information to flow freely in different locations of the image, establishing long-range dependencies and providing better robustness in recognizing obscured sheep faces. The Transformer architecture is a more general form of attention mechanism, which can give different attention to different features, meaning it can suppress the interference of background information on recognition results to some extent. However, the Transformer's division of the whole image into small patches destroys the local continuity of the image. Additionally, without the local inductive bias of the convolutional structure, the Transformer usually requires a large amount of data for training to achieve the same results as the CNN-based model.

Therefore, it is a natural choice to combine Mobilenetv2, based on the convolutional structure, with Transformer. In this paper, we combine the advantages of Mobilenetv2 and Vision Transformer to propose MobileViTFace, a lightweight sheep face recognition model that makes a good trade-off between recognition accuracy and model size. The main contributions of this paper are summarized as follows:

(1) We propose a sheep face recognition model MobileViTFace based on the CNN and Transformer structure. MobileViTFace fully combines the advantages of the convolutional structure and Transformer structure to extract more effective features for the final sheep face recognition, which is a general form of combining convolutional structure and attention structure.
(2) Improving the effectiveness of sheep face recognition. MobileViTFace not only has high recognition accuracy, but also the number of parameters and FLOPs of the model is significantly reduced due to the lightweight design.
(3) The application of the deep learning-based sheep face recognition model in practical production is promoted. The proposed MobileViTFace sheep face recognition model is deployed on the Jetson Nano edge computing-based platform to develop a sheep

face recognition system, which effectively improves the informationization of sheep farms and can provide a reference for other sheep farms.

2. Materials and Methods

2.1. Self-Built Dataset

The sheep face image data were collected in June 2020 at a sheep breeding base in Ningxia, China. The Sony DSC-RX100M2 camera was used to track the sheep in the scenes, and each video was about 1 min long with a frame rate of 60 fps and a resolution of 1920 × 1080. In total, 105 categories of 5490 images of sheep containing sheep faces were obtained by intercepting key frames from the captured videos and excluding blurred and poorly lit images. Some of the sheep data in different scenes are shown in Figure 1.

Figure 1. (**a**–**f**) are some samples of the dataset.

The obtained images were divided into the training set and validation set according to the ratio of 7:3. Considering that there may be angle and brightness changes in the actual shooting, the image data in the training set were enhanced by including angle transformation of the image, adjusting the brightness and darkness, etc., for data expansion. Finally, 13,176 training images were obtained.

2.2. Model

2.2.1. Overall Flow Chart

The overall flow chart of the sheep face recognition system with the sheep face recognition algorithm as the core is shown in Figure 2.

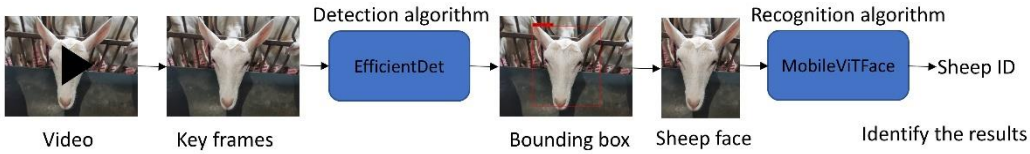

Figure 2. The flow chart of the sheep face recognition model in this paper.

The video stream data was acquired through the webcam installed at the sheep farm. The acquired video was decoded by FFMPEG (Fast Forward Moving Picture Expert Group) and video frame interception to obtain sheep image data. We performed sheep face detection and recognition and compared the obtained sheep face features with the registered sheep face features in the database to obtain the recognition results. The recognition results are visualized and displayed, and the sheep face detection and recognition models are introduced in the following sections.

2.2.2. Sheep Face Detection Module

Through experimental comparison, EfficientDet-D1 [28] was selected as the sheep face detection model, and the detailed experimental results are presented in Section 3.1. The commonly used object detection frameworks were divided into two major categories, where one category is the two-stage object detection algorithms such as those found in [29–31]. The two-stage object detection algorithms have higher accuracy but are slower and cannot meet the real-time requirements. The other category is the single-stage object detection algorithms such as those found in [32,33]. The detection speed of the single-stage algorithm can meet the real-time requirements. However, its detection accuracy is lower than that of the two-stage detector. Balancing the speed and accuracy of detection algorithms is the problem that needs to be solved. To solve the above issues, the EfficientDet network was selected as this paper's sheep face detection model. EfficientDet is a object detection model proposed by the Google Brain team in 2020. It is a collective name for a series of efficient and scalable object detection models, and its network architecture is shown in Figure 3. The feature extraction network is EfficientNet and follows the compound scaling (composite scaling) method. Bi-FPN (Bi-Feature Pyramid Network), a feature fusion network, introduces learnable weights to learn the importance of different input features. The prediction network consists of a classification prediction network and a regression prediction network, and both networks share the weights of the feature network. However, among the eight models in the EfficientDet series d0-d7, as the accuracy of the network gradually increases, the computational effort increases accordingly. EfficientDet-D1 was selected as the sheep face detection network in this paper to strike a balance between detection accuracy and speed.

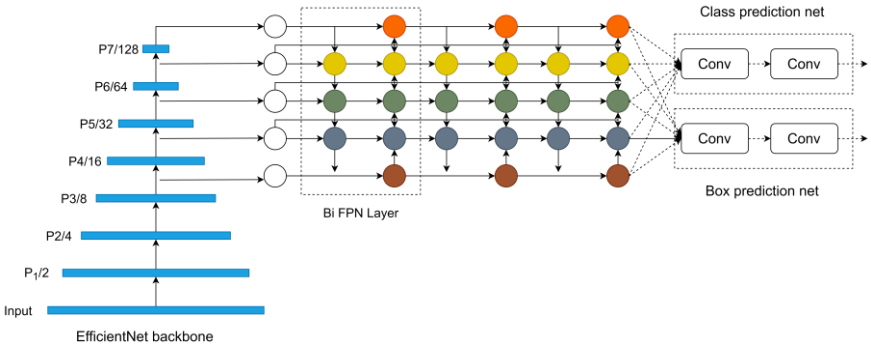

Figure 3. EfficientDet network structure.

2.2.3. Improved Sheep Face Recognition Module

MobileNetV2 is an improvement of mobileNetV1, a lightweight neural network. MobileNetV2 retains the depthwise separable convolution of the MobileNetV1 version with the addition of a linear bottleneck and inverted residual. For the input image, after a series of convolution and bottleneck operations, the height and width of the input image are continuously halved, the number of channels is increased, and the number of channels remains constant until each downsampling. Finally, the feature vectors for prediction are generated by averaging pooling and point convolution. Unlike the original Mobilenetv2, MobileViTFace was based on the MobileNetV2 network structure, adding 2 LinearViT structures to its third downsampling, 4 LinearViT structures to its fourth downsampling, and 3 LinearViT structures to its fifth downsampling. The final feature vector was fixed to 128 dimensions through the fully connected layer, which was used to represent the sheep face, and other settings remain unchanged. The network structure of MobileViTFace is shown in Table 2.

This design is based solely on existing studies [34,35] and our experiments [36]. In general, the Transformer structure is considered more powerful than the convolutional structure because the Transformer aims to establish global connections between features. In contrast, the convolutional structure captures only local information. However, in terms of

efficiency, the convolutional structure consistently outperforms the Transformer structure at all input resolutions. However, as the input resolution decreases, the efficiency gap between the Transformer and convolutional structures shrinks. This observation inspired us to design the network so that the Transformer structure can be placed at a later stage to balance performance and efficiency. Although the above improvements achieve significant results, their efficiency is still not comparable to that of the convolutional network alone. Therefore, we used a mixture of convolutional and Transformer structures. The combination of the convolutional structure and Transformer block is shown in Figure 4, where the input features $X = H \times W \times C$ (H represents the height of the input image, W represents the width, and C represents the number of channels) are first locally modeled by 3×3 convolution and 1×1 convolution to obtain the output $X = H \times W \times d$ (H represents the height of the input image, W represents the width, and d represents the number of channels). Then, each feature map is divided into N patches, and each patch is flattened as a 1-dimensional vector with dimension P by linear projection. These patches are input into M LinearViTs for global representation. Next, the patches are restored to the shape they had before being input to the LinearViT. Feature fusion is performed with the input feature map through residual connectivity, and the final output feature map shape is the same size as the input feature map. In this way, based off of the convolution structure and Transformer structure, the model has the ability of both local representation and global representation.

Table 2. MobileViTFace: Each line describes one or more identical operation sequences repeated for n times. All layers in the same sequence have the same output channels. The step length of the last layer in each sequence is the stride, and the step length of all other layers is 1. All spatial convolutions use a 3×3 convolution.

Input	Operator	n	Stride	Output Channels
$224^2 \times 3$	Conv2d	1	2	16
$112^2 \times 16$	bottleneck	1	1	32
$112^2 \times 32$	bottleneck	3	2	64
$56^2 \times 64$	bottleneck	1	2	96
$28^2 \times 96$	LinearViT	2	-	96
$28^2 \times 96$	bottleneck	1	2	128
$14^2 \times 128$	LinearViT	4	-	128
$14^2 \times 128$	bottleneck	1	2	160
$7^2 \times 160$	LinearViT	3	-	160
$7^2 \times 160$	Conv2d 1×1	1	1	640
$7^2 \times 640$	Avgpool 7×7	1	7	1000
$1^2 \times 1000$	FC	-	-	128

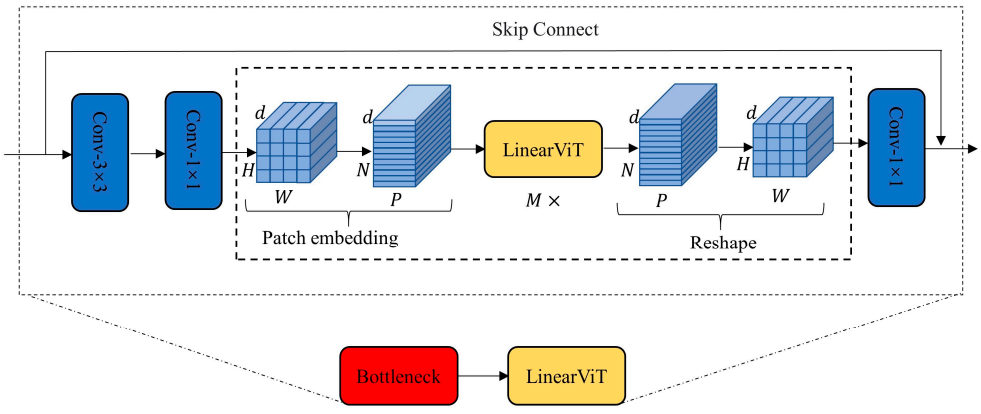

Figure 4. Bottleneck and LinearViT connection design.

2.2.4. The Design of the Bottleneck

Inverted residual is the basic component of the bottleneck. Different from the residual structure, 1×1 convolution is first used to realize dimensionality enhancement, and then 3×3 DW convolution (channel by channel convolution) is used to extract the features. Finally, 1×1 convolution is used to achieve dimensionality reduction. The order of descending and ascending dimensions is changed, and the standard convolution of 3×3 is changed into DW convolution to reduce the number of parameters of the model. In the inverted residuals structure, the ReLU6 activation function is used for the first two convolutions, and the linear activation function is used for the last one. The purpose of replacing ReLU with ReLU6 is to ensure good numerical resolution even in the low-precision Float16 on the mobile end. Moreover, leap connections are added to inverted residuals when the step size is 1. The inverted residuals structure is shown in Figure 5.

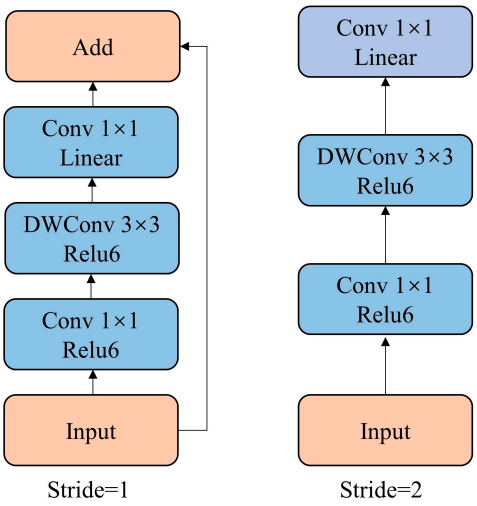

Figure 5. The design of the bottleneck.

2.2.5. The Design of the LinearViT

The MHA allows tokens to interact with each other and is the key to learning global representation. However, the Transformer block has a self-attention complexity of $O(N^2)$, i.e., it is quadratic in the number of tokens N. The deployment of Transformer-based models on resource-constrained devices is of particular concern. Therefore, the necessary optimization of MHA is a must. To this end, this paper takes a cue from PVT v2 [37] and replaced the original Transformer with LinearViT to reduce the computational complexity of the model. A comparison of the two is shown in Figure 6.

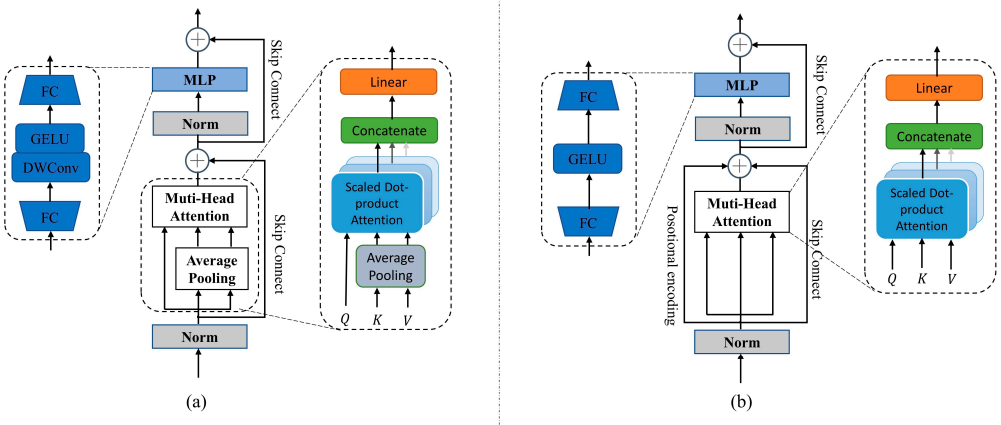

Figure 6. The linear versus the original Transformer. (**a**) is the LinearViT, (**b**) is the orginal Vision Transformer.

Compared with the original Transformer, LinearViT has two main improvements.

(1) Multi-head attention layer with linear computation complexity:

As shown in Figure 6, the original MHA receives a query Q, a key K, and a value V as the input and output of the optimized feature map. The difference is that LinearViT's MHA reduces the spatial scale of K and V before the attention operation, which significantly reduces the computation/memory overhead. For an image or intermediate feature layer with an input of space dimension $h \times w \times c$, the original MHA complexity and LinearViT's MHA complexity are Equations (1) and (2), respectively, where P is the pooling size of LinearViT's MHA, which is set to 7. It can be seen that the computational complexity of LinearViT is almost negligible compared with the original computational complexity.

$$\Omega(MHA) = 2h^2w^2c + 2wc^2 \tag{1}$$

$$\Omega(LinearViT) = 2hwP^2c \tag{2}$$

(2) Replacing the position encoding with depthwise separable convolution:

The original Transformer divides the input image into patches that destroy the spatial location information of the input. It requires the addition of location encoding to retain spatial location information. The size of the location encoding is generally the same as the number of patches. Still, a change in the size of the input image or the size of the patches causes a change in the location encoding information, which is not conducive to model reuse. However, the convolutional structure does not have such a concern. To solve the above problem, in this paper, following the PVTv2, we added a 3×3 deep convolution between the first fully connected (FC) layer and GELU with a 0 padding size of 1 in the feedforward network.

2.3. Evaluation Indicators and Experimental Environment

2.3.1. Evaluation Indicators

The evaluation metric of the sheep face detection model is AP (Average Precision), which is expressed as the area under the Precision-Recall curve. The larger the value of AP, the better the detection model is. The formula for calculating AP is shown in Equation (3).

The evaluation metrics of the sheep face recognition model MobileViTFace include Accuracy, Recall, Precision, F1-score, Params, and FLOPs, and they are calculated in Equations (3)–(7), where TP, FP, TN, and FN are the number of true positives, false positives, true negatives, and false negatives, respectively. The Accuracy represents the number of correctly classified samples as a percentage of the total number of samples. The Recall represents the ratio of the number of samples correctly retrieved to the number of samples that should have been retrieved. Precision is the ratio of the number of samples correctly retrieved to the total number of samples retrieved. The F1-score considers both Precision and Recall. FLOPs are used to measure the model's runtime, which refers to the number of floating-point operations performed throughout the forward process. The lower the FLOPs, the less computation and execution time the model requires. Params determines the model's size. The smaller the model, the lower the hardware requirements and the higher the model's applicability, provided that the task requirements are met.

$$AP = \int_0^1 P(r)dr \tag{3}$$

$$accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{4}$$

$$precision = TP/(TP + FP) \tag{5}$$

$$recall = TP/(TP + FN) \tag{6}$$

$$\text{F1-score} = 2/((1/\text{precision}) + (1/\text{recall})) \tag{7}$$

2.3.2. Experimental Parameter Setting

The experiments in this study were conducted in the Ubuntu 20.04 environment (processor: Intel Core i9-10900X, RAM: 48 GB, graphics card: GeForce RTX 3090 × 2), and the deep learning framework was Pytorch, combined with Cuda 11.1 for training.

Jetson Nano is a low-spec computing platform launched by Nvidia. The size of Jetson Nano is 100 mm × 80 mm × 29 mm, with a 4-core, 64-bit ARM CPU and a 128-core Maxwell GPU. The CPU and GPU share 4 GB of memory. The power consumption is 5–10 watts.

The parameters of the sheep face detection model were set as follows: the pre-trained model on the ImageNet dataset was used to initialize the parameters of the feature extraction network, the batch size was set to 64, the SGD optimizer was used, and the momentum was set to 0.9. The image size of the input network model was 640 × 640, and 100 epochs were trained.

In the sheep face recognition model experiment, the Arcface loss function was used, the dynamic initialization learning rate was set as 1×10^{-3}, the AdamW optimizer was used, and the cosine annealing algorithm was used to adjust the learning rate. To prevent overfitting of the training model, the dropout was set to 0.6, learning momentum to 5×10^{-4}, iteration epoch to 50, and batch size to 64.

3. Results

3.1. Comparison of Sheep Face Detection Results

This section aims to screen out the models that make a good trade-off between recognition effectiveness and recognition speed from the currently dominant object detection networks in preparation for the subsequent execution of sheep face recognition. Sheep face detection experiments were executed on the constructed dataset to detect as many sheep faces as possible, and the experimental results are shown in Figure 7. It can be seen that, basically, all models converge within 100 epochs, but Yolov3 performs less consistently, which may be related to the learning rate. The detailed experimental results are shown in Table 3, from which it can be seen that EfficientDet-D1 not only detects well, though not the best, but has lower Params and FLOPs. Therefore, using it as a model for sheep face detection is reasonable.

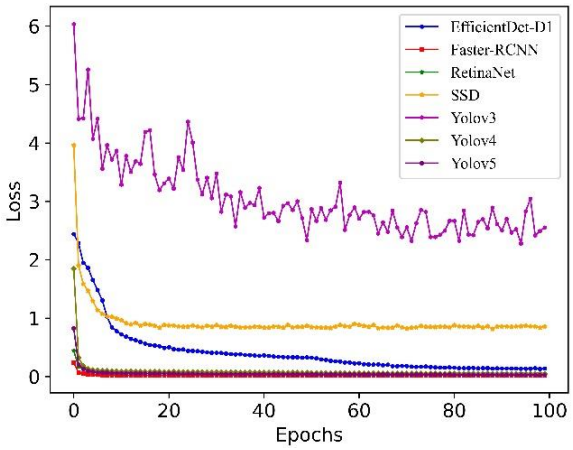

Figure 7. Loss comparison of sheep face detection models.

3.2. Comparison of Recognition Results Using MobileViTFace and Other Methods

To verify the effectiveness of the proposed MobileViTFace, ten network models, such as Resnet18 [38], were selected for comparison experiments on the self-built sheep face dataset. The training process is shown in Figures 8 and 9.

Table 3. Comparison of experimental results of detection models.

Model	AP(%)	Model Size (MB)	FLOPs (G)
SSD	99.90	52.20	15.00
Faster RCNN	99.90	158.80	135.30
RetinaNet [36]	99.00	140.00	20.60
YOLO V3	91.40	234.00	155.20
YOLO V4 [38]	98.90	244.00	25.40
YOLO V5 (S)	98.70	27.00	16.40
EfficientDet-D1	98.90	25.00	5.60

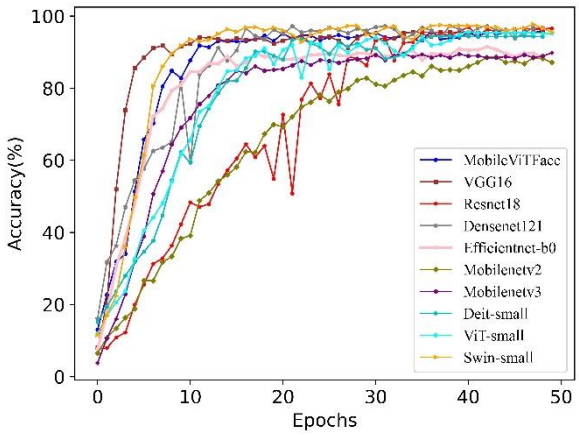

Figure 8. Comparison of validation accuracy of sheep face recognition models.

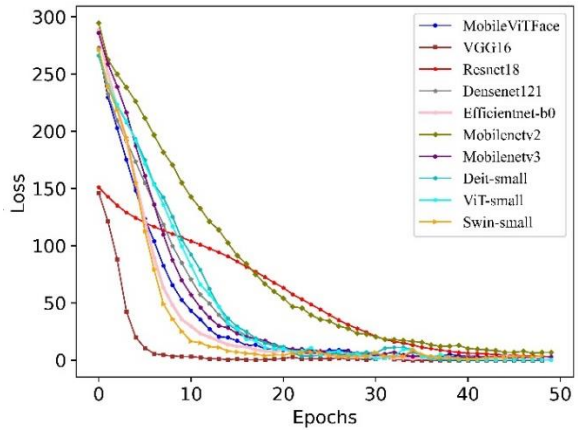

Figure 9. Loss comparison of sheep face recognition models.

From Figure 8, it can be seen that the recognition accuracy of MobileViTFace improves rapidly, tends to be stable at the 11th epoch, does not show significant changes during the subsequent iterations, and no longer changes at 50 epochs. This can be compared with the dramatic changes in recognition accuracy of Mobilenetv2 and ViT-small; by combining the advantages of both, MobileViTFace has the characteristics of good stability, easy training, and faster convergence. Another remarkable phenomenon is that EfficientNet-b0, MobilenetV2, and MobilenetV3 have similar Params and FLOPs as our model. Still, the recognition effect significantly lags behind our model, which indicates that the addition of Transformer significantly improves the recognition effect of Mobilenetv2. From Figure 9, we can observe that MobileViTFace converges quickly, tends to stabilize at the 45th epoch, and converges significantly faster than Mobilenetv2 and ViT. Additionally, it is worth noting that VGG16 [39] converges the fastest, although it has higher Params and FLOPs, which can be attributed to its use of smaller convolutional kernels.

Table 4 shows more detailed experimental results. Two significant patterns can be found in the table: (1) With similar Params and FLOPs, the recognition accuracy of MobileViTFace is much higher than that of EfficientNet-b0, MobilenetV2, and other models; for example, the recognition accuracy of MobileViTFace is 9.79% higher than that of MobilenetV2. (2) With similar recognition accuracy, MobileViTFace's Params and FLOPs are much lower than other models; for example, MobileViTFace's Params only account for 9.6% of Swin-small's, but the recognition accuracy is only 0.64% lower for MobileViTFace. This indicates that MobileViTFace makes a good trade-off between recognition accuracy and Params and FLOPs, which is significantly better than MobilenetV2 and Transformer before improvements.

Table 4. Detailed experimental result comparison of MobileViTFace with other SOTA models.

Model	Pre (%)	Recall (%)	F1 (%)	Acc (%)	Params (MB)	FLOPs (G)
VGG16	95.87	95.66	95.76	95.32	138.30	15.50
Resnet18	97.64	97.19	97.42	96.61	11.60	1.80
DenseNet121	97.80	97.18	97.49	96.84	7.90	2.80
EfficientNet-b0	92.03	91.25	91.64	90.88	5.30	0.39
MobilenetV2	89.72	87.83	88.76	87.15	3.50	0.31
MobilenetV3	90.84	89.58	90.21	89.83	5.40	0.22
ViT-small	96.00	95.44	95.72	95.21	22.00	4.24
DeiT-small	95.26	94.39	94.82	94.39	22.00	4.24
Swin-small	98.78	98.57	98.68	97.58	49.60	8.50
Ours	97.85	97.05	96.55	96.94	4.80	0.90

3.3. How to Deal with the Newly Added Sheep?

The above experiments demonstrate the effectiveness of the MobileViTFace proposed in this paper. Still, the above experiments were conducted in a closed-set environment, i.e., new additions of sheep were not considered, which is a problem that must be considered because sheep are often released or added to the sheep farm. To solve this problem, two approaches were used, i.e., experiments in the open-set environment and fine-tuning of the already trained sheep face recognition model. The experimental results are shown in Figures 10 and 11. As can be seen from the figures, both the model in the open-set environment and the model fine-tuned with the newly added sheep face images can converge quickly within ten epochs. Still, the model in the open-set environment only obtains a recognition accuracy of about 88%, which cannot meet the actual demand. The recognition accuracy of the model after fine-tuning is as high as 98%, thus fine-tuning the trained sheep face recognition model is still a feasible solution at this stage.

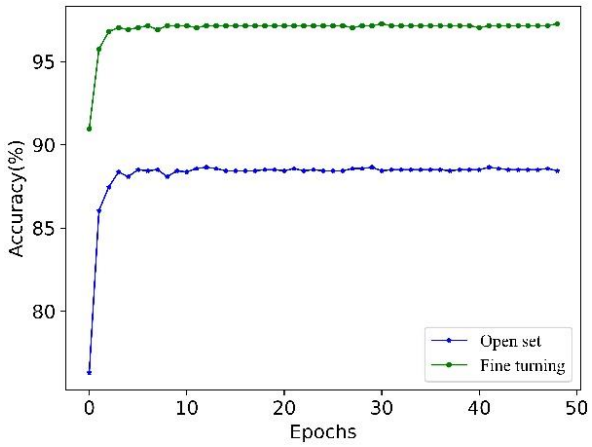

Figure 10. Comparison of recognition accuracy between open-set and fine-tuning experiments.

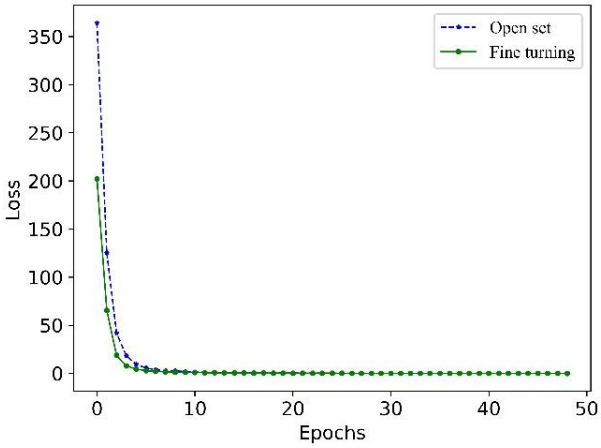

Figure 11. Loss comparison between open-set experiment and fine-tuning experiment.

3.4. Visualization of Recognition Results

The recognition effect of the sheep face recognition system is shown in Figure 12. From Figure 12a,b, it can be seen that the sheep face recognition system can effectively detect the faces of two breeds of sheep and give their corresponding identity information in the system. Figure 12c shows that the sheep face recognition system can also accurately recognize sheep faces on a small scale from a distance. For sheep with unknown identities, the system developed in this paper can also identify them accurately, as shown in Figure 12d, which shows the unknown identity. In the actual test, the average delay from sending the video stream to receiving and decoding the key frame is 0.33 s. The single frame inference time of the detection model and recognition model optimized by TensorRT is 68.22 ms and 15.46 ms, respectively, and the FPS can reach 13. The system can meet the requirements of real-time sheep face recognition.

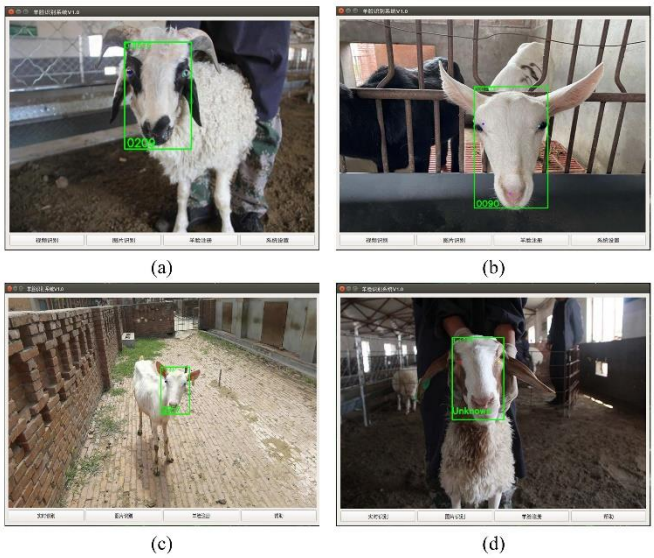

Figure 12. (**a**–**c**) are the correct identification results, and (**d**) is the identification result of unregistered sheep.

3.5. Failure Case

Although the MobileViTFace proposed in this paper shows good recognition results, some cases still fail to recognize faces correctly. The reasons for incorrect recognition or non-recognition are analyzed in this subsection. As shown in Figure 13, some failure cases are shown. Among them, Figure 13a,b are the face images of the same sheep, but image Figure 13a is blurrier and the detailed information of the sheep's face is lost more seriously,

which in turn leads to the model recognition error. Figure 13c,d are two sheep faces with black hair on both faces, and the black color makes other features inconspicuous, which in turn leads to the model failing to extract enough features for correct classification.

Figure 13. Sample examples of identifying errors, (**a**,**b**) are the same sheep, and (**c**,**d**) are sheep with black facial hair.

4. Discussion

This study proposes a sheep face recognition model with high recognition accuracy, low Params, low FLOPs, and easy deployment on edge devices. This paper combined Mobilenetv2 with Vision Transformer to obtain MobileViTFace, a lightweight sheep face recognition model with a good trade-off between recognition accuracy and model size. MobileViTFace achieves accurate and real-time recognition results on edge devices and meets the expected goals. MobileViTFace significantly reduces the model parameters and computation while maintaining high recognition accuracy, improves the robustness of the model to occlusions and background noise, and facilitates the application of deep learning models to individual livestock recognition. The model's design is more complex than the pure CNN structure. MobileViTFace combines the advantages of CNN and Vision Transformer, and compared to the pure CNN structure [14,17,20], MobileViTFace has higher recognition accuracy because the addition of the Vision Transformer structure gives the model a higher performance ceiling. The Params is small, and the computation is simple. Compared to the pure Vision Transformer structure [40], MobileViTFace has faster inference and requires less data volume. Compared with other lightweight recognition models [41], our model is more robust and has faster inference.

This study focused more on practical applications and did not pursue higher recognition accuracy, which leads to excessive Params and FLOPs of the model and cannot be promoted for use in practice. MobileViTFace significantly reduces the Params and FLOPs of the model while maintaining high recognition accuracy and can achieve real-time accurate sheep face recognition with a resource-constrained edge device. The model design is generally relatively complex, and the Params and FLOPs need to be further reduced compared to the lightweight CNN model.

5. Conclusions

In this paper, we found that Transformer can improve the performance of lightweight convolutional neural networks. With the appropriate lightweight design of Transformer, the proposed MobileViTFace sheep face recognition model possesses both the high efficiency of the convolutional structure and the high performance of Transformer, which promotes the deep learning-based sheep face recognition model being deployed on edge devices. In the future, we will develop sheep face recognition models that are more suitable for use in complex contexts.

Author Contributions: Conceptualization, X.L.; methodology, X.L.; software, X.L.; validation, X.L.; formal analysis, X.L.; investigation, X.L.; resources, X.L.; data curation, J.D.; writing—original draft preparation, X.L.; writing—review and editing, X.L.; visualization, J.Y.; supervision, S.L.; project administration, S.L.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (Grants number 2020YFD1100600, Grants number 2020YFD1100601). The authors appreciate the funding organization for its financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author. The data are not publicly available due to the privacy policy of the authors' institution.

Acknowledgments: We thank all of the funders.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ait-Saidi, A.; Caja, G.; Salama, A.A.K.; 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. [CrossRef] [PubMed]
2. Kumar, S.; Tiwari, S.; Singh, S.K. Face recognition for cattle. In Proceedings of the 2015 Third International Conference on Image Information Processing (ICIIP), Waknaghat, India, 21–23 December 2015; pp. 65–72.
3. Yan, H.; Cui, Q.; Liu, Z. Pig face identification based on improved AlexNet model. *INMATEH-Agric. Eng.* **2020**, *61*, 97–104. [CrossRef]
4. Gaber, T.; Tharwat, A.; Hassanien, A.E.; Snasel, V. Biometric cattle identification approach based on Weber's Local Descriptor and AdaBoost classifier. *Comput. Electron. Agric.* **2016**, *122*, 55–66. [CrossRef]
5. Zaorálek, L.; Prilepok, M.; Snášel, V. Cattle identification using muzzle images. In Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015, Villejuif, France, 9–11 September 2015; pp. 105–115.
6. Hou, J.; He, Y.; Yang, H.; Connor, T.; Gao, J.; Wang, Y.; Zhou, S. Identification of animal individuals using deep learning: A case study of giant panda. *Biol. Conserv.* **2020**, *242*, 108414. [CrossRef]
7. Salama, A.Y.A.; Hassanien, A.E.; Fahmy, A. Sheep identification using a hybrid deep learning and bayesian optimization approach. *IEEE Access* **2019**, *7*, 31681–31687. [CrossRef]
8. Khaldi, Y.; Benzaoui, A.; Ouahabi, A.; Jacques, S.; Taleb-Ahmed, A. Ear recognition based on deep unsupervised active learning. *IEEE Sens. J.* **2021**, *21*, 20704–20713. [CrossRef]
9. Gadekallu, T.R.; Rajput, D.S.; Reddy, M.; Lakshmanna, K.; Bhattacharya, S.; Singh, S.; Alazab, M. A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU. *J. Real-Time Image Process.* **2021**, *18*, 1383–1396. [CrossRef]
10. Yang, H.; He, X.; Jia, X.; Patras, I. Robust face alignment under occlusion via regional predictive power estimation. *IEEE Trans. Image Process.* **2015**, *24*, 2393–2403. [CrossRef] [PubMed]
11. Wang, N.; Gao, X.; Tao, D.; Yang, H.; Li, X. Facial feature point detection: A comprehensive survey. *Neurocomputing* **2018**, *275*, 50–65. [CrossRef]
12. Yang, H.; Carlone, L. In perfect shape: Certifiably optimal 3D shape reconstruction from 2D landmarks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 621–630.
13. Hansen, M.F.; Smith, M.L.; Smith, L.N.; Salter, M.G.; Baxter, E.M.; Farish, M.; Grieve, B. Towards on-farm pig face recognition using convolutional neural networks. *Comput. Ind.* **2018**, *98*, 145–152. [CrossRef]
14. Hitelman, A.; Edan, Y.; Godo, A.; Berenstein, R.; Lepar, J.; Halachmi, I. Biometric identification of sheep via a machine-vision system. *Comput. Electron. Agric.* **2022**, *194*, 106713. [CrossRef]
15. Wang, Z.; Liu, T. Two-stage method based on triplet margin loss for pig face recognition. *Comput. Electron. Agric.* **2022**, *194*, 106737. [CrossRef]
16. Meng, X.; Tao, P.; Han, L.; CaiRang, D. Sheep Identification with Distance Balance in Two Stages Deep Learning. In Proceedings of the 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 4–6 March 2022; pp. 1308–1313.
17. Billah, M.; Wang, X.; Yu, J.; Jiang, Y. Real-time goat face recognition using convolutional neural network. *Comput. Electron. Agric.* **2022**, *194*, 106730. [CrossRef]
18. Xu, F.; Gao, J.; Pan, X. Cow face recognition for a small sample based on Siamese DB Capsule Network. *IEEE Access* **2022**, *10*, 63189–63198. [CrossRef]
19. Marsot, M.; Mei, J.; Shan, X.; Ye, L.; Feng, P.; Yan, X.; Zhao, Y. An adaptive pig face recognition approach using Convolutional Neural Networks. *Comput. Electron. Agric.* **2020**, *173*, 105386. [CrossRef]
20. Xu, B.; Wang, W.; Guo, L.; Chen, G.; Li, Y.; Cao, Z.; Wu, S. CattleFaceNet: A cattle face identification approach based on RetinaFace and ArcFace loss. *Comput. Electron. Agric.* **2022**, *193*, 106675. [CrossRef]

21. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
22. Adjabi, I.; Ouahabi, A.; Benzaoui, A.; Taleb-Ahmed, A. Past, present, and future of face recognition: A review. *Electronics* **2020**, *9*, 1188. [CrossRef]
23. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
24. Yang, F.; Yang, H.; Fu, J.; Lu, H.; Guo, B. Learning texture transformer network for image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 5791–5800.
25. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 10012–10022.
26. Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Guo, B. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 12124–12134.
27. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Guo, B. Swin transformer v2: Scaling up capacity and resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 12009–12019.
28. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, The Washington State Convention Center, Seattle, WA, USA, 16–18 June 2020; pp. 10781–10790.
29. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic seg-mentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
30. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
31. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
32. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
33. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
34. Xia, X.; Li, J.; Wu, J.; Wang, X.; Wang, M.; Xiao, X.; Wang, R. TRT-ViT: TensorRT-oriented Vision Transformer. *arXiv* **2022**, arXiv:2205.09579.
35. Mehta, S.; Rastegari, M. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv* **2021**, arXiv:2110.02178.
36. Li, X.; Li, S. Transformer Help CNN See Better: A Lightweight Hybrid Apple Disease Identification Model Based on Transformers. *Agriculture* **2022**, *12*, 884. [CrossRef]
37. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Shao, L. Pvt v2: Improved baselines with pyramid vision transformer. *Comput. Vis. Media* **2022**, *8*, 415–424. [CrossRef]
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
39. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
40. Zhong, Y.; Deng, W. Face transformer for recognition. *arXiv* **2021**, arXiv:2103.14803.
41. Li, Z.; Lei, X.; Liu, S. A lightweight deep learning model for cattle face recognition. *Comput. Electron. Agric.* **2022**, *195*, 106848. [CrossRef]