Postprint: Multi-Scale Feature Fusion Yak Face Recognition Algorithm Based on Transfer Learning

Authors: Chen Zhanqi, Zhang Yu' an, Wang Wenzhi, Li Dan, He Jie, Song Rende

Date: 2023-02-17T00:00:00+00:00

Abstract

Individual identity marking of yaks is a prerequisite for establishing individual records, behavior monitoring, precision feeding, disease prevention and control, and food traceability. To address the application requirements of animal individual identification technology in intelligent and information-based breeding platforms for smart livestock farming, this study proposes a yak face recognition algorithm based on transfer learning and multiscale feature fusion, named Transfer Learning-Multiscale Feature Fusion-VGG (T-M-VGG). Using the pretrained Visual Geometry Group Network (VGG) as the backbone network, a convolutional neural network model based on transfer learning is constructed. Feature maps from Block3, Block4, and Block5 outputs are obtained, denoted as F3, F4, and F5 respectively. F3 and F5 are processed through a parallel dilated convolution module composed of three dilated convolutions with different dilation rates to enlarge the receptive field, then fed into an improved feature pyramid for multiscale feature fusion. Finally, global average pooling is used to replace the fully connected layer for classification output. Experimental results demonstrate that the proposed T-M-VGG algorithm achieves a recognition accuracy of 96.01% on a dataset of 38,800 images from 194 yaks, with a model size of 70.75 MB. Randomly selecting 12 images of different classes for face occlusion testing yields a recognition accuracy of 83.33%. This algorithm can provide a reference for yak face recognition research.

Full Text

Preamble

Multiscale Feature Fusion Yak Face Recognition Algorithm Based on Transfer Learning

CHEN Zhanqi[1], ZHANG Yu'an[1]*, WANG Wenzhi[1], LI Dan[1], HE Jie[1], SONG Rende[2]
[1]Department of Computer Technology and Application, Qinghai University, Xining, China
[2]Animal Disease Prevention and Control Center of Yushu Tibetan Autonomous Prefecture, Yushu, China

Abstract: Individual identification of yaks is a prerequisite for establishing animal records, behavior monitoring, precision feeding, disease prevention and control, and food traceability. To address the application requirements of animal individual identification technology in intelligent and information-based livestock farming platforms, this study proposes a yak face recognition algorithm based on transfer learning and multiscale feature fusion (T-M-VGG). Using the pre-trained Visual Geometry Group Network (VGG) as the backbone, a convolutional neural network model based on transfer learning is constructed. The feature maps from Block3, Block4, and Block5 are extracted, where Block3 and Block5 are processed by a parallel dilated convolution module composed of three dilated convolutions with different dilation rates to enlarge the receptive field before being fed into an improved feature pyramid for multiscale feature fusion. Finally, the feature maps output from Block4 and Block5 are used for classification, with global average pooling replacing the fully connected layer. Experimental results demonstrate that the proposed algorithm achieves a recognition accuracy of 96.01% on a dataset of 38,800 images, with a model size of 70.75 MB. Randomly selecting 12 images for occlusion testing yields an identification accuracy of 83.33%. This algorithm can provide a reference for yak face recognition research.

Keywords: yak; face recognition; transfer learning; feature pyramid; T-M-VGG

Journal Information: Smart Agriculture, 2022, 4(2): 77-85.

---

1 Introduction

Yaks, known as the "ships of the plateau," play a crucial role in poverty alleviation for herders in the Qinghai-Tibet Plateau region. However, yak farming in Tibetan areas still relies on traditional practices, lacking widespread application of information-based and intelligent management equipment. For small and medium-scale farmers, livestock identification primarily depends on conventional methods such as ear tagging, color marking, and hot iron branding. Ear tags, installed through ear perforation, pose risks of bacterial infection and can be lost due to friction among animals. These traditional contact-based identification techniques cannot provide lasting reliability for the identification process.

Currently, domestic and international researchers have conducted extensive

studies on individual and behavioral identification of cattle, sheep, pigs, and other animals using Convolutional Neural Networks (CNNs), but research on yak identification remains limited. Qin Xing and Song Gefang [1] employed a bilinear CNN with Visual Geometry Group Network (VGG) as a feature extractor to test 2,110 photos of 200 pigs, achieving 95.7% face recognition accuracy, though the model required 4 days to run in the experimental environment and had a large size, limiting its applicability in real-time scenarios. He Yutong et al. [2] improved YOLOv3 for pig face detection and recognition, enhancing model accuracy but still facing challenges in boundary localization for small samples. Liu Zhongchao and He Dongjian [3] adapted the LeNet-5 CNN for dairy cow estrus behavior recognition, achieving 98.25% accuracy with a 5.8% miss rate, enabling real-time monitoring of mounting behavior. Yang Qiumei et al. [4] used GoogleNet to classify pig head, back, and tail regions to identify drinking behavior, achieving 92.11% accuracy and effectively improving farm management efficiency. Zhang Hongming et al. [5] proposed a sheep face recognition model incorporating spatial attention mechanisms, achieving 88.06% accuracy in open-set verification. Wei Zheng [6] employed a 2D linear discriminant analysis algorithm based on locality preserving projection for imperfect cattle iris recognition, though iris collection proved inconvenient for yaks. He Dongjian' s team [7,8] utilized CNNs to extract features from dairy cow backs and torsos, but this approach suits cattle with distinct body characteristics. Chen Zhengtao et al. [9] developed a parallel CNN yak face recognition algorithm based on transfer learning, achieving 91.2% accuracy with 2 days of training time, but the parallel transfer learning VGG16 structure increased model storage costs. Hansen et al. [10] and Marsot et al. [11] used CNNs for pig face recognition, achieving 96.7% and 83.0% accuracy, respectively. Kumar et al. [12] studied cattle muzzle print recognition based on deep learning, attaining 98.9% accuracy. Jung et al. [13] used CNNs for cattle sound classification and behavior analysis, achieving 94.1% accuracy. Salama et al. [14] employed Bayesian optimization to find optimal CNN architectures for sheep face recognition, reaching 98% accuracy.

Given the similarities between human face recognition and animal identification tasks, researchers have migrated face detection and recognition technologies to animal applications. Wada et al. [15] used the Eigenfaces algorithm to identify 10 pigs with 77% accuracy. Rashid et al. [16] trained a network to find similarity mapping spaces between human and animal facial features based on transfer learning. However, yak facial hair presents challenges for detection and recognition. While iris, muzzle print, and sound recognition offer uniqueness and stability, the high equipment installation costs and difficulty controlling yaks during iris and muzzle print collection make these methods unsuitable for small and medium-scale farms in plateau pastoral areas. Yaks have minimal variation in coat color, and seasonal shedding makes body trunk identification difficult. Although the non-contact biometric identification techniques proposed in the aforementioned studies face implementation challenges, using captured facial images for recognition offers certain advantages. Drawing from these methods

and leveraging the fast training speed of transfer learning and strong feature extraction capability of VGG16, this study combines transfer learning with a feature pyramid to achieve yak individual facial image recognition.

---

2 Methods

2.1 Dataset Collection

Data collection was conducted in Yushu Tibetan Autonomous Prefecture, Qinghai Province. The dataset includes facial information from 194 yaks, with each yak treated as a distinct class. Data were collected over two consecutive mornings, with approximately 2 minutes of video footage captured per yak using a GoPro HERO8 Black camera at 1920$×$1080 resolution. Each video segment was converted into image frames. A sample of the dataset is shown in Figure 1 [Figure 1: see original paper].

2.2 Dataset Processing

To avoid excessive similarity between consecutive video frames, the Structural Similarity (SSIM) algorithm [17] was applied. SSIM calculates variance, covariance, and mean intensity between image pairs to determine similarity and filter out highly redundant images. To balance inter-class sample distribution and enhance model robustness, data augmentation was performed for classes with fewer samples.

First, processed data were cropped to resolutions of 512$×512*and*320×320*usingPythonimagingtools*,*thenresiz*
Second, OpenCV functions were used for random data augmentation: clockwise rotation of 5° and 10°, brightness reduction factor of 0.85, brightness enhancement factor of 1.3, salt-and-pepper noise coefficient of 0.15, and Gaussian noise coefficient of 0.2. Finally, for classes with excessive samples, random removal was performed to maintain consistent validation set sizes across categories. The final dataset comprised 38,800 images, with each class sequentially numbered. The training set contained 31,040 images (80%), with 170–210 images per class, while the validation set contained 7,760 images (20%), with 40 images per class. Training and validation sample IDs were mutually exclusive.

2.3 Experimental Conditions

The experimental environment consisted of: Ubuntu 18.04.5 LTS 64-bit operating system, GeForce GTX 1080Ti GPU acceleration, CUDA 10.0, CUDNN 7.4.1.5, Python 3 programming language, and Keras deep learning framework based on TensorFlow for training and validation.

2.4 T-M-VGG Network Construction

Transfer learning based on CNNs can migrate knowledge from data-rich source domains to data-scarce target domains, improving learning effectiveness in the target domain [18,19]. This study employs VGG16 network-based transfer learning as the basic architecture, augmented with parallel dilated convolution modules and an improved feature pyramid structure.

2.4.1 Parallel Dilated Convolution Dilated convolution can increase the receptive field without changing parameter count. The dilated convolution calculation is given by equation (1):

$$Out(x, y) = Input(x + dr \times m, y + dr \times n) \times K(m, n) \tag{1}$$

where $Input(x, y)$ is the input feature map, dr is the dilation rate, $K(m, n)$ is the kernel function, and h and w are the feature map dimensions.

Assuming a dilated convolution kernel size of k and dilation rate of dr, the equivalent kernel size k' is given by equation (2):

$$k' = k + (k - 1) \times (dr - 1) \tag{2}$$

The current receptive field size is calculated by equation (3):

$$F(i + 1) = F(i) + (k' - 1) \times L \tag{3}$$

where $F(i + 1)$ is the current receptive field size, $F(i)$ is the previous layer' s receptive field size, and L is the product of stride values for the first i layers excluding the current $F(i + 1)$ layer.

Liu and Huang [20] incorporated dilated convolutions into InceptionNet [21] to expand the receptive field and enhance feature extraction capability. Drawing inspiration from this approach, we constructed a parallel dilated convolution module called P_{DConv} (Parallel Dilated Convolution), as shown in Figure 2 [Figure 2: see original paper]. Input feature maps undergo 1$×$1 convolution to reduce channel dimensions, followed by three parallel dilated convolutions with different dilation rates. The resulting features are fused and combined with a shortcut structure to output the final feature map.

2.4.2 Feature Pyramid Feature pyramid networks extract feature maps from different layers, fusing high-level and low-level information to form multi-dimensional enhanced features. Common feature pyramid structures are shown in Figure 3 [Figure 3: see original paper]. The feature pyramid structure constructed in this study is illustrated in Figure 4 [Figure 4: see original paper].

In Figure 4, F3 through F6 represent feature layers output by the feature extraction network. The blue nodes indicate execution order: P65, P54, P43, P53, P64, and P. Implementation details follow reference [24]. The feature pyramid algorithm pseudocode is as follows:

```
Algorithm: Feature Pyramid Implementation
Input: F3, F4, F5, F6
Output: P

def OurFPN(Input):
    P3, P4, P5, P6 = F3, F4, F5, F6
    Features = [P3, P4, P5, P6]

    for j in range(len(Features)):
        C_{Feature}[j] = Conv2D(channels_{num}, kernel_{size}=1)(Feature[j])

    P3_{in}, P4_{in}, P5_{in}, P6_{in} = C_{Feature}[0], C_{Feature}[1], C_{Feature}[2], C_

    P6_{UP} = UpSampling2D()(P6_{in})
    P65 = Add([P6_{UP}, P5_{in}])
    P65 = SeparableConv()(P65)

    # Similar operations for P54, P43, P53

    P53_{MaP} = MaxPooling2D()(P53)
    P64 = Add([P53_{MaP}, P54, P65_{UP}])
    P64 = SeparableConv()(P64)

    P64_{UP} = UpSampling2D()(P64)
    P = Add([P64_{UP}, P53])
    P = SeparableConv()(P)

    return P
```

The parallel dilated convolution and improved feature pyramid are combined with the VGG16 transfer learning network to form the final T-M-VGG (Transfer Learning-Multiscale Feature Fusion-VGG) model, as shown in Figure 5 [Figure 5: see original paper]. The VGG16 network' s third, fourth, and fifth convolutional layer outputs (F3, F4, F5) are extracted. F3 and F5 are fed into parallel dilated convolution modules P_{DConv}. P3, P4, P5, and P6 are then input to the feature pyramid for fusion (P6 is generated via max pooling of P5). The fused features are finally passed to the classifier for output.

2.5 Experimental Parameter Settings

The transfer learning implementation freezes all pre-trained convolutional layers, training only custom fully connected layers while replacing them with global

average pooling layers. To verify the proposed structure' s effectiveness, comparative experiments were conducted against CNN structures [9-11,25], VGG16 [26], MobileNetV3 [27] (Large and Small versions), InceptionV3 [28], FaceNet (Inception-ResNetV2), and pre-trained transfer learning VGG16 (denoted as Tr-L-VGG16). MobileNetV3 Large and Small are denoted as Mb-Net-L and Mb-Net-S, respectively.

To prevent overfitting, early stopping was applied for all methods, monitoring validation accuracy (val_{acc}) with min_{delta}=0.001 and patience=3. Controlled variable methods were used to compare network performance. Parameter settings are detailed in Table 1 .

2.6 Evaluation Metrics

F1-score (Macro_{f1}) and accuracy (Accuracy) suitable for multi-class evaluation were adopted. The metrics are defined as:

$$Macro\_f1 = \sum_{i=1}^{N} f1\_scorei \times 100\%$$

where N is the number of classes and $f1\_scorei$ is the F1-score for class i.

$$Accuracy = \frac{Num_{TRUE}}{Num_{TOTAL}} \times 100\%$$

where Num_{TRUE} is the number of correctly predicted samples and Num_{TOTAL} is the total number of validation samples.

---

3 Results and Analysis

3.1 VGG Series Algorithm Performance

Early stopping resulted in varying training and validation processes across models. Figures 6 [Figure 6: see original paper] and 7 [Figure 7: see original paper] show validation accuracy and loss values for different experimental schemes. In the legends, T-M-VGG (train) indicates training set results, while T-M-VGG (val) indicates validation set results.

For comprehensive evaluation, model size and trainable parameters were included as auxiliary metrics, as shown in Table 2 .

Figures 6 and 7 demonstrate that all methods in this study show increasing accuracy and decreasing loss with iteration count, plateauing after a certain number of epochs. Tr-L-VGG16, with frozen convolutional layers and few trainable parameters in the custom global average pooling layer, exhibits slow loss reduction and gradual accuracy improvement, resulting in poor recognition performance.

VGG16 initialized with pre-trained parameters converges quickly, reaching stability at 7 iterations. The method from reference [9], using parallel transfer learning VGG16 for yak recognition, approaches convergence at 14 iterations in our experimental environment, with loss values around 0.5. Table 2 reveals that Tr-L-VGG16 has the fewest trainable parameters and lowest accuracy. VGG16 achieves the largest model size at 502.48 MB, impacting loading speed and storage overhead. The method from reference [9] yields a model size of 166.33 MB with 88.03% accuracy, showing that parallel transfer learning structures increase model size. In contrast, T-M-VGG achieves 96.01% accuracy with a model size of only 70.75 MB, representing improvements of nearly 3 and 68 percentage points over VGG16 and Tr-L-VGG16, respectively, and an 8 percentage point accuracy gain with approximately 96 MB size reduction compared to reference [9], demonstrating the effectiveness of parallel dilated convolutions and feature pyramids.

3.2 Other Algorithm Performance

Figure 6 shows that network structures from references [10], [11], and [25] exhibit similar trends under our hyperparameter settings, converging near 83% accuracy. FaceNet and InceptionV3 converge more rapidly. Table 2 indicates that T-M-VGG maintains advantages in both accuracy and trainable parameters compared to references [10], [11], and [25]. While InceptionV3 and FaceNet show similar accuracy (within ~1 percentage point), their model sizes are approximately 100 MB and 348 MB larger, respectively. MobileNetV3-Large and MobileNetV3-Small, though compact, achieve accuracy nearly 2 percentage points lower than T-M-VGG.

3.3 Visualization of Recognition Effects

To better evaluate model robustness, 12 images were randomly selected from the dataset and subjected to partial occlusion. Occlusion masks were applied to generate pseudo-occluded images, which were then fed into the T-M-VGG model for prediction. Visualization results are shown in Figure 8 [Figure 8: see original paper].

For yak IDs 1, 42, 49, 76, 83, 161, 168, 172, and 192, occlusions in non-facial regions (body, ear tags, background) resulted in only ID 76 being misclassified as 73, confirming that the model learns facial rather than environmental features. For IDs 75 and 78, occluding minor facial regions (non-critical feature areas) still yielded correct predictions. However, ID 180 was misclassified when occlusions altered prominent facial features, representing expected model behavior. Among 12 categories, T-M-VGG correctly predicted 10, achieving 83.33% occlusion test accuracy.

4 Conclusion and Outlook

This study developed parallel dilated convolution modules and an improved feature pyramid structure, integrating them with transfer learning to create a yak face recognition algorithm. Key conclusions are:

1. The proposed T-M-VGG model achieves 96.01% recognition accuracy on a dataset of 194 yaks (38,800 images) with a model size of 70.75 MB.
2. Comparative validation on our dataset demonstrates the superiority of combining multiscale fusion with transfer learning for yak face recognition.
3. From perspectives of accuracy and model size, the T-M-VGG architecture with 256$\times$256 input resolution reduces storage requirements while improving accuracy, meeting practical identification needs.

Current limitations include: (1) The need to increase yak categories and sample diversity to optimize recognition of yaks with highly similar facial features; (2) Investigation of how physiological changes across growth cycles affect recognition; (3) Integration with object detection algorithms for real-time recognition.

Future work will address these challenges to further enhance model performance and applicability.

References

[1] QIN X, SONG G. Pig face recognition algorithm based on bilinear convolution neural network[J]. Journal of Hangzhou Dianzi University (Natural Sciences), 2019, 39(2): 12-17.

[2] HE Y, LI B, ZHANG F, et al. Pig face recognition based on improved YOLOv3[J]. Journal of China Agricultural University, 2021, 26(3): 53-62.

[3] LIU Z, HE D. Recognition method of cow estrus behavior based on convolutional neural network[J]. Transactions of the CSAM, 2019, 50(7): 186-193.

[4] YANG Q, XIAO D, ZHANG G. Automatic pig drinking behavior recognition with machine vision[J]. Transactions of the CSAM, 2018, 49(6): 232-238.

[5] ZHANG H, ZHOU L, LI Y, et al. Sheep face recognition method based on improved mobilefacenet[J/OL]. Transactions of the CSAM, 1-10 [2022-05-13].

[6] WEI Z. Research on iris recognition technology of imperfect bull' s eye based on the combination of global and local features[D]. Nanjing: Southeast University, 2017.

[7] ZHAO K, HE D. Recognition of individual dairy cattle based on convolutional neural networks[J]. Transactions of the CSAE, 2015, 31(5): 181-187.

[8] HE D, LIU J, XIONG H, et al. Individual identification of dairy cows based on improved YOLOv3[J]. Transactions of the CSAM, 2020, 51(4): 250-260.

[9] CHEN Z, HUANG C, YANG B, et al. Parallel convolutional neural network yak face recognition algorithm based on transfer learning[J]. Journal of Computer Applications, 2020, 41(5): 1332-1336.

[10] HANSEN M F, SMITH M L, SMITH L N, et al. Towards on-farm pig face recognition using convolutional neural networks[J]. Computers in Industry, 2018, 98: 145-150.

[11] MARSOT M, MEI J, SHAN X, et al. An adaptive pig face recognition approach using convolutional neural networks[J]. Computers and Electronics in Agriculture, 2020, 173: 105386.

[12] KUMAR S, SINGH S K, SINGH R, et al. Deep learning framework for recognition of cattle using muzzle point image pattern[J]. Measurement, 2018, 116: 1-17.

[13] JUNG D H, KIM N Y, MOON S H, et al. Deep learning-based cattle vocal classification model and real-time livestock monitoring system with noise filtering[J]. Animals, 2021, 11(2): 357.

[14] SALAMA A, HASSANIEN A E, FAHMY A. Sheep identification using a hybrid deep learning and Bayesian optimization approach[J]. IEEE Access, 2019, 7: 9267-9275.

[15] WADA N, SHINYA M, SHIRAISHI M. Pig face recognition using eigenspace method[J]. ITE Transactions on Media Technology & Applications, 2013, 1(4): 376-381.

[16] RASHID M, GU X, LEE Y J. Interspecies knowledge transfer for facial keypoint detection[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2017: 1600-1609.

[17] WANG Z, BOVIK A C, SHEIKH H R, et al. Image quality assessment: From error visibility to structural similarity[J]. IEEE Transactions on Image Processing, 2004, 13(4): 600-612.

[18] CHEN G, ZHAO S, CAO L, et al. Corn plant disease recognition based on migration learning and convolutional neural network[J]. Smart Agriculture, 2019, 1(2): 34-44.

[19] LI M, WANG J, LI H, et al. Method for identifying crop disease based on CNN and transfer learning[J]. Smart Agriculture, 2019, 1(3): 46-55.

[20] LIU S, HUANG D. Receptive field block net for accurate and fast object detection[C]//The European Conference on Computer Vision. Cham, Switzerland: Springer, 2018: 385-400.

[21] SZEGEDY C, WEI L, JIA Y, et al. Going deeper with convolutions[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2015: 1-9.

[22] LIN T Y, DOLLAR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2017: 2117-2125.

[23] LIU S, QI L, QIN H, et al. Path aggregation network for instance segmentation[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2018: 8759-8768.

[24] TAN M, PANG R, LE Q V. Efficientdet: Scalable and efficient object detection[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2020: 10781-10790.

[25] FAN X, ZHOU J, XU Y, et al. Corn disease recognition under complicated background based on improved convolutional neural network[J]. Transactions of the CSAM, 2021, 52(3): 210-217.

[26] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J/OL]. arXiv:1409.1556.2014.

[27] HOWARD A, SANDLER M, CHEN B, et al. Searching for Mobilenetv3[C]//The IEEE International Conference on Computer Vision. Piscataway, USA: IEEE, 2019: 1314-1324.

[28] SZEGEDY C, VANHOUCKE V, IOFFE S, et al. Rethinking the inception architecture for computer vision[C]//The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2016: 2818-2826.

Note: Figure translations are in progress. See original paper for figures.