Dog Nose-Print Verification Using Deep Metric Learning

Wei Zhang*
Xidian University
Xi’an, Shannxi,China
zhangwei3.0@stu.xidian.edu.cn

Jinwei Zhang
Xidian University
Xi’an, Shannxi,China
fermioner@outlook.com

Yifan Zhao
Xidian University
Xi’an, Shannxi,China
21171213895@stu.xidian.edu.cn

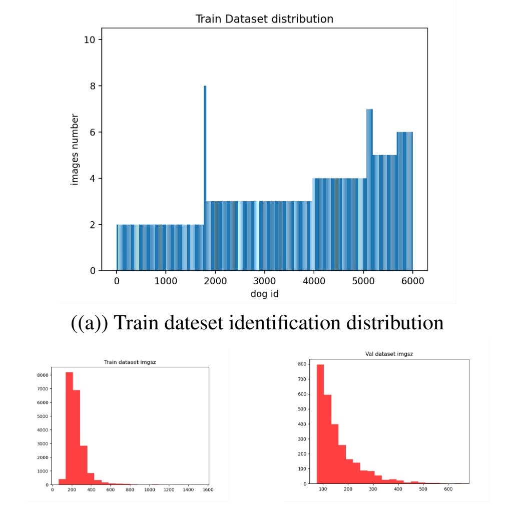
Abstract

We propose a representation learning network by combining some work in the fields of metric learning, face recognition, and ReID. We use Google’s proposed BiT as a feature extractor for images and combine multiple stage feature maps to preserve richer detail features, and use metric loss and classification loss to learn jointly to obtain better performance. The use of pseudo-labeling techniques on the validation dataset can further improve the performance, and finally the use of fusion based on ranking results can overcome the differences in the inference distribution of different models. Based on these techniques we achieve the score of 0.9406 on the validation dataset and the socre of 0.8603 on the test dataset. Our code is available in `https://github.com/kali20gakki/CVPR2022-Biometrics-Workshop-Pet-Biometric-Challenge`

1. Introduction

Pet identification is a challenging problem, and this pet biometric challenge is dedicated to finding solutions that balance accuracy, cost and usability well. In this biometric challenge, we argue that there are several key problems that need to be addressed.

The first challenge comes from the dataset. The information on each dog identity in the dataset is rare. The size of the training samples corresponding to each dog identity information is very small compared to the huge number of dog identities. As shown in Figure 1(a), there are no more than 8 training samples corresponding to each dog identity in the training dataset. In addition, the images of the dog nose print region in our dataset are from the object detection algorithm, which causes a large variation in the size of the region of interest(ROI) of the dog nose print. The minimum length of each ROI varies from 70 to 600 pixels. As shown in Figure 1(b) and Figure 1(c), there is a large difference in the image quality resolution distribution between the training and validation datasets. The training dataset is concentrated around 200 × 200, and the validation dataset is concentrated around 100 × 100. Also, the quality of the images from the test dataset is also much lower than that of the training and validation datasets. In general, the real-world dog nose print images from the real world have very complex patterns. We need to improve the robustness and effectiveness of the model in terms of feature extraction and identity discrimination of the model.



((a)) Train dateset identification distribution

((b)) Train dataset image size ((c)) Validation dataset image size

Figure 1. Statistics of the dataset.

The second challenge comes from the task itself, where our task is 1 vs 1 identity verification of dog nose print images. This identity verification problem can be viewed as a metric learning task. Simple classification tasks are not competent in scenarios where individual identity information is lacking and do not perfectly satisfy the criterion that the maximum intra-class distance is less than the minimum inter-class distance for a given metric space. This criterion is often difficult to learn features because dog nose

[*] Wei Zhang is the leader of this team.

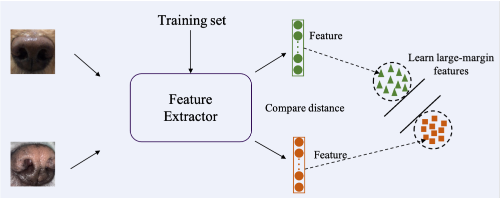


Figure 2. Dog Nose-Print identity verification

prints exhibit inherently large intra-class differences and high inter-class similarity. The key to the dog nose print image verification task is to learn discriminative large-margin features.

Based on the above proposed problems, our solutions are summarized as follows:

- Muti-Similarity Loss and Sub-center ArcFace Loss are introduced into our model. The objective is to learn an embedding space in which the embedding vectors of similar samples are drawn closer together and the embedding vectors of different samples are pushed farther apart.
- Pseudo labeling techniques are introduced into our training. The pseudo labeling technique improves the model performance in a supervised process by leveraging the unlabeled data.
- The idea of model ensemble is applied to our inference phase. ensemble learning can effectively reduce the variance and improve the robustness of the model.

2. The proposed approach

2.1. Overview

We use a metric learning approach to learn reliable representations. As shown in the Figure 3, the backbone network uses BiT-101x1 [3] pre-trained on ImageNet1K for feature extraction. We use the feature maps from the last two stages of the backbone network, and the stride of the last stage is set to 1 to retain more detailed features. Instead of using a global averaging pool for the output feature maps, we use generalized averaging pooling (GeM Pooling) to obtain two global feature vectors. The two global vectors are concatenated together to obtain the image embedding after the nonlinear mapping, and this embedding is used to compute the Multi-Similarity Loss. Finally, the embedding after BN-Neck is used to calculate ArcFace Loss. The loss function of the model can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_{\text{MS}} + \gamma \mathcal{L}_{\text{Arc}} \tag{1}$$

where $\mathcal{L}_{\text{MS}}$ represents Muti-Similarity Loss and $\mathcal{L}_{\text{Arc}}$ denotes Sub-center ArcFace Loss. γ is a hyperparameter to balance the different loss functions.

In the inference phase we use the embedding used in the calculation of Multi-Similarity Loss to obtain the cosine similarity of the two images

2.2. Loss fuction

Muti-Similarity Loss. Muti-Similarity Loss [6] considers the self-similarity ,negative relative similarity and positive relative similarity . Multi-Similarity loss consider all three perspectives by implementing a new pair weighting scheme using two iterative steps: mining and weighting. (i) informative pairs are first sampled by measuring positive relative similarity; and then (ii) the selected pairs are further weighted using self-similarity and negative relative jointly.

We first select informative pairs by computing positive relative similarity, which measures the relative similarity between negative↔positive pairs having a same anchor. Specifically, a negative pair is compared to the hardest positive pair (with the lowest similarity), while a positive pair is sampled by comparing to a negative one having the largest similarity. Formally, assume $\boldsymbol{x}_i$ is an anchor, a negative pair $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$ is selected if S_{ij} satisfies the condition:

$$S_{ij}^{-} > \min_{\boldsymbol{y}_k = \boldsymbol{y}_i} S_{ik} - \epsilon, \tag{2}$$

where ϵ is a given margin. If $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$ is a positive pair, the condition is:

$$S_{ij}^{+} < \max_{\boldsymbol{y}_k \neq \boldsymbol{y}_i} S_{ik} + \epsilon. \tag{3}$$

For an anchor $\boldsymbol{x}_i$, we denote the index set of its selected positive and negative pairs as $\mathcal{P}_i$ and $\mathcal{N}_i$ respectively.

Pair mining with positive relative similarity can roughly select informative pairs, and discard the less informative ones. Specifically, given a selected negative pair $\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \in \mathcal{N}_i$, its weight w_{ij}^{-}can be computed as:

$$\begin{aligned} w_{ij}^{-} &= \frac{1}{e^{\beta(\lambda - S_{ij})} + \sum_{k \in \mathcal{N}_i} e^{\beta(S_{ik} - S_{ij})}} \\ &= \frac{e^{\beta(S_{ij} - \lambda)}}{1 + \sum_{k \in \mathcal{N}_i} e^{\beta(S_{ik} - \lambda)}} \end{aligned} \tag{4}$$

and the weight $w_{ij}^{+}$of a positive pair $\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \in \mathcal{P}_i$ is:

$$w_{ij}^{+} = \frac{1}{e^{-\alpha(\lambda - S_{ij})} + \sum_{k \in \mathcal{P}_i} e^{-\alpha(S_{ik} - S_{ij})}}, \tag{5}$$

where α, β, λ are hyper-parameters.

In Eq. (4), the weight of a negative pair is computed jointly from its self-similarity by $e^{\beta(\lambda - S_{ij})}$– self-similarity,and its relative similarity -negative relative similarity , by comparing to its negative pairs. Similar rules are applied for computing the weight for a positive pair, as in In Eq. (5).

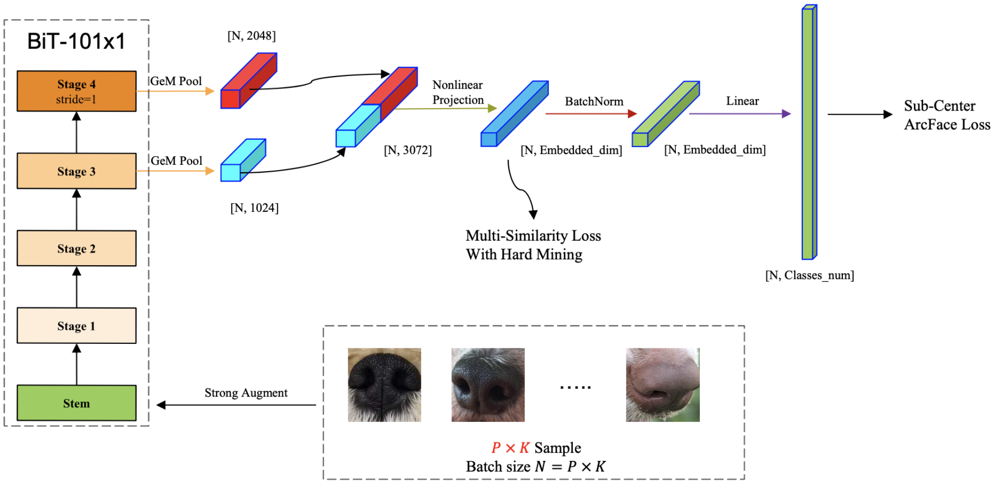


Figure 3. The structure of our model.For every mini-batch, we randomly choose P classes, and then randomly sample K instances from each class(each dog identity)

Finally, Muti-Similarity Loss is formulated as,

$$\mathcal{L}_{MS} = \frac{1}{m}\sum_{i=1}^{m}\left\{\frac{1}{\alpha}\log\left[1+\sum_{k\in\mathcal{P}_i} e^{-\alpha(S_{ik}-\lambda)}\right]\right. \\ \left. + \frac{1}{\beta}\log\left[1+\sum_{k\in\mathcal{N}_i} e^{\beta(S_{ik}-\lambda)}\right]\right\} \tag{6}$$

where $\mathcal{L}_{MS}$ can be minimized with gradient descent optimization, by simply implementing the proposed iterative pair mining and weighting.

Sub-center ArcFace Loss. Sub-center ArcFace loss [1] is an extended version of ArcFace loss [2] developed from the softmax loss function. ArcFace is directly maximizing the classification bounds in the angle space θ.ArcFace assumes that the training data is clean,which is not possible in large-scale training data. Therefore a modified version of the ArcFace loss is introduced into our model.

As shown in Figure 3, We set the number of sub-centers to K_s. Based on a ℓ_2 normalisation step on both embedding feature $\mathbf{x}_i \in \mathbb{R}^{3072\times 1}$ and all sub-centers $W \in \mathbb{R}^{C\times K_s\times 3072}$, we get the subclass-wise similarity scores $\mathcal{S} \in \mathbb{R}^{C\times K_s}$ by a matrix multiplication $W^T\mathbf{x}_i$. Then, we employ a max pooling step on the subclass-wise similarity score $\mathcal{S} \in \mathbb{R}^{C\times K}$ to get the classwise similarity score $\mathcal{S}' \in \mathbb{R}^{C\times 1}$. The proposed sub-center ArcFace loss can be formulated as:

$$\mathcal{L}_{\text{Arc}} = -\log \frac{e^{s\cos\left(\theta_{i,y_i}+m\right)}}{e^{s\cos\left(\theta_{i,y_i}+m\right)} + \sum_{j=1,j\neq y_i}^{C} e^{s\cos\theta_{i,j}}} \tag{7}$$

where $\theta_{i,j} = \arccos\left(\max_k\left(W_{i_n}^T\mathbf{x}_i\right)\right), k \in \{1,\cdots,K_s\}$. m is the angular margin parameter, s is the feature re-scale parameter, and C is the total class number.

2.3. BNNeck

Two different loss function designs are used in our study. The simultaneous use of both loss functions suffers from the problem of not converging at the same time [5], which can cause a degradation in model performance. We introduce a structure called BNNeck to to separate the two losses in order to alleviate this problem. The specific structure is implemented by introducing BatchNorm layer as shown in Figure 3.

2.4. Pseudo labeling

First, the model is trained on the labeled data, and then the trained model is used to predict the labels of the unlabeled data to create pseudo-labels [4]. In addition, the labeled data and the newly generated pseudo-labeled data are combined as the new training data.

We add the top p pairs of inference results on the validation set to the training set. The pseudo-labeling will be made more and more accurate through multiple rounds of iterations.

2.5. Ensemble

Ensemble Learning builds and combines multiple models to accomplish learning tasks. We fuse the inference results of multiple models. In particular, simply averaging the inference results of multiple models may not be an appropriate approach. When the means of the inference distributions of two models differ significantly, samples with high scores in inference distributions with large means may be pulled down by results with inference distributions with small means. Therefore, we introduce the ensemble strategy called sorted averaging. We first sort the predicted results.

Then we average the sorted numbers. Finally, we normalize the average sorted number.

2.6. Inference

We determine whether different pairs of images belong to the same dog identity by comparing the cosine similarity distance between different individual dog noses. The embedding used to calculate Muti-Similarity Loss is used to obtain the cosine similarity $\mathcal{S}_{ij}$ as shown in Eq. (8).

$$\mathcal{S}_{ij} = \frac{\boldsymbol{e}_i^T \boldsymbol{e}_j}{\|\boldsymbol{e}_i\| \|\boldsymbol{e}_j\|} \tag{8}$$

where the $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ represent the embedding of different images.

3. Experiments

3.1. Settings

Hyperparameters. For every mini-batch, we randomly choose P classes, and then randomly sample K instances from each class(each dog identity) with $K = 6$ for all datasets in all experiments.The model is trained using AdamW's pytorch implementation with the learning rate 5×10^{-5} and the weight decay 0.05. The batch size N of model is 150. The total epochs are 250 and the wramup epochs are 10. After the warm-up phase, we use cosine decay learning rate scheduling.

ϵ in Eq. (2) and Eq. (3) is set to 0.1 and the hyperparameters in Eq. (6) are: $\alpha = 2, \lambda = 0.5, \beta = 50$. For $\mathcal{L}_{Arc}$, the hyperparameters in Eq. (7) are : $m = 28.6$, $s = 64$ and $K_s = 2$. γ is set to 0.01 in Eq. (1).

Data augmentation. All the input images were resized to 224 × 224. For data augmentation,We used ColorJitter, RandomBrightnessContrast, ShiftScaleRotate, Cutout, ImageCompression and GaussianBlur for the training dataset. For the test dataset we do not perform any data augmentation.

Pseudo labeling and Ensemble The pseudo-labeling technique was only used for the validation dataset and was not applied to the test dataset. In the model ensemble phase, we select models by their performance on the validation dataset

3.2. Performence

As shown in Table 1, We experimented on the validation dataset and find that the backbone network of the BiT series is much better than other CNNs and Vision Transformer. The best result in terms of metric loss is Multi-Similarity Loss, and classification loss Sub-Center Arcface Loss is slightly better than ArcFace Loss in terms of model performance. Performance. Better performance can be obtained by using BNNeck to separate the two losses. The higher the number of pseudo labels the better the performance on the validation dataset, but the same number of pseudo labels performs poorly on the test dataset. We believe that the severe overfitting occurs on the validation dataset. Therefore, the number of pseudo labels should be moderate and we believe that a number of about 700 pseudo-labels is appropriate. In terms of model fusion, we use a fusion based on sorted averaging which is much better than averaging the predictions directly. We find that the same model has a large difference in results between the validation dataset and the test dataset. The quality of the images in the test dataset was found to be lower than the training dataset and the test dataset by visualizing the dataset. Therefore, we use the stronger data augmentation approach to reduce the number of pseudo labels and the set of models to alleviate the overfitting problem.

Method	Backbone	Score
Triplet Loss	ResNet50	0.8407
Triplet Loss	BiT-101x1	0.8568
Multi-Similarity Loss	BiT-101x1	0.8632
+Sub-Center ArcFace Loss	BiT-101x1	0.8701
+BNNeck	BiT-101x1	0.8923
+PxK Sample	BiT-101x1	0.9012
+Pseudo label	BiT-101x1	0.9299
+Ensemble by sort	BiT-101x1	0.9406

Table 1. Leaderboard scores for different model variants on the validation dataset

References

[1] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*, page 741–757, Berlin, Heidelberg, 2020. Springer-Verlag. 3

[2] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[3] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pages 491–507. Springer, 2020. 2

[4] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2013. 3

[5] Hao Luo, Wei Jiang, Youzhi Gu, Fuxu Liu, Xingyu Liao, Shenqi Lai, and Jianyang Gu. A strong baseline and batch normalization neck for deep person re-identification. *IEEE Transactions on Multimedia*, 22(10):2597–2609, 2019. 3

[6] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair

weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019. 2