1st Place Solution to Pet Biometric Challenge 2022

HUST423

Abstract

This paper presents the 1st place solution to the Pet Biometric Challenge (CVPR 2022 Workshop). The solution is based on the ensemble of models with effective training techneques, including various advanced CNN backbones, data augmentation and loss functions. Moreover, a novel and effective Instance-level Contrastive Loss (ICL) is proposed for the pet nose retrieval task. With above contributions, we achieve 91% rank-1 accuracy in Phase 2.

1. Introduction

Nowadays pattern identification based on vision has been widely applied in human biometrics. As a contrast, pet identification is still not fully explored due to the lack of labeled pet images. This task is basically an instance-level retrieval task, which aims at pinpointing pet noses images similar to the query image. The pet nose identification is closely related to person re-identification, which mainly focuses on fine-grained image texture.

In this report, we will introduce the training techniques used in the pet nose retrieval competition. Many techniques used in person re-identification can be applied in pet nose identification due to the similarity between the two tasks. We adopt various backbones, including advanced CNN-based models and vision transformer based models, to get discriminative pet nose representation.

Simple but effective data augmentation methods are applied to further boost the robustness of our deep models. We also explore tons of loss functions [2, 5, 7, 11, 12] to facilitate discriminative feature learning, including classification loss, advanced memory-based contrastive loss and pair-wise ranking losses. Moreover, we propose a novel Instance-level Contrastive Loss (ICL), which conducts well-designed hard example mining. Since the challenge is regarded to 1 vs 1 identity verification which requires pair-wise similarity, we also propose a siamese network to get the pair-wise similarity.

Many effective training tricks [6, 8–10], which have been well explored in various computer vision tasks, are also enrolled into our models.

Our contributions can be summurazied as follows:

- We introduce a bag of tricks which are proven to be effective for pet nose-print images retrieval.
- We propose a novel and effective training loss function — Instance-level Contrastive Loss (ICL), which can fully mine hard examples.
- We propose two methods which are contrastive learning and Siamese network methods. Both methods utilize instance-level memory to mine hard samples for enhancing feature discrimination.

2. Method

2.1. Overview

We adopt two methods for pet identification, which are feature learning by contrastive loss and image validation by siamese framework. The two methods are illustrated in the following.

2.2. Baseline Network

We adopt several large CNN networks including SEResNet101 [4], ResNetXt101 [13] and ResNetSt101 [14] as backbones. IBN extension [8] is added to ResNet101 and ResNetXt101. The input image size is set as 256 in the first few training epochs and 384 in the rest of training epochs. Generalized Mean Pooling (GeM) [9] is utilized in the pooling procedure. We choose 3.5×10^{-4} as the initial learning rate and set weight decay as 5×10^{-4}. Adam optimizer is utilized for training.

Several loss functions are selected in our method including OIM loss [12], Weighted-Instance constrastive loss, Proxy NCA [5], Hard Triplet Loss [3], contrasitve loss and MSLoss [11].

2.3. Data Augmentation

Some tricks in data augmentation are explored. In the first phase of the challenge, we use autoaugment method [1]. It can search appropriate composed augmentation including flip, rotate, bright/color change $\cdots$ etc. In the second phase of the challenge, considering the blurred images in the test data which show pseudo-shadow brought by JPEG compress, we add a JPEG compressing, gaussian blur and motion blur data augmentation methods.

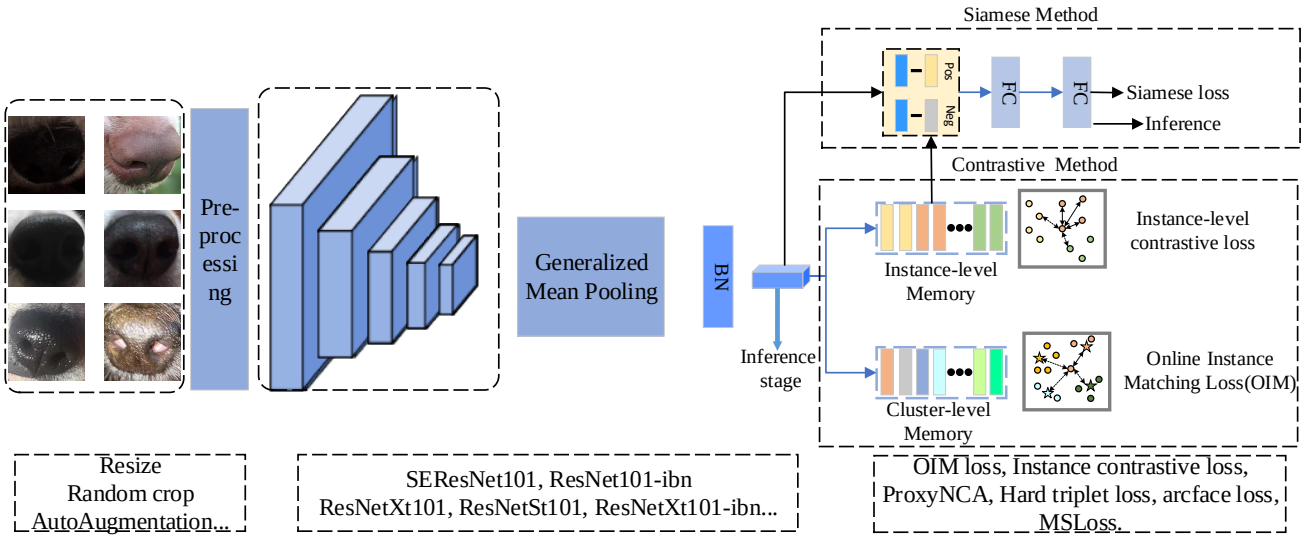

Figure 1. Overall Framework

2.4. Contrastive Learning Method

In this module, we illustrate the loss functions used in our methods, which include Online Instance Matching (OIM) loss, instance-level contrastive loss, contrastive loss and etc. The OIM loss is adopted for leaning the relationship across different ids. A memory bank is set to aggregate the corresponding id features in each iteration, and the centroid c_j of each class in the memory bank can be updated by the following equation:

$$\boldsymbol{c}_j \leftarrow m\boldsymbol{c}_j + (1-m)\bar{f}\left(x_i^t \mid E[\theta]\right) \tag{1}$$

where c_j is the j-th entry of the memory, storing the updated feature centroid of class j and m is the momentum factor of updating centroids. Therefore the similarity between the sample and class centroids can be measured by dot product. The OIM loss can be denoted as:

$$L_{\text{oim}} = -\sum_{i=1}^{N}\sum_{k=1}^{K}\log\frac{\exp\left(< f\left(x_{i,k} \mid \theta\right), c^{+} >\right)/\tau}{\sum_{j=1}^{J}\exp\left(< f\left(x_{i,k}^{t} \mid \theta\right), c_j >\right)/\tau} \tag{2}$$

where c^+ is the class centroid which has the same class as sample x_i has, and τ is the temperature parameter.

The OIM loss is a proxy-based loss, which can maintain the discrimination of each identity class. However it neglects the relationship between samples in the dataset. To learn the relationship between samples, we propose an instance-level contrastive loss, which can reduce the intra-class distance and increase the inter-class distance. In each epoch, we construct a instance memory $\mathcal{V}' \in R^{d\times N}$ containing all the instance features, where d is the feature dimension and N is the total number of images. Given an images x_i and its ID Label y_i, we consider all instances with the same label y_i as positive samples and others as negative samples. Considering the large number of negative samples, we can only select a few of them to train. The instance-level contrastive loss can be denoted as:

$$\mathcal{L}_{\text{ins}} = -\sum_{i=0}^{N}\sum_{j=0}^{N_{pos}}\log\frac{\exp\left(< v_j \cdot E\left(x_i\right) > /\tau_v\right)}{\sum_{k=1}^{N_{neg}+1}\exp\left(< v_k \cdot E_{id}\left(x_i\right) > /\tau_v\right)} \tag{3}$$

Where N_{pos} denotes positive samples, and N_{neg} denotes the selected hard negative samples. In our experiment, we sort negative samples according to their similarity with x_i and select 500 of them with the largest similarity. Moreover, to mine hard positive samples, we adopt a weighted instance-level contrastive loss to learn more discriminate features by weighting different positive samples:

$$\mathcal{L}_{\text{w-ins}} = -\sum_{i=0}^{N}\sum_{j=0}^{N_{pos}} w_j \log\frac{\exp\left(< v_j, E\left(x_i\right) > /\tau_v\right)}{\sum_{k=1}^{N_{neg}+1}\exp\left(< v_k, E_{id}\left(x_i\right) > /\tau_v\right)} \tag{4}$$

where weights $\mathbf{w} = softmax([< v_1, E\left(x_i\right) >, ..., < v_{N_{pos}}, E\left(x_i\right) >])$

Finally, we combine the OIM loss and weighted instance loss as the final loss. We also combine the OIM with other pair-wise loss such as: triplet loss, contrastive loss, and MS loss, etc.

2.5. Siamese Methods

We use two methods to train the siamese branch, which are divided into the online method and the offline method respectively. The online method is to select a random pair of positive and negative samples in each batch, while the offline method is to select top 100 hard samples in memory

Table 1. Alative Results

Loss Functions	IBN	Non-local	mAP	Rank1	Rank5	Rank10
CE + Triplet			83.1	87.0	91.0	92.0
OIM			87.2	90.0	93.8	94.8
OIM	✓	✓	90.3	92.0	95.6	96.0
OIM + CE	✓	✓	90.0	91.6	95.2	96.3
OIM + CE	✓	✓	89.4	91.0	95.2	95.7
OIM + ICL(HN)	✓	✓	91.4	92.5	96.4	96.8
OIM + ICL(HN+PW)	✓	✓	**92.5**	**94.0**	**96.7**	**98.1**

bank and then randomly select a random pair of positive and negative samples. We describe the procedure as below.

Assume a sample feature $v_i \in R^{1\times d}$ is produced from GeM. Then we select a positive sample feature $v_j \in R^{1\times d}$ which has the same id as v_i, and a negative sample feature $v_k \in R^{1\times d}$ which comes from the different identity. Then we compute the substraction of (v_i, v_j) and (v_i, v_k) respectively, and feed them into the siamese branch, which is composed of several fully connect layers to get similarity logits. A sigmoid function is used to normalize the logits into $(0, 1)$ to get similarity score. This can be represented as:

$$score = Sigmoid(FC(abs(v_i - v_j)))$$

. where FC is the fully connect layer. Then we use a BCE loss to optimize network which can be denoted as:

$$L_{bce} = -(y\log(score) + (1-y)\log(1-score))$$

$y \in \{0, 1\}$ is ground truth of pair features. $y = 1$ denotes the same id and different id otherwise.

2.6. Inference and Ensemble

2.6.1 Inference.

A pair of image X_i and X_j is fed into our backbone, then the corresponding features v_i and v_j are extracted. The pair-wise cosine similarity is calculated. We also compute the subtraction of v_i and v_j and fed the subtract vector into siamese branch described in Section 2.3. to obtain the siamese similarity. The above two similarity are used to ensemble final result.

2.6.2 Ensemble

Model Ensemble We compute the average weights of the model of last several epochs to obtain the ensembled model. The output of the ensembled model is taken as the final predicted result.

Score Ensemble We get different similarity scores with multiple backbones and use their mean value as the result.

3. Experiment

3.1. Training Detail

We divide our training procedure into three stages in a contrastive learning way. In the first stage, we train our network on 224×224 images for the first 70 epoches, and then fintune the network on 384×384 for the last 30 epoches. Then, we generate pseudo labels on validation dataset by clustering methods, and finetune the model on both training and validation datasets. In the siamese branch, we first use the model pretrained by contrastive learning method as weight initialization. And then finetune the network on both traning and validation datasets.

3.2. DataSet

To conduct ablative experiments, we divide the dataset in the fist phase into training set and validation set. Specificly, 90% identities are randomly selected as the training set and the rest 10% identities are chosen as the validation set.

3.3. Ablation Study

In this part, we evaluate the effectiveness of loss functions and modules, including IBN [8] and Non-local [10]. The experiment is conducted on the above dataset. In Table. 1, "CE" and "Triplet" denote cross-entropy loss and triplet loss, respectively. "ICL(HN)" means that ICL only mines the hard negative pairs. "ICL(HN+PW)" is the proposed weighted ICL which simultanuously mines the hard negative samples and adopts adaptive weights on positive pairs. It can be observed that: 1) IBN and Non-local are beneficial. 2) Memory-based contrastive loss are extremly helpful. OIM loss brings 4.1% and 3.0% improvement of mAP and Rank-1 accuracy. It is worth noting that the proposed weighted ICL improve mAP and Rank-1 accuracy by 2.2% and 2.0%, respectively.

4. Conclusion

In this paper, we present our methods for Pet Biometric Challenge 2022 in detail. For the pet identification task,

we adopt two solutions, feature learning by contrastive loss and image validation by siamese framework. We adopt a class-level contrastive loss and a instance-level constrastive loss (ICL) for feature learning, which is used to mine hard positive and negative samples for training. To learn the relationship between different samples, we propose a weighted instance contrastive loss (weighted ICL) for enhancing the training for hard samples. Moreover, we select hard positive and negative pairs from the instance-level memory for the siamese framework. Finally, we ensemble all trained methods to produce the final result.

References

[1] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 1

[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*, pages 4690–4699, 2019. 1

[3] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 1

[4] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 1

[5] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3238–3247, 2020. 1

[6] 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*, 2019. 1

[7] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1

[8] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 464–479, 2018. 1, 3

[9] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018. 1

[10] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 1, 3

[11] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019. 1

[12] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3415–3424, 2017. 1

[13] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 1

[14] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 1