1st Place Technical Report for MMSports '23 Player Reidentification Challenge

Jun Yu[1], Renda Li[1], Renjie Lu[1], Leilei Wang[1], Shuoping Yang[1], Lei Wang[1]
Minchuan Chen[2], Qingying Zhu[2], Shaojun Wang[2], Jing Xiao[2]
[1]University of Science and Technology of China, [2]Ping An Technology

1. Introduction

This technical report mainly introduces the technical details of the second phase of the MMSports 2023 Player Re-Identification challenge. In the first phase, for this challenge, we have published a paper "Exploring Loss Function and Rank Fusion for Enhanced Person Re-identification". It is worth noting that although we tried many new methods in the second stage, their accuracy did not improve significantly on the challenge set. Therefore, our second-stage approach is identical to that described in the first-stage paper, with only some details explained in this technical report.

2. Implementation Details

Our code is improved based on the repository of last year's championship program. The use of training and inference scripts is explained in the GitHub repository we submitted.

3. Dataset

The DeepSportradar Re-Identification dataset comes from short tracking sequences of basketball games, each sequence is composed by 20 frames. For the validation and test sets, the query images are persons taken at the first frame, while the gallery images are identities taken from the 2nd to the last frame.

You can download the data set using the `download.py` script, or you can download it from the official link. In addition, you need to execute `preprocess_data.py` to preprocess the data to obtain the csv partition file of the training/validation data set. These two scripts only need to be executed once. It should be noted that a new challenge set will be adopted in 2023, so the challenge set needs to be re-downloaded using the script in the official repository.

4. Method

The framework for Re-Identification using CLIP proposed in the repository is a strong baseline, and we have followed their overall model architecture. We use two weight-sharing image encoders as the Siamese network. Specifically, several groups of query image and its corresponding gallery image will be encoded into feature vectors. We use InfoNCE loss function to maximize the cosine similarity between the query image and its corresponding gallery image, and minimize the cosine similarity between the query image and other gallery images.

4.1. Data augmentation

As shown in the experiments in our paper, we finally used two effective data augmentations, random erasing and sharpening, with the hyperparameter settings consistent with those in the code. These data augmentations are used during training, but no TTA strategy is used during inference.

4.2. Post process

We mainly use two methods for post-processing, k-reciprocal re-ranking and model fusion. The hyperparameter settings in k-reciprocal re-ranking are consistent with those in the `train_our_post.py` script. When training using this script, the results of two k-reciprocal re-ranking implementations will be output. Our implementation uses hard weight. For specific ablation experiments, please refer to the results in the paper. For the model fusion part, we used three predicted distance matrices of eva_giant, eva02 and beitv2. The pre-training weights used by these three models can be found in the paper or the `train_our_post.py` script. First, we use the `sim_fuse.py` and `dissim_fuse.py` scripts to obtain two fusion results for the three distance matrices, and then use the `sim_fuse.py` script to fuse the two results.

4.3. Training and Evaluation

We use `train_our_post.py` for training of the main models and `train_our_post_resnet.py` for fast iteration of the method with small models. We use four RTX 3090 for training. Generally speaking, the batchsize is set to 4x32, but for larger models such as eva_giant or beitv2, the batchsize is adjusted to 4x8 or 4x4. For all models, the total number of epochs of training is set to 8 , take the best weight in the validation set. The maximum learning rate is 4x10e-5, the minimum learning rate is 10e-5, the scheduler uses polynomial, and the warm-up epochs are 1. We use the adamw optimizer, and its parameters are all default values. It should be noted that when using different models for training, you may need to modify the final input layer of the model, because some models output features by default and some models output category logits. This modification should be simple. For the evaluation phase, predict.py can be used for inference.

5. Results

Since our second stage method is basically the same as the first stage, the relevant test set and challenge set results can be found directly from the paper.

6. Conclusions

In this work, we work to solve the problem of player re-identification for Synergy Re-Identification dataset. Specifically, we explore suit- able data augmentation methods for this dataset and analyze the impact of different data augmentations. In addition, we use class- independent InfoNCE loss and conduct comparative experiments with ID loss commonly used in person re-identification tasks. Through experiments, we found that the use of InfoNCE loss can avoid the gap generated by the model during the training phase and the inference phase. In the post-processing part, we adopted the k-reciprocal re-ranking method widely used in the field of person re-identification and proposed a model/rank fusion method that utilizes the similarity and dissimilarity between different models. This enables better retrieval performance when the accuracy of the single model is high. Finally, we concluded that the key to achieving excellent retrieval performance on the Synergy re-identification dataset is the pre-training of large-scale datasets, a suitable loss function, and post-processing strategies for sorting optimization and model/rank fusion.

7. Acknowledgments

This work was supported by the Natural Science Foundation of China (62276242), National Aviation Science Foundation (2022Z071078001), CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2021-016B, CAAIXSJLJJ-2022-001A), Anhui Province Key Research and Development Program (202104a05020007), USTC-IAT Application Sci. & Tech. Achievement Cultivation Program (JL06521001Y), Sci. & Tech. Innovation Special Zone (20-163-14-LZ-001-004-01).