Journal of Computers Vol. 31 No. 2, 2020, pp. 277-286
doi:10.3966/199115992020043102023

Application of Knowledge Distillation in Representation Learning Person Re-identification Model

Chang Liu, Hang Ma, Jun-Jie Jin, Xin-Lun Zhou, Wen-Bai Chen*

School of Automation, Beijing Information Science & Technology University, Beijing 100192, China
chenwb@bistu.edu.cn

Received 20 January 2020; Revised 20 February 2020; Accepted 30 March 2020

Abstract. The lock targets in monitoring system is important for fully exerting the surveillance capability of mobile devices and saving working time. To save the time required and huge amount of computing resources, a fast person re-identification (Re-ID) method is proposed. In this paper, we use knowledge distillation to make a large teacher model (ResNet50) guide a small but effective student model (MobileNet v2) for representation learning. Experimental results demonstrate that the proposed method is feasible. Compared with the teacher model and the student model, the system applied the knowledge distillation method can save more 55.4% of time and increase mAP 12.73% and Rank-1 8.63%, respectively.

Keywords: deep neural network, knowledge distillation, person Re-ID, representation learning

1 Introduction

Person re-identification (Re-ID) [1] aims at retrieving specified pedestrian across multiple non-overlapping cameras network, which has attracted many researchers in recent years. This task is particularly important in monitoring applications, because the security systems of public areas such as airports and train stations are constantly being improved to ensure the welfare of the people. Person Re-ID mainly relies on video images taken by the camera to obtain person representation information. Despite many years of research on person Re-ID task, it still faces great challenges such as the quality of the cameras deployed has been uneven, the environment for information acquisition is constantly changing, and the person video image is affected by lighting changes, shooting angles, and posture. Therefore, the realization of Re-ID is facing enormous challenges [2], and the current research is also far from practical application.

Recent advances in person Re-ID consist in the good performance of deep learning methods [3]. Deep convolutional neural networks have led to a series of breakthroughs for Re-ID. He et al. [4] proposed a residual learning framework to ease the training of networks. Liu et al. [5] proposed a new framework called Hybrid Task Convolutional Neural Network (HTCNN) which employs the modified ResNet-50 as the base network for person Re-ID in camera sensor networks. Zheng et al. [6] proposed the pose invariant embedding (PIE) as pedestrian descriptor and using Resnet-50 as backbone in the experiment, so that solved the pedestrian misalignment problem in Re-ID. For person Re-ID, ResNet50 is the widely used as their backbone and achieves high accuracy performance. Although by using a rapidly developing convolutional neural network, the accuracy and model stability of person Re-ID technology based on representation learning methods have reached a very high level. However, Re-ID models with good performance often contain complex model structures and huge computing resources. The most important deployment position of person Re-ID is precisely on embedded and mobile devices with weak computing performance. Therefore, the development of a small-scale but sufficient performance model has become the key to promote the application of person Re-ID technology.

This paper proposes a fast person Re-ID method, which uses knowledge distillation to make a large teacher model guide a small but effective student model for representation learning. The knowledge

* Corresponding Author

distillation [7] method solves the problem of the balance between performance and resources of the deep learning model, and has broad application prospects in deep learning engineering applications.

The remainder sections are organized as follows. Section2 describes the structure of person Re-ID model bases on representation learning, and Section3 introduces the application of knowledge distillation in Re-ID model. Experiment results and analysis are given in Sections4. Conclusions are provided in Section5.

2 Re-ID Model Structure Based on Representation Learning

This section briefly introduces the basic principles and structure of student network and teacher network.

2.1 Student Network

MobileNets are presented as efficient light weight models suitable for training and reducing calculations. In order to significantly reducing the amount of operation and memory required for training while keeping it fairly accurate, we use MobileNet [8] as the student network. MobileNet uses deep separable convolutions [9], which can be understood as solving a standard volume integral into a point convolution kernel and a deep convolution. A deep convolution maps each convolution kernel to each channel, while a point convolution is used to fit the output of the channel convolution. This greatly improves the computational efficiency of the convolutional neural network and the problem of excessive parameter complexity.

The deep separable convolution is a fundamental component of the entire network. When training the neural network, the parameters are constantly updated. In addition to the data of the input layer, the input data distribution of each layer of the subsequent network is always changing, so we use the batch normalization [10] layer to solve this problem. Relu as the activation function, the basic structure of depthwise separable convolution is shown in Fig. 1.

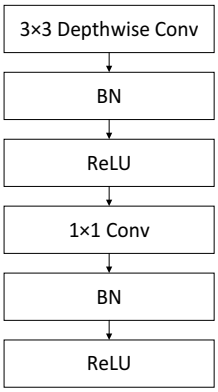


Fig. 1. Depthwise separable convolution structure

The entire neural network is a combination of multiple depthwise separable convolutions. The beginning of the network is a 3x3 standard convolution, followed by a deep separable convolution of the stack, and a partial deep separable convolution in the network is downsampled using a 2 unit step size. The feature is then converted to 1x1 using the average pooling layer, plus the fully connected layer based on the predicted category size, and the final softmax [11] layer as the output. The calculation amount of the entire network is shown in Table 1.

Table 1. Network parameters and calculations

Type	Mult-Adds	Parameters
Con 1×1	94.86%	74.59%
Con DW 3×3	3.06%	1.06%
Con 3×3	1.19%	0.02%
Fully Connection	0.18%	24.33%

On the 1x1 convolution, it basically concentrates on all calculations. Generally, convolution requires memory recombination, but the 1x1 convolution kernel does not require memory reorganization, so it can be implemented faster. The 1x1 convolution also concentrates most of the parameters, and of course some of the parameters are on the fully connected layer.

2.2 Teacher Network

To make deep learning approaches computationally efficient, the ResNet50 [12] neural network structure as our teacher network and extracts the feature data from the beginning of the model training. In general, the deeper the layer of deep neural networks, the richer the features and information obtained by training, and the more advantageous in image classification and recognition. Deep neural network can solve the problem of gradient disappearance through Batch Normalization, but as the number of layers of our neural network structure continues to increase, there will be a problem of reduced accuracy, that is, the accuracy begins to decrease after saturation. He et al. [4] proposed a residual network structure to solve this problem. The residual module is shown in Fig. 2.

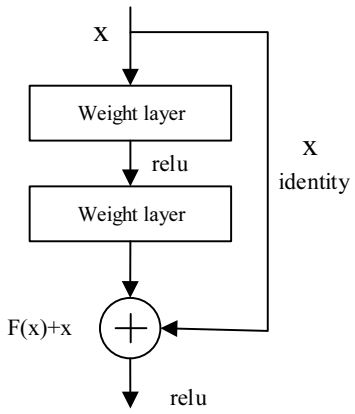


Fig. 2. Residual structure

ResNet50 is a network composed of such residual blocks, and its structure includes five convolution blocks, each of which contains a different number of residual units. ResNet50 is a commonly used deep neural network in the research of Re-ID. Although it proposed to be earlier than MobileNet V2, the accuracy and performance level brought by his huge network structure is that small networks like MobileNet V2 can't reach it. So using ResNet50 to conduct knowledge distillation on MobileNet V2 is a more scientific choice.

3 Application of Knowledge Distillation in Re-ID Model

This section uses knowledge distillation to make a large teacher model guide a small but effective student model for representation learning. The small neural network model trained by the knowledge distillation method can obtain the performance close to the large-scale neural network model while maintaining the original small scale.

3.1 Working Principle

Neural networks usually use the "softmax" output layer to generate class probabilities [13]. The output layer Z_i will be compared to other logit values. Convert the logit value Z_i of each class calculation to the probability q_i, as shown in formula (1).

$$q_i = \frac{e^{z_i}}{\sum je^{z_j}} \tag{1}$$

Complex neural networks are trained by 0-1 encoding, but the last layer uses the softmax layer to generate the probability distribution, so it is a softer target than the original 0-1 encoded hard target. This distribution is made up of a number of values between (0, 1). The same specimen, when used to train a small neural network using soft targets generated on complex neural networks, learns to converge more quickly. When a large and complex neural network saves learning information, the small neural network can even obtain knowledge directly from the obtained soft target.

$$q_i = \frac{e^{z_i/T}}{\sum j e^{z_j/T}} \tag{2}$$

As shown in formula (2), where T is a parameter that is usually set to be greater than 1 [14], in the concept of knowledge distillation, the image is referred to as temperature. As shown in Fig. 3, when a higher value is used for T, the neural network produces a softer probability distribution, that is, a softer target with a more uniform output distribution. Then, the soft target generated by the complex neural network is used to train the small-scale neural network, and the T values of the two neural networks are kept the same, so that the small-scale network approaches the soft target and learns the data structure distribution characteristics of the large-scale network output. In this paper, data and data structure information is treated as a mixture, and the data structure distribution information is separated by probability distribution. When the value of T is large, it is equivalent to separating the key distribution information from the original data at a high temperature, and then merging the data distribution by the new model by the same temperature, and finally let the temperature recover, so that both can be fully integrated to achieve the purpose of knowledge distillation.

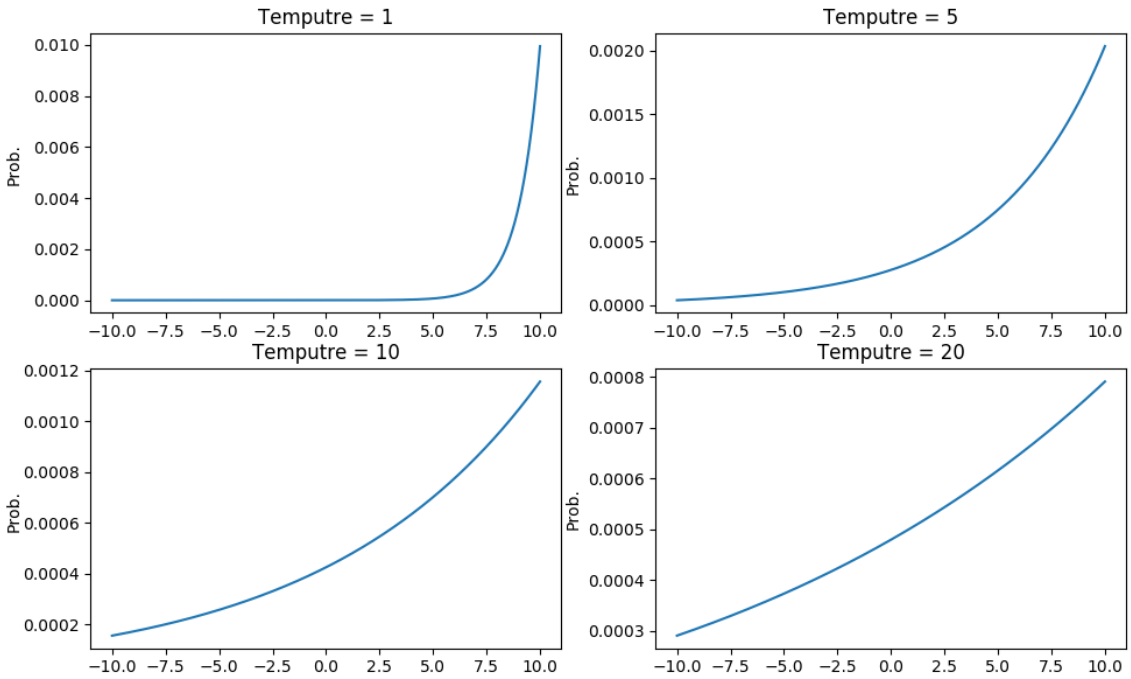


Fig. 3. Relationship between temperature T and probability distribution

3.2 Loss Function

This paper uses the teacher model and real sample tags to help the student model build a loss function and use the loss function to help the student model train.

The first loss function is the cross entropy [15] with a soft target. The second loss function is the cross entropy with the real sample label, which is calculated from the exact same logits in the softmax layer of the distillation model, however, the temperature T is 1. Generally, setting the weight of the second part of the loss function relatively low can get better results, because the first part of the loss function will have a

$1/T^2$ term after the gradient is calculated. At this time, the first part and the second part of the loss function are different in magnitude, so we must first ensure that the relative contribution of the first and second parts to the loss function is almost constant, so this paper first multiplies the first part of the loss function by T^2, and then calculates the weighted average of the two terms.

Relative entropy: also known as Kullback-Leibler divergence, refers here to the first part of the loss function above. It measures the ineffectiveness of the distribution q when the true distribution is p. When the difference between the two probability distributions becomes larger, their relative entropy also becomes larger. This paper uses it to make the prediction probability of the student model as close as possible to the output of the teacher model.

Cross entropy: recorded as $CHE(p,q) = H(p) + D_{KL}(p \| q)$. It can be seen that it differs from the relative entropy by only H(p). When p is known, H(p) can be regarded as a constant. At this time, the cross entropy and KL distance are equivalent in behavior, which reflects the similarity of the distribution p, q. When the difference between the predicted probability and the real sample label increases, their cross entropy also increases, which can be used to measure the difference between the probability of the student model prediction and the real sample label. In model distillation we use it to make the predictive probability of the student network close to the real label. Combine the two to get our knowledge distillation loss function, as in formula (3).

$$Loss = KL(p_2, q) \times alpha \times T^2 + CE(label, p_1) \times (1 - alpha) \quad \textbf{(3)}$$

Where T is temperature, KL [16] is relative entropy, CE is cross entropy, q is the result of the teacher network output after distillation, p_1, p_2 are the output in the student network, label is the real tag information given by the data set, and alpha is the proportional parameter of KL and CE in KDLoss. When alpha=0, the student network is equivalent to a deep convolutional neural network using cross entropy as a loss function. The first part of KDLoss is designed to optimize the student network to a softened distribution, while the second part, as traditional, allows the student network to optimize for real tag values. After obtaining the specific structure of the loss function, we can use it to train the deep neural network [17] to achieve the purpose of our knowledge distillation. The specific implementation is shown in Fig. 4.

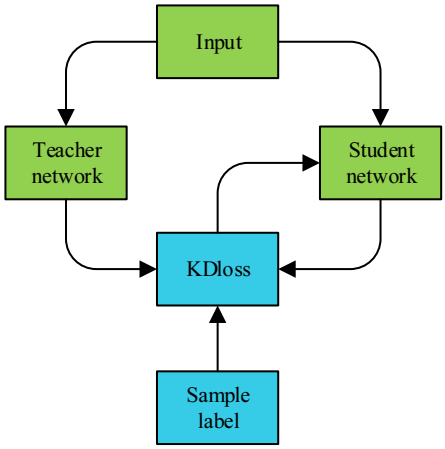


Fig. 4. Knowledge distillation training process

In order to make the work look more visual, in the experiment, the big network is called the teacher network, and the small network is called the student network. Using the KDloss loss function, the knowledge gained by the teacher network learning is ingeniously guided to train the student network. In this way, a Re-ID model with both performance and resource advantages is obtained.

4 Experimental Validation

We evaluate our method on the Market-1501 dataset, whose images come from 1501 pedestrians and 32668 pedestrian rectangles captured by 6 cameras. The section is arranged as follows. Firstly, the

experimental settings were presented and evaluate our method. Then, the two models training process were analyzed and the performance comparison of main models were shown.

4.1 Experimental Setup

The proposed fast Re-ID method is built on the PyTorch framework. In the experiments, NVIDIA GeForce TITAN XP and cuda9.0 were used as the GPU computing platform in this paper. The operating system was Ubuntu 18.04.2 LTS, and the development environment was python3.6 and Pytorch1.1.0. First, the ResNet50 and MobileNet V2 Re-ID models were used for training and testing. After all the data has been saved, a knowledge distillation method is introduced in the student model. Testing and evaluation are not fundamentally different from the teacher network when training and testing student networks that introduce knowledge distillation, but in the training section we will change the loss function structure in model training.

The loss function constructed by the knowledge distillation method can enable the student network to obtain the knowledge guidance of the trained teacher network, which can enable the student network to obtain the performance close to the teacher network on the dataset.

4.2 Experimental Results

As shown in Fig. 5, this paper can use the trained model to realize the person image recognition function, and can output the sorting result with the similarity of the query image from high to low.

Fig. 5. Query image similarity ranking results

In order to observe the change of the loss and model recognition accuracy during the experiment. The number of iterations in the experiment is 60. The situation of the teacher model is shown in Fig. 6.

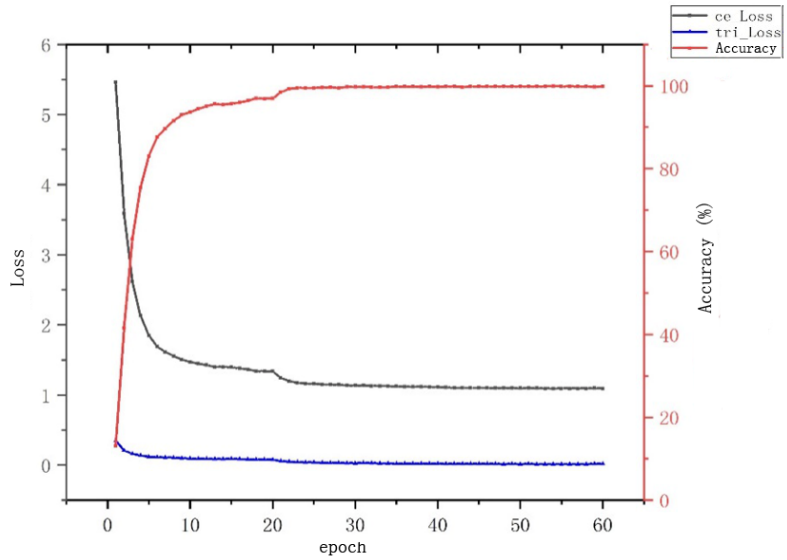
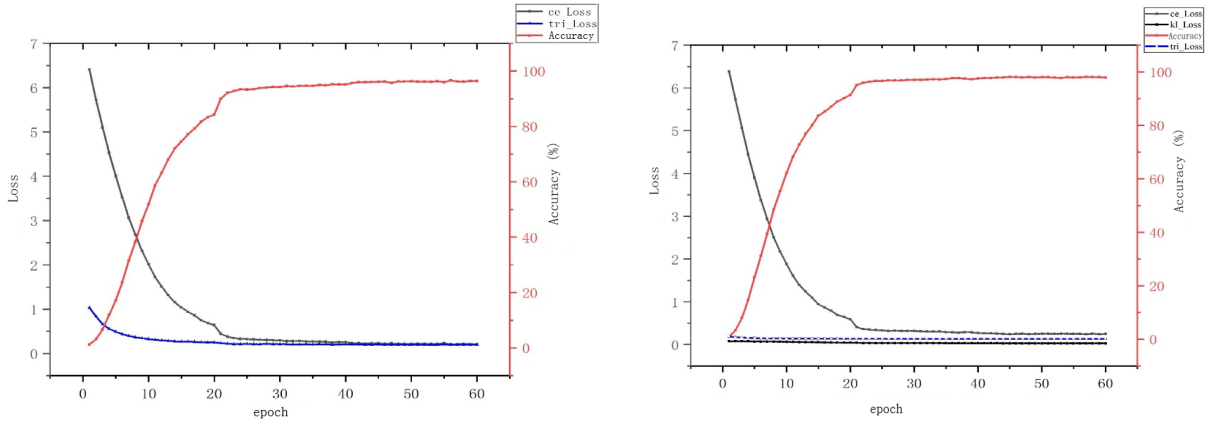


Fig. 6. Teacher model training process

Where ce_Loss is the cross-entropy loss function trained using the real labels of the dataset. Tri_Loss is a loss function designed specifically for sorting tasks. Its advantage lies in the distinction of details, that is, when the two inputs are similar, the details can be better modeled, which is equivalent to adding a

measure of the difference between the two inputs and learning a better representation of the inputs. However, its convergence speed is relatively slow.



(a) Unused knowledge distillation method (b) Used knowledge distillation method

Fig. 7. Student model training process

The student network without using knowledge distillation method training situation is shown in Fig. 7(a), which is roughly the same as the teacher network. The difference in specific training situations is caused by different network structures. However, it can be clearly seen in this paper that the student model is far inferior to the teacher model in both the height of the final accuracy and the speed of the climb. This paper uses the loss function constructed by introducing the knowledge distillation method in the training process of the student model. The resulting training situation is shown in Fig. 7(b).

The above picture shows the training model of the student model. It can be seen that the height of the final accuracy of the student model and the speed of the climb are improved compared to the original case, but the relative entropy loss function in the model not obvious. Although the relative entropy loss function occupies a small amount in the loss of the entire training iteration, its function is huge. At the end of the training, the re-recognition accuracy of pedestrian pictures with the knowledge distillation method introduced by the student network improved by more than ten percent compared with that before distillation.

For the teacher model and the student model, we set five iterations to observe the re-identification precision change for one cycle. The evaluation criteria are mAP and CMC, and the teacher model accuracy changes as shown in Fig. 8.

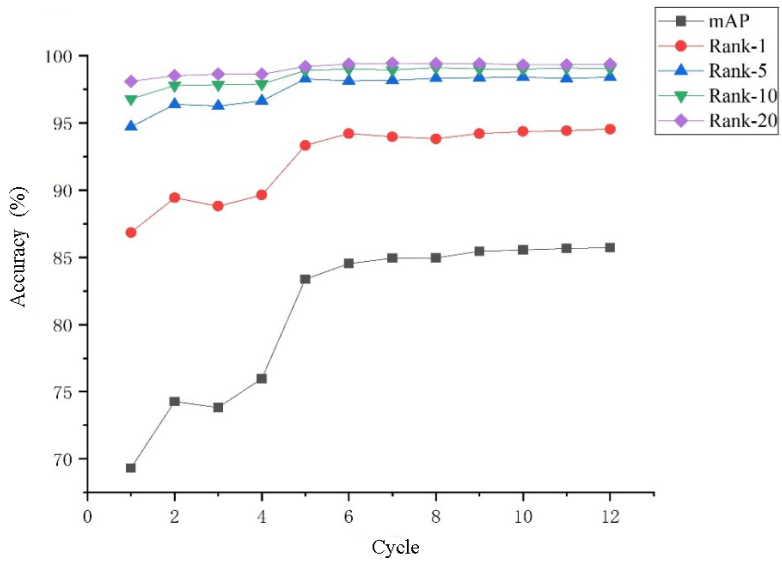


Fig. 8. Teacher model Re-ID accuracy change

Fig. 9 shows the change in re-identification accuracy of the student model (without introduction of knowledge distillation). The change of the accuracy of the student model re-identification after the introduction of knowledge distillation is shown in Fig. 10. It can be seen that after the introduction of knowledge distillation, the final recognition accuracy of the student model has been greatly enhanced. Moreover, unlike the teacher model and the original student model, the student model introduced with knowledge distillation has a more uniform change in the accuracy of recognition.

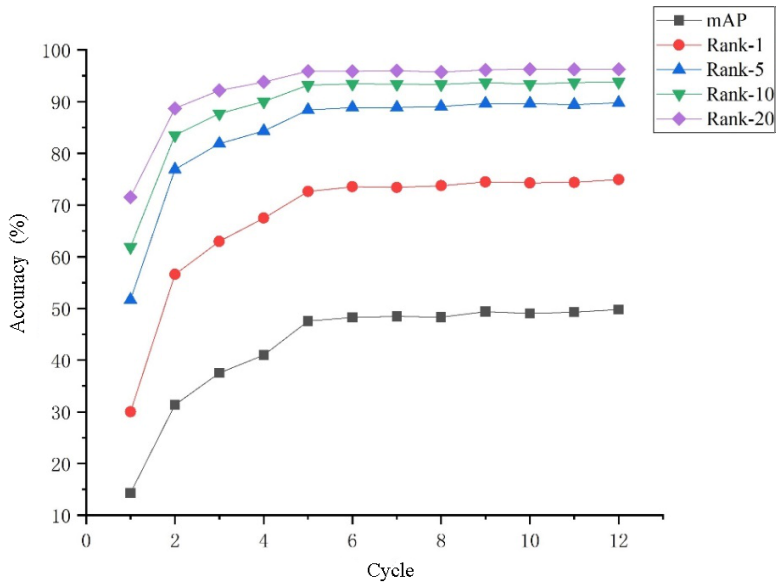


Fig. 9. Student model Re-ID accuracy change

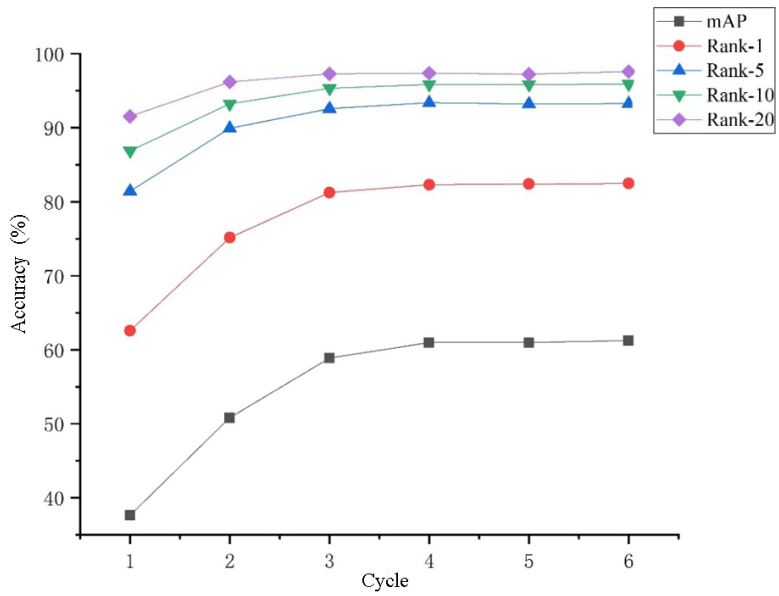


Fig. 10. Changes in the accuracy of student models with knowledge distillation

It is worth mentioning that the average training time of the model using the teacher network is 56 minutes, and the training time of the students who introduced the knowledge distillation method is about the same as the original network, both of which are 25 minutes. In this paper, the parameters T of the loss function are set to [1, 2, 5, 10] and alpha are set to [0.1, 0.3, 0.5] respectively in the training of student models using knowledge distillation. As shown in Table 2.

Table 2. Performance comparison of student models trained under different T and alpha parameters

parameter settings	mAP	Rank-1	Rank-5	Rank-10	Rank-20
T=1, alpha=0.1	61.04%	80.32%	93.26%	95.90%	97.57%
T=1, alpha=0.3	60.09%	80.13%	92.28%	94.87%	96.43%
T=1, alpha=0.5	60.01%	80.07%	91.19%	93.56%	95.23%
T=2, alpha=0.1	61.24%	82.48%	93.26%	95.90%	97.57%
T=2, alpha=0.3	61.01%	80.31%	93.15%	95.57%	97.24%
T=2, alpha=0.5	60.89%	80.23%	92.48%	94.32%	96.23%
T=5, alpha=0.1	62.53%	83.57%	94.01%	96.23%	98.76%
T=5, alpha=0.3	62.12%	83.02%	93.69%	95.89%	98.02%
T=5, alpha=0.5	61.78%	82.95%	92.94%	95.03%	97.87%
T=10, alpha=0.1	61.06%	80.36%	93.28%	95.92%	97.59%
T=10, alpha=0.3	60.18%	80.17%	92.36%	94.90%	96.51%
T=10, alpha=0.5	60.19%	80.56%	91.32%	93.76%	95.89%

The student model after knowledge distillation has the strongest performance at T=5, alpha=0.1, which is roughly the same as the optimal parameter direction we envision. We then compare the optimal performance (T=5, alpha=0.1) obtained by the student knowledge distillation from the above table with the teacher model and the original student model, as shown in Table 3.

Table 3. Performance comparison of main models in experiments

model	mAP	Rank-1	Rank-5	Rank-10	Rank-20
MobileNet V2	49.80%	74.94%	89.76%	93.82%	96. 23%
MobileNet V2 + Knowledge Distillation	62.53%	83.57%	94.01%	96.23%	98. 76%
ResNet50	85.72%	94.54%	98.40%	99.05%	99. 38%

It can be seen that the performance of the student network with knowledge distillation method has been greatly improved compared with original model, which improved mAP 12.73% and Rank-1 8.63%. Although it is still not as good as the teacher network, time reduced by 55.4%. All the above observations and conclusions demonstrate that the proposed method can have an effective ability to person Re-ID.

5 Conclusion

This work proposes a fast person Re-ID method that improves the real time and saves computing resources. Compared with the ResNet-50 which has too many parameters to conduct Re-ID in real time and MobileNet v2 which shows poor accuracy, the improved small network has faster training speed and higher matching accuracy. In the training phase, a large teacher model was used to guide a student model for representation learning with knowledge distillation. Experimental results demonstrate that the proposed method has improved mAP 12.73% and Rank-1 8.63% than the classic student network. What's more, training time of the method is 55.4% faster than the teacher model. In the future, we will bring the proposed method into mobile systems for practically testing. And further improving its accuracy with intelligent algorithms.

Acknowledgements

This work is supported by Cross-Training Plan of High Level Talents and Training Project of Beijing.

References

[1] M.A. Saghafi, A. Hussain, H.B. Zaman, M.H.M. Saad, Review of re-identification techniques, IET Computer Vision 8(6)(2014) 455-474.

[2] I. Prodaiko, Person re-identification in a top-view multi-camera environment, [thesis] Lviv, Ukraine: Ukrainian Catholic University, 2020.

[3] S. Lin, C.T. Li, Person Re-identification with Soft Biometrics Through Deep Learning, Deep Biometrics, Springer, Cham, 2020.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[5] S. Liu, W. Huang, Z. Zhang, Person re-identification using hybrid task convolutional neural network in camera sensor networks, Ad Hoc Networks 97(2020). DOI:10.1016/j.adhoc.2019.102018.

[6] L. Zheng, Y. Huang, H. Lu, Y. Yang, Pose-invariant embedding for deep Re-ID, IEEE Transactions on Image Processing 28(9)(2019) 4500-4509.

[7] J. Yim, D. Joo, J. Bae, J. Kim, A gift from knowledge distillation: fast optimization, network minimization and transfer learning, in: Proc. IEEE Conference on Computer Vision & Pattern Recognition, 2017.

[8] Z. Qin, Z. Zhang, X. Chen, C. Wang, Y. Peng, FD-MobileNet: improved MobileNet with a fast downsampling strategy, in: Proc. IEEE International Conference on Image Processing, 2018.

[9] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: Proc. IEEE Conference on Computer Vision & Pattern Recognition, 2017.

[10] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proc. International Conference on International Conference on Machine Learning, 2015.

[11] Z. Yang, Z. Dai, R. Salakhutdinov, W.W. Cohen, Breaking the softmax bottleneck: a high-rank rnn language model. <https://arxiv.org/abs/1711.03953>, 2017.

[12] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, P.L. De Geus, Malicious software classification using transfer learning of ResNet-50 deep neural network, in: Proc. IEEE International Conference on Machine Learning & Applications, 2018.

[13] D.S. Young, D.R. Hunter, Mixtures of regressions with predictor-dependent mixing proportions, Computational Statistics & Data Analysis 54(10)(2010) 2253-2266.

[14] L. Xiang, G. Ding, Learning from multiple experts: self-paced knowledge distillation for long-tailed classification. <https://arxiv.org/abs/2001.01536>, 2020.

[15] P.T. De Boer, D.P. Kroese, S. Mannor, R.Y. Rubinstein, A tutorial on the cross-entropy method, Annals of Operations Research 134(1)(2015) 19-67.

[16] J.E. Contreras-Reyes, Asymptotic form of the kullback-leibler divergence for multivariate asymmetric heavy-tailed distributions, Physica A Statistical Mechanics & Its Applications 395(4)(2014) 200-208.

[17] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications. Neurocomputing 234(2017) 11-26.