Journal of Computers Vol. 31 No. 6, 2020, pp. 284-296
doi:10.3966/199115992020123106022

The Influence of Different Normalization Methods on Person Re-ID

Hang Ma[1], Yan Tong[1], Ding-Qian Liu[1], Yue Lu[1],
Guo-Hong Yi[1], Ming-Deng Su[2], Wen-Bai Chen[1*]

[1] School of Automation, Beijing Information Science and Technology University, Beijing, China

[2] Beijing towards Intelligence Technology Innovation Center, Beijing, China
chenwb@bistu.edu.cn

Received 15 August 2020; Revised 25 September 2020; Accepted 24 October 2020

Abstract. At present, person re-identification (re-ID) has become a hot research topic in the field of computer vision because of its wide application prospect and the difficulty of concrete implementation. In recent years, the method based on deep learning has developed rapidly, which has become the main research direction in this field. As an important technology in deep learning, normalization is applied to person re-ID models by more and more researchers. However, at present, there are still no researchers who analyze and evaluate the impact of various normalization methods on person re-ID model, which may ignore the optimization potential of normalization methods on person re-ID model. In this paper, we first design and build an end-to-end person re-ID model based on deep learning. By adding various normalization layers before the classifier, and training, testing and evaluating on Market-1501, DukeMTMC-reID and MSMT17 datasets, we comprehensively and deeply analyze and summarize the impact of normalization methods on person re-ID model performance.

Keywords: deep learning, neural network, normalization methods, person re-ID

1 Introduction

With the increasing demand for public security and the rapid development of image acquisition and transmission technology, the camera network monitoring system is widely used in daily life, and the pedestrian re-identification (re-ID) technology has gradually become the focus of attention [1-2]. The main task of person re-ID is to determine whether the pedestrians in two or more different cameras have the same identity, in order to replace the manual processing of the monitoring network data [3]. In practical applications, person re-ID has become a research hotspot in the field of computer vision due to the influence of pedestrian posture, shooting angle, lighting conditions, environmental occlusion, image resolution and other factors. In order to simplify the process of pedestrian recognition, pedestrian detection is generally not included in the task of pedestrian recognition.

However, in practical applications, the appearance of pedestrians is easily affected by wearing, occlusion, visual angle and occlusion. At the same time, the video image has the influence of low resolution and light transformation, which makes pedestrian recognition become one of the most challenging problems in the field of computer vision. For example, the surface of surveillance video is generally fuzzy and the resolution is relatively low, so face recognition and other methods cannot be used for recognition. Only the human appearance information outside the head can be used for recognition, but different pedestrians may have the same body shape and clothing, which brings great challenges to the accuracy of person re-ID. The images of pedestrian are often taken from different cameras. Due to different shooting scenes and camera parameters, there are generally problems in pedestrian recognition, such as changes in light and perspective, which results in large differences in the same pedestrian under different cameras. The appearance characteristics of different pedestrians may be more similar than the

* Corresponding Author

appearance characteristics of the same person. The pedestrian image for recognition may be taken at different times, and the posture and clothing of pedestrians will change to different degrees. In addition, the appearance characteristics of pedestrians will be very different under different lighting conditions. The scene under the actual video monitoring is very complex. Many of the monitoring scenes have large traffic, complex scene and so on. In this case, it is difficult to re-identify by gait and other features. All of the above have brought great challenges to the research of person re-ID, so the current research is far from the practical application.

In recent years, with the rapid development of deep learning technology, person re-ID based on deep learning has gradually become the mainstream research direction in this field [4-5], and has achieved good results. Typical process of pedestrian re-identification is shown in Fig. 1. Convolutional neural networks (CNNs) have recently become increasingly predominant choices in person re-ID thanks to their strong representation power and the ability to learn invariant deep embeddings. As an important part of deep learning, batch normalization (BN) [6] is widely used in the construction of deep neural network. At present, many excellent re-ID models have also applied the batch normalization method in the process of building, but there is no further comprehensive analysis and evaluation on the impact of various normalization layers on re-ID model. It is possible that the normalization method has a higher optimization potential for the performance of re-ID model, so it is very important to analyze the influence of various normalization layers on it. How to extract more representative and robust features, and how to learn more discriminative similarity measures are the main research directions of person re-ID. Normalization methods are effective ways to achieve those goals. However, different normalization methods will have different effects on different tasks. Which normalization method is most suitable for person re-ID task? We are committed to carrying out detailed experiments on different normalization methods in person re-identification task, so as to explore the most suitable normalization method for that, and improve the performance of the re-ID model.

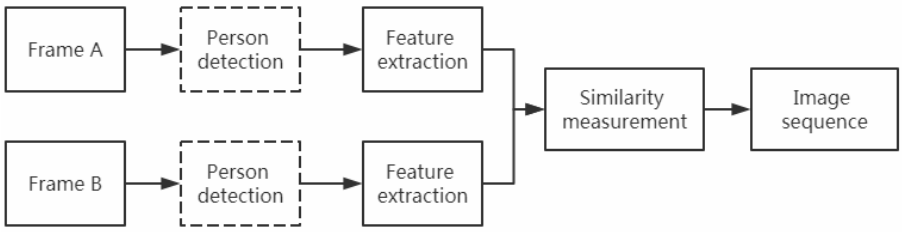

Fig. 1. Typical process of person re-identification

Based on resnet-50 [7], this paper builds an end-to-end re-ID model based on deep learning as a test model. It mainly analyzes and evaluates the impact of various normalization methods on the performance of re-ID model when all kinds of normalization layers are added before the final classifier of the model. In the experiment part of this paper, model training and testing are carried out on three open datasets Market-1501 [4], DukeMTMC-reID [8] and MSMT17 [9] respectively, and the corresponding loss curve of training process and the final test results are analyzed, then the corresponding conclusions are drawn. Finally, through experiments, the effects of adding various normalization methods on the performance of the person re-identification model before the final fully connected layer of the neural network are comprehensively analyzed and evaluated, which proves that normalization methods can effectively improve the performance of the model. In summary, we introduce four normalization methods in detail, and conduct thorough experiments on three commonly used datasets for three of the methods that applicable to person re-ID tasks. The experimental results show that the Batch Normalization achieves the best performance in our person re-ID task.

2 Methods of Re-ID

2.1 Handcraft Features

Most of the early person re-ID work is based on the method of hand-made design features. Firstly, the image features are extracted, and then the similarity is measured [1-3]. For example, in the literature [10], Farenzena et al. Calculated the maximum stable color region (MSCR), the high repetition structure color block (RHSP) and the weighted color histogram (WH) of the pedestrian foreground separated from the background based on the body configuration, and combined the three results as the total face of the pedestrian image description of color features.

2.2 Deep Neural Network

Although the method based on the characteristics of manual design has achieved good results after years of research, it is far from meeting the requirements of practical application in terms of efficiency and generalization ability. Since Krizhevsky et al. won the ILSVRC12 classification competition by a large margin [11], the deep learning model based on CNN has set off a new wave in the field of computer vision. The deep learning model can integrate feature expression and similarity measurement, and get the performance far beyond the traditional method through the joint optimization of them. According to the difference between the two integration methods, the re-ID methods based on deep learning are divided into three types: end-to-end, hybrid and independent [1], and the end-to-end structures are studied more by researchers. In recent years, the person re-ID methods based on deep learning have gradually become the mainstream research direction and has achieved great success. Although the method based on deep learning has been developed rapidly, it still faces the challenges such as the change of pedestrian's posture, the difference of illumination conditions, the influence of environment occlusion on the recognition accuracy and the small scale of current dataset.

There are two classical models to solve person re-ID task using convolutional neural network.

(1) SCNN: In 2014, Yi et al. [12] took the lead in using three siamese convolutional neural networks (SCNN) to determine whether the pedestrians in the two pictures belong to the same ID. Yi et al. First divide the two input pedestrian images into three parts: top, middle and bottom, then input the same part of the two images into the same SCNN, and finally integrate the similarity of each SCNN measurement output to complete pedestrian identification. The architecture of triplet SCNN can be seen in Fig. 2. Good test results have been achieved on the Viper dataset.

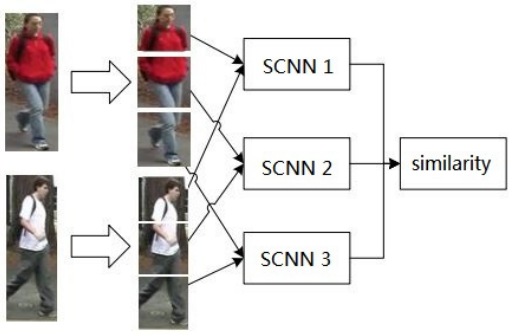

Fig. 2. The architecture of triplet SCNN

(2) FPNN: In the same year, Li et al. [13] also proposed a new network structure filter pairing neural network (FPNN) to handle the same tasks. As shown in Fig. 3, FPNN is divided into six layers, the first layer is convolution and maximum pooling layer, which is mainly used for feature extraction and convolution filtering; the second layer is block matching layer, which is mainly used to match the filter response of local blocks across the horizon. The third layer is maxout-grouping layer, which is mainly used to improve the robustness of matching the previous layer. The fourth layer has the same network type as the first layer, which is mainly used to extract the displacement matrix of pedestrian body parts. The fifth layer full connection layer converts the previously extracted features into one-dimensional

feature vectors to input the last layer softmax layer to determine whether the pedestrians in the two input images have the same ID. The advantage of FPNN is that it can deal with the problems such as misalignment, ray and geometry transformation, occlusion and background interference in a unified deep structure, which effectively improves the model recognition effect.

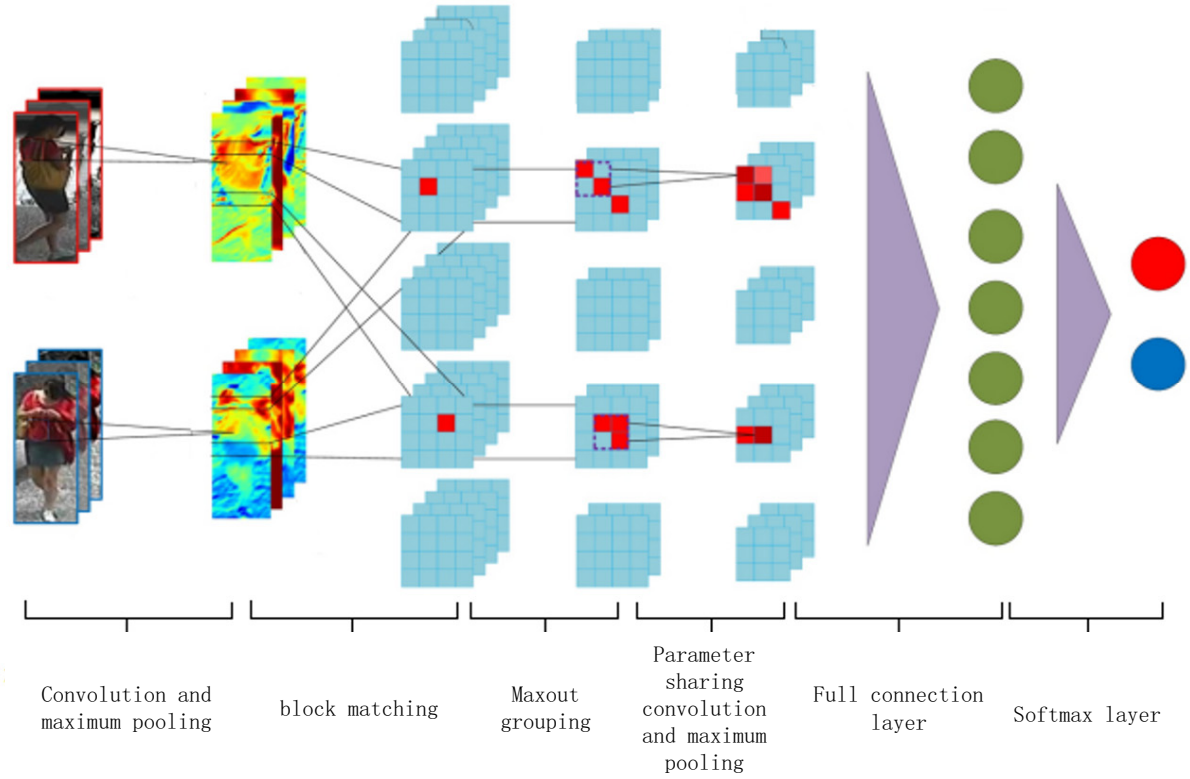

Fig. 3. Structure of residual connection unit

3 Normalization

In terms of normalization, in reference [14], the author confirmed that adding BN layer to deep neural network can greatly improve the training stability and generalization of the network. However, BN has the disadvantage of over dependence on the size of input batch size, that is, when the batch size is small, BN error will increase dramatically. In this regard, some researchers proposed layer normalization (LN) [15] and instance normalization (IN) [16], both of these normalization methods avoid normalization along the dimension of batch, so that their performance is little affected by batch size. The training of sequential model or generative model is effective, but the effect is limited in the task of image classification. Later, Wu et al. put forward group normalization (GN) [17]. GN is a combination of IN and LN methods, and has the advantage that its performance is less affected by batch size. It can still maintain good stability in a large range of batch changes, and has achieved good results in image classification tasks. In the case of small size, GN is better than BN. More details will be introduced in the following section.

In the aspect of neural network structure normalization, the literature [28] confirmed that adding batch normalization (BN) layer to the deep neural network can greatly improve the training stability and generalization of the network. Normalization layer is usually inserted behind convolution layer and before nonlinear layer. The residual unit is shown in Fig. 4, and a normalization layer is usually inserted after the output of the residual unit.

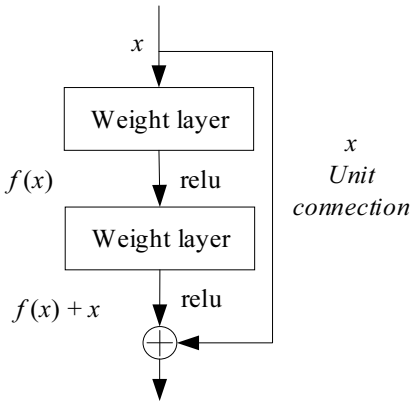

Fig. 4. Structure of residual connection unit

In deep learning, normalization is mainly used as a separate network layer, placed before or between the nonlinear activation layer and the next network layer, to convert the input into a normal distribution with 0 as the mean and 1 as the variance. The common normalization methods are BN, GN, IN and LN [6].

All of the four normalization methods can be represented by formula (1):

$$\hat{x}_i = \frac{1}{\sigma_i}(x_i - \mu_i), \tag{1}$$

in which x_i is the feature calculated by a certain network layer to be normalized, the subscript i in formula (1) represents the index. In case the input image is 2D, $i = (i_N, i_C, i_H, i_W)$ is a four-dimensional vector, where N is batch size, C is the number of channels, H and W corresponding to the height and width. After normalization $\hat{x}_i$ is in accordance with the standard normal distribution.

In formula (1) μ and σ is the mean value and standard deviation calculated by formula (2) and formula (3), respectively.

$$u_i = \frac{1}{m}\sum_{k \in s_i} x_k , \tag{2}$$

$$\sigma_i = \sqrt{\frac{1}{m}\sum_{k \in s_i}(x_k - \mu_i)^2 + \varepsilon} , \tag{3}$$

where ε is a small constant, s_i is a set of pixels whose average is μ_i and standard deviation is σ_i, m refers to the size of the collection.

After normalization, the final features are as follows (4), where γ and β are two learnable variables (to simplify the notation, their respective I index).

$$y_i = \gamma \hat{x}_i + \beta . \tag{4}$$

Note that we abbreviate Batch Normalization, Layer Normalization, Instance Normalization and Group Normalization to BN, LN, IN and GN, respectively. The main difference between the four normalization (BN, LN, IN, GN) methods mentioned above is that the definitions of s_i are different, as shown in Fig. 5.

BN. $S_i = \{k | k_C = i_C\}$. For each channel C, BN follows (N, H, W) axes to calculate μ and σ for normalization. BN normalizes a batch of data along the channel dimension.

LN. $S_i = \{k | k_N = i_N\}$. For each sample N, LN follows the (C, H, W) axes to calculate μ and σ for normalization. LN normalizes each sample including all channels along the dimension of sample size.

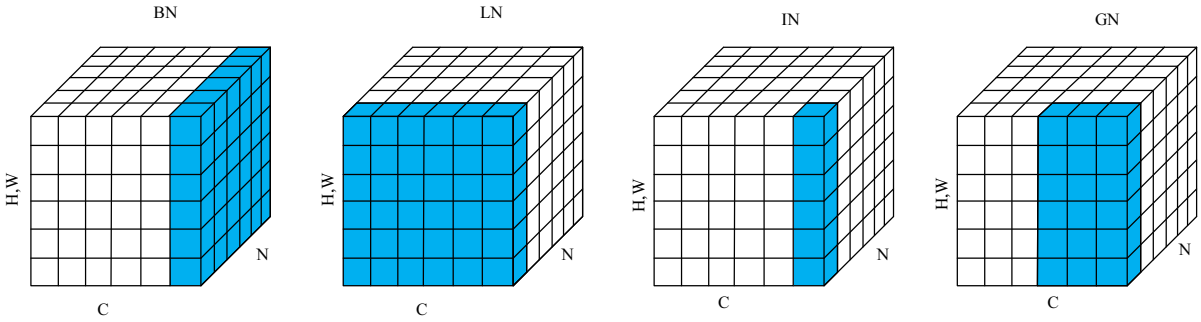

Fig. 5. Definition of s_i in the four normalization methods

IN. $S_i = \{k | k_N = i_N, k_C = i_C\}$. For each channel C and each sampling N, IN follows (H,W) axes to calculate μ for normalization. IN is equivalent to a separate normalization for each sample of each channel.

GN. $S_i = \left\{k \middle| k_N = i_N, \left\lfloor \frac{k_C}{C/G} \right\rfloor = \left\lfloor \frac{i_C}{C/G} \right\rfloor \right\}$, where G is a pre-defined super parameter (32 by default), representing the number of groups. C/G is the number of channels per group. $\lfloor . \rfloor$ indicates rounding down. GN takes all pixels in the same sampling N as per C/G A group along the (C,H,W) to calculate μ and σ for normalization. The right most GN in Fig. 3. is for $C=6, G=2$ case. GN is similar to IN and LN, except that GN groups LN, or it can be viewed as a combination of IN.

4 Experimental Model

4.1 Model Building

What is used in this testing re-ID models are Python3 and PyTorch [20] deep learning framework. The model uses ResNet-50 [7] pre-trained on ImageNet [21] as the feature extraction network, which can be used by removing the last classifier for 1000 classes and add a new fully connected layer as our classifier. The input dimension before the new fully connected layer is still 2048, and the output dimension is the number of pedestrian IDs corresponding to the dataset. The whole architecture of our model is shown as Fig. 6, and the details of ResNet-50 is shown as Table 2. The output dimension 751 is corresponding to Market-1501 dataset, which should be changed according to the number of IDs in the dataset.

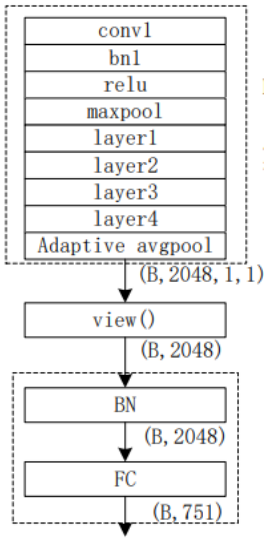

Fig. 6. The model we used in the experiments

The structure details of ResNet -50 are shown in Table 1.

Table 1. Details of ResNet-50

Layer	Output size	Detailed architecture
conv1, pool1	112×112	7×7×64, stride 2, 3×3 max pooling, stride 2
conv2_x	56×56	[1 × 1, 64 3 × 3, 64 1 × 1, 256] × 3
conv3_x	28×28	[1 × 1, 128 3 × 3, 128 1 × 1, 512] × 4
conv4_x	14×14	[1 × 1, 256 3 × 3, 256 1 × 1, 1024] × 6
conv5_x	7×7	[1 × 1, 512 3 × 3, 512 1 × 1, 2048] × 3
pooling5	1×1	Average pooling; Fully connected layer; Softmax activation

Finally, it is replaced by the adaptive average pooling layer in PyTorch, and its (W, H) parameter is set to (1, 1). The output dimension 751 corresponds to the number of pedestrian identities in the Market-1501 dataset, which needs to be changed to the corresponding number according to different data sets.

4.2 Model Training

Firstly, the input image is expanded. We resize the input images to 256×128, and then use 0 to fill the 10 pixels with 276×148 size. We crop the filled image to 256×128, whose position of the center point is selected randomly. Then the image is randomly flipped horizontally. The number of iterations of the whole dataset in the training process (epoch) is set to 60, and the number of pictures per batch (batch size) is 32. We use cross entropy loss as our loss function and Adam [22] as our optimizer. The initial learning rate is set to 3×10^{-4}, and weight decay is set to 5×10^{-4}. We take learning rate decay strategy and every 20 epochs it decays to 0.1 of the original. The overall training setup is shown in Table 2.

Table 2. Training setting

Training parameter setting	Value
batch size	32
epoch	60
optimizer	Adam
learning rate	0.0003
weight decay	0.0005
lr_scheduler	MultiStep
milestones	[20, 40]
gamma	0.1

5 Experimental Results and Analysis

5.1 Experiment Dataset

In this paper, three open datasets, Market-1501, DukeMTMC-reID and MSMT17, are tested. The details of the three datasets are introduced below.

The Market-1501 [4] dataset uses a total of six cameras, including five high-resolution cameras and one low-resolution camera. The dataset consists of 1501 identities (IDs) and 32668 detected images of pedestrian rectangular boxes, which are obtained by DPM [16]. The training set in the dataset contains 751 IDs and 12936 training images in total; the gallery during the test contains 750 IDs and 19732

images in total. In addition, there are 3368 query images used in the test. Each ID in the training set corresponds to a pedestrian, and the image of each pedestrian is captured by at least two and at most six cameras.

The DukeMTMC-reID [8] dataset contains 36411 detected pedestrian rectangular boxes, which are manually labeled by people. There are 1404 IDs in two or more cameras in the dataset, and 408 IDs (interference IDs) in only one camera. The dataset training set contains 16522 images from 702 IDs. The test set contains all the images of 702 other IDS and 408 interference IDs, of which the query set contains 2228 images and the gallery set contains 17661 images. 2228 query images selected from another 702 IDs contain a gallery of 1110 IDs (702 IDs + 408 interference IDs) and 17661 images in total.

The MSMT17 [9] dataset uses a camera network (including 12 outdoor cameras and 3 indoor small cameras) with 15 cameras in the campus, mainly composed of 4101 IDs and 126441 detected pedestrian rectangular frame pictures. Bounding boxes are detected by Faster RCNN [17] and marked with pedestrian label manually. The training set of this dataset contains 1041 IDs, with 32621 training images in total; the test set contains 3060 IDs, with 93820 images in total. 11659 images are randomly selected from the test set as query images, and the remaining 82161 images as gallery. Compared with the re-ID dataset proposed before, the dataset adopts a more reliable bounding box detector, which has more IDs, images and cameras, and the scene and background of the collected photos are more complex. The collected images also have great illumination changes due to different time periods, which is more close to the real scene of person re-ID. Some examples of MSMT17 dataset are shown in Fig. 7.

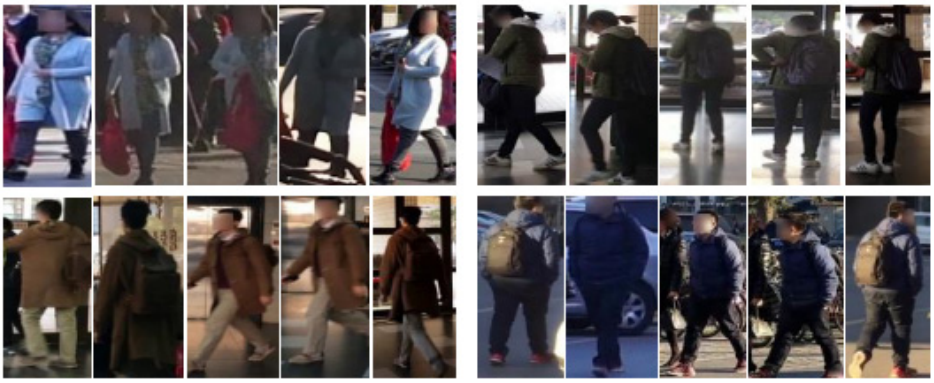

Fig. 7. Some examples of MSMT17 dataset

The comparisons of the three datasets are shown in Table 3. The MSMT17 dataset is more complex than the other two datasets.

Table 3. The comparisons of the three datasets

Dataset	Persons	Images	Cameras	Detector	Scene
Market-1501	1501	32688	6	DPM	Outdoor
DukeMTMC-reID	1812	36411	8	Handmade	Outdoor
MSMT17	4101	126441	15	Faster RCNN	Outdoor+Indoor

5.2 Evaluation Criterion

In this paper, the cumulative match characteristic (CMC) curve and the average precision mAP are used as the evaluation criteria. The CMC curve indicates that for the provided query picture, the first k returned results contain the probability that the row person in the query picture has the same ID. Due to space constraints, our experimental results only show the cumulative matching accuracy of the selected rank (rank-1, rank-5), rather than drawing the whole CMC curve. The mAP represents the area under the curve drawn with precision and recall as the horizontal and vertical coordinates.

5.3 Experiment Methods

First, regarding to the initial re-ID model (that is, no normalization layer is added before the last full connection layer) after training and testing on three open datasets, the loss curve of training process is drawn and the test results are recorded. Then we add normalization layers respectively before the last full connection layer (classifier) of the original model. In each case, the model is trained and tested on the three open datasets, and the results are recorded to evaluate the additional effect of model performance of various normalization layer pairs at the last fully connected layer of the Re-ID model. The input dimensions of each normalization layer equals to the output dimensions of the average pooling layer i.e. 2048, and other parameters are acquiescently selected by PyTorch. The number of groups in GN layer is 32. After the original model and adding all kinds of normalization layers, the training and test phase settings on each dataset are exactly the same. The overall flow of the experiment is shown in Fig. 8.

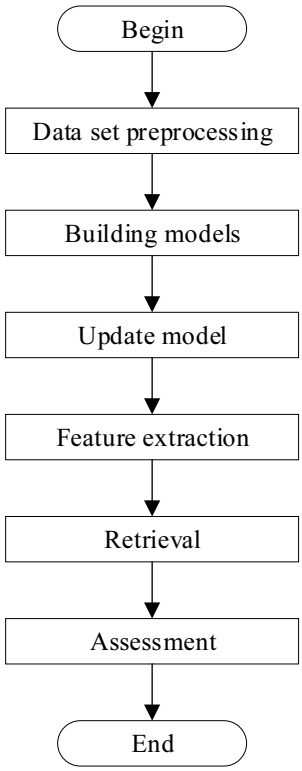

Fig. 8. Schematic diagram of the whole experimental process

The input image is resized to 256×128 when extracting image features. We first extract the original image features once, then flip the image horizontally to extract the second feature, and take the average of the two features as the final feature of the image.

5.4 Experiment Results

Market-1501. The training process loss curve of the original model (i.e. without adding the normalization layer), adding BN, GN and LN layers in the Market-1501 dataset before the classifier of the model is shown in Fig. 9. In order to better display the trend of loss curve, we select the first 30 epochs of results to show, and do the same treatment for the loss curve of the training process of the later two datasets. It can be seen from the figure that compared with the original model, adding normalization layers before the classifier of the model can significantly reduce the initial loss value of the model in the training process, and the loss reduction of the model is obviously increased rapidly, and it converges to the final loss value more quickly, and the final loss value results are smaller.

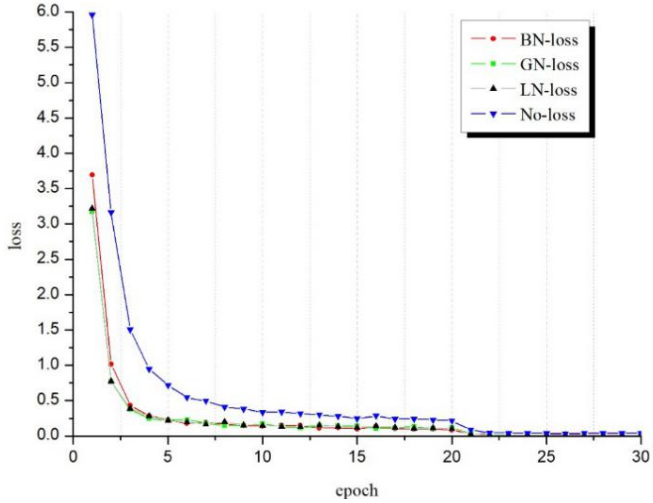

Fig. 9. Loss curve of Market-1501 in training process

The final test results of the four cases on Market-1501 are shown in Table 4. From the table, we can see that compared with the original model without adding normalization layer, in the Market-1501 dataset, the rank-1, rank-5 and mAP of the model after adding normalization layer are significantly improved. Adding BN layer increases the rank-1, rank-5 and mAP by 7%, 3% and 10% respectively on the left and right; adding GN layer is similar to BN, but slightly worse than BN in the improvement effect of mAP; adding LN layer improves the rank-1, rank-5 and mAP of the model. The effect of LN layer on model performance is very similar to that of GN. The above results show that adding these three normalization layers can effectively improve the performance of the model, and in the case of batch size of 32, BN has the best effect on the performance of the model.

Table 4. Test results of Market-1501

Method	Rank-1 (%)	Rank-5 (%)	MAP (%)
ResNet50	84.0	93.8	67.7
ResNet50+BN	91.4	96.8	77.1
ResNet50+GN	90.8	96.9	75.8
ResNet50+LN	91.1	96.6	75.9

DukeMTMC-reID. The loss curve of training process in the DukeMTMC-reID dataset in four cases is shown in Fig. 10. It can be seen from the figure that the initial loss value is lower and the loss curve converges faster after adding the normalized layer.

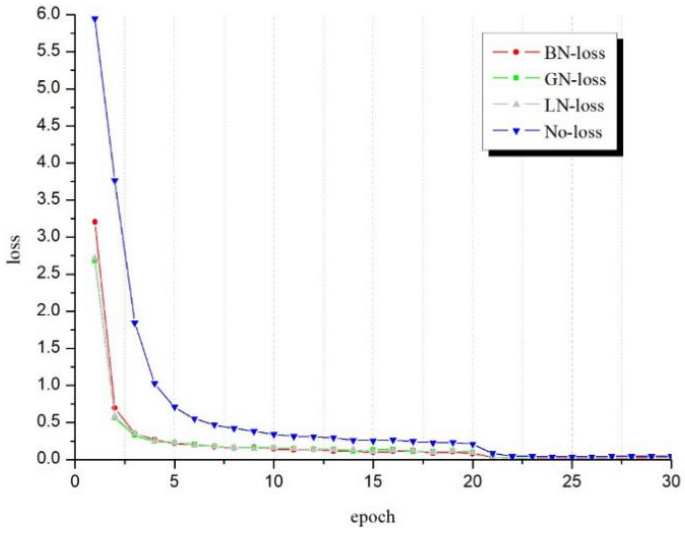

Fig. 10. Training process loss curve on DukeMTMC-reID

The final test results of the four cases are shown in Table 5. In the DukeMTMC-reID dataset, which is a little more complex, compared with the original model, adding various normalization layers has a stronger effect on the performance improvement of re-ID model. Adding BN layer can improve the model rank-1, rank-5 and mAP by 7%, 5% and 9% respectively; adding GN layer can improve the model rank-1, rank-5 and mAP by 6%, 5% and 7% respectively; adding LN layer has a very similar effect on the model performance. It can be seen that when batch size is 32, the BN improvement effect is still the best on this dataset.

Table 5. Test results of DukeMTMC-reID

Method	Rank-1 (%)	Rank-5 (%)	MAP (%)
ResNet50	75.0	85.8	57.7
ResNet50+BN	82.4	91.2	66.5
ResNet50+GN	81.0	90.5	63.3
ResNet50+LN	80.8	90.4	64.0

MSMT17. We further evaluated the impact of adding various kinds of normalization before the final full connection layer on model performance on the recently published larger dataset MSMT17. The loss curve of four cases in the training process is shown in Fig. 11, and the final test results are shown in Table 6. From the test results in the table, it can be found that the effect of adding various normalization layers on MSMT17 dataset with larger data scale and more complex picture environment is more powerful.

Table 6. Test results of MSMT17

Method	Rank-1 (%)	Rank-5 (%)	MAP (%)
ResNet50	52.2	69.7	29.5
ResNet50+BN	68.4	81.2	39.4
ResNet50+GN	61.8	76.3	32.1
ResNet50+LN	63.2	77.2	33.0

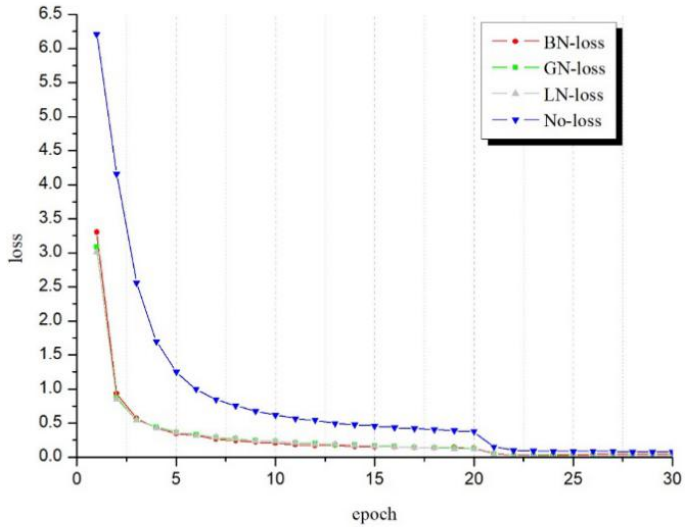

Fig. 11. Training process loss curve on MSMT17

Adding BN layer can improve the model rank-1, rank-5 and mAP by 16%, 10% and 10% respectively; adding GN layer can improve the model rank-1, rank-5 and mAP by 9%, 7% and 3% respectively; adding LN layer has a very similar effect on the model performance, but slightly better than GN effect. In this dataset adding BN layer is still the best.

6 Conclusion

This paper analyzes and evaluates the influence of adding various normalization layers before the classifier of the re-ID model. We draw some conclusions here. In our work, we carry out experiments on three datasets of different scales, Market-1501, DukeMTMC-ReID and MSMT17 respectively. The conclusions we get are as follows. Firstly, adding normalization layer does effectively improve the performance of re-ID model, and can make the model have lower initial loss value and faster loss reduction speed in the training stage, and can be convergent more quickly. Second, when the batch size is 32, adding BN layer is the best way to improve the performance of re-ID model, followed by using LN and GN. Thus, adding the normalization layers can effectively improve the performance of the model in person re-ID tasks. This study provides a new way for other researchers to optimize re-ID models. In addition to the four normalization methods introduced in this article, we expect more practical normalization methods in the person re-ID field.

Acknowledgements

This work is supported by Cross-Training Plan Project of University High Level Talents of Beijing and the Natural Science Foundation of Beijing under Grant 4202026.

Reference

[1] Y.-J. Li, L. Zhuo, J. Zhang, J.-F. Li, H. Zhang, A survey of person re-identification, Acta Automatica Sinica (2018) 1554-1568.

[2] L. Zheng, Y. Yang, A.G. Hauptmann, Person re-identification: past, present and future. <https://arxiv.org/abs/1610.02984>, 2016.

[3] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, S.C. Hoi, Deep learning for person re-identification: a survey and outlook. <https://arxiv.org/abs/2001.04193>, 2020.

[4] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable person re-identification: A benchmark, in: Proc. IEEE International Conference on Computer Vision, 2015.

[5] K. Zhou, Y. Yang, A. Cavallaro, T. Xiang, Omni-scale feature learning for person re-identification, in: Proc. IEEE International Conference on Computer Vision, 2019.

[6] J.-W. Liu, H.-D. Zhao, X.-L. Luo, J. Xu, Research progress on batch normalization of deep learning and its related algorithms, Acta Automatica Sinica 46(6)(2020) 1090-1120.

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

[8] Z. Zheng, L. Zheng, Y. Yang, Unlabeled samples generated by GAN improve the person re-identification baseline in vitro, in: Proc. IEEE International Conference on Computer Vision, 2017.

[9] L. Wei, S. Zhang, W. Gao, Q. Tain, Person transfer GAN to bridge domain gap for person re-identification, in: Proc. 31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[10] M. Farenzena, Person re-identification by symmetrydriven accumulation of local features, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[11] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, in: Proc. 26th Annual Conference on Neural Information Processing Systems, 2012.

[12] D. Yi, Z. Lei, S.Z. Li, Deep metric learning for person re-identification, in: Proc. 22nd International Conference on Pattern Recognition, 2014.

[13] W. Li, R. Zhao, T. Xiao, X. Wang, Deep ReID: deep filter pairing neural network for person re-identification, in: Proc. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[14] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proc. 32nd International Conference on Machine Learning, 2015.

[15] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer Normalization. <https://arxiv.org/abs/1607.06450>, 2016.

[16] D. Ulyanov, A. Vedaldi, V. Lempitsky, Instance normalization: the missing ingredient for fast stylization. <https://arxiv.org/abs/1607.08022>, 2016.

[17] Y. Wu, K. He, Group normalization, in: Proc. 15th European Conference on Computer Vision, 2018.

[18] P.F. Felzenszwalb, D.A. Mcallester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: Proc. 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2008.

[19] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Proc. 29th Annual Conference on Neural Information Processing Systems 2015, 2015.

[20] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: Proc. NIPS 2017 Autodiff Workshop, 2017.

[21] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[22] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization. <https://arxiv.org/abs/1412.6980>, 2014.