Optimizing Person Re-Identification using Generated Attention Masks

Leonardo Capozzi[1,2][0000−0002−5162−7124], João Ribeiro Pinto[1,2][0000−0003−4956−5902], Jaime S. Cardoso[1,2][0000−0002−3760−2473], and Ana Rebelo[1][0000−0003−4776−6057]

[1] INESC TEC, Porto, Portugal
{leonardo.g.capozzi, joao.t.pinto, jaime.cardoso, arebelo}@inesctec.pt
[2] Faculdade de Engenharia da Universidade do Porto, Porto, Portugal

Abstract. The task of person re-identification has important applications in security and surveillance systems. It is a challenging problem since there can be a lot of differences between pictures belonging to the same person, such as lighting, camera position, variation in poses and occlusions. The use of Deep Learning has contributed greatly towards more effective and accurate systems. Many works use attention mechanisms to force the models to focus on less distinctive areas, in order to improve performance in situations where important information may be missing. This paper proposes a new, more flexible method for calculating these masks, using a U-Net which receives a picture and outputs a mask representing the most distinctive areas of the picture. Results show that the method achieves an accuracy comparable or superior to those in state-of-the-art methods.

Keywords: Attention · Person Re-Identification · Feature Extraction · Deep Learning

1 Introduction

Person re-identification (re-ID) consists of matching a query image of an individual to other pictures of that same individual. The pictures are taken from a system of different cameras, each with a different view. Hence, this problem has important applications in security and surveillance systems. This task can be quite challenging, due to the differences in the position of the cameras, where the face of the person might not be visible, the possibility of occlusions, the large variation in poses, and differences in the background.

In the past, the problem of person re-ID was treated as a classification problem, and a common strategy was to use handcrafted features [11–13, 15, 26, 32]. This was due to the small size of the datasets available in the past. With the rising interest in solutions for this problem, more complex datasets have been created. These new datasets include more identities and more pictures per identity. The approach to the problem has also changed with the use of Deep Learning, where instead of using handcrafted features, the networks encode the images by

extracting relevant information into feature vectors that can be used to compare different images, using simple distance metrics [4, 17, 21, 23, 28].

The use of the same datasets, with the same train-test splits, allows to benchmark different methods, in order to compare their performance. Measures such as the Mean Average Precision (mAP) and the Cumulative Match Characteristic (CMC) curve are commonly used to compare the results.

One of the most commonly used loss functions is the Triplet Loss, which modifies the embedding space to pull together pictures belonging to the same person, and to push apart pictures of different identities [6]. This loss is also used with a batch mining process, which makes training more effective by choosing triplets where the error of the network is higher [8].

Many works also use attention mechanisms to force the network to focus on less distinctive areas, in order to improve performance in situations where important information may be missing [4, 10, 17, 21, 22].

In [4], the authors propose a method that randomly drops part of the input images in order to force the network to include less distinctive areas in the creation of the feature maps. This leads to a more robust network capable of performing better in situations where information may be missing or occlusions might occur.

In order to improve results, Quispe and Pedrini [17] proposed a method that instead of dropping random parts of the image, created an activation map. This allowed them to drop areas of the image with highly distinctive information, in order to force the network to perform better in situations where information was missing, by using less distinctive information. Although this method calculates which parts of the image it should drop, it can only drop horizontal stripes, which is limited and might lead to suboptimal results.

In this paper, we present a more flexible method, capable of calculating such masks using an auxiliary segmentation model, which allows it to define masks of any shape. This also removes the constraint of deriving the mask directly from the activations, which allows us to calculate the mask in a less predetermined manner.

2 Proposed Methodology

The proposed method is comprised of two networks. The feature extraction network uses a ResNet-50 backbone [7], pretrained on ImageNet [5]. This network receives as input an image and outputs a vector containing 512 features. This feature vector can be used to compare different pictures by computing their euclidean distance. After the network is trained, pictures containing the same individuals will have features vectors with a small distance and pictures with different individuals will have feature vectors with a large distance. These feature vectors allow us to sort a gallery of images. This is achieved by taking a query image and computing its distance to each image in the gallery.

This method also makes use of a mask generation network, to make the feature extraction network more robust. The mask generation network is trained to

output a mask that highlights the parts of the image that the feature extraction network is using to generate the feature vector. To make the feature extraction network more robust we hide 30% of the image, by choosing the pixels where the mask has the highest values. This forces the feature extraction network to extract relevant information from other parts of the image. We set the drop ratio to 30% as it was the recommended value in [4].

2.1 Network Architecture

Feature Extractor The backbone is comprised of a ResNet-50 Network, pretrained on ImageNet. Using a pretrained backbone helps our model to more easily extract relevant features in an image, by using previously learnt filters. The last layer of the ResNet-50 network was removed, therefore the output from the ResNet-50 model is a feature vector of shape 2048x1x1 (channels, height and width respectively). Then we added a convolutional layer with 512 filters, with a kernel size of 1×1 and stride 1×1, followed by batch normalization and ReLU activation.

The output of the network is a template that can be used to match individuals. This template is a unidimensional vector containing 512 features, that describes the picture of the person that was passed to the network. The picture is matched against a gallery of pictures containing a variety of different individuals, by computing the euclidean distance between the feature vectors and sorting the gallery based on the distance metric.

Mask Generator The mask generation part of the model is performed by a U-Net [20], with an encoder that reduces the resolution at each level, and a decoder that increases the size back to its original resolution. The U-Net features skip connections between layers of the encoder and the decoder that have the same resolution. This allows the transmission of information between layers, which avoids information loss during the encoding and decoding process.

The encoder is composed of 9 Convolutional layers and 4 Max-Pooling layers distributed over 5 resolution levels. Convolutional layers have $32 - 512$ filters, with a size of 3×3 and stride 1×1, each followed by a Batch Normalization layer and a ReLU activation function. Max-Pooling layers use a window size of 2×2 and a stride of 2×2, which reduces the resolution by a factor of 2 at each level.

The decoder mirrors the architecture of the encoder with convolutional and deconvolutional layers. Convolutional layers have 32 to 512 filters, with a size of 3×3 and stride 1×1, each followed by batch normalization and ReLU activation. Deconvolution Layers have $32 - 256$ filters with a size of 2×2 and stride 2×2, which increases resolution by a factor of 2.

The final layer is a Convolutional layer and has one filter with size 1×1 and stride 1×1, followed by a sigmoid activation function to have pixel values between 0 and 1 for the mask.

2.2 Loss

Feature Extractor The feature extraction model uses the triplet loss, the goal of this loss function is to reduce the distance between feature vectors of pictures belonging to the same individual, and to increase the distance between feature vectors of pictures belonging to different individuals. In the experimental settings section, we go over the selection process of the triplets.

The triplet loss can be written as:

$$\mathcal{L}_{triplet}(F) = \mathbb{E}_{a,p,n}[max(||F(a) - F(p)||_2 - ||F(a) - F(n)||_2 + m, 0)]. \quad (1)$$

where F is the feature extractor, a is the anchor image, p is the positive image, n is the negative image and m is the margin.

When both models are being trained simultaneously we take a weighted sum of the triplet loss of the regular images and the triplet loss of the images with the mask applied.

$$\mathcal{L}_{extractor}(F) = \lambda_1 \mathcal{L}_{triplet}(F) + (1 - \lambda_1)\mathcal{L}_{tripletwithmask}(F). \quad (2)$$

where $\mathcal{L}_{tripletwithmask}$ is the triplet loss applied to masked images (where some parts of the image has been removed to force the network to get relevant information from less distinctive areas).

Mask Model In order to generate a mask that highlights the areas of the image that the feature extractor is using to calculate the feature vector, we use several losses combined.

The first component can be written as:

$$\mathcal{L}_{identity}(M) = \mathbb{E}_a[mse(F(a), F(M(a) \times a)]. \quad (3)$$

where mse is the mean squared error between two vectors, a is an image, M is the U-Net that generates the masks and F is the feature extractor. $M(a)$ is the mask generated for image a, and $M(a) \times a$ is the resulting image after applying the mask to image a.

The problem with this loss is that it will make the output of the mask model a mask filled with ones. This will make $M(a) \times a$ equal to a, minimizing the previous loss. This is not what we want, as our goal is to make the relevant pixels equal to one while making non-relevant pixels equal to zero. In order to achieve this we add a second loss component:

$$\mathcal{L}_{sparsity}(M) = \mathbb{E}_a[mean(M(a))]. \quad (4)$$

where $M(a)$ is the mask generated for image a. This loss function aims to reduce the mean value of the masks, in order to solve the previous problem.

In addition to both of these losses, we need to make the mask contiguous, in order to select an area of the image, and not single and separated pixels.

Hence, we add the third loss component:

$$\mathcal{L}_{contiguity}(M) = \mathbb{E}_a[\frac{1}{h \times w} \sum_{i,j} |M(a)_{i+1,j} - M(a)_{i,j}| + |M(a)_{i,j+1} - M(a)_{i,j}|]. \quad (5)$$

where $M(a)_{i,j}$ is the value of the mask at index (i, j). This loss calculates the difference of consecutive pixels of the mask (vertically and horizontally), then it calculates the absolute value and finally the mean.

The final loss function becomes:

$$\mathcal{L}_{mask}(M) = \lambda_2 \mathcal{L}_{identity}(M) + (1 - \lambda_2)(\mathcal{L}_{sparsity}(M) + \mathcal{L}_{contiguity}(M)), \quad (6)$$

where $\mathcal{L}_{sparsity}$ and $\mathcal{L}_{contiguity}$ losses are adapted from [18].

3 Experimental Settings

3.1 Data

The proposed model was trained on three datasets that are commonly used in the person re-identification problem.

Market1501 dataset [30] contains data collected from six cameras, on an open environment, in Tsinghua University. It contains images of 1501 individuals. 751 individuals are used for training and 750 individuals for testing. There are a total of 12936 images in the training set, 19732 images in the test set and 3368 query images.

DukeMTMC-reID dataset [19, 34] is a subset of the DukeMTMC dataset. The original dataset contains 85-minute videos of high-resolution captured from 8 different cameras. It contains images of 1404 individuals. 702 individuals are used for training and 702 individuals for testing. This dataset contains 16522 images in the training set, 17661 images in the testing set and 2228 query images.

CUHK03 dataset [9] is comprised of images collected from The Chinese University of Hong Kong (CUHK) campus. It contains images from 1467 identities collected from 5 different pairs of camera views. This dataset contains 7368 images for training, 5328 images for testing and 1400 query images. This dataset has two versions: labelled and detected. Labelled means that the bounding boxes were labelled by a human. Detected means that the bounding boxes were estimated by a pedestrian detector. This dataset is prone to missing body parts, misalignments and occlusions, especially on the "detected" version, which makes it more challenging.

3.2 Data Augmentation

There are a couple of pre-processing steps that are applied to the images during training. These improve training, as they make the model more robust to

changes and avoid overfitting. During training, images are resized to a resolution of 234×117 pixels (height and width, respectively). Then a random section of the image is cropped, with a size of 224×112 pixels. The image is then randomly flipped horizontally, with a probability of 0.5. After all these steps are applied we normalize the images using the mean and standard deviation from ImageNet. This is needed because we use a pretrained ResNet-50 model, which was trained with normalized images.

3.3 Training

The proposed model was trained in three stages. The first stage consists of training the feature extraction model using $\mathcal{L}_{triplet}$. The second stage consists of training the mask model (while keeping the feature extraction model frozen) using $\mathcal{L}_{mask}$. The third stage consists of training both models simultaneously using $\mathcal{L}_{extractor}$ for the feature extractor model and $\mathcal{L}_{mask}$ for the mask generation model. After many experiments we set $\lambda_1 = 0.90$ and $\lambda_2 = 0.95$.

Since we are using the triplet loss we need a strategy to select the triplets that maximize the error of the network. Choosing triplets randomly is not a viable strategy, as the network can easily adapt itself to most pictures. This would result in a loss of progress, as the majority of the selected triplets would already have a difference in distances larger than the margin. To overcome this issue we used batch hard triplet mining [8].

Due to GPU memory constraints we used a batch size of 60 pictures (20 individuals and 3 pictures per individual).

We train the model for 50 epochs on the first stage, 25 epochs on the second stage and 100 epochs on the third stage.

For the feature extraction model we used the Adam optimizer with a learning rate of 2×10^{-4} as the default learning rate of 1×10^{-3} would make the pretrained ResNet-50 model unstable and led to a collapsed model, where every feature vector outputted by the model had the same values. The learning rate of 2×10^{-4} worked well for every dataset tested.

For the mask model we used the Adam optimizer with the default values.

4 Results and Discussion

To evaluate the performance of the model we used the mean average precision (mAP) and the rank-1 accuracy. We computed these values for the test sets of the Market1501, DukeMTMC-reID dataset and CUHK03 dataset. There are two versions of the CUHK03 dataset, labelled (L) and detected (D).

After training the model with the Market1501 dataset the model attained a mAP of 63.9% and a rank-1 accuracy of 97.4%. On the DukeMTMC-reID dataset, the model reached an mAP of 61.1% and a rank-1 accuracy of 90.2%. On the CUHK03(L) dataset the model attained an mAP of 69.0% and a rank-1 accuracy of 93.4%. On the CUHK03(D) dataset the model reached an mAP of 66.9% and a rank-1 accuracy of 92.7%.

Table 1. Comparison to state-of-the-art approaches

	Market1501		DukeMTMC-ReID		CUHK03 (L)		CUHK03 (D)	
Method	mAP	rank-1	mAP	rank-1	mAP	rank-1	mAP	rank-1
IDE [31]	46.0	72.5	47.1	67.7	21.0	22.2	19.7	21.3
PAN [33]	63.4	82.8	51.5	71.6	35.0	36.9	34.0	36.3
DPFL [3]	73.1	88.9	60.0	79.2	40.5	43.0	37.0	40.7
HA-CNN [10]	75.7	91.2	63.8	80.5	41.0	44.4	38.6	41.7
PyrNet [14]	86.7	95.2	74.0	87.1	68.3	71.6	63.8	68.0
Auto-ReID [16]	85.1	94.5	–	–	73.0	77.9	69.3	73.3
MGN [24]	86.9	95.7	78.4	88.7	67.4	68.0	66.0	66.8
DenSem [27]	87.6	95.7	74.3	86.2	75.2	78.9	73.1	78.2
MHN [1]	85.0	95.1	77.2	89.1	72.4	77.2	65.4	71.7
ABDnet [2]	88.2	95.6	78.5	89.0	–	–	–	–
SONA [25]	**88.6**	95.6	78.0	89.2	**79.2**	81.8	**76.3**	79.1
OSNet [35]	84.9	94.8	73.5	88.6	–	–	67.8	72.3
Pyramid [29]	88.2	95.7	**79.0**	89.0	76.9	78.9	74.8	78.9
Top-DB-Net [17]	85.8	94.9	73.5	87.5	75.4	79.4	73.2	77.3
Proposed	63.9	**97.4**	61.1	**90.2**	69.0	**93.4**	66.9	**92.7**

Comparing the results to other state-of-the-art methods (table 1) we can see that our model has the best results regarding the rank-1 accuracy metric.

Regarding the mAP score, our method is slightly inferior, but aligned, to the alternative methods. This might suggest that our method is better at making a single prediction (as it has a better rank-1 accuracy than other methods) but feels greater difficulty in sorting the whole gallery. To solve this problem we could add multiple streams (or networks), as in [4, 17]. This would make the training more stable, since training with the masked images could have the opposite effect of what we want to achieve, which is to force the network to use less distinctive features to compute the feature vector, making the model more accurate.

Figure 1 shows the masks that were generated by our mask generation model. We select 30% of the pixels with the highest values and hide them, therefore black areas represent zones of high importance to our feature extraction model, while white areas represent zones of low importance. During training, we hide the high importance zones, in order to force our feature extraction model to use other parts of the image to generate the feature vectors. In figure 1 we can see that the mask generation model almost always tries to hide the person's face, as this is one of the most important attributes of the person for identification. It also hides information about clothing and other things that the feature extraction model finds useful.

Fig. 1. Masks generated by our method (the hidden parts are areas of high importance).

5 Conclusion

The proposed model generates masks that represent areas of the image with important information for the identification process. In order to improve the performance of our feature extraction algorithm, we train it with original images, and images with missing information. This forces the model to use less distinctive areas to compute the feature vector, which leads to better results.

Upon evaluation on three popular datasets, we show that the performance of the algorithm is superior to the alternatives regarding the rank-1 accuracy. However, when using the mAP metric the proposed algorithm was slightly inferior, but aligned, to the alternatives.

The process of removing information from the images could add noise and have the opposite effect that we want to achieve, making the training process unstable and decreasing the performance of the model. We tried to balance the losses to avoid this, but further efforts should be devoted to improving the architecture of the model, adding different streams to minimize the effects of removing information [4, 17]. These changes could improve the performance of the model on the mAP metric.

Acknowledgements

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020, within project UIDB/50014/2020, and within the PhD grant “SFRH/BD/137720/2018”.

References

1. Chen, B., Deng, W., Hu, J.: Mixed high-order attention network for person re-identification. CoRR **abs/1908.05819** (2019), http://arxiv.org/abs/1908.05819
2. Chen, T., Ding, S., Xie, J., Yuan, Y., Chen, W., Yang, Y., Ren, Z., Wang, Z.: Abd-net: Attentive but diverse person re-identification. CoRR **abs/1908.01114** (2019), http://arxiv.org/abs/1908.01114
3. Chen, Y., Zhu, X., Gong, S.: Person re-identification by deep learning multi-scale representations. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). pp. 2590–2600 (2017). https://doi.org/10.1109/ICCVW.2017.304
4. Dai, Z., Chen, M., Zhu, S., Tan, P.: Batch feature erasing for person re-identification and beyond. CoRR **abs/1811.07130** (2018)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
6. Dong, X., Shen, J.: Triplet loss in siamese network for object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015)
8. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. CoRR **abs/1703.07737** (2017)
9. Li, W., Zhao, R., Xiao, T., Wang, X.: Deepreid: Deep filter pairing neural network for person re-identification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 152–159 (2014). https://doi.org/10.1109/CVPR.2014.27
10. Li, W., Zhu, X., Gong, S.: Harmonious attention network for person re-identification. CoRR **abs/1802.08122** (2018)
11. Li, Z., Chang, S., Liang, F., Huang, T.S., Cao, L., Smith, J.R.: Learning locally-adaptive decision functions for person verification. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3610–3617 (2013). https://doi.org/10.1109/CVPR.2013.463
12. Liao, S., Hu, Y., Xiangyu Zhu, Li, S.Z.: Person re-identification by local maximal occurrence representation and metric learning. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2197–2206 (2015). https://doi.org/10.1109/CVPR.2015.7298832
13. Ma, A.J., Yuen, P.C., Li, J.: Domain transfer support vector ranking for person re-identification without target camera label information. In: 2013 IEEE International Conference on Computer Vision. pp. 3567–3574 (2013). https://doi.org/10.1109/ICCV.2013.443
14. Martinel, N., Foresti, G.L., Micheloni, C.: Aggregating deep pyramidal representations for person re-identification. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1544–1554 (2019). https://doi.org/10.1109/CVPRW.2019.00196
15. Pedagadi, S., Orwell, J., Velastin, S., Boghossian, B.: Local fisher discriminant analysis for pedestrian re-identification. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3318–3325 (2013). https://doi.org/10.1109/CVPR.2013.426
16. Quan, R., Dong, X., Wu, Y., Zhu, L., Yang, Y.: Auto-reid: Searching for a part-aware convnet for person re-identification. CoRR **abs/1903.09776** (2019), http://arxiv.org/abs/1903.09776

17. Quispe, R., Pedrini, H.: Top-db-net: Top dropblock for activation enhancement in person re-identification. arXiv preprint arXiv:2010.05435 (2020)
18. Rio-Torto, I., Fernandes, K., Teixeira, L.F.: Understanding the decisions of cnns: An in-model approach. Pattern Recognition Letters **133**, 373 – 380 (2020). https://doi.org/https://doi.org/10.1016/j.patrec.2020.04.004, http://www.sciencedirect.com/science/article/pii/S0167865520301240
19. Ristani, E., Solera, F., Zou, R.S., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. CoRR **abs/1609.01775** (2016), http://arxiv.org/abs/1609.01775
20. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. CoRR **abs/1505.04597** (2015)
21. Shen, Y., Li, H., Xiao, T., Yi, S., Chen, D., Wang, X.: Deep group-shuffling random walk for person re-identification. CoRR **abs/1807.11178** (2018)
22. Si, J., Zhang, H., Li, C., Kuen, J., Kong, X., Kot, A.C., Wang, G.: Dual attention matching network for context-aware feature sequence based person re-identification. CoRR **abs/1803.09937** (2018)
23. Sun, Y., Zheng, L., Deng, W., Wang, S.: Svdnet for pedestrian retrieval. CoRR **abs/1703.05693** (2017)
24. Wang, G., Yuan, Y., Chen, X., Li, J., Zhou, X.: Learning discriminative features with multiple granularities for person re-identification. CoRR **abs/1804.01438** (2018), http://arxiv.org/abs/1804.01438
25. Xia, B.N., Gong, Y., Zhang, Y., Poellabauer, C.: Second-order non-local attention networks for person re-identification. CoRR **abs/1909.00295** (2019), http://arxiv.org/abs/1909.00295
26. Yang, Y., Yang, J., Yan, J., Liao, S., Yi, D., Li, S.Z.: Salient color names for person re-identification. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 536–551. Springer International Publishing, Cham (2014)
27. Zhang, Z., Lan, C., Zeng, W., Chen, Z.: Densely semantically aligned person re-identification. CoRR **abs/1812.08967** (2018), http://arxiv.org/abs/1812.08967
28. Zhao, L., Li, X., Wang, J., Zhuang, Y.: Deeply-learned part-aligned representations for person re-identification. CoRR **abs/1707.07256** (2017)
29. Zheng, F., Sun, X., Jiang, X., Guo, X., Yu, Z., Huang, F.: A coarse-to-fine pyramidal model for person re-identification via multi-loss dynamic training. CoRR **abs/1810.12193** (2018), http://arxiv.org/abs/1810.12193
30. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person re-identification: A benchmark. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1116–1124 (2015). https://doi.org/10.1109/ICCV.2015.133
31. Zheng, L., Yang, Y., Hauptmann, A.G.: Person re-identification: Past, present and future. CoRR **abs/1610.02984** (2016)
32. Zheng, W., Gong, S., Xiang, T.: Reidentification by relative distance comparison. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(3), 653–668 (2013). https://doi.org/10.1109/TPAMI.2012.138
33. Zheng, Z., Zheng, L., Yang, Y.: Pedestrian alignment network for large-scale person re-identification. CoRR **abs/1707.00408** (2017)
34. Zheng, Z., Zheng, L., Yang, Y.: Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. CoRR **abs/1701.07717** (2017), http://arxiv.org/abs/1701.07717
35. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Omni-scale feature learning for person re-identification. CoRR **abs/1905.00953** (2019), http://arxiv.org/abs/1905.00953