J Supercomput
https://doi.org/10.1007/s11227-017-2221-5

Deep feature learning for person re-identification in a large-scale crowdsourced environment

Seon Ho Oh[1] · **Seung-Wan Han**[1] · **Bum-Suk Choi**[1] · **Geon-Woo Kim**[1] · **Kyung-Soo Lim**[1]

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract Finding the same individual across cameras in disjoint views at different locations and times, which is known as person re-identification (re-id), is an important but difficult task in intelligent visual surveillance. However, to build a practical re-id system for large-scale and crowdsourced environments, the existing approaches are largely unsuitable because of their high model complexity. In this paper, we present a deep feature learning framework for automated large-scale person re-id with low computational cost and memory usage. The experimental results show that the proposed framework is comparable to the state-of-the-art methods while having low model complexity.

Keywords Person re-identification · Deep feature learning · Large-scale environment · Crowdsourcing · Visual surveillance

✉ Kyung-Soo Lim
luke.kyungsoo@gmail.com; lukelim@etri.re.kr

Seon Ho Oh
seonho@etri.re.kr

Seung-Wan Han
hansw@etri.re.kr

Bum-Suk Choi
bschoi@etri.re.kr

Geon-Woo Kim
kimgw@etri.re.kr

[1] Electronics and Telecommunications Research Institute, Daejeon, Korea

Published online: 01 February 2018

1 Introduction

With the increasing demand of security and public safety, automated person re-identification has received much attention in the past five years [2,8,9,12,13,16,21,23,28,29]. Person re-identification (re-id) is the problem of finding the same individuals across multiple cameras in disjoint views at different locations and/or time, or across time within a single camera. It has many applications such as video surveillance for security, public safety, human-computer interaction, robotics and content-based video or image retrieval [10]. Despite the best efforts from the computer vision and pattern recognition research community, re-id remains an unsolved problem because of the dramatic variations in visual appearance and ambient environment caused by different view points from different cameras, significant body poses across time and space, illumination changes, background clutter and occlusions. Furthermore, the problem becomes more difficult if different individuals have similar appearance, e.g., they wear similar clothes. Some examples are shown in Fig. 1.

Fig. 1 Samples of pedestrian images observed in different camera views in person re-identification. Two adjacent images have identical identities. Each row shows examples from the same dataset: CUHK01, CUHK03 and Market-1501

The advances in mobile technology have enabled a new paradigm to accomplish large-scale sensing, which is known in the literature as participatory sensing [4]. The key idea of participatory sensing is to enable ordinary citizen to use their mobile phones to collect and share the data from their surrounding environments. Crime stoppers [22], which enable a member of the community to provide anonymous information about a criminal activity, is the most well-known participatory program. From this viewpoint, crowdsourced participatory sensing has great potential for public security and safety area [15]. The automated person re-id with crowdsourced videos or images will improve and provide public safety opportunities that can take advantage of the citizen participation in cities worldwide. Meanwhile, it is interesting to note that unpaid crowdsourcing yields results of similar or higher quality than its paid counterpart[3].

Numerous studies have been conducted on person re-id, but the existing approaches are not applicable to large-scale crowdsourced environments because of the higher model complexity [21]. For example, VGG-16 model [17] has 138.34 million parameters, taking up more than 500 MB storage space, and needs 15.47 billion floating point operations (FLOPs) to process a single 224×224 RGB image. AlexNet [11] has 60.95 million parameters which require more than 240 MB storage space, and 726.79 million FLOPs. As a result, it is difficult to expect a fast response from large-scale crowdsourced environments, and it is unsuitable for preprocessing the input image or video in crowdsourced devices.

The current person re-id methods are typically solved by either complex iterative optimization [2,13] or costly generalized eigenproblems [14,18,27]. Thus, they require long time to train when the data size grows. Moreover, most models are restricted to learning with a fixed number of gallery set images; they cannot handle the situation when new data become available. In a real-world scenario, a human operator may generate new data in the deployment process.

To make practical person re-id more suitable for large-scale and crowdsourced environments, a model must be simple with a notably fast inference algorithm and scalable to new data. Thus, we propose a deep feature learning framework for person re-id. In order to reduce the model complexity while maintaining the network deeper, we use the Inception concept [19]. We borrow the idea from [19] to use unit hypersphere embedding to jointly learn the feature and distance metric. For the sake of simplicity and robustness of matching, we also introduce a notably simple but effective minimum average distance matching.

The main contributions of this work are as follows:

- A compact deep neural network for person re-id in large-scale environments, which can run on crowdsourcing devices.
- A minimum average distance matching strategy, which enables fast and robust matching.
- An extensive experiment on three public person re-identification benchmarks including CUHK01 [12], CUHK03 [13] and Market-1501 [29], which shows that the proposed simple and computationally efficient approach achieves comparable performance to the state-of-the-art methods except CUHK01.

This paper is structured as follows: Sect. 2 briefly overviews previous studies on person re-id. Sect. 3 describes the overall framework of our person re-id method in detail. Sect. 4 discusses the experimental results. Finally, Sect. 5 concludes the paper.

2 Related work

Given a query image, the typical person re-id pipeline extracts a feature to describe the query image, and subsequently finds the individual by comparing the features across gallery images. Thus, existing works on person re-id commonly focus on designing invariant and discriminant features [6,28], learning robust and discriminative similarity metrics to compare the features [5,8,9], or both [2,13,26].

The first group focuses on designing a good feature representation, which is discriminative and invariant for describing the appearance under various changes and conditions. The symmetry-driven accumulation of local feature (SDALF) [6] exploits both symmetric and asymmetric properties of a person by representing each part of a person with a weighted color histogram, maximally stable color regions and texture information. Saliency information has been investigated [28] by estimating rare patches among different images to match rare appearances such as rare-colored coats, baggage and folders.

The second group focuses on a supervised metric/distance learning. The basic idea of these approaches is to find a projection from the feature space to the distance space so that the projected Mahalanobis-like distance is small when the feature vectors represent the same person and large otherwise. These metric learning methods include Mahalanobis metric learning (KISSME) [9], Large Margin Nearest Neighbor Learning (LMNN) [8] and Information Theoretic Metric Learning (ITML) [5]. KISSME [9] exploits the equivalence constraints, which consider a log-likelihood ratio test of two Gaussian distributions.

The third group jointly learns the feature and distance metric. With recent great success on deep learning in various computer vision and pattern recognition tasks, deep learning-based approaches have become the main trend in person re-id. Yi et al. [26] proposed a Siamese neural network with a symmetric structure that comprised two independent subnets to learn the pairwise similarity. In their work, images are partitioned into three overlapped parts to train three independent networks. Finally, the three networks are fused at the score level, and the cosine distance is used as their metric. Li et al. [13] proposed the filter pairing neural network (FPNN), which jointly handles the problem of misalignment, photometric and geometric transforms, occlusion and black cluster, etc. They used a patch-matching layer to match the filter responses of local patches across the views. Ahmed et al. [2] presented another architecture called JointRe-id that took a pair of images as its input and output a similarity value, which indicated whether the two input images depicted the same person. They introduced a layer to compute the cross-input neighborhood differences to capture local relationships between two input images based on their mid-level features and a patch summary layer to obtain high-level features after two layers of convolution and max pooling. Wu et al. [23] enhanced the architecture of JointRe-id [2] by using deeper stack of tied convolutional layers that have small filter size before cross-input

neighborhood difference layer. However, these deep models taking a pair of images as its input learn a network with a binary classification, which is tending to predict most of inputs pairs as negative due to the great imbalance of training data. [24]. Moreover, image pairs may not be available in a crowdsourced environment.

Our architecture differs from these previous approaches. We adopt the **nn4** model from FaceNet [19] to tackle the problem of the person re-id while keeping the network deeper and having fewer parameters for crowdsourced environments. Moreover, joint learning of the representation and distance metric with unit hypersphere embedding simplifies the training and matching tasks. Consequently, our network achieves comparable performance on the CUHK03 dataset [13] and Market-1501 dataset [29] while having low model complexity. To the best of our knowledge, this paper is the first work to use deep feature learning for person re-identification for large-scale crowdsourced environments.

3 Methodology

In this section, we present person re-id method in detail. First we describe the network architecture of the proposed deep feature learning framework. Then, we elaborate the training strategy to train the proposed model.

3.1 Architecture

To use a deep learning-based approach in a large-scale crowdsourced environment, the complexity of the deep neural network should be considered. In order to reduce the model complexity while maintaining the network deeper, we use the inception layer of GoogleNet [19]. The inception layers [19] performs cross-channel correlations while ignoring spatial dimensions through a 1×1 convolution; this dramatically reduced the dimensionality in the filter dimension. In addition to the 1×1 convolution, concatenating the responses from convolutional filters with different sizes that represent cross-spatial and cross-channel correlations can handle different clusters of information. Moreover, max pooling before convolution allows both deeper and larger convolutional layers and more efficient computation by reducing the dimension.

The proposed network model mainly consists of the following distinct layers: three convolution layers, three pooling layers, five inception layers and one embedding layer. We use a 60×160 RGB image as input of the proposed network model. Table 1 illustrates the network model for person re-identification in details. Each row describes a layer in the network. The total number of parameters for our model including batch normalization is 5.21 million, and needs 237.13 million FLOPs. The proposed network is 11.7 and 26.5 times smaller than AlexNet and VGG-16 model, respectively. And our network requires 3.07 and 65.24 times less FLOPs than AlexNet and VGG-16 model, respectively.

Our model has a 7×7 convolutional layer in front (named conv1), followed by max pooling layer. Two convolutional layers (named conv2, conv3) which have 1×1 and 3×3 filters followed by max pooling layer are added. We exploit conv1 to conv3

followed by max pooling as the stem part. Given an input image, the stem part will produce 192 channels of features map, which have 1/16 resolutions of the input image.

On top of these feature maps, stack of inception layers is added. Our network uses two types of inception layer with small variation. The last seven columns of Table 1 describe the parameters of the inception layers from [19] and the number of parameters for each layer. The columns starting with "#N $\times$ N" denote the depth of the output feature map, and "#3 $\times$ 3 reduce" and "#5 $\times$ 5 reduce" represent the number of 1 $\times$ 1 filters that were used in the reduction layer before 3 $\times$ 3 and 5 $\times$ 5 convolutions. The "pool proj." column describes pooling type, the size of the dimensions to be projected or pooling kernel size and stride. The average pooling layer summarizes the features map into 8 $\times$ 2 $\times$ 736.

The embedding layer is a composition of the fully connected layer and the L_2 normalization layer. A fully connected layer linearly combines 8 $\times$ 2 $\times$ 736 feature maps into d-dimensional vector ($d = 256$ in this paper). Then, the following L_2 normalization layer constrains the embedding vector x to live on a d-dimensional hypersphere, i.e., $\|x\|_2 = 1$, and it enables a simple nearest neighbor matching.

3.2 Training strategies

The choice of the proper loss function is very important to training the network. FPNN [13], JointRe-id [2], and PersonNet [23] used softmax losses because they only distinguish whether the given pair of images is the same or different. GatedSiames [20] used contrastive loss to train deep Siamese CNN architecture. In [19], they used triplet loss, which enforces that the embeddings of the same person are closer together, and the embeddings of different people are farther apart in the learned embedding space. However, the triplet loss requires the three input images (anchor, positive, negative) and it restricts the batch size and difficult to converge. The deep models with a binary classification [2,13,23] uses the softmax loss. However, they require a pair of images as its input, and suffer from the imbalance of training data. Recently proposed Online Instance Matching (OIM) [25] loss does not have any restrictions on the input batch size and converges quickly while minimizing the features difference among the instances of the same person and maximizing the distance among different people. Thus, we use OIM loss to train our network model. In addition, the softmax loss with multi-class classification is also used to compare the results.

Suppose there are L different target people in the training data. During the training, we maintain a lookup table (LUT) $V \in \mathbb{R}^{D \times L}$ to store the features of all identities, where D is the feature dimension. Following the definition of the OIM loss in [25], we define the probability of the feature x being recognized as the identity with class-id i by Softmax function

$$p_i = \frac{\exp v_i^T x \tau}{\sum_{j=1}^{L} \exp(v_j^T x / \tau)}, \tag{1}$$

where the higher temperature τ leads to softer probability distribution. The update of the LUT is as follows. During the forward propagation, we compute the cosine similarities between the feature x and all the identities by $V^T x$. During the backward

Table 1 Details of the network model for the compact deep feature learning

Name	Output size	#1 × 1	#3 × 3 reduce	#3 × 3	#5 × 5 reduce	#5x5	Pool proj.	Param.
Conv1 (7 × 7 × 3,2)	80 × 30 × 64							9K
Max pool	40 × 15 × 64							
Conv2 (1 × 1 × 64,1)	40 × 15x64							4K
Conv3 (3 × 3 × 64,1)	40 × 15 × 192							110K
Max pool	20 × 8 × 192							
Inception (3a)	20 × 8 × 256	64	96	128	16	32	m, 32p	164K
Inception (3b)	20 × 8 × 320	64	96	128	32	64	m, 64p	64K
Inception (3c)	10 × 4 × 640		128	256, 2	32	64	m, 3 × 3, 2	398K
Inception (5a)	10 × 4 × 544	256	96	192			m, 96p	791K
Inception (5b)	10 × 4 × 736	256	96	384			m, 96p	662K
Avg pool	8 × 2 × 736							
Embedding	256							3.01M
Total								5.21M

propagation, if the target class-id is t, then we will update t-th column of the LUT by $v_t \leftarrow \gamma v_t + (1-\gamma)x$, where $\gamma \in [0, 1]$, and then normalize v_t to have unit l_2 norm.

To increase the volume of training data and alleviate the over-fitting problem, we augment the data by performing random crop and resize. To determine the width and height of the crop region from an original image of the size $W \times H$, we randomly select the crop area size and aspect ratio drawn from a uniform distribution in the range [0.64, 1.0] and [2, 3], respectively. Once the width and height of cropping region is obtained, we randomly crop the region from an original image. The cropped image is resized to 160×60. We also horizontally flip each image randomly.

4 Experimental results

4.1 Experimental settings

We implemented our architecture using the TensorFlow [1] deep learning framework. Network training converged in roughly 1–2 h on two NVIDIA Titan Xp GPUs. The training was carried out by optimizing loss functions using online sampling of the dataset with stochastic gradient descent (SGD). The temperature scalar τ in 1 for OIM loss was set to 1/30. The mini-batch size was set to 256 and train the network for 50 epochs for each of loss functions. Dropout was used before the embedding layers with a probability of 0.5 to alleviate over-fitting. The learning rate was initially set to 0.1 and subsequently then exponentially decayed with a factor of 0.1 every 40 epochs. The weights were initialized from zero-mean Gaussian distributions with the standard deviations of 0.01. The bias terms were set to 0. We used ReLU as the activation function and batch normalization on all convolution layers, including the inception layers.

We performed experiments on public benchmarks: the CUHK03 dataset [13], the Market-1501 dataset [29] and the CUHK01 dataset [12]. We adopted the widely used single-shot modality in our experiment to enables an extensive comparison. We employed two kinds of evaluation metrics: cumulative matching characteristics (CMC) and mean average precision (mAP) [29]. The former includes only the first match in the CMC calculation no matter how many ground truths match the gallery. Thus, CMC represents the probability that a query identity appears in different-sized candidate lists and rank-k is the k-th value of CMC curve. The latter first computes the area under the precision–recall curve for each query, which is known as an average precision (AP). Then, the mean value of APs of all queries, i.e., mAP, is calculated. Unlike CMC, mAP considers both precision and recall, thus providing more comprehensive evaluation.

Note that CUHK03 and Market-1501 dataset calculate CMC curve and CMC rank-k accuracy quite differently. The CUHK03 assumes that query and gallery are from different camera views. For each query, they randomly sample a single instance from each gallery's identity and compute a CMC curve from sampled gallery set. This random sampling process is repeated for N times, and the average CMC curve is reported. The Market-1501 assumes that the query and gallery sets could have same camera views. But for each individual query identity, his/her gallery samples from

the same camera are excluded. Consequently, the query will find the closest positive sample in the gallery.

For robust matching, we can use the class-wise matching strategy instead of the individual image-wise comparison. The matching distance is defined as

$$d(x_q, C) = \frac{1}{N_c} \sum_{i \in C} \left\| f(x_i) - f(x_q) \right\|_2^2 \tag{2}$$

where C is the images of a person, N_c is the number of images for that person, and x_q is the query image. Thus, the matching is simply finding a person that has the minimal average distance. The formulation can be further simplified with the mean and variance in each class, which helps in efficient matching.

4.2 Experiments on CUHK03 dataset

The CUHK03 dataset contains 13,164 images of 1,360 identities. The dataset was captured with six surveillance cameras. Each identity is observed by two disjoint camera views and has 4.8 images in each view on average. This dataset provides both manually cropped images and auto-detected ones using a prevailing pedestrian detector [7]. We report the results of a trained model using both data.

Following the protocol used in [13], we randomly divided 1360 identities into 1260 identities for training set and remaining 100 identities for the test set. We compared our method against eSDC [28], KISSME [9], FPNN [13], JointRe-id [2], and PersonNet [23]. Table 2 shows the performance comparisons of our model with other state-of-the-art methods. Our deep network outperforms the state-of-the-art methods on the CUHK03 dataset in the both of losses. The model trained with softmax loss shows that the proposed deeper, but lightweight architecture can successfully learn representation and distance metric jointly. And the performance gap between softmax loss and OIM loss shows the effectiveness of OIM loss.

4.3 Experiments on the Market-1501 dataset

The Market-1501 dataset contains 32,643 images of 1501 identities. Each identity was captured by at most six cameras, and boxes of person were obtained by DPM [7]. The dataset was divided into training and testing sets, which contained 751 and 750 identities, respectively. For training, 12,936 images were used. For testing, 19,732 and 3,368 images were used for the gallery set and probe set, respectively. We compared our model with state-of-the-art methods in Table 2. The results were reported using single-shot and single-query. Our deep network achieves comparable performance with the state-of-the-art methods on the Market-1501 dataset in the both of losses. Unlike the CUHK03, the performance gap between the two losses are only 0.9%, but the model trained with OIM loss gives better result.

Table 2 Method comparison on the CUHK03 and the Market-1501 datasets

Method	mAP	Rank@1	Method	mAP (%)	Rank@1 (%)
eSDC [28]	–	7.7%	LOMO [14]	7.7	26.1
KISSME [9]	–	11.7	eSDC [28]	13.5	33.5
FPNN [13]	–	19.9%	BoW+KISSME [9]	17.7	39.6
JointRe-id [2]	–	54.7%	PersonNet [23]	18.57	37.2
PersonNet [23]	–	64.8%	GatedSiamese [20]	39.6	65.9
Ours (softmax)	58.2%	65.0%	Ours (softmax)	46.9	70.8
Ours (OIM)	66.0%	72.6%	Ours (OIM)	47.6	71.7

Table 3 Method comparison on the CUHK01 dataset

Method	mAP	Rank@1 (%)
FPNN [13]	–	27.8
JointRe-id [2]	–	65.0
PersonNet [23]	–	71.1
Ours (softmax, scratch)	27.0%	29.9
Ours (OIM, scratch)	20.5%	23.2
Ours (softmax, fine-tuned)	26.8%	29.0
Ours (OIM, fine-tuned)	36.9%	53.6

4.4 Experiments on CUHK01 dataset

The CUHK01 dataset contains 971 identities with 2 images per person in each view. Each image is manually cropped and normalized to 160 × 64 pixels. We report the results in the setting where 486 identities were used for training, and the remaining 485 identities were used for testing. Since the 486 identities for training gives only 1940 images for training, thus it is practically impossible for a deep architecture of reasonable size not to under-fit if trained from scratch on these data. To solve this issue, we used fine-tuning which initializes the model by training on a larger dataset and then adopt it on the small dataset. In our experiment, we pre-trained a network on CUHK03 dataset and adopted it for CUHK01 dataset. The performance of the network trained from scratch and after fine-tuning was 29.9 and 53.6%, respectively. Table 3 compares the performance of our model with other state-of-the-art methods. Unlike the CUHK03 and Market-1501 results, the CUHK01 result fell short of the state-of-the-art performance. These results are presumed to be due to lack of proper regularization mechanism and insufficient number of samples in the training dataset.

4.5 Execution time

The execution time was evaluated on a GPU server (3.0 GHz CPU and 128 GB memory with NVIDIA Titian Xp) and two mobile devices: Samsung Galaxy S6 (1.5 GHz octa-core CPU, 3 GB RAM) and LG G5 (1.6 GHz quad-core CPU, 4 GB RAM)

Table 4 Comparison of the execution time on different devices

Devices	Feature extraction (ms)	Classification (ms)
Samsung Galaxy S6	182.22 ± 16.46	< 1
LG G5	89.29 ± 1.29	< 1
GPU server	1.74 ± 0.38	< 1

with Android 6.0.1. Table 4 compares the execution time on different devices. The execution time on the mobile device was obtained as an average of 50 iterations, and the execution time on the GPU server was obtained as an average of 1000 iterations to minimize the measurement error. Table 4 shows that the proposed model can be applied to crowdsourced participatory sensing while reducing the computational burden of the re-id system in practice.

5 Conclusion

In this paper, we have presented a deep feature learning framework for person re-identification in large-scale crowdsourced environments. We have designed deep architecture with low complexity by learning the features and distance metrics jointly through the unit hypersphere embedding, and adopting Inception concept. We introduced the minimum average distance matching strategy that enables simple but robust matching. We demonstrate the effectiveness of our method by conducting a comprehensive evaluation of our approach on various benchmark datasets. On two public benchmark datasets, CUHK03 and Market 1501, our method outperforms the state-of-the-art by a large margin. We also show that the proposed architecture can be run in real-time on the mobile devices. It shows that the proposed model can be applied to crowdsourced participatory sensing while reducing the burden of computation of the person re-identification system in practice.

Acknowledgements This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0717-16-0107, Development of Video Crowd Sourcing Technology for Citizen Participating-Social Safety Services and No. B0126-16-1007, Development of Universal Authentication Platform Technology with Context-Aware Multi-Factor Authentication and Digital Signature).

References

1. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/, software available from tensorflow.org
2. Ahmed E, Jones M, Marks TK (2015) An improved deep learning architecture for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3908–3916

3. Borromeo RM, Toyama M (2016) An investigation of unpaid crowdsourcing. Hum Cent Comput Inf Sci 6(1):11
4. Burke JA, Estrin D, Hansen M, Parker A, Ramanathan N, Reddy S, Srivastava MB (2006) Participatory sensing. Center for Embedded Network Sensing, Los Angeles
5. Davis JV, Kulis B, Jain P, Sra S, Dhillon IS (2007) Information-theoretic metric learning. In: Proceedings of the 24th International Conference on Machine Learning. ACM, pp 209–216
6. Farenzena M, Bazzani L, Perina A, Murino V, Cristani M (2010) Person re-identification by symmetry-driven accumulation of local features. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 2360–2367
7. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intel 32(9):1627–1645
8. Hirzer M, Roth PM, Bischof H (2012) Person re-identification by efficient impostor-based metric learning. In: 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS). IEEE, pp 203–208
9. Koestinger M, Hirzer M, Wohlhart P, Roth PM, Bischof H (2012) Large scale metric learning from equivalence constraints. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 2288–2295
10. Koteswara Rao L, Venkata Rao D (2015) Local quantized extrema patterns for content-based natural and texture image retrieval. Hum Cent Comput Inf Sci 5(1):26
11. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp 1097–1105
12. Li W, Zhao R, Wang X (2012) Human reidentification with transferred metric learning. In: Asian Conference on Computer Vision. Springer, pp 31–44
13. Li W, Zhao R, Xiao T, Wang X (2014) Deepreid: deep filter pairing neural network for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 152–159
14. Liao S, Hu Y, Zhu X, Li SZ (2015) Person re-identification by local maximal occurrence representation and metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2197–2206
15. Ogie RI (2016) Adopting incentive mechanisms for large-scale participation in mobile crowdsensing: from literature review to a conceptual framework. Hum Cent Comput Inf Sci 6(1):24
16. Paisitkriangkrai S, Shen C, van den Hengel A (2015) Learning to rank in person re-identification with metric ensembles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1846–1855
17. Parkhi OM, Vedaldi A, Zisserman A (2015) Deep face recognition. In: Proceedings of the British Machine Vision Conference 2015, vol 1, p 6
18. Pedagadi S, Orwell J, Velastin S, Boghossian B (2013) Local fisher discriminant analysis for pedestrian re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3318–3325
19. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 815–823
20. Varior RR, Haloi M, Wang G (2016) Gated siamese convolutional neural network architecture for human re-identification. In: European Conference on Computer Vision. Springer, pp 791–808
21. Wang H, Gong S, Xiang T (2016) Highly efficient regression for scalable person re-identification. arXiv preprint arXiv:1612.01341
22. Wikipedia (2017) Crime stoppers—Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Crime_Stoppers. Online; Accessed 2 March 2017
23. Wu L, Shen C, Hengel Avd (2016) Personnet: person re-identification with deep convolutional neural networks. arXiv preprint arXiv:1601.07255
24. Wu L, Shen C, van den Hengel A (2017) Deep linear discriminant analysis on fisher networks: a hybrid architecture for person re-identification. Pattern Recognit 65:238–250. https://doi.org/10.1016/j.patcog.2016.12.022
25. Xiao T, Li S, Wang B, Lin L, Wang X (2017) Joint detection and identification feature learning for person search. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
26. Yi D, Lei Z, Liao S, Li SZ (2014) Deep metric learning for person re-identification. In: 2014 22nd International Conference on Pattern Recognition (ICPR). IEEE, pp 34–39

27. Zhang L, Xiang T, Gong S (2016) Learning a discriminative null space for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1239–1248
28. Zhao R, Ouyang W, Wang X (2013) Unsupervised salience learning for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3586–3593
29. Zheng L, Shen L, Tian L, Wang S, Wang J, Tian Q (2015) Scalable person re-identification: a benchmark. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1116–1124