Distributed Implementation for Person Re-identification

Saurav Sthapit*, John Thompson*, James R. Hopgood* and Neil M. Robertson†
*Institute of Digital Communications, School of Engineering, University of Edinburgh
Emails:{s.sthapit, john.thompson, james.hopgood}@ed.ac.uk
†Vision Lab, School of Engineering and Physical Sciences, Heriot-Watt University
Email: n.m.robertson@hw.ac.uk

***Abstract*—Person re-identification is to associate people across different camera views at different locations and time. Current computer vision algorithms on person re-identification mainly focus on performance, making it unsuitable for distributed systems. For a distributed system, computational complexity, network usage, energy consumption and memory requirement are as important as the performance. In this paper, we compare the merits of current algorithms. We consider three key algorithms, Keep It Simple and Straightforward MEtric (KISSME), Symmetry-Driven Accumulation of Local Features (SDALF) and Unsupervised Saliency Matching (USM). The advantage of SDALF, and USM is that they are unsupervised methods so training is not required but computationally many time expensive than KISSME. The Saliency based method is superior in performance but also has the largest feature size. As the features needs to be transmitted from one camera to other in distributed system, this mean higher energy consumption and longer time delay. Among these three, KISSME offers a balance between performance, complexity and feature lengths and hence more suitable for distributed systems.**

I. Introduction

Person re-identification refers to associating people across camera views at different locations and times [1]. It can have huge impact on surveillance and security because manual identification is not only tedious and costly but the results may also be received too late. The main challenges it faces is that the Field Of View (FOV) of the cameras can be non-overlapping, background and pose can change, as well as the occurrence of occlusion. A particular individual can look dissimilar in different views, while different individuals can look similar from different angles. Figure1 shows some sample pedestrian images from the VIPeR dataset [2] taken by two cameras illustrating these difficulties.

Fig. 1. Samples of pedestrian images from VIPeR dataset [2]

Person re-identification algorithms can broadly be classified into supervised and unsupervised algorithms. Supervised methods include algorithms like Mid-level features [3], Keep It Simple and Straightforward MEtric (KISSME) [4], Locally Aligned Featrue Transform (LAFT) [5], Information Theoretic Metric Learning (ITML) [6]. They mostly focus on metric learning, whereas unsupervised algorithms focus on feature design. Some of the unsupervised methods include Symmetry-Driven Accumulation of Local Features (SDALF) [7], Bio-inspired Covariance based features (BiCov) [8] and spatio-temporal [9]. For a more detailed review of recent approaches, refer to these papers [1], [10], [11], [12].

Current research in this area, however, focusses on implementing their algorithm on a single system [7], [4], [13], [3]. Implementing person re-identification on a distributed system has numerous benefits which will be illustrated with the example shown in Fig. 2. The system comprises of multiple smart cameras which may be static or moving. They are shown in the Fig. 2 by black and white camera icons respectively. The cameras are connected to each other and their field of view may be non-overlapping. The targets 1 and 2 are moving along the path shown by the arrows.

In a centralised system, all the sensor nodes would have been connected to a single computer with immediate access to data from all the sensor nodes. But on the downside, it has to process the data itself, which may be challenging particularly in real-time applications. In the distributed case, each sensor node has access to its own data only but offers more flexibility for signal processing. Running it on wireless embedded platform such as smartphone could be possible, which means the cameras could be deployed and scaled easily. In a military context, this means the camera may be embedded within a soldier's uniform to monitor targets without raising suspicion in conflict zones. We can think of light cameras in Fig.2 as these soldiers monitoring target 2. however, along with the algorithm's accuracy, there are several other factors to think about such as feature data length, computational complexity etc.

In this paper, we discuss the advantages and the disadvantages of current person re-identification algorithms when implemented on a distributed platform. The paper is structured as follows. Section II describes the basic workflow in person re-identification. Then we analyse various algorithms in section III. Section IV describes the experiments carried out and their results. Finally section V discusses the results and concludes the paper.

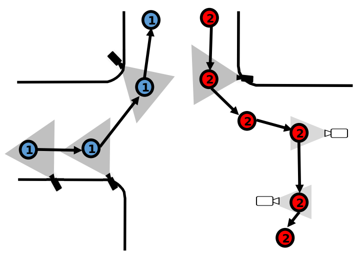

Fig. 2. Scenario of multi-camera person re-identification. Shaded cameras are fixed, white cameras are moving and grayed area represent Field of View (FOV)

II. SYSTEM DESCRIPTION

Person re-identification algorithms generally follow the basic workflow depicted in Fig. 3. Images are taken from each camera and preprocessed. The pre-processing step may include background subtraction and a person detection algorithm. To create a unique signature of each person, features are extracted. Popular features include combination of low level features such as colour histograms, Local Binary Patterns (LBP) [14], Scale Invariant Feature Transform (SIFT) [15] and Histogram of Gradient(HOG) [16]. Metric distance between signatures is calculated to verify if the images belong to the same individual or not. Alternatively, the test signature may be compared with the gallery set containing signatures of a seen individual to find the correct match. Some researchers have defined the person identification problem as a ranking problem [17].

In the distributed case, the signature has to be communicated from one camera to another as shown in the Fig. 3. Very often, these camera are connected with wireless networks such as Wi-Fi or cellular system. We know that the time taken and energy required to send the data across the network is directly proportional to the length of the data [18]. We conduct an experiment to quantize the energy and time required for such system in section IV-A.

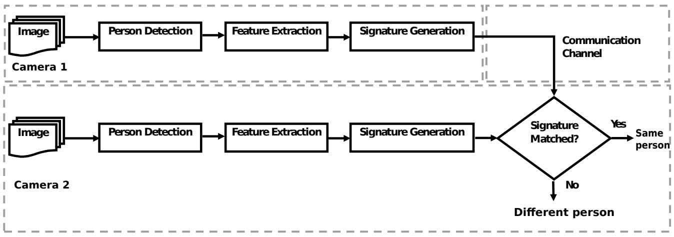

Fig. 3. Person Re-identification workflow.

Depending upon the number of images used, algorithms can be classified into single-shot and multi-shot algorithms. Single-shot algorithms take into account only one image per person (class) whereas multiple-shot algorithms uses multiple images. Multi-shot algorithms tries to keep the signature data size low and keep the matching considerably fast by throwing away redundant information.

A. Distributed Scenario

For implementing the re-identification system on a distributed system, let us assume each camera in Fig.3 has its own processing capability. So each sensor node can generate signature for the people in its FOV. For signature matching, one device has to send their signature to its neighbour so that it can be matched with its camera views. These are often battery powered devices, such as a smartphone, so longevity of the battery is desired. As it is desirable to keep the signature size as small as possible, we analyse the size of descriptors of the algorithms in consideration. Distributed systems are equipped with less powerful processors and have less memory resources, so the complexity of the algorithm is desired to be as low as possible. In order to measure complexity, we measure the time taken to run. For this paper, we have run our experiments on a desktop computer.

B. Datasets

Popular publicly available datasets for person re-identification are listed in Table.I. VIPeR is the most widely used and challenging dataset, one of the reason being limited samples per subject. We have used the VIPeR dataset in our experiments because many published algorithm comparisons are available.

III. PERSON RE-IDENTIFICATION ALGORITHMS

Among many algorithms, we have selected three key ones owing to their significance in person re-identification and availability of their source code. We go through them very briefly here.

A. KISS MEtric Learning

Keep It Simple and Straightforward Metric (KISSME) [4] focusses on learning the metric rather than complicated descriptor design. For the descriptor, images are divided into overlapping blocks and histograms are extracted in HSV and LAB colour-space. Local Binary Patterns (LBP) [14] are extracted to capture the texture information. For the VIPeR dataset, based on the code and data[1] provided by authors [4], each image has 22154 dimension features. Principal Component Analysis (PCA) is used by the authors to shorten the length of the descriptor to 34 experimentally chosen dimensions.

The Mahalanobis Metric learning is a widely used method in classification and in computer vision. It is defined as the squared distance between two points x_i and x_j as

$$d^2_M(x_i, x_j) = (x_i - x_j^T)\boldsymbol{M}(x_i - x_j) \tag{1}$$

where $\boldsymbol{M} \succeq 0$ is a positive semi-definite matrix.The main approach of Mahalanobis based algorithms is to define and learn the matrix $\boldsymbol{M}$ such that distance between images of same class is minimised and distance between images of different classes are maximised. KISSME [4], ITML, [6], LDML [21] and LAFT [5] are based on these methods. A detailed review of Mahalanobis based methods can be found in Roth et al's paper [22]. KISSME tries to address the metric learning approach from a statistical inference point of view. They test the hypothesis H_0 that the pair is dissimilar versus the alternative hypothesis H_1 that the pair is similar.

$$\delta(\mathbf{x}_{ij}) = \log\left(\frac{p(\mathbf{x}_{ij}|H_0)}{p(\mathbf{x}_{ij}|H_1)}\right) = \log\left(\frac{f(\mathbf{x}_{ij}|\theta_0)}{f(\mathbf{x}_{ij}|\theta_1)}\right) \tag{2}$$

where $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ is the pairwise difference with zero mean. A high value of $\delta(\mathbf{x}_{ij})$ means the pair are dissimilar and vice-versa. Assuming a Gaussian structure of the difference space, Eq. 2 can be written as

$$\delta(\mathbf{x}_{ij}) = \log\left(\frac{\frac{1}{\sqrt{2\pi|\Sigma_{y_{ij}=0}|}}\exp(-1/2\,\mathbf{x}_{ij}^T\Sigma^{-1}_{y_{ij}=0}\,\mathbf{x}_{ij})}{\frac{1}{\sqrt{2\pi|\Sigma_{y_{ij}=1}|}}\exp(-1/2\,\mathbf{x}_{ij}^T\Sigma^{-1}_{y_{ij}=1}\mathbf{x}_{ij})}\right) \tag{3}$$

where,

$$\Sigma_{y_{ij=0,1}} = \sum_{y_{ij}=0,1}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{4}$$

They arrive at the Mahalanobis distance metric in Eqn.1 that reflects the properties of the log-likelihood ratio test by re-projecting $\hat{\boldsymbol{M}} = \left(\Sigma^{-1}_{y_{ij}=1} - \Sigma^{-1}_{y_{ij}=0}\right)$ onto the cone of positive semi-definite matrices.

[1]accessible from https://lrs.icg.tugraz.at/research/kissme/

TABLE I. POPULAR PERSON RE-IDENTIFICATION DATASETS

Dataset	No. of Person	No. of Images	Features
VIPeR [2]	632	1264	pose, background, only 1 image per subject per camera
CAVIAR4REID [19]	72	1220	pose, background, varying resolution, multiple images per subject per camera
CUHK01 [20]	971	3884	pose, background, multiple images per subject per camera

B. Symmetry-Driven Accumulation of Local Features(SDALF)

SDALF [7] is suitable for single-shot and multi-shot images. The pedestrian image is divided into the head, torso and leg region and three types of features Weighted Color Histograms(WHSV), Maximally Stable Color Region(MSCR) and Recurrent High-Structured Patches (RHSP) are extracted. Each of these features are extracted from the torso and leg region and optionally from the head region. The histograms feature is built with 12 bins channel per region, totalling to $12 \times 3 \times 3 = 108$ dimensions[2]. The MSCR feature of a blob is represented by 9 dimensional feature but these blobs per image is variable. Similarly, the feature length of RHSP features is variable as well. Similarity between two images is calculated as weighted sum of euclidean distance between their features. As the algorithm is unsupervised, it doesn't require any training and is also scalable to videos.

C. Unsupervised Saliency

Saliency is defined as "distinct features that 1) are *discriminative* in making a person standing out from their companions, and 2) are *reliable* in finding the same person across different views" [23]. Zhao et al. have developed a few variants of supervised and unsupervised methods using saliency [13], [23], [3] but we will mostly focus on Unsupervised Salience Matching [13]. Each image is densely divided into overlapping patches. For each patch, 32 bin LAB colour histograms are computed in three scales for three channels. So the colour feature is of length $32 \times 3 \times 3 = 288$. Similarly for SIFT features, each patch is further divided into 4×4 cells to obtain $4 \times 4 \times 8 = 128$ dimensional feature per channel. So total feature length for each patch is $288 + 128 \times 3 = 672$ dimensions. For an image, these DenseFeats features is represented as $X^{A,u} = \{x^{A,u}_{m,n} | m = 1..., M, n = 1..., N\}$ where (A, u) denotes the u^{th} image in camera A, (m, n) denotes the patch centred at the m^{th} row and the n^{th} column of the image. Total size of feature for an image is $M \times N \times 672$.

Once, the features are extracted for each patch, the key steps of the algorithm is briefly listed in Table II. Fig.4 illustrates the adjacency constrained search set of the patch in yellow box which is used in computing the Nearest Neighbour set. One of the two approaches is based nearest neighbour distances. A score is assigned for each patch using Eq. 5.

$$\mathbf{score}_{knn}(x^{A,u}_{m,n}) = D_k(X_{NN}(x^{A,u}_{m,n})) \tag{5}$$

where D_k denotes the distance of the k-th nearest neighbour. Similarity between two images is calculated using Eq.6

$$\mathbf{Sim}(\mathbf{x}^{A,u}, \mathbf{x}^{B,v}) = \sum_{m,n} \frac{\mathbf{score}_{knn}(x^{A,u}_{m,n}).s(x^{A,u}_{m,n}, x^{B,v}_{i,j}).\mathbf{score}_{knn}(x^{B,v}_{i,j})}{\alpha_{sdc} + |\mathbf{score}_{knn}(x^{A,u}_{m,n}) - \mathbf{score}_{knn}(x^{B,v}_{i,j})|} \tag{6}$$

IV. SIMULATION RESULTS

In the ideal scenario, the algorithms would be implemented on a real distributed system such as Android smartphone and results could be measured. However, the algorithms are initially written in MATLAB to simulate a distributed system scenario and the simulations were carried out on MATLAB running on a desktop PC. In future, we can experiment with implementing the algorithms on embedded device to check their performance.

[2]reduced to 72 if head region is not used

Fig. 4. Illustration of adjacency constrained search. Green region represents the adjacency constrained search set of the patch in yellow box. The patch in red box is the target match [23]

TABLE II. ALGORITHM FOR UNSUPERVISED HUMAN SALIENCY LEARNING

Algorithm for learning Unsupervised Human saliency
Input: image $X^{A,u}$ and a reference image set $\mathbb{R} = \{X^{B,v}, v = 1, ...N_r\}$
Output: saliency probability map $P(l^{A,u}_{m,n} = 1 \| x^{A,u}_{m,n})$
for each patch $x^{A,u}_{m,n}$ **do**
compute Nearest Neighbour (NN) set $X_{NN}(x^{A,u}_{m,n})$
compute $score_{knn}(\mathbf{x}^{A,u}_{m,n})$ based on NN distances,
end for

Experiments were carried out on a desktop PC with an Intel Xeon processor (X5650) with 12 cores and 24 gigabytes of RAM running Scientific Linux 6.5 unless specified. Some of the algorithms have parallel implementation as well but we have turned it off for these experiments for two reasons. 1) To make the comparisons fair, 2) Parallel MATLAB instances run within their own Java Virtual Machine (JVM) environments accounting for increased memory allocations. This caused some algorithms to fill the RAM to fill quickly and slowing down the execution.

For the experiments, the VIPeR dataset was randomly split into two sets of 316 image pairs each. One set was used for training and other for testing. We do this following the testing conventions in these papers [7], [4], [13].

A. Cost of sending data in wireless network

In the distributed case, the signature of a person extracted in one camera has to be transmitted to another via a communication channel as shown in Fig. 3. The implication of transferring data to a neighbour node has a cost in terms of energy and time, particularly in the case of wireless transmission. We conducted simple experiment to analyse how much energy and time is required in order to data to other nodes. We developed a simple application(app) for the Android platform which sends files of various sizes to the server using WiFi or mobile data (see Fig.5). The application was built using Google's Android Development Kit (ADK) and Android Studio. The experiments were conducted in a LG G2 smartphone. Time is measured using the system clock. Initial time is noted when data sending commences. The final time is noted after an acknowledgement is received from the server and the time taken

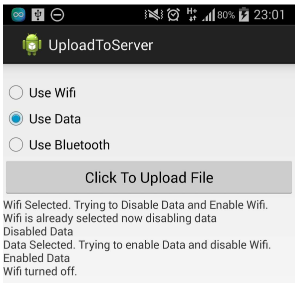

Fig. 5. Android application for calculating time and energy cost of transmitting data

is the difference of these two. Measuring energy consumed is however complicated than measuring time, because by default Android reports battery level in percentage only.It is too crude for our purpose and also as many processes are running simultaneously in background, it's hard to calculate the exact energy consumed for the communication. We used a third party application called Trepn profiler [24]. It is developed by Qualcomm for their Snapdragon processors and has access to hardware counters in the processor which are not available for public use. It isolates the energy used by an application, by collecting baseline energy consumption before starting the test application. Similar to the counter for measuring time, we flag the start and the end of the communication event to the Trepn application using Android Intent. Trepn then logs the energy consumption for each event.

As expected, the evaluations show in Fig.6 that the cost rises as the size of data goes up. WiFi has generally lower energy consumption than the phone networks. The difference becomes notable as the size of data goes up. Surprisingly, the speed of 4G was even faster than the WiFi albeit at higher energy cost. The test were done in Edinburgh with the WiFi provided by router connected to the Virgin Network and 4G by Everything Everywhere (EE) Network. But we didn't take into account many factors such as the load on the network, Signal strength etc.

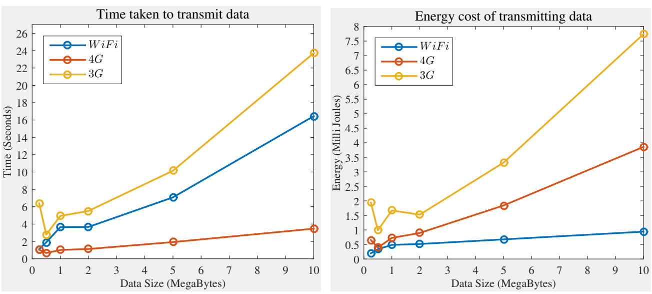

Fig. 6. Time and energy required to send data across the network

B. Runtime and Feature Length

1) KISSME: Among all the methods, KISSME was the fastest to train and learn the metric and it performed well too. The length of the feature before and after dimensionality reduction was determined from the source code and feature dataset provided. However, to calculate the time taken for feature extraction, we wrote the code as per their paper [4]. We divide the image into overlapping blocks of size 8×16 and stride of 8×8 to get 105 patches. We took histograms of 24 bins per channel and uniform LBP of 59 bins. So in total, the feature size is $105 \times 3 \times 2 \times 24 + 105 \times 59 = 21315$ dimensions. The histogram extraction of HSV and LAB and LBP features

TABLE III. SDALF Execution time

Step	Time(sec)
Division into 3 parts	162.15
MSCR Extraction	138.21
WHSV Extraction	123.17
RHSP Extraction	4824.6
MSCR Matching	6095.3
WHSV Matching	214.74
RHSP Matching	423.00
Total	11981.17

TABLE IV. Feature length, runtime and Rank 1 results.

Algorithm	Feature Length(PCA)	Time(sec)	Rank 1
KISSME	22154(34)	260.05	18.03
SDALF	5359	11981.00	19.80
Unsupervised Saliency	201600	11737.90	27.22

took approximately 260 seconds, which is very high compared to its training time of around 0.05 seconds. But still, feature extraction per image would take about $260/1264 \approx 0.2$ seconds.After dimensionality reduction, the feature dimension is reduced to just 34 which is highly desirable.

2) SDALF: As discussed in section III, the feature length of SDALF is not fixed but dependent on the number of RHSP patches and MSCR regions found in the image. Table III shows the breakdown of average time spent per step for the VIPeR dataset. RHSP features took the longest to compute so we experimented with removing it. The result showed there was only marginal degradation of performance. It can be seen in Fig. 7. But as the test has been done only in one dataset, it may not be true for all.

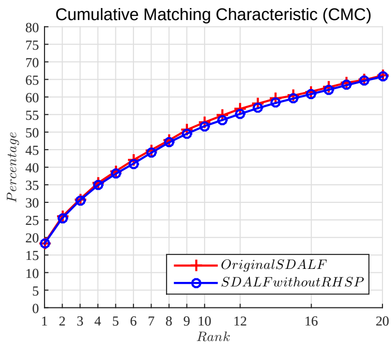

Fig. 7. Performance of SDALF with and without RHSP

3) Saliency: Saliency learning has the highest feature size per image. Each feature is of 201600 dimensions, if we suppose it is of MATLAB double precision, it's size is approximately 1.5 Megabytes which is not huge. However, each probe patch has it own adjacency search area for each image in the gallery set. If we assume 10 patches per row and constrained search area to be ± 2 rows, and there are 100 images in the gallery then. For each patch, we need to calculate the distance between itself and $10 \times 5 = 5000$ patches[3]. If there are 300 patches per image, it amounts to $5000 \times 300 = 1,500,000$ distances per image, which is more than 11 Megabytes in MATLAB double precision. In terms of running on embedded devices, memory is often a limited resource.

C. Cumulative Matching Characteristics (CMC) curves

Cumulative Matching Characteristics(CMC) [25] is widely used in person re-identification performance evaluation. It treats person re-identification as a ranking problem. Rank-1

[3] except for two top and two bottom rows

implies that the correct match has been found whereas Rank-k implies there were $k-1$ wrong classes ahead of the correct class. CMC(k) measures the probability that the correct match has a rank equal or higher than k [10]. TableIV shows Rank-1 score of various algorithms. It shows Saliency has better performance although it is computationally expensive and high data size. KISSME on the other hand looks the best to be implemented on distributed system as it is shown to be fast and computationally inexpensive as well.

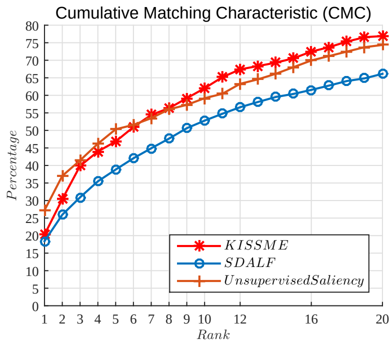

Fig. 8. Performance of the algorithms in VIPeR dataset

V. Conclusion

In this paper, we explored the possibilities of implementing person re-identification algorithms on distributed systems. We studied KISSME, SDALF and Unsupervised Saliency matching in terms of their runtime, size of descriptor, along with their person re-identification performance. We also looked at time and energy cost of communicating with neighbouring systems using various wireless technologies. Unsupervised Saliency has better Rank-1 result but it is computationally the most expensive and the memory requirement is also the highest. Even though we did not mention the energy cost for computing on the distributed platform, this would also consume high amount of energy. SDALF on the other hand has smallest signature before dimensionality reduction and potentially could be made even smaller by removing RHSP features. In theory at least, SDALF and Saliency features may be reduced using dimensionality reduction as well. But based on our experiments, without any modifications, KISSME is the best algorithm for a distributed system owing to its low complexity and shortest signature length. The only drawback is that it has to be trained and the large covariance matrices has to be computed and communicated to the neighbours.

This paper explored only the consequences of using distributed systems for person re-identification systems where communication between the sensor nodes is a requirement. But in some cases there might be a question between communicating or processing on its own. Even with communicating between nodes, there is a question of which node to communicate to when multiple nodes are available. In future, we are interested in answering these questions.

VI. Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) [EP/K014277/1] and the MOD University Defence Research Centre (UDRC) on Signal Processing

References

[1] S. Gong, M. Cristani, S. Yan, and C. C. Loy, *Person Re-Identification, Advance in Computer Vision and Pattern Recognition*. 2014.

[2] D. Gray, S. Brennan, and H. Tao, "Evaluating appearance models for recognition, reacquisition, and tracking," *Perform. Eval. Track. Surveill. (PETS), 10th Int. Work.*, vol. 3, pp. 41–47, 2007.

[3] R. Zhao, W. Ouyang, and X. Wang, "Learning Mid-level Filters for Person Re-identification," *2014 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 144–151, June 2014.

[4] M. Kostinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, no. Ldml, pp. 2288–2295, 2012.

[5] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith, "Learning Locally-Adaptive Decision Functions for Person Verification," *2013 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3610–3617, June 2013.

[6] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," *Icml*, pp. 209–216, 2007.

[7] M. Farenzena, L. Bazzani, a. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2360–2367, 2010.

[8] H. Ma, C. Zeng, and C. X. Ling, "A Reliable People Counting System via Multiple Cameras," *ACM Trans. Intell. Syst. Technol.*, vol. 3, pp. 1–22, Feb. 2012.

[9] T. B. T. Sebastian, P. H. Tu, J. Rittscher, R. Hartley, and N. Gheissari, "Person Reidentification Using Spatiotemporal Appearance," in *2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Vol. 2*, vol. 2, pp. 1528–1535, IEEE, 2006.

[10] X. Wang and R. Zhao, "Person Re-Identification: System Design and Evaluation Overview," in *Pers. Re-Identification* (S. Gong, M. Cristani, S. Yan, and C. C. Loy, eds.), London: Springer London, 2014.

[11] M. H. M. Saad, H. B. Zaman, M. A. Saghafi, and A. Hussain, "Review of person re-identification techniques," *IET Comput. Vis.*, pp. 1–20, Feb. 2014.

[12] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image Vis. Comput.*, vol. 32, pp. 270–286, 2014.

[13] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised Salience Learning for Person Re-identification," *2013 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3586–3593, June 2013.

[14] T. Ojala, M. Pietikäinen, T. Maenpaa, T. Mäenpää, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 24, no. 7, pp. 971–987, 2002.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, 2004.

[16] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, pp. 886–893, 2005.

[17] B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang, "Person Re-Identification by Support Vector Ranking," *Procedings Br. Mach. Vis. Conf. 2010*, vol. 1, pp. 21.1–21.11, 2010.

[18] K. Kumar and Y.-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?," *Computer (Long. Beach. Calif).*, vol. 43, no. 4, pp. 51–56, 2010.

[19] L. B. Dong Seon Cheng, Marco Cristani, Michele Stoppa and V. Murino, "Custom Pictorial Structures for Re-identification," *Proc. Br. Mach. Vis. Conf.*, pp. 68.1—-68.11, 2011.

[20] W. Li and X. Wang, "Locally aligned feature transforms across views," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3594–3601, 2013.

[21] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," *2009 IEEE 12th Int. Conf. Comput. Vis.*, pp. 498–505, 2009.

[22] P. M. Roth, M. Hirzer, M. Köstinger, C. Beleznai, and H. Bischof, "Mahalanobis Distance Learning for Person Re-identification," in *Pers. Re-Identification*, pp. 247–267, Springer London, 2014.

[23] R. Zhao, W. Ouyang, and X. Wang, "Person Re-identification by Salience Matching," *2013 IEEE Int. Conf. Comput. Vis.*, pp. 2528–2535, 2013.

[24] Q. Technologies, "Trepn Profiler Starter Edition User Guide." 2014.

[25] J. R. Beveridge, D. S. Bolme, B. A. Draper, and M. L. Teixeira, "The CSU face identification evaluation system - Its purpose, features, and structure," *Mach. Vis. Appl.*, vol. 16, no. April 2003, pp. 128–138, 2005.