Enhancing Person Re-identification by Late Fusion of Low-, Mid-, and High-Level Features

Lejbølle, Aske Rasch; Nasrollahi, Kamal; Moeslund, Thomas B.

Published in:
IET Biometrics

DOI (link to publication from Publisher):
10.1049/iet-bmt.2016.0200

Publication date:
2018

Document Version
Accepted author manuscript, peer reviewed version

Enhancing Person Re-identification by Late Fusion of Low-, Mid-, and High-Level Features

Aske R. Lejblle[1,*], Kamal Nasrollahi[1], Thomas B. Moeslund[1]

[1]Visual Analysis of People (VAP), Aalborg University, Rendsburggade 14, 9000 Aalborg, Denmark
[*]E-mail: asrl@create.aau.dk

Abstract: Person re-identification is the process of finding people across different cameras. In this process, the focus often lies in developing strong feature descriptors or a robust metric learning algorithm. While the two aspects are the most important steps in order to secure a high performance, a less explored aspect is late fusion of complementary features. For this purpose, this paper proposes a late fusing scheme that, based on an experimental analysis, combines three systems that focus on extracting features and provide supervised learning on different abstraction levels. To analyse the behaviour of the proposed system, both ranking aggregation and score-level fusion are applied. Our proposed fusion scheme increased results on both small and large datasets. Experimental results on VIPeR show accuracies 5.43% higher than related systems, while results on PRID450S and CUHK01 increase state-of-the-art results by 10.94% and 14.84%, respectively. Furthermore, a cross dataset test show an increased rank-1 accuracy of 28.26% when training on CUHK02 and testing on VIPeR. Finally, an analysis of the late fusion shows aggregation to be better when individual results are unequally distributed within top-10 while score-level fusion provides better results when two individual results lie within top-5 while the last lies outside top-10.

1. Introduction

Person re-identification is of great importance in biometrics and surveillance systems [1, 2, 3]. It is defined as the task of comparing images of persons captured by cameras with different views for the purpose of finding matches. Given an image (probe) from camera *A*, it is compared against all other images (gallery) captured by camera *B*. The results of the comparison are ranked according to an employed similarity measure to find the most similar images in the gallery for the given probe. Since the viewpoint of each camera is different along with the environment in which they are placed, problems like, viewpoint variations, lighting, scale, pose, and occlusion make re-identification a very challenging research topic (see Figure 1).

When dealing with re-identification, two steps are very important in order to reach good performance: 1) feature extraction and representation and 2) metric learning. Regarding the first step, feature extraction and representation, the employed feature descriptors should not only be discriminative but also need to be fast to extract as they are compared against a potentially large database [4, 5]. Features can either be extracted globally by looking at larger areas such as body parts [6] or locally by sampling minor local patches [7]. Furthermore, features can be extracted on different abstraction levels by dividing into low-, mid- and high-level depending on representation of the features. Low-level features are per-pixel based and typically cover colour and texture histograms.

(a) (b) (c) (d)

Fig. 1. *Examples of image pairs from different datasets, including; (a) VIPeR, (b) PRID450S, (c) CUHK01 and (d) CUHK03.*

Examples of more advanced low-level features include creating a covariance matrix from image derivatives [8] or looking at local key points as in SIFT [9]. Mid- and high-level features are both learned from low-level features and are defined by representing either parts of objects in the image (mid) or the entire object in the image (high) by sparse feature representations. Examples of mid-level features are Bag of Words (BoW) models [10] used to quantize low-level features to visual words while the late layers in a Convolution Neural Network (CNN) is an example of a high-level feature representation [11].

Regarding the second step, metric learning, supervised learning is often utilized in which a selection of image pairs from a number of persons are used to compute a distance matrix used when calculating the similarities [12] or compute a projection matrix used to project features to a common subspace [13]. Common for all methods is emphasizing on keeping features from dissimilar pairs apart while keeping features from similar pairs as close as possible.

In order to enhance the performance of a re-identification system, fusion can be applied on different levels. In the multi-shot case i.e., when several images of each identity are available, data-level fusion can be applied by combining information from all images of the same identity. To have a more robust representation of each identity, fusion on feature level can be applied instead. Feature fusion is typically applied by fusing different types of colour and texture features, often by simple concatenation. Fusing of different feature types makes sense if the features do not result in redundant information and complement each other. Dimensionality reduction techniques, such as PCA, are therefore often used to reduce such fused features. Finally, fusion can be applied on late level [14] as well. In this case, the results of different re-identification problems are fused, usually using Bayesian decision theory to combine the outputs by either summing or multiplying the scores [15]. In such a situation, a weight can be assigned to each of the scores depending on the correctness computed from training data. Another way is to combine the ranked lists by aggregation, either by looking at the mean position of each ranked identity or using order statistics [16].

In this paper, we analyse the complementarity of features extracted at different abstraction levels by applying late fusion on different combinations of outputs from three re-identification systems,

each working at different abstraction levels. This ends up in a system that advances state-of-the-art re-identification results on public benchmark datasets. For a more extensive analysis, we apply two different well known late fusion techniques, score-level fusion based on Bayesian theory [15] and rank aggregation [16]. We analyse the scenarios in which late fusion improves the results in order to define situations in which either of the late fusion techniques provides better results. This include an analysis of how different information captured by different feature types affect the late fused result. Finally, we measure processing time of late fusion and compare it with the increased accuracy. The contributions of this paper are therefore as follows:

1. We show that fusion of low-, mid- and high-level features is of potential when late fusion is applied.
2. We analyse how different feature types positively affect late fusion.
3. We conclude the cases in which it is better to apply score-level fusion than rank aggregation and vice versa.
4. We show that late fusion does not add particular processing time compared to increased accuracy.

The rest of this paper is organized as follows: first, the related work within re-identification is reviewed in Section 2. Then, the proposed system, including the fusion techniques and the three chosen re-identification features which are used in the fusion are explained in Section 3. In Section 4, experimental results on different public benchmark datasets along with an analysis and measured processing times are reported. Finally, the paper is concluded in Section 5.

2. Related Work

Low-level features In current re-identification systems, different low-level features are typically fused in order to take both colour and texture information into account. In order to make the features more discriminative, features are extracted more locally as those by Zhao *et al.* [9], that are obtained by sampling of local patches from which SIFT descriptors and colour histograms are extracted. The patches are then used to learn a set of SVMs by clustering similar patches. Local patches are also used in [17], in which a Gaussian function is used to calculate a similarity score along with KNN to find the most similar reference patches for each test patch in an unsupervised manner. Finally, Liao *et al.* [18] extract colour and texture histograms from overlapping patches and use a metric learning algorithm based on Mahalanobis distance to learn a projection matrix used to keep distance between similar image pairs closer. An example of low-level features that are extracted more globally is given in [19], in which the body is horizontally split into six equally sized stripes and colour name features are extracted and used with KISSME [12] metric learning. Similar regions are used in [6, 20, 21] where different colour and texture based features are extracted and used for cross-view metric learning.

Mid-level features Few systems utilize low-level features to learn mid-level features. An example of this is the BoW model learned in [22], which is used to extract mid-level features that are used together with a cosine similarity in an unsupervised manner. Another example of mid-level features is given in [23], where dictionaries based on HSV colour and LBP texture histograms are learned in order to represent features as *atoms* that are contained in the dictionary.

High-level features As the popularity of CNN increases, they have also been proposed for person re-identification as they are able to learn high-level feature representations by training on

images without the need for hand-crafted low- or mid-level features. Usually, Siamese networks are constructed, taking an image pair as input and outputting whether they match or not [24, 4, 25]. Another idea has been proposed in [26] where a pre-trained CNN extracts high-level features used along with hand-crafted low-level features. Finally, [27] extracts hand-crafted low-level features that are included in training a CNN in order to make a more robust image representation. High-level features are then extracted using the trained CNN and applied to a metric learning algorithm.

Late fusion In the context of re-identification, only a few systems propose late fusion of results in order to increase the overall performance. In [28], the product rule from Bayesian probability theory is used to fuse scores computed as the dot product between two feature vectors. Each feature is furthermore assigned a weight which is calculated from the area under the feature's score curve. Score-level fusion is also applied in [29] where the outputs of different metric learning systems are fused. In this case, they are all trained using similar low-level features. A different way of combining scores was proposed in [30], in which linear combinations of scores are computed using a weight, which is learned through SVM. Rank aggregation was used in [31] where ranking lists, calculated using both locally and more globally extracted low-level features, are combined. Aggregation is also used in [32] where different ranking lists computed using both individual and concatenated low-level features are aggregated.

3. Proposed System

The block diagram of the proposed system is shown in Figure 2. First, features on different abstraction levels are extracted from a given probe and transformed using the trained metric learning algorithms. Next, similarities are calculated between the features extracted from the probe and the gallery features in each of the learned subspaces. The resulting outputs are then either fused using the scores or the ranks of the identities. The output is a new ranked list which is re-ordered. These steps are explained in the following sub-sections, along with the used features.

3.1. Low-level features

The first part of the fusing system emphasizes on low-level features extracted from local patches to make the features more discriminative. A benchmark is provided by Schwarz *et al.* which compares results from different systems tested on different datasets[1]. By a review of the systems that show the best results, the system proposed by Liao *et al.* [18] is chosen as it not only has decent results on most of the listed datasets with single-shot rank-1 accuracies of 40% and 52.20% on VIPeR and CUHK03, respectively, but also shows fast feature extraction and metric learning.

The system, as shown in Figure 3, works by preprocessing each image using the Retinex algorithm [33], in order to enhance colour information, especially in shadowed regions. Next, features of overlapping patches of size 10×10 are extracted, including a joint HSV histogram with a bin size of 8, along with Scale Invariant Local Ternary Patterns (SILTP) [34] to handle illumination changes. For patches located at the same horizontal level, histograms within each channel are compared and maximized to deal with viewpoint changes between camera views. Furthermore, the image is downsampled two times by applying a 2×2 average pooling kernel and a similar feature extracting procedure is carried out. Finally, a log transformation is applied to each histogram to suppress large bin values. Due to feature maximization, the method is called Local Maximal Occurrence Representation (LOMO).

[1] Available at http://www.ssig.dcc.ufmg.br/reid-results/

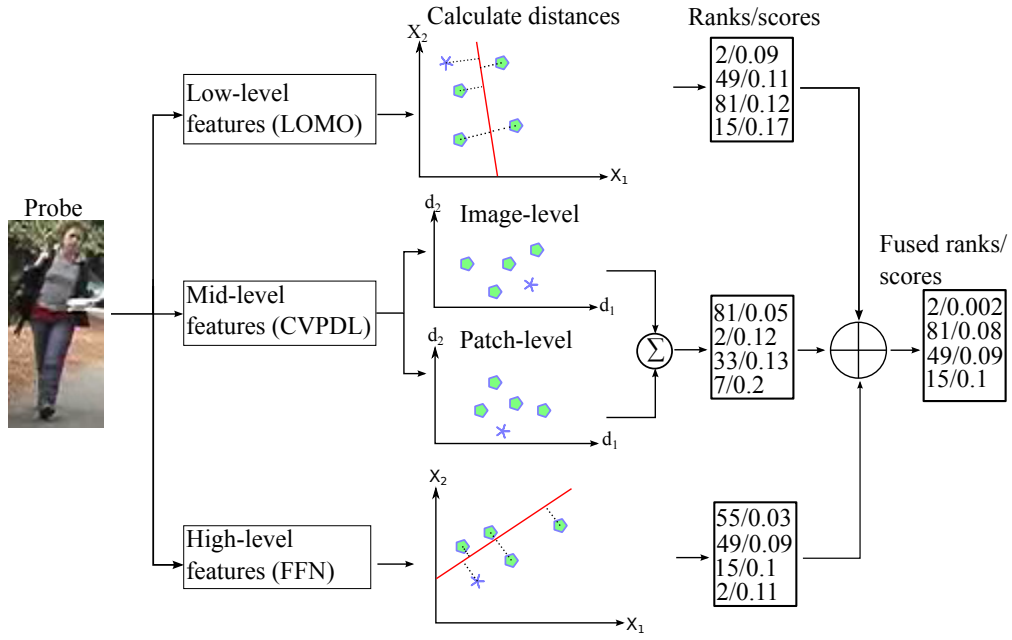

Fig. 2. *Overview of fusion system. Given a probe, low-, mid- and high-level features are first extracted. Then, extracted features (star) are transformed either to a subspace (LOMO and FFN) given by the solid red lines or changed to a different representation (CVPDL) and matched with the gallery (polygons). For simplicity, only features in two dimensions are shown. The outputs are then fused by a specified late fusion technique (symbolised by $\oplus$), producing a fused output.*

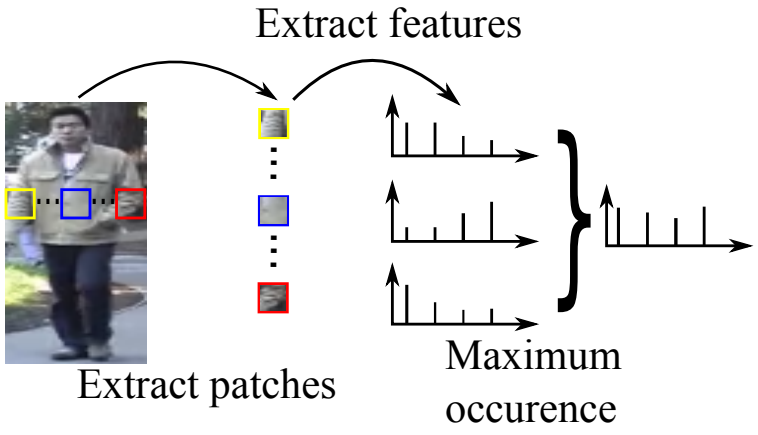

Fig. 3. *Used low-level features are from, LOcal Maximal Occurrence Representation (LOMO), that are composed of colour and texture features [18].*

As metric learning, following [18], an extended version of KISSME, shown in Equation 1, is used, which is originally based on Mahalanobis distance:

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

where the distance matrix $M = \Sigma_S^{-1} - \Sigma_D^{-1}$ with Σ_S calculated from similar image pairs and Σ_D from dissimilar pairs. These two matrices represent the intra-class and inter-class differences.

The metric learning is called Cross-view Quadratic Discriminant Analysis (XQDA) and considers a projection matrix, W, which maps features from two different views to a subspace before calculating the distance, as given by Equation 2:

$$d_M^2(x_i, x_j) = (x_i - x_j)^T W M' W^T (x_i - x_j), \tag{2}$$

where $M' = \Sigma_S'^{-1} - \Sigma_D'^{-1}$ and the two matrices, Σ_S' and Σ_D', are computed by Equation 3:

$$\Sigma_S' = W^T \Sigma_S W \quad \Sigma_D' = W^T \Sigma_D W \tag{3}$$

The matrix W is calculated by maximizing the ratio of the intra-class and inter-class variance. As this can be calculated from an eigenvalue decomposition of the two matrices in M, training time is only 0.5 seconds slower than KISSME.

3.2. Mid-level features

The second part of the fusing system utilizes mid-level features. From the few proposed systems that make use of sparse coding, the features developed by Li *et al.* [35] are chosen as the method utilizes dictionary learning from patches on both patch- and image-level. Furthermore, the system achieves decent results with rank-1 accuracies of 33.99% and 59.47% on VIPeR and CUHK01, respectively.

It extracts features in Lab colour space from overlapping patches of size 10×10 with a step size of 5. For each patch, 32-dimensional colour histograms and 128-dimensional SIFT descriptors are extracted in each channel. In addition, colour features are extracted from down sampled patches using scaling factors of 0.5 and 0.75, resulting in a total 672-dimensional feature vector for each patch. Finally, all features are L_2 normalized before the training set is used for dictionary learning. The mid-level test features are extracted by utilizing the learned dictionaries along with Orthogonal Matching Pursuit to transform from low-level features as shown in Figure 4. While features are transformed for each patch at patch-level, all patch features are concatenated and transformed on image-level.

Instead of solving the usual objection function when dealing with dictionary learning, defined by
$\min_{D,Z} \|X - DZ\|_F^2$ s.t. $\|d_i \leq 1\|_2$, where X is the feature matrix, D is the dictionary and Z is the coefficient matrix, Li *et al.* [35] proposes a projection matrix, P, in order to ease the NP-hard problem. Given features X, dictionaries D and projection matrices P in camera views *1* and *2*, the objection function defined in Equation 4 is solved as:

$$\begin{aligned} \min_{D_1, D_2, P_1, P_2} & ||X_1 - D_1 P_1 X_1||_F^2 + \\ & ||X_2 - D_2 P_2 X_2||_F^2 + \lambda f(D_1, D_2, P_1, P_2) \\ & \text{s.t. } ||d_{1,i} \leq 1||_2, \quad ||d_{2,i} \leq 1||_2, \end{aligned} \tag{4}$$

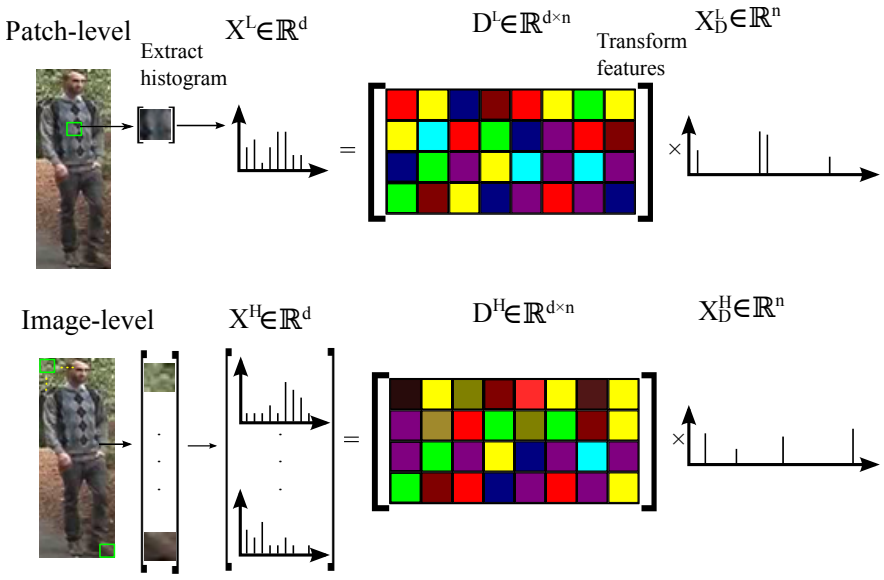

Fig. 4. *Mid-level features by Cross-View Projective Dictionary Learning (CVPDL) [35]. Each patch feature is transformed using a learned dictionary at patch-level while all patch features are concatenated and transformed on image-level.*

where $f(D_1, D_2, P_1, P_2)$ is a regularization function which affects the similarity between dictionary or projection matrices in the two views. As dictionary and projection matrices are learned on both patch- and image-level, the superscripts *L* and *H* are used to represent patch-level and image-level, respectively.

At patch-level, patches at the same spatial location are assumed to share the same dictionary, making the regularization function $\|D_1^L - D_2^L\|_F^2$ and by splitting each of the first two terms in Equation 4 by adding a relaxation variable, A, the dictionary and projection matrices are calculated by solving the objection function defined in Equation 5:

$$\begin{aligned} \min_{D_1^L, P_1^L, A_1^L} ||X_1 - D_1^L A_1^L||_F^2 + \beta ||P_1^L X_1 - A_1^L||_F^2 + \\ \lambda_1 ||D_1^L - D_2^L||_F^2 \\ \text{s.t. } ||d_{1,i}^L \leq 1||_2, \quad ||d_{2,i}^L \leq 1||_2, \end{aligned} \tag{5}$$

where β is a balance parameter. Similar objection function is used to calculate D_2^L, A_2^L and P_2^L.

At image-level, all patch features are concatenated to a single feature representation and features in the two views are instead assumed to share the same subspace i.e., projection matrix, resulting in a regularization function $\|P_1^H X_1 - P_2^H X_2\|$. By once again introducing the relaxation variable, the objection function on the image-level in one view is defined by Equation 6:

$$\begin{aligned} \min_{D_1^H, P_1^H, A_1^H} ||X_1 - D_1^H A_1^H||_F^2 + \alpha ||P_1^H X_1 - A_1^H||_F^2 + \\ \lambda_2 ||A_1^H - A_2^H||_F^2 \\ \text{s.t. } ||d_{1,i}^H \leq 1||_2, \quad ||d_{2,i}^H \leq 1||_2, \end{aligned} \tag{6}$$

As in the case of patch-level, similar objection function is defined to calculate D_2^H, P_2^H and A_2^H.

When matching at patch-level, each patch in view 1 is compared with every patch at the same horizontal level in view 2 due to misalignment. The shortest distance is then defined as the distance for that particular patch. Having calculated distances for all patches, the scores are accumulated to determine the score, $score_P$, between a probe in view 1 and a query in view 2. At image-level, the score between a probe and a query, $score_I$, is calculated as the cosine similarity.

Finally, the patch- and image-level scores are fused by $Score = score_P + \lambda score_I$, where λ is a pre-defined weight parameter between [0,1].

3.3. High-level features

The third and final part of the fusing system emphasizes on high-level features. As CNN's are both fast and have shown decent results, utilizing such a network is desired. Most of the already proposed CNN's produce a binary output to whether an image pair match or not. This is not suitable in late fusion as the probability often will be large for either being similar or dissimilar and the CNN will therefore overrule decisions made by other systems. Wu *et al.* [27] proposed a system which combines CNN and low-level hand-crafted features in a new architecture called Fused Feature Network (FFN). This way, a new type of feature is learned by both taking low-level colour and texture features along with more high-level CNN features into account. The system achieved rank-1 accuracies of 41.69%, 47.53% and 58.02% on VIPeR, CUHK01 and PRID450s, respectively. Furthermore, the new feature type was fused with LOMO features on feature level, increasing rank-1 accuracies to 51.06%, 55.51% and 66.62%, respectively.

The architecture, as shown in Figure 5, consists of two parts, the top being a CNN and the bottom being extraction of hand-crafted features.

The CNN architecture is similar to the ImageNet presented by Krizhevsky *et al.* [11] and consists of five convolution layers, all but the third followed by a MAX pooling and normalization layer. The network takes a randomly cropped image of size $227\times227\times3$ as input and outputs a 4096-dimensional CNN feature vector from the last pooling layer.

The other part horizontally divides the input image to 18 equally sized images and extracts colour histograms using colour spaces RGB, HSV, Lab, YCbCr and YIQ along with Gabor texture features. All histograms are represented in 16 bins and are L_1-normalized before concatenated to a 8064-dimensional feature vector.

Having two different feature vectors, they are each fed to a fully connected layer called **buffer** layer that output two 4096-dimensional feature vectors. The two outputs are then concatenated and fed to the new **fusion** layer which weights are learned based on the features types. Connecting the two types of features, Wu *et al.* [27] show that the weight update for the CNN is influenced by the output of the hand-crafted features. The output from the fusion layer is then defined as the new FFN feature type. In the training phase, a softmax layer is used to determine the corresponding label based on the input.

The FFN is trained using a fine-tuning scheme in which a pre-trained ImageNet model is utilized. The network is fine-tuned for 50,000 iterations using Stochastic Gradient Descent (SGD) with a mini-batch size of 25. After fine-tuning, FFN features are extracted from the fusion layer. As shown in the bottom of Figure 2, a projection matrix is then learned to map features to a subspace. In this case, Mirror Kernel Marginal Fisher Analysis (Mirror-KMFA) training scheme proposed by Chen *et al.* [36], with a chi-square kernel, is used as metric learning algorithm because of its high performance. After converting the features to kernel space, the algorithm aligns the feature distributions from features in two different views and computes mirror transformed features using

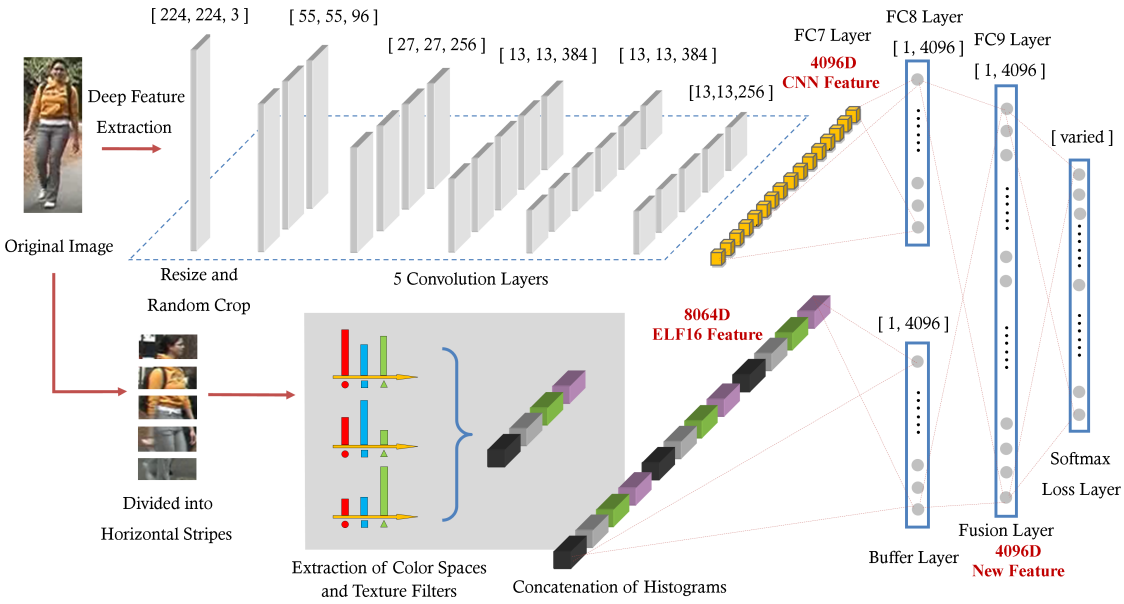

Fig. 5. *High-level features from Feature Fusion Network (FFN) [27]. Top part outputs CNN features from a pre-trained CNN while the bottom part outputs hand-crafted features. The second fully connected layer (FC9) outputs the new FFN feature. The figure is from [27].*

a projection matrix as defined in Equation 7:

$$X = \Lambda^{-\frac{1}{2}} U^T X^k_{aug}, \tag{7}$$

where X^k_{aug} is the kernalized and augmented feature matrix while Λ and U are matrices containing eigenvalues and corresponding eigenvectors, respectively, from an eigenvalue decomposition of a matrix $C = [K, -\beta K; -\beta K, K]$ with K being a matrix containing features from both views while β is a regularization term.

Next, MFA is utilized by calculating a projection matrix M, calculated by solving the generalized eigenvalue problem $S^w m_i = \lambda_i S^b m_i$ where S^w and S^b are intra- and inter-class scatter matrices, respectively. Finally, similarity between features is calculated in kernel space using Equation 8:

$$\mathcal{D}_{x_{1,i}, x_{2,j}} = x^2_{1,i} + x^2_{2,j} + 2x_{1,i}x_{2,j} \tag{8}$$

3.4. The proposed late fusion

Two different late fusion techniques are used in this paper, as they both have shown decent results when applied in other contexts.

Score-level fusion: The first late fusion method is based on the computed similarity scores, for each method. The algorithm was proposed by Zheng *et al.* [28] and is based on Bayesian theory of combining classifiers. In this case, the product rule is used since it has shown to be superior to other ways of combining outputs [15].

Having a number of computed similarity scores, $s^{(i)}_{p,q}$, where i denotes the method and p, q denotes a probe and a query image, the late fused output is calculated following Equation 9:

$$sim(p,q) = \prod_{i=1}^{K} (s_{p,q}^{(i)})^{w_q^{(i)}}, \qquad \sum_{i=1}^{K} w_q^{(i)} = 1, \tag{9}$$

wherein $w_q^{(i)}$ is the weight assigned for the i^{th} method and originally calculated using Equation 10:

$$w_q^{(i)} = \frac{\frac{1}{AUC_i}}{\sum_{k=1}^{K} \frac{1}{AUC_k}}, \tag{10}$$

where AUC_i is the area under the score curve (AUC) for the i^{th} method.

In the original context, higher scores are better and they therefore find an equal reference score, calculated from the training samples, in order to remove the tail of the score curve and decrease the AUC. This is done by calculating the euclidean distance between a predefined number of reference curves and the test curve, The k-nearest reference curves are then averaged and subtracted the test curve.

Although, in our case, a lower score value is better, leaving us with another way to deal with the reference curves and weight assignment, following same procedure by finding the most similar reference curves to the test score curves. An example of a score curve and the averaged nearest reference curves is shown in Figure 6 along with the resulting curve after subtraction.

As we desire to keep the AUC large by having small scores for the most similar pairs and large for the rest, we flip the reference curve before subtraction. This way, we keep the score for most similar pairs to a minimum while increasing the total AUC as shown in Figure 6 (b).

The weight assignment is, hence, also changed to be calculated following Equation 11:

$$w_q^{(i)} = \frac{AUC_i}{\sum_{k=1}^{K} AUC_k} \tag{11}$$

Lastly, in order to properly make use of this method, the outputs to be fused need to have a common metric to avoid any bias. Therefore, the scores are min-max normalized by Equation 12, before they are fused:

$$\hat{d}_M^2(x_i, x_j) = \frac{d_M^2(x_i - x_j) - \min d_M^2(\mathbf{x})}{\max d_M^2(\mathbf{x}) - \min d_M^2(\mathbf{x})} \tag{12}$$

Rank aggregation: Instead of looking at scores, rank aggregation makes use of the ranking lists for each method. The technique was successfully used by Ye *et al.* [31] who combines ranking lists from locally and globally based features using KNN to achieve a more correct list. Some of the most common ways of re-ranking is either by looking at the median or mean ranking or looking at the maximum ranking among all ranking lists.

In [37], different ranking aggregation techniques are compared, including Mean, Max, Stuart [38] and Robust Ranking Aggregation (RRA) [39]. It is shown that the Stuart method is slightly better than the others. The technique was originally used for aggregation of lists of genes which would contain noisy information and was therefore made robust to this challenge. It is therefore suited for person re-identification and is used in this context.

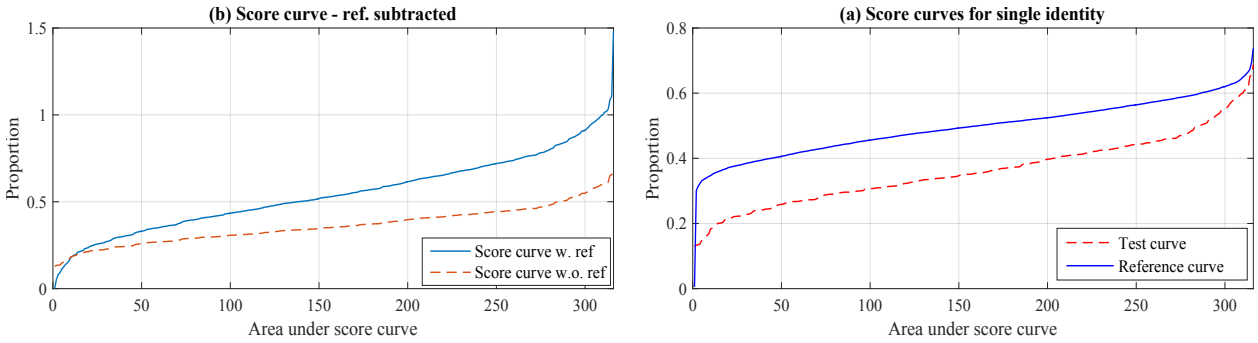

Fig. 6. *The impact of subtracting a reference curve: (a) shows the original test curve along with the average of the most similar reference curve. Flipping the curve and subtracting it from the test curve results in the difference shown in (b).*

The technique uses the statistically placement of the ranks, $r_{(i)} = \{r_{i,1}, ..., r_{i,m}, ..., r_{i,M}\}$ to calculate a new order of the ranks $r^{new}_{(i)}$ where $r_{i,m}$ is the rank for the i^{th} person at the m^{th} output, following Equation 13:

$$r^{new}_{(i)} = M! \cdot V_{M+1}, \tag{13}$$

with $V_{M+1} = \sum_{m=1}^{M} \sum_{l=1}^{m} (-1)^{l-1} \frac{V_{m-l}}{l!} r^{l}_{M-m+1}$ and $V_0 = 1$.

Using this formula, ranks are first normalized to the interval [0,1] where smaller values indicate a higher rank.

4. Experimental results

In this section we first give the details of the datasets and protocols that are used for evaluation of the proposed system. Then, the obtained results from the experimental analysis in Section 4.2 are explained and analysed. The best results are compared against state-of-the-art re-identification systems and finally, the processing time of the system is compared to the increase in accuracy. Throughout all tests, we refer to our results from rank aggregation and score-level fusion by the subscripts *agg* and *sco*, respectively.

4.1. Datasets and Protocol

Datasets: Tests are conducted on four different public datasets, two minor and two larger. Common for all is the challenges of change in viewpoint and illumination which make the task more difficult. Examples of image pairs in each dataset are shown in Figure 1. Further, a cross-dataset test is conducted for a more realistic performance evaluation. The first dataset is VIPeR [6] which consists of 1264 images of 632 identities captured in two different camera views, thus, each person has one image from each view. The second dataset, PRID450S [40] contains 900 images of 450 identities.

The two larger datasets, CUHK01 and CUHK03 [41, 4], contain multiple images of each person in each camera view. CUHK01 contains 3884 images of 971 different identities. For each identity, two images are captured in each camera view. CUHK03 contains 13,164 images of 1360 different identities. Images are captured using three camera pairs and images from one identity are captured by one camera pair. One to five images are captured from each identity in each camera view with an average of 4.8. Both manually labeled bounding boxes and automatically detected are included.

Evaluation protocols: In all experiments, single-shot setting is used. For VIPeR and PRID450S, identities are randomly divided in a training set of 316 and 225 identities, respectively, while the other half is used for testing. For CUHK01, 485 identities are randomly used for training while 486 are used for testing. For each identity, one image in each camera view is randomly chosen in each iteration. For CUHK03, the protocol defined in [4] is used, having 1160 identities for training and 100 for testing. As for CUHK01, one image in each camera view is randomly chosen in each iteration. For VIPeR, PRID450S and CUHK01, 10 iterations are run, each with randomly split data. For CUHK03, 20 iterations are run following the protocol. The tests on this dataset is made on the manually labelled bounding boxes. The mean accuracy over all iterations is calculated for each dataset and the results are presented by Cumulated Matching Characteristic (CMC) curves which show the accumulated ranked similarities for all identities, having the rank-1 accuracy indicating the number of probes that have their corresponding gallery image as the most similar.

4.2. The results of late fusion

Initial tests are conducted by evaluating different combination of features when late fusion is applied to conclude which combinations benefit mostly from late fusion. To take both minor and large datasets into account, tests are conducted on CUHK01 and VIPeR using the protocols defined in Section 4. Tables 1-2 summarize the results. All_{agg} and All_{sco} indicate late fusion of all three systems while the results from LOMO, FFN and CVPDL are reproduced results and therefore differs from original.

Table 1 Results on VIPeR (p=316). The best results are in bolt.

System/Rank	r = 1	r = 5	r = 10
All_{agg}	**45.63**	**75.06**	85.16
All_{sco}	45.24	75.02	**85.70**
FFN+LOMO_{agg}	43.86	73.89	84.34
FFN+LOMO_{sco}	43.73	73.81	84.08
FFN+CVPDL_{agg}	39.40	68.04	79.59
FFN+CVPDL_{sco}	37.85	67.56	78.96
LOMO+CVPDL_{agg}	37.22	69.84	81.61
LOMO+CVPDL_{sco}	39.94	71.06	83.01
LOMO	37.72	67.59	80.06
FFN	30.70	57.72	69.15
CVPDL	28.23	55.41	70.54

Table 2 Results on CUHK01 (p=486). The best results are in bolt.

System/Rank	r = 1	r = 5	r = 10
All_{agg}	**70.35**	**88.46**	**93.29**
All_{sco}	64.67	83.81	89.35
FFN+LOMO_{agg}	47.35	73.19	80.88
FFN+LOMO_{sco}	47.17	71.39	78.99
FFN+CVPDL_{agg}	65.86	86.85	93.02
FFN+CVPDL_{sco}	58.60	80.11	86.78
LOMO+CVPDL_{agg}	67.06	87.16	91.89
LOMO+CVPDL_{sco}	63.86	84.42	89.74
LOMO	41.77	66.19	74.86
FFN	32.28	56.95	66.73
CVPDL	53.44	78.85	86.95

In the case of VIPeR, the best pairwise combination is FFN+LOMO_{agg} with a rank-1 accuracy which is 3.89% higher than the next best result of LOMO+CVPDL_{sco}. Though, when rank aggregation is applied using all three systems, the rank-1 accuracy increase by 1.77% compared to FFN+LOMO_{agg}.

For CUHK01, the best pairwise combination is LOMO+CVPDL_{agg}, while late fusion of all systems achieves a rank-1 accuracy which in comparison is 3.29% higher when rank aggregation is utilized. From this it can be concluded that the best results are achieved when applying late fusion to all three systems while, generally, rank aggregation is shown to provide the best results. Though, in the case of combining FFN and LOMO, accuracies are similar for both type of late fusion.

In both cases it is clear that individual results are important for late fusion, although, there is an indication that mid- and high-level features complement each other better than other combinations. For VIPeR, FFN+CVPDL$_{agg}$ result in the highest increase in accuracy of 8.7% compared to individual results of FFN and CVPDL while showing an increase of 12.42% in the case of CUHK01. While LOMO+CVPDL$_{agg}$ show an increase of 13.62% in the case of CUHK01, there is a decrease of 0.50% in the case of VIPeR. Comparisons between the late fused system and individual results are furthermore visualized in Figure 7.

For VIPeR, PRID450S and CUHK03, LOMO provides the best individual results while CVPDL show the best accuracy on CUHK01 53.44%. While FFN and CVPDL show close to similar results on both VIPeR and PRID450S, CVPDL seem to be better at handling change in texture across camera views as it is superior to FFN in the cases of CUHK01 and CUHK03. When applying score-level fusion, rank-1 accuracies of 45.24%, 68.76%, 64.67 and 54.72% on VIPeR, PRID450S, CUHK01 and CUHK03, respectively, are achieved. Meanwhile, when rank aggregation is applied, rank- accuracies of 45.63%, 77.56%, 70.35% and 52.25% for VIPeR, PRID450S, CUHK01 and CUHK03, respectively, are achieved.

For VIPeR, PRID450S and CUHK01, the largest improvement is achieved by utilizing rank aggregation when compared to the individual results. In these cases, the rank-1 accuracies are increased by 7.91%, 17.83% and 16.91%, respectively. In the case of CUHK03, score-level fusion provides the best rank-1 accuracy that increases the results of LOMO by 11.51%.

Comparing score-level fusion and rank aggregation, the latter achieves a rank-1 accuracy which is 0.39% higher in the case of VIPeR. For PRID450S and CUHK01, differences are more significant with rank aggregation achieving rank-1 accuracies that are 8.8% and 5.68% higher than score-level fusion. Finally, score-level fusion achieves a accuracy 2.47% higher than rank aggregation in the case of CUHK03. This is most likely due to the much worse result by CVPDL, showing score-level fusion to be more robust to single bad performing features.

Overall, large increase in accuracies are achieved on both minor and larger datasets, showing the benefit of applying late fusion to features at different abstraction levels. In addition, the system of [29] increases the rank-1 accuracy by 5.82% compared to their individual results, while we show an increased accuracy of 7.91%.

4.3. The importance of late fusion

In order to analyse the affection of late fusion, three examples are provided, on how the fused result is improved or maintained by having different individual results. This includes an explanation of how different feature types contribute to the late fusion and how this affect the result. Examples are created by looking at the results from a single test iteration on VIPeR dataset.

The first example is shown in Figure 8 where all individual features rank the true match as the fourth most similar, while the remaining queries are ranked differently by each system. For FFN, the contours of the jeans and dark shirt seem to dominate the matched queries, while the results of LOMO and CVPDL show a higher dependency on the colors, especially seen by the impact of change in color for the jeans. Furthermore, the two latter also seem to be more affected textures created from shadows present in the probe image. Due to these differences, the aggregated output provides a better result. For the case of score-level fusion, the calculated distances for each system affect the fused result causing the score-fused to be similar.

In the second example shown in Figure 9, neither of the individual systems ranks the true match within top-5 and while rankings differ even more than in the first example. Here, FFN not only captures information on the contours, but also the colors, since hand-crafted color features were

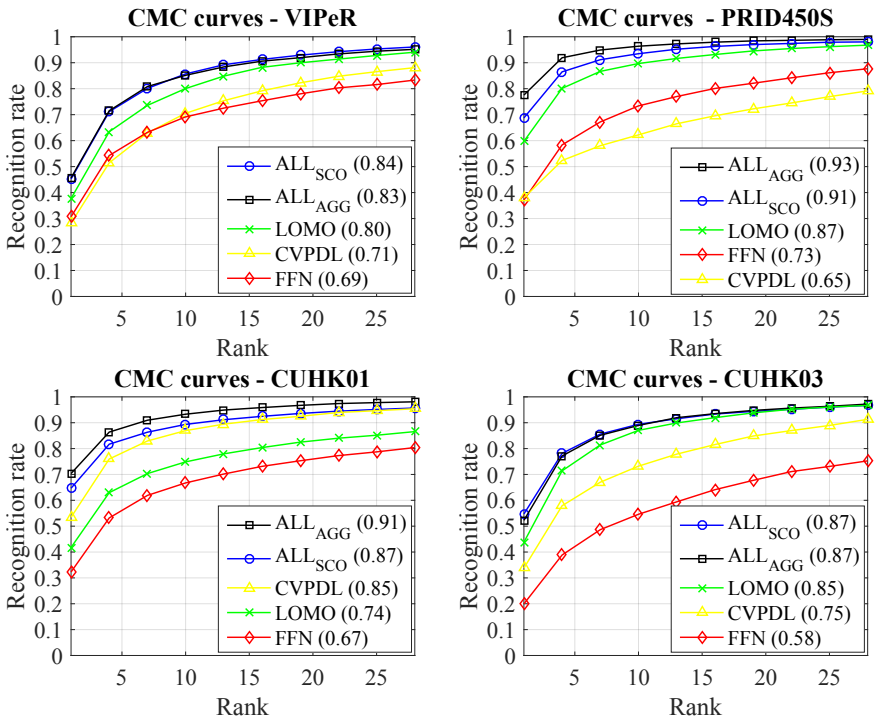

Fig. 7. *CMC curves of results using score-level fusion on (a) VIPeR (p=316); (b) PRID450S (p=225); (c) CUHK01 (p=486) and (d) CUHK03, manually labeled (p=100).*

also used to train the feature type. Again, LOMO and CVPDL seem to capture information from the street, such as shadows. But while CVPDL seem to capture more information on the lower part of the image, including the colors from jeans and texture from the street, LOMO emphasize more on the combination of colors such as the color of the left handbag and coat. After applying late fusion, the results are improved in both cases and the output of the rank aggregation now ranks the true match within top-4.

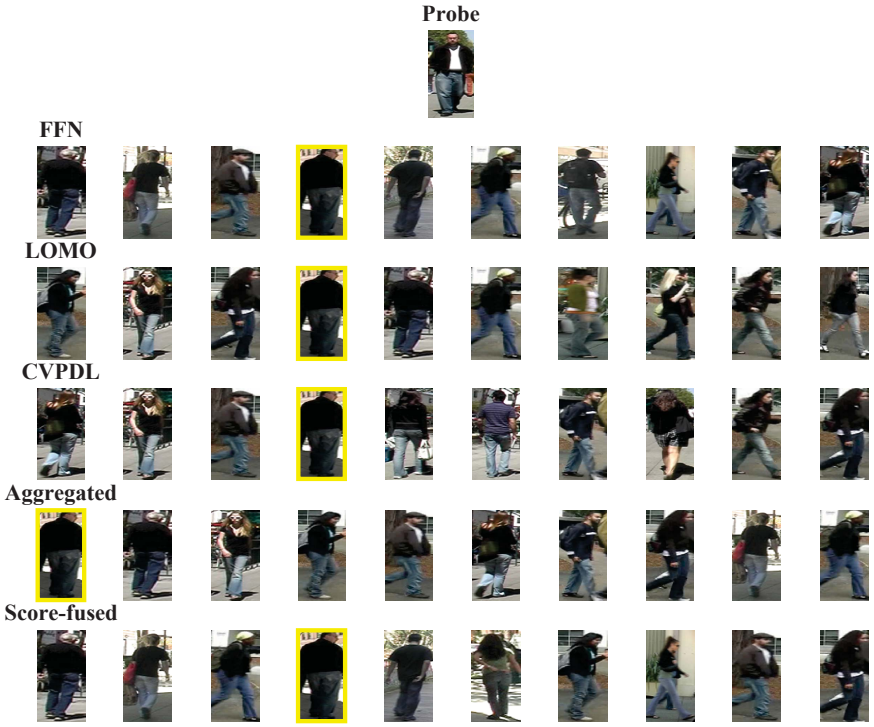

Fig. 8. *First case in which all individual features rank the true match similar. True match is shown by the green rectangle.*

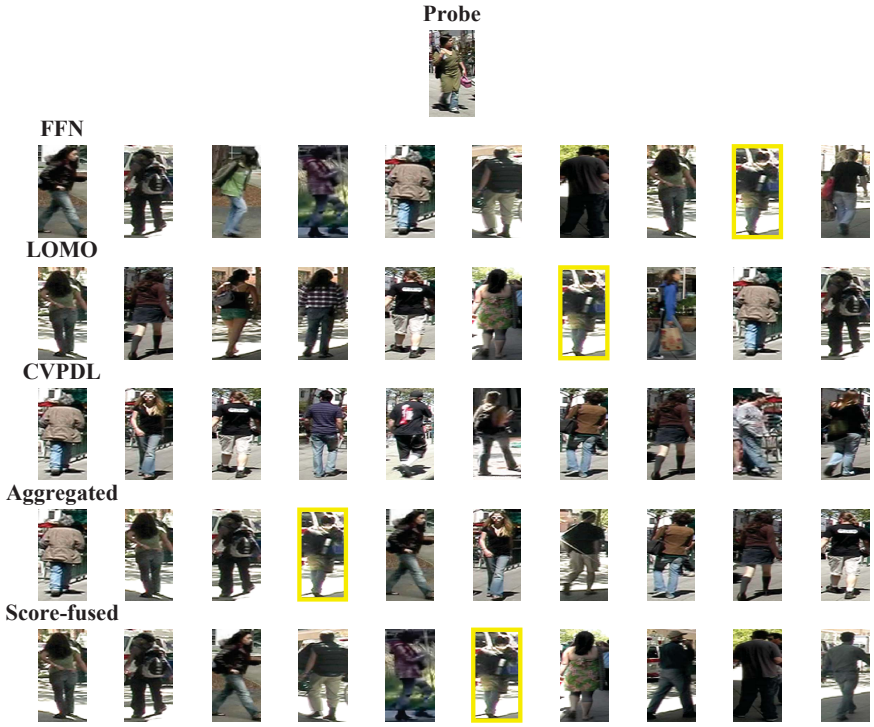

Fig. 9. *Second case in which the true match is ranked differently by each individual feature. True match is shown by the green rectangle.*

In the third and last example shown in Figure 10, LOMO and FFN rank the true match within top-5. While LOMO is affected by the gray-scale change of the trousers due to lighting changes, FFN emphasize more on the broad body contour, making it more robust to such challenges. Once again, CVPDL seem to capture the texture from shadows, while also capturing the white colors of the shirt. In this case, the rank aggregation is affected by the bad result from CVPDL and the aggregated result is therefore similar to that of LOMO. Meanwhile, score-level fusion is not similarly affected due to the weight assignment and use of distances and the result is therefore similar to that of FFN.

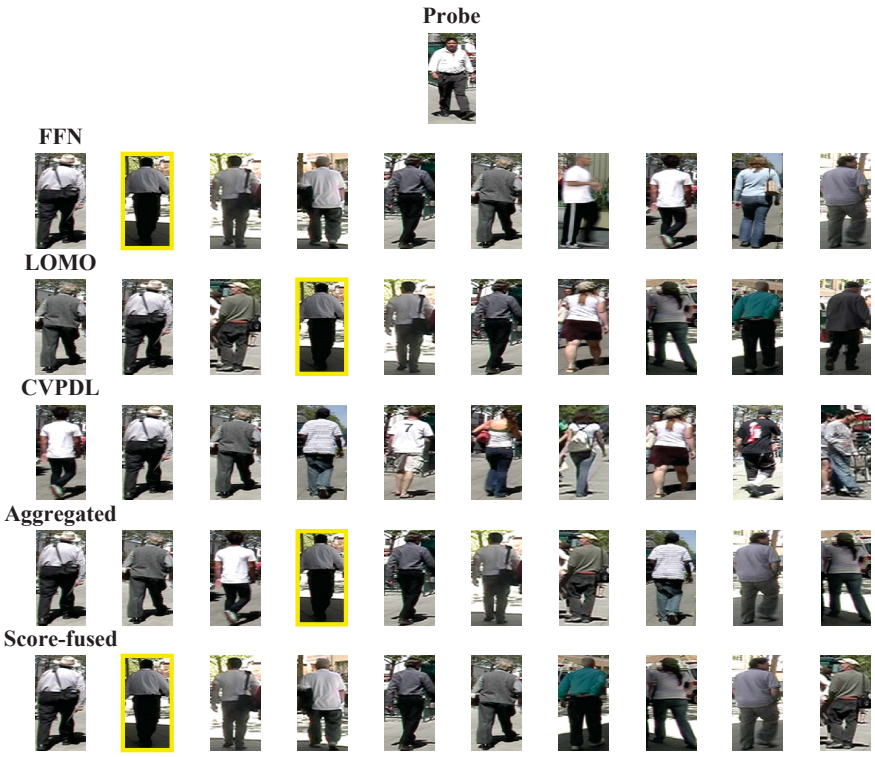

Fig. 10. *Third case in which the true match is ranked high by two features and outside top-10 by the last. True match is shown by the green rectangle.*

Generally, aggregation performs better than score-level fusion when all three individual features ranks the true match within top-10 while score-level fusion has better results in situations where one feature performs badly, though, this cannot be completely defined as the rankings of other identities affects the fused result. FFN takes the overall contours into account and is therefore not affected much by the background. Contrary, as color features on a global scale are also used for training the feature, it is affected by changes in color as seen in Figure 9. Meanwhile, LOMO performs better when similar colors are preserved in parts of the image, due to utilization of patches. Finally, CVPDL seem to suffer from situations with changing backgrounds because of to its matching scheme, especially due to changes in shadows, while performing better in situations where the color of the clothing is more uniform or distinct texture is visible in both camera views. Overall, information captured by each feature type positively affect late fusion as corresponding output lists differ. Few cases exist in which false identities are ranked high by all three systems as seen in Figure 10, where lighting changes, uniform colors and similarities in the shape of the body, cause a false identify to be ranked higher.

4.4. Comparison to state-of-the-art

The proposed system is compared with other state-of-the-art systems on all four datasets. Tables 3-6 contain the results for our system compared to the state-of-the-art. *Ours*$_{agg}$ and *Ours*$_{sco}$ indicate our system by fusion of both low-, mid-, and high-level features using rank aggregation and score-level fusion, respectively.

Table 3 Comparison between our system and state-of-the-art systems on the VIPeR dataset (p=316).

Method/Rank	r = 1	r = 5	r = 10
Ours$_{agg}$	45.63	75.06	85.16
Ours$_{sco}$	45.24	75.02	85.70
FFN+LOMO [27]	**51.06**	**81.01**	**91.39**
LOMO+XQDA [18]	40.00	67.40	80.51
Mirror-KMFA [36]	42.97	75.82	87.28
MuRE [29]	42.72	–	88.04
SCNCD [19]	37.80	68.50	81.20
Deep Re-id [24]	34.81	63.72	76.24
KISSME [12]	24.75	53.48	67.44
ECM [30]	38.90	67.80	78.40
MLF+LADF [9]	43.39	73.04	87.28

Table 4 Comparison between our system and state-of-the-art systems on the PRID450S dataset (p=225).

Method/Rank	r = 1	r = 5	r = 10
Ours$_{agg}$	**77.56**	**93.47**	**96.09**
Ours$_{sco}$	68.76	88.49	93.47
FFN+LOMO [27]	66.62	86.84	92.84
Mirror-KMFA [36]	55.42	63.72	87.72
KISSME [12]	36.31	65.11	75.42
ECM [30]	41.90	66.30	76.90
SCNCD [19]	41.60	68.90	79.40

Table 5 Comparison between our system and state-of-the-art systems on the CUHK01 dataset (p=486).

Method/Rank	r = 1	r = 5	r = 10
Ours$_{agg}$	**70.35**	**88.46**	**93.29**
Ours$_{sco}$	64.67	83.81	89.35
FFN+LOMO [27]	55.51	78.40	83.68
Deep Re-id [24]	47.53	72.10	80.53
Mirror-KMFA [36]	40.40	64.63	75.34
MLF [9]	34.30	55.12	64.91

Table 6 Comparison between our system and state-of-the-art systems on the CUHK03 dataset (p=100).

Method/Rank	r = 1	r = 5	r = 10
Ours$_{agg}$	52.25	80.40	88.95
Ours$_{sco}$	54.72	81.25	89.40
LOMO+XQDA [18]	52.20	82.23	**92.14**
Deep Re-id [24]	**54.74**	**86.42**	91.50
FPNN [4]	20.65	51.32	68.74

It is clearly shown that our introduced fusion results in a system that outperforms previous systems. For VIPeR, the feature fused system of LOMO and FFN achieves a rank-1 accuracy 5.43% better than the proposed system indicating that FFN and CVPDL might share the same difficulties when classifying certain identities which is also indicated by their performance being very similar. This causes fusing of the two features to not improve accuracy by much. Compared to the related system of [29], our system achieves a rank-1 accuracy which is 2.91% higher, showing the importance of training on features at different abstraction levels.

For PRID450S and CUHK01 our system clearly beats the feature fused systems by a rank-1 increase of 10.94% and 14.84%, respectively.

For CUHK03, our results outperforms the rank-1 accuracy of [18] with 2.52% while having almost similar accuracy compared to the state-of-the-art CNN of [24]. Looking at Figure 7 (d), this is probably due to the performance of FFN with a rank-1 accuracy of 20%. The CNN by Ahmed *et al.* was trained on CUHK03 while FFN was trained on the Market-1501 dataset [5] which, along with the architecture, might be the reason for the almost similar performance.

4.5. Cross-dataset test

In a real world scenario, it is desired to have a system that adapts well to new data. Furthermore, it is undesired to label new training data in each new application. As a result, a decent accuracy is desired, independently of which dataset is used for training. To test this scenario, training and testing are performed on different datasets.

For training, the extended CUHK01, CUHK02 [41], is utilized. This dataset consists of 1816 different identities, each with two images in two different views, bringing the total number of images to 7264. In the training phase, all identities are included using one image from each view. In the test, VIPeR is used, using the same identities in each iteration as in the intra-dataset test. Similar to the previous tests, 10 iterations are run and the accuracies are the averaged. The resulting CMC curves are shown in Figure 11. As in the test on CUHK03, score-level fusion provides the highest accuracies, having a rank-1 accuracy 5.95% higher than rank aggregation. Compared to the individual results, dominated by LOMO, the rank-1 accuracy is increase by 10.95%, once again showing the benefit of late fusion.

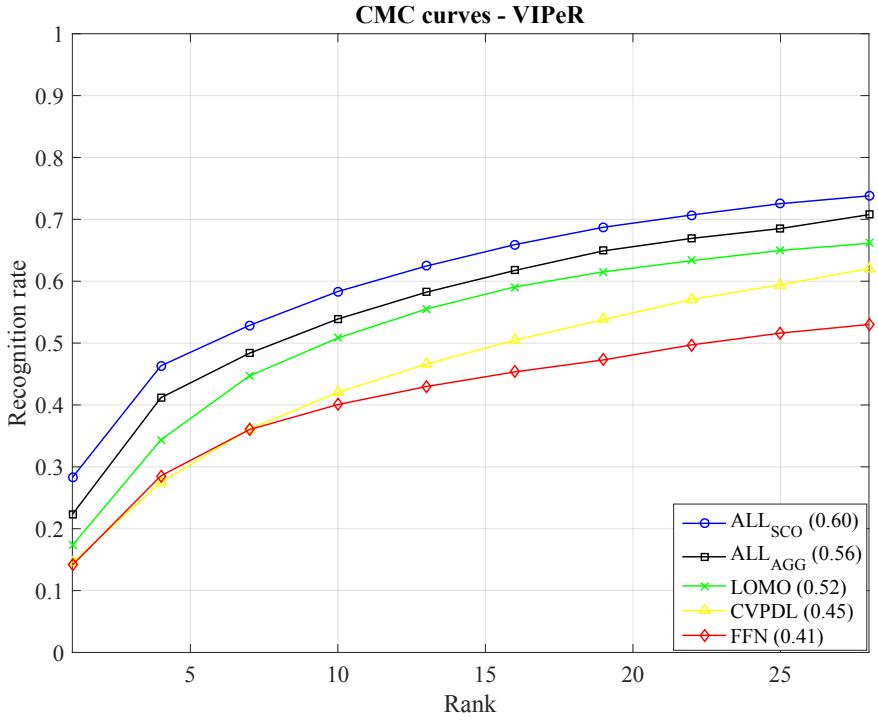

Fig. 11. *CMC curves of results using cross-dataset settings on VIPeR (p=316).*

Table 7 shows our results compared to previous systems tested under similar settings. Both previous systems utilize CNN's to learn similarity between identities. Our fusion system increase the rank-1 accuracy by 5.85% compared to the previous best result, despite the fact that the two other systems utilize multiple images from each identity in the training phase compared to only one in our case.

4.6. Processing time

In order to analyse whether it makes sense to make use of the introduced late fusion, the processing time when matching for each individual feature along with the processing time for the fusion

Table 7 Comparing to other state-of-the-art results when training on CUHK02 and testing on VIPeR (p=316). "–" indicates non available results. Best results are in bold.

System/Rank	r = 1	r = 5	r = 10	r = 20
Ours$_{agg}$	22.31	43.89	53.89	65.66
Ours$_{sco}$	**28.26**	**49.10**	**58.34**	69.35
DeepRank[42]	22.41	–	56.39	**72.72**
DML[25]	16.17	–	45.82	57.56

techniques are examined and compared to the increase in the accuracy.

A test is made by averaging the timings of 10 iterations on the VIPeR dataset using an Intel i7-4700MQ CPU @ 2.4GHz and the results are shown in Table 8.

Table 8 Average timings for matching and late fusion in seconds over 10 iterations on VIPeR dataset.

FFN	LOMO	CVPDL	Aggregation	Score-level
10.30	0.18	292.38	6.26	0.12

As shown in Table 8, matching for CVPDL takes up 96.5% of the total processing time if score-level fusion is employed. In reality, this might not be suitable if the system is running real-time and a different way of matching patches should be developed if the algorithm should be kept in the fusing system.

Furthermore, rank aggregation and score-level fusion only takes up 2.1% and 0.04%, respectively, of the total processing time.

5. Conclusion

Throughout this paper, we have proposed a novel method to combine three state-of-the-art features developed for re-identification through late fusion. In order to get the most proper results, the features are extracted at different abstraction levels, namely low-, mid- and high-level. Two types of late fusion techniques are utilized, score-level fusion and rank aggregation, focusing on different ways to fuse outputs. The score-level fusion is re-defined to fit the scores we calculate.

Experimental results on four different datasets showed a clear improvement in rank-1 accuracies on two with rank-1 accuracies of 10.94% and 14.84% for PRID450S and CUHK01, respectively, compared to previous systems and an increase on VIPeR of 5.43% compared to related systems. The results on CHUK03 are almost similar in performance compared to the state-of-the-art CNN of [24] with a potential to be increased by training or fine-tuning on similar dataset. Overall, rank aggregation provided the best results being up to 8.8% better than score-level fusion, though, with the latter being faster in processing time. Further, an analysis indicated that rank aggregation performed better when individual results are within a certain range of one another while score-level fusion is better when one result is much worse than the other two. In addition, the analysis showed that FFN mostly captures the overall contour of the body, LOMO mostly the combination of colors while CVPDL mostly the texture. The different focus points cause different ordering of matches which positively affect late fusion when the true match is ranked high. When looking at the processing time, late fusion only takes up at most 2.1% of the total processing time, making late fusion beneficial when compared to the increased accuracies. Finally, patch matching showed a large increase in processing time, leaving it to be modified if kept in the fusing system.

6. References

[1] Z. Liu, Z. Zhang, Q. Wu, Y. Wang. "Enhancing person re-identification by integrating gait biometric", *Neurocomputing*, **168**, pp. 1144 – 1156, (2015).

[2] B. DeCann, A. Ross. "Modelling errors in a biometric re-identification system", *IET Biometrics*, **4(4)**, pp. 209–219, (2015).

[3] A. Schumann, E. Monari. "A soft-biometrics dataset for person tracking and re-identification", *Proc. AVSS*, pp. 193–198, (2014).

[4] W. Li, R. Zhao, T. Xiao, X. Wang. "Deepreid: Deep filter pairing neural network for person re-identification", *Proc. CVPR*, pp. 152–159, (2014).

[5] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian. "Scalable person re-identification: A benchmark", *Proc. ICCV*, pp. 1116–1124, (2015).

[6] D. Gray, H. Tao. "Viewpoint invariant pedestrian recognition with an ensemble of localized features", *Proc. ECCV*, pp. 262–275, (2008).

[7] L. Bazzani, M. Cristani, V. Murino. "Symmetry-driven accumulation of local features for human characterization and re-identification", *Computer Vision and Image Understanding*, **117(2)**, pp. 130–144, (2013).

[8] S. Bak, F. Brémond. "Re-identification by covariance descriptors", S. Gong, M. Cristani, S. Yan, C. C. Loy (Eds.), *Person Re-Identification*, volume 1 of *Advances in Computer Vision and Pattern Recognition*, 1 edition, chapter 4, pp. 71–91, (Springer, London, 2014).

[9] R. Zhao, W. Ouyang, X. Wang. "Learning mid-level filters for person re-identification", *Proc. CVPR*, pp. 144–151, (2014).

[10] S. Lazebnik, C. Schmid, J. Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories", *Proc. CVPR*, volume 2, pp. 2169–2178, (2006).

[11] A. Krizhevsky, I. Sutskever, G. E. Hinton. "Imagenet classification with deep convolutional neural networks", *Proc. NIPS*, pp. 1097–1105, (2012).

[12] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, H. Bischof. "Large scale metric learning from equivalence constraints", *Proc. CVPR*, pp. 2288–2295, (2012).

[13] F. Xiong, M. Gou, O. Camps, M. Sznaier. "Person re-identification using kernel-based metric learning methods", *Proc. ECCV*, pp. 1–16, (2014).

[14] M. Pantic, L. J. Rothkrantz. "Toward an affect-sensitive multimodal human-computer interaction", *Proceedings of the IEEE*, **91(9)**, pp. 1370–1390, (2003).

[15] J. Kittler, M. Hatef, R. P. Duin, J. Matas. "On combining classifiers", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20(3)**, pp. 226–239, (1998).

[16] C. Dwork, R. Kumar, M. Naor, D. Sivakumar. "Rank aggregation methods for the web", *Proc. WWW*, pp. 613–622, (2001).

[17] R. Zhao, W. Ouyang, X. Wang. "Unsupervised salience learning for person re-identification", *Proc. CVPR*, pp. 3586–3593, (2013).

[18] S. Liao, Y. Hu, X. Zhu, S. Z. Li. "Person re-identification by local maximal occurrence representation and metric learning", *Proc. CVPR*, pp. 2197–2206, (2015).

[19] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, S. Z. Li. "Salient color names for person re-identification", *Proc. ECCV*, pp. 536–551, (2014).

[20] B. Prosser, W.-S. Zheng, S. Gong, T. Xiang, Q. Mary. "Person re-identification by support vector ranking", *Proc. BMVC*, pp. 21.1–21.11, (2010).

[21] W.-S. Zheng, S. Gong, T. Xiang. "Person re-identification by probabilistic relative distance comparison", *Proc. CVPR*, pp. 649–656, (2011).

[22] L. Zheng, L. Shen, L. Tian, S. Wang, J. Bu, Q. Tian. "Person re-identification meets image search", *CoRR*, **abs/1502.02171**, pp. 4321–4330, (2015), [Online]. Available: `http://arxiv.org/abs/1502.02171`, retrieved from: http://arxiv.org/abs/1502.02171.

[23] X. Liu, M. Song, D. Tao, X. Zhou, C. Chen, J. Bu. "Semi-supervised coupled dictionary learning for person re-identification", *Proc. CVPR*, pp. 3550–3557, (2014).

[24] E. Ahmed, M. Jones, T. K. Marks. "An improved deep learning architecture for person re-identification", *Proc. CVPR*, pp. 3908–3916, (2015).

[25] D. Yi, Z. Lei, S. Liao, S. Z. Li. "Deep metric learning for person re-identification", *Proc. ICPR*, pp. 34–39, (2014).

[26] S. Paisitkriangkrai, C. Shen, A. van den Hengel. "Learning to rank in person re-identification with metric ensembles", *Proc. CVPR*, pp. 1846–1855, (2015).

[27] S. Wu, Y.-C. Chen, W.-S. Zheng. "An enhanced deep feature representation for person re-identification", *Proc. WACV*, pp. 1–8, (2016).

[28] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, Q. Tian. "Query-adaptive late fusion for image search and person re-identification", *Proc. CVPR*, pp. 1741–1750, (2015).

[29] N. Martinel, C. Micheloni, G. L. Foresti. "A pool of multiple person re-identification experts", *Pattern Recognition Letters*, **71**, pp. 23–30, (2016).

[30] X. Liu, H. Wang, Y. Wu, J. Yang, M.-H. Yang. "An ensemble color model for human re-identification", *Proc. WACV*, pp. 868–875, (2015).

[31] M. Ye, J. Chen, Q. Leng, C. Liang, Z. Wang, K. Sun. "Coupled-view based ranking optimization for person re-identification", *Proc. MMM*, pp. 105–117, (2015).

[32] R. F. de Carvalho Prates, W. R. Schwartz. "Cbra: Color-based ranking aggregation for person re-identification", *Proc. ICIP*, pp. 1975–1979, (2015).

[33] D. J. Jobson, Z.-u. Rahman, G. A. Woodell. "A multiscale retinex for bridging the gap between color images and the human observation of scenes", *IEEE Transactions on Image Processing*, **6(7)**, pp. 965–976, (1997).

[34] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, S. Z. Li. "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes", *Proc. CVPR*, pp. 1301–1306, (2010).

[35] S. Li, M. Shao, Y. Fu. "Cross-view projective dictionary learning for person re-identification", *Proc. AAAI*, pp. 2155–2161, (2015).

[36] Y.-C. Chen, W.-S. Zheng, J. Lai. "Mirror representation for modeling view-specific transform in person re-identification", *Proc. IJCAI*, pp. 3402–3408, (2015).

[37] R. F. de Carvalho Prates, W. R. Schwartz. "Appearance-based person re-identification by intra-camera discriminative models and rank aggregation", *Proc. ICB*, pp. 65–72, (2015).

[38] J. M. Stuart, E. Segal, D. Koller, S. K. Kim. "A gene-coexpression network for global discovery of conserved genetic modules", *Science*, **302(5643)**, pp. 249–255, (2003).

[39] R. Kolde, S. Laur, P. Adler, J. Vilo. "Robust rank aggregation for gene list integration and meta-analysis", *Bioinformatics*, **28(4)**, pp. 573–580, (2012).

[40] P. M. Roth, M. Hirzer, M. Köstinger, C. Beleznai, H. Bischof. "Mahalanobis distance learning for person re-identification", S. Gong, M. Cristani, S. Yan, C. C. Loy (Eds.), *Person Re-Identification*, volume 1 of *Advances in Computer Vision and Pattern Recognition*, 1 edition, chapter 12, pp. 247–267, (Springer, London, 2014).

[41] W. Li, R. Zhao, X. Wang. "Human reidentification with transferred metric learning.", *Proc. ACCV*, pp. 31–44, (2012).

[42] S.-Z. Chen, C.-C. Guo, J.-H. Lai. "Deep ranking for person re-identification via joint representation learning", *IEEE Transactions on Image Processing*, **25(5)**, pp. 2353–2367, (2016).