Pedestrian Recognition with a Learned Metric

Mert Dikmen, Emre Akbas, Thomas S. Huang, and Narendra Ahuja

Beckman Institute,
Department of Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign
{mdikmen,eakbas2,t-huang1,n-ahuja}@illinois.edu

Abstract. This paper presents a new method for viewpoint invariant pedestrian recognition problem. We use a metric learning framework to obtain a robust metric for large margin nearest neighbor classification with rejection (i.e., classifier will return no matches if all neighbors are beyond a certain distance). The rejection condition necessitates the use of a uniform threshold for a maximum allowed distance for deeming a pair of images a match. In order to handle the rejection case, we propose a novel cost similar to the Large Margin Nearest Neighbor (LMNN) method and call our approach Large Margin Nearest Neighbor with Rejection (LMNN-R). Our method is able to achieve significant improvement over previously reported results on the standard Viewpoint Invariant Pedestrian Recognition (VIPeR [1]) dataset.

1 Introduction

Viewpoint invariant recognition of pedestrians is a problem that appears in numerous contexts in computer vision scenarios such as multi-camera tracking, person identification with an exemplar image or re-identification of an individual upon re-entering the scene after some time. This is a key problem and has been drawing attention in recent years with the advance of visual tracking and widespread deployment of surveillance cameras, which necessitated the need for continuous tracking and recognition across different cameras even with significant time and location differences. Our approach handles the long time delay case: recognition of the same individual without the temporal and spatial information associated with the images of the pedestrians. By learning an appropriate distance metric we achieve high recognition with high accuracy. Although we demonstrate it in the context of this problem, the learned metric is general and can be applied to aid data association in other tracking scenarios.

This paper assumes that the pedestrians in the scene has been successfully detected and consequently cropped. Pedestrian detection is an active research topic, but fortunately this problem is easier than the problem of general object detection and has been met with reasonable success with the emergence of several advanced methods in recent years. The relative success of pedestrian detection can be attributed to several limiting factors on the complexity of the

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part IV, LNCS 6495, pp. 501–512, 2011.

problem. Pedestrians are by definition upright people figures with limited configurations. Therefore template based approaches with a sliding window classifier produce favorable results [2,3]. In addition, there exists a number of strong and relatively easy to detect contextual cues, such as the presence of ground and other rigid objects (e.g., cars), which can be integrated into the decision process to significantly improve the detection performance [4].

Several attempts have been made for tackling the recognition problem in the context of matching pedestrians by their appearance only. Park et al. [5] perform recognition by matching color histograms extracted from three horizontal partitions of the person image. Hu et al. [6] have modeled the color appearance over the silhouette's principal axis. However, finding the principal axis requires robust background subtraction and is error prone in crowded situations. Matching spatio-temporal appearance of segments have been considered by Gheissari et al. [7]. Yu et al. [8] introduced a greedy optimization method for learning a distance function. Gray and Tao [1] defined the pedestrian recognition problem separate from multi-camera tracking context and provided a benchmark dataset (VIPeR, see Fig. 1) for standardized evaluation. Their method transforms the matching problem into a classification problem, in which a pair of images is assigned a positive label if they match (i.e., belong to the same individual) or negative label otherwise. This classifier is learned in a greedy fashion using AdaBoost. The weak classifiers are decision stumps on individual dimensions of histograms of various features within a local rectangle in the person image. The rectangles span the entire horizontal dimension, while they are densely sampled vertically over all positions and sizes. Note that in the context of nearest neighbor classification, the $\{+1, -1\}$ labeling scheme of the matches vs non-matches

Fig. 1. Representative image pairs from the VIPeR dataset (images on each column are the same person). The dataset contains many of the challenges observed in realistic conditions, such as viewpoint and articulation changes as well as significant lighting variations.

creates a naturally unbalanced learning problem with N vs N^2 samples in two classes respectively (N = number of training points). Also worth noting is that the two methods [8, 1], which learn the pairwise comparison function, achieve this through greedy optimization, which is not globally optimal and furthermore makes indirect use of covariances in the feature space. Our method is both globally optimal and also has an explicit covariance modeling of features.

The contributions of this paper are the following: (1) We apply a large margin nearest neighbor approach to the pedestrian recognition problem to achieve significantly improved results, (2) we define a novel cost function for learning a distance metric specifically for nearest neighbor problems with rejection. In addition we show that despite using only color as the appearance feature, our method is robust under significant illumination changes.

2 Metric Learning

In this section, we briefly introduce the metric learning framework of Weinberger and Saul [9] for large margin nearest neighbor (LMNN) classifier. The goal is to learn a Mahalanobis metric where the squared distances are denoted by:

$$\mathcal{D}_{\mathbf{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathsf{T}}\mathbf{M}(\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{1}$$

$\mathcal{D}_{\mathbf{M}}^{1/2}$ is a valid distance iff $\mathbf{M}$ is a symmetric positive-semidefinite matrix. In this case $\mathbf{M}$ can be factored into real-valued matrices as $\mathbf{M} = \mathbf{L}^{\mathsf{T}}\mathbf{L}$. Then, an equivalent form for (1) is

$$\mathcal{D}_{\mathbf{L}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = ||\mathbf{L}(\boldsymbol{x}_i - \boldsymbol{x}_j)||^2. \tag{2}$$

LMNN learns a real-valued matrix $\mathbf{L}$ that minimizes the distance between each training point and its K nearest similarly labeled neighbors (Eq. 3), while maximizing the distance between all differently labeled points, which are closer than the aforementioned neighbors' distances plus a constant margin (Eq. 4).

$$\varepsilon_{pull}(\mathbf{M}) = \sum_{i,j \rightsquigarrow i}^{N} \mathcal{D}_{\mathbf{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{3}$$

$$\varepsilon_{push}(\mathbf{M}) = \sum_{i,j \rightsquigarrow i} \sum_{k=1}^{N} (1 - y_{ik}) \left[1 + \mathcal{D}_{\mathbf{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \mathcal{D}_{\mathbf{M}}(\boldsymbol{x}_i, \boldsymbol{x}_k)\right]_+ \tag{4}$$

Here, y_{ik} is an indicator variable which is 1 if and only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same class, and $y_{ik} = 0$ otherwise. The $j \rightsquigarrow i$ notation means that $\boldsymbol{x}_j$ is one of the K similarly labeled nearest neighbors of $\boldsymbol{x}_i$ (i.e., $\boldsymbol{x}_j$ is a target neighbor of $\boldsymbol{x}_x$). Note that for ε_{pull} to be a continuous and convex function, it is necessary that the K target neighbors of each training sample be fixed at the initialization. In practice they are determined by choosing the K nearest neighbors by Euclidean distance.

The $\boldsymbol{x}_k$ in Eq.4 for which $y_{ik} = 0$ are called the impostors for $\boldsymbol{x}_i$. The expression $[z]_+ = max(z, 0)$ denotes the standard hinge loss. Although this hinge loss is not differentiable at $z = 0$, we did not observe any convergence issues. Nevertheless it is always possible to replace the standard hinge loss with a smooth approximation [10].

The affine combination of ε_{pull} and ε_{push} through the tuning parameter μ[1] (Eq. 5) defines the overall cost, which essentially maximizes the margin for K nearest neighbor classifier by pulling together same-labeled points and repelling differently-labeled ones (impostors).

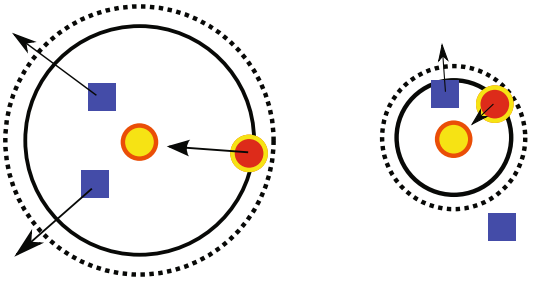
$$\begin{aligned}\varepsilon_{\mathbf{LMNN}}(\mathbf{M}) =&(1-\mu)\sum_{i,j\rightsquigarrow i}\mathcal{D}_{\mathbf{M}}(\boldsymbol{x}_i,\boldsymbol{x}_j)\\ &+\mu\sum_{i,j\rightsquigarrow i}\sum_{k=1}^{N}(1-y_{ik})\left[1+\mathcal{D}_{\mathbf{M}}(\boldsymbol{x}_i,\boldsymbol{x}_j)-\mathcal{D}_{\mathbf{M}}(\boldsymbol{x}_i,\boldsymbol{x}_k)\right]_+.\end{aligned} \tag{5}$$

2.1 Nearest Neighbor with Rejection

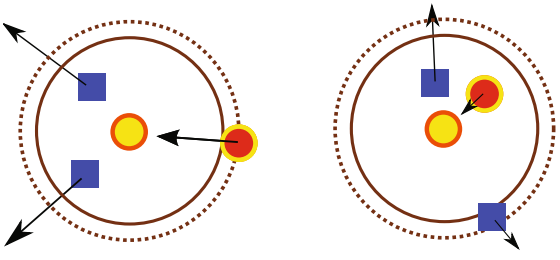
In this section we introduce our LMNN-R framework for doing K nearest neighbor classification with the option of rejection. As a practical example for this problem, consider the person re-identification task, where given an image of a pedestrian, one would like to determine whether the same person is in the current scene or not. The target set of the people in the scene may not contain the query person. One way to adapt the nearest neighbor classifier to the problem of re-identification is to adopt a universal threshold (τ) for maximum allowed distance for matching image pairs. If the distance of the nearest neighbor of the query in the target set is greater than τ, one would deem that the query has no match in the target set (rejection). Conversely, if there is a nearest neighbor closer than τ, then it is called a match. What we have just described is the 1 nearest neighbor with rejection problem. This problem can be extended to K nearest neighbor case, in which a label is assigned through majority voting of P nearest neighbors within τ, where $P \leq K$. If $P = 0$ the classifier will refuses to assign a label.

The introduction of the option to refuse label assignment necessitates a distance metric that allows the use of a global threshold in all localities of the feature space. One method would be to assume unimodal class distributions as proposed by Xing et al. [11]. Their objective function maximizes the distance between all sample pairings with different labels, while a constraint is imposed on the pairs of similarly labeled points to keep them closer than a universal distance. This model was proposed for learning a distance metric for k-means clustering. It does not directly apply to our problem formulation. One drawback is the situation when similarly labeled samples do not adhere to a unimodal distribution (e.g., two islands of samples with same labels). Another problem is the lack of margin in their formulation, which is essential for good generalization

[1] All reported experiments in this paper use $\mu = 0.5$ for both LMNN and LMNN-R.



(a) LMNN



(b) LMNN-R

Fig. 2. Illustration contrasting our proposed approach with [9] Note that the point configurations for a) and b) are the same. For a given training point (yellow), the target neighbor (red) is pulled closer, while the impostors (blue) are pushed away. a) To determine the impostors, the LMNN cost function uses a variable distance from the training point depending on the proximity of the target neighbors; b) LMNN-R on the other hand, forces the impostors out of a universal distance from the training point, while simultaneously attracting target neighbors.

performance in classification. A cost function, which emphasizes local structure is more suitable in our case.

We adopt the LMNN cost function (Eq. 5), which minimizes the distance between each training point and its K nearest similarly labeled neighbors (Eq. 3), while maximizing the distance between all differently labeled points, which are closer than the aforementioned neighbors' distances plus a constant margin (Eq. 4). The margin imposes a buffer zone to ensure good generalization. It is this local property that makes the LMNN metric learning very suitable to nearest neighbor classification. Note that the distance to determine the impostors is varying for each training point $\boldsymbol{x}_i$ (Eq. 4). We replace this with a universal distance: the average distance of all K nearest neighbor pairs in the training set (Eq. 6). LMNN-R cost function forces the closest impostors of a training point to be at least a certain distance away, determined by this average which is only weakly affected by where its own K nearest neighbors are (Fig. 2). The net effect of this modification is that now we can use a universal threshold on pairwise distances for determining rejection, while still approximately preserving the local

structure of the large margin metric learning. The only requirement for the loss function to be convex is that the K nearest neighbor structure of the training points need to be pre-defined. However, extensions such as multi-pass optimization [9] proposed to alleviate this problem for LMNN apply to LMNN-R also.

$$R = \frac{1}{NK} \sum_{m,l \rightsquigarrow m} \mathcal{D}_{\mathbf{M}}(\boldsymbol{x}_m, \boldsymbol{x}_l) \tag{6}$$

$$\varepsilon_{\textbf{LMNN-R}}(\mathbf{M}) = (1-\mu)\varepsilon_{pull}(\mathbf{M}) + \mu\varepsilon^*_{push}(\mathbf{M}) \tag{7}$$

$$\varepsilon^*_{push}(\mathbf{M}) = \sum_{i=1}^{N}\sum_{k=1}^{N}(1-y_{ik}) \left[1 + \frac{1}{NK} \left(\sum_{m,l \rightsquigarrow m} \mathcal{D}_{\mathbf{M}}(\boldsymbol{x}_m, \boldsymbol{x}_l) \right) - \mathcal{D}_{\mathbf{M}}(\boldsymbol{x}_i, \boldsymbol{x}_k) \right]_+, \tag{8}$$

The LMNN-R cost (Eq. 7) can be minimized as a semidefinite program, which is formulated by writing ε^*_{push} as a constraint through the introduction of slack variables, or it can be minimized by following the gradient directly and projecting $\mathbf{M}$ back to the semidefinite cone at each iteration (iterative sub-gradient projection as in [9]).

3 Experiments

We demonstrate the performance of our method on the VIPeR dataset [12] which is a specifically constructed dataset for the viewpoint invariant pedestrian recognition problem. This dataset contains images of 632 unique pedestrians and a total of 1264 images composed of two views per pedestrian seen from different viewpoints. The images are captured outdoors under uncontrolled lighting. Therefore there is a great deal of illumination variance in the dataset, including between the images belonging to the same pedestrian (e.g., the first and last columns in Figure 1). Compared to the previously available datasets (see [1]), the VIPeR dataset has many more unique subjects and contains a higher degree of viewpoint and illumination variation, which makes it realistic and more challenging (Figure 1).

3.1 Methodology

As done in [1], we randomly split the set of pedestrians into two halves: training and testing. The LMNN and LMNN-R frameworks learn their respective distance metric using the training set. For testing, each image pair of each pedestrian in the test set is randomly split to query and target sets. The results are generated using the pairwise distance matrix between these query and target subsets of the images in the test set. For thoroughness, we report our results as an average over 10 train-test splits. When reporting an average is not appropriate, we report our best result out of the 10 splits.

We follow the same evaluation methodology of [1] in order to compare our results to theirs and other benchmark methods. We report results in the form of cumulative matching characteristics curve (CMC), re-identification rate curve and expected search time by a human operator. In addition, we also provide an average receiver operator characteristic curve to demonstrate the improvement of the LMNN-R method over LMNN for automated recognition.

3.2 Image Representation

The images in the dataset are 128 pixels tall and 48 pixels wide. We use color histograms extracted from 8x24 rectangular regions to represent the images. The rectangular regions are densely collected from a regular grid with 4 pixel spacing in vertical and 12 pixel spacing in horizontal direction. This step size is equal to half the width and length of the rectangles, providing an overlapping representation.

For the color histograms, we use RGB and HSV color spaces and extract 8-bin histograms of each channel separately. We tried several combinations for all of the mentioned parameters found that these numbers worked reasonably well through our preliminary experiments. We concatenate the histograms extracted from an image and obtain a feature vector of size 2232 for RGB and HSV representations each. The combined representation is simply the concatenation of these two. Dimension reduction through PCA is applied to these high-dimensional vectors to obtain subspaces of specific dimensionality. This step is necessary to reduce redundancy in the color based representation and to filter out some of the noise. The reported results are obtained with 20, 40 and 60 dimensional representations. We have observed that we get diminished returns above 60 dimensions.

To account for the illumination changes we experiment with a simple color correction technique where each RGB channel of the image is histogram-equalized independently to match a uniform distribution as close as possible in ℓ_1 norm. Since in the cropped images, a significant number of the pixels belong to the pedestrian, this is a reasonable way of performing color correction. We also experimented with brightness and contrast correction methods, as well as histogram equalizing the V channel of the HSV images. However, they were not able to perform as good as the described RGB histogram equalization method.

3.3 Results

Recognition. We present the recognition performances as CMC curves in Figure 3. This curve, at rank score k, gives us the percentage of the test queries whose target (i.e. correct match) is within the top k closest match. As it is not appropriate to take the average of CMC curves over different random splits of the dataset, we report the CMC of a single split where the normalized area under the curve is maximum. This corresponds to using "RGB+HSV" features reduced to 60 dimensions via PCA and using our proposed approach LMNN-R. We outperform

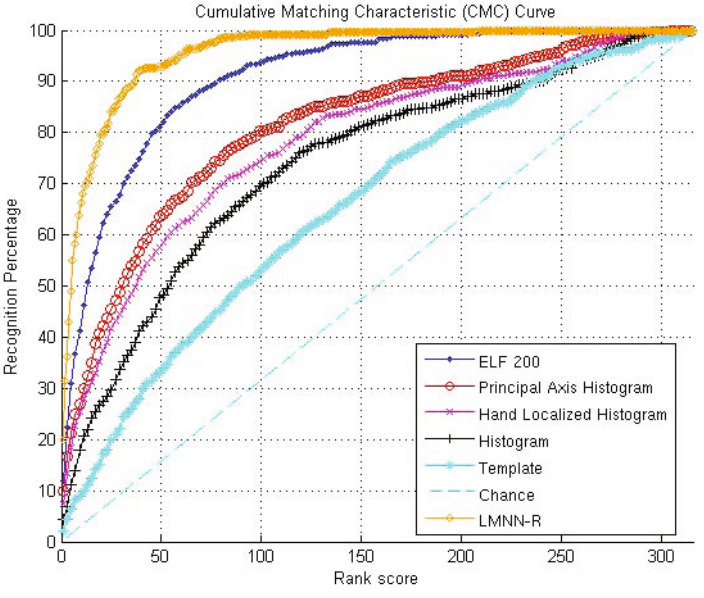


Fig. 3. Cumulative matching characteristics (CMC) curve for our method and others'. This result is obtained using a combined HSV and RGB representation in a 60 dimensional subspace learned with PCA.

all previously reported results[2]. An explanation of the methods used to obtain these previous results is as follows. "Chance" refers to random matching, "Template" refers to pixelwise sum-of-squared distances matching. "Histogram" and "Hand Localized Histogram" refer to the method by Park et al. [5], and "Principal Axis Histogram" refers to the method of Hu et al. [6]. "ELF 200" (or just "ELF" in the remaining of the text) refers to the work of Gray et al. [1].

CMC curves can be summarized using the "expected search time" measure defined in [1]. Assuming a human operator reviews a query image's closest matches sequentially according to their distance from the query. Assuming an average review time of 1s per image, the total expected search time for finding the correct match would be the average rank of the target. Our method's expected target rank is 23.7 which is an improvement of over 15% with respect to the state-of-the-art 28.9 (see Table 1).

To evaluate the performance of LMNN and LMNN-R over all different combinations of parameter and feature choices, we use the normalized area under the CMC curves. Table 2 shows the mean and standard deviation of these values over 10 random splits of the dataset. Best results are obtained using RGB and HSV together on original (non-color corrected) images. RGB alone performs the worse than HSV alone, which is expected because HSV is more robust to variations in intensity of the lighting.

[2] Results of other methods are from [1] as a courtesy of D. Gray.

Table 1. Expected search times for LMNN-R and other methods

Method	Expected Search Time (in seconds)
Chance	158.0
Template	109.0
Histogram	82.9
Hand Localized Histogram	69.2
Principal Axis Histogram	59.8
ELF	28.9
LMNN-R	**23.7**

Since the dataset has a significant degree of illumination variation, one expects that color correction should help increase the matching accuracy. While this is true for the plain ℓ_2 norm (i.e. no learning), it is not the case for learned metrics of LMNN and LMNN-R. A possible explanation for this can be made by realizing that the histogram equalization process is a non linear transformation of the data. While improving the performance of the marginal cases for simple matching by Euclidean distances, this procedure may affect the average transformation that image pairs undergo in realistic scenarios, such that this transformation cannot be reliably modeled by LMNN and LMNN-R methods anymore. Therefore we suggest letting the learning algorithm handle the color correction issues.

For the number of reduced dimensions, 60 is slightly better than 40. And LMMN-R gives slightly better results than LMNN in general.

In the previous re-identification experiments, we assume that the target set will have a match for the query image. This is not the case in many practical scenarios as often it is not known whether the query person is in view. Therefore

Table 2. Table of results averaged over 10 random splits of the dataset. 20, 40 and 60 denote the number of dimensions (of the reduced subspace found by PCA) used, L_2 refers to the regular ℓ_2 norm which, in our case, corresponds to "no learning". "corr'd" means "color corrected" and "orig" indicates that no modification was done to the original image. We obtain our best average results using RGB and HSV together on original images with the proposed learning approach LMNN-R. The overall best result, i.e. the one given in Figure 3, has a normalized area of 95.88 under its CMC curve, which is comparable to the average results.

		RGB+HSV		HSV		RGB	
		corr'd	orig	corr'd	orig	corr'd	orig
20	L_2	76.61 ± 0.88	72.54 ± 0.77	80.09 ± 0.59	77.97 ± 0.81	67.85 ± 1.13	60.63 ± 0.79
	LMNN	91.81 ± 0.39	93.46 ± 0.36	92.11 ± 0.47	92.90 ± 0.34	82.06 ± 0.69	86.39 ± 0.72
	LMNN-R	92.14 ± 0.37	93.59 ± 0.37	92.35 ± 0.47	92.87 ± 0.51	82.47 ± 0.83	86.63 ± 0.68
40	L_2	77.48 ± 0.87	73.73 ± 0.81	80.79 ± 0.66	78.89 ± 0.80	68.73 ± 1.11	60.90 ± 0.92
	LMNN	92.68 ± 0.44	94.54 ± 0.42	92.82 ± 0.33	94.40 ± 0.32	83.81 ± 1.27	87.14 ± 0.86
	LMNN-R	93.13 ± 0.48	94.76 ± 0.47	93.04 ± 0.45	94.64 ± 0.43	84.71 ± 1.24	87.49 ± 0.92
60	L_2	77.85 ± 0.86	74.14 ± 0.79	80.97 ± 0.67	79.17 ± 0.80	68.83 ± 0.91	61.20 ± 0.91
	LMNN	92.27 ± 0.50	94.67 ± 0.55	92.52 ± 0.28	94.54 ± 0.29	84.23 ± 0.63	87.56 ± 1.01
	LMNN-R	92.56 ± 0.53	$\mathbf{94.95 \pm 0.46}$	92.62 ± 0.43	94.69 ± 0.37	84.94 ± 0.57	87.79 ± 1.04

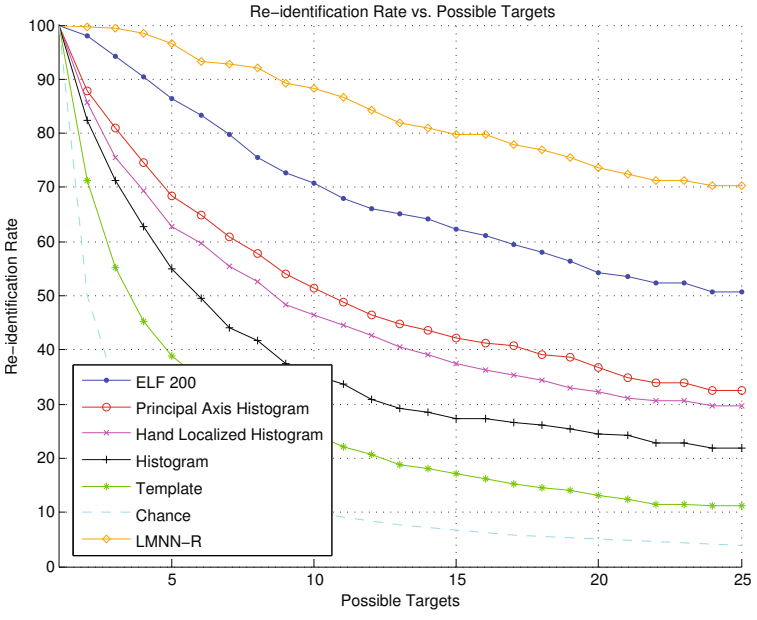


Fig. 4. Re-identification rate vs. the number of targets for our method and others

we also show the receiver operator characteristic curve (ROC) for such kind of cases where one would like to detect the query pedestrian in a target set of pedestrians. The detection performance is measured by comparing the true positive rate vs. the false positive rate, which shows for a given recall rate (true positive), what fraction of non matching images in the target set will be returned as false positives. Due to the universal threshold, the LMNN-R method was able to outperform LMNN by about 1% at a false positive rate of 10% (Fig. 5).

Re-identification. This is another measure for evaluating the performance of pedestrian matching methods. It is the probability of finding a correct match as a function of the number of possible targets. A formal definition could be found in [12]. Figure 4 shows the re-identification rates of our method and the previous methods.

Execution times. We implemented LMNN and LMNN-R in MATLAB[3] and although we have not employed the active set method which was designed to make LMNN more efficient (described in [9]), our code runs reasonably fast in practice. For the VIPeR dataset, a typical training session takes 160 seconds and finding the target of a query pedestrian takes only 1.2 milliseconds on a 2GHz Intel Core2-Duo PC.

[3] The MATLAB code for LMNN and LMMN-R optimization as well as replicating the experiments in the paper is available in the supplementary material of the paper.

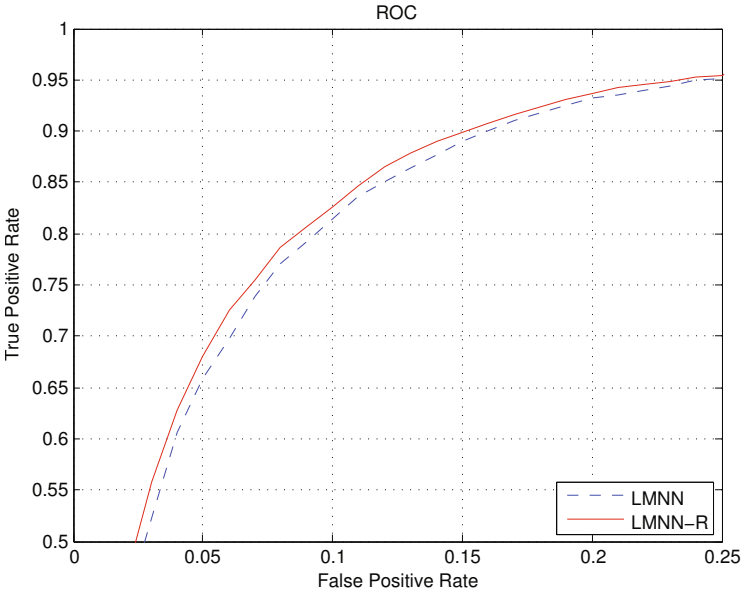


Fig. 5. The receiver operator characteristic curve showing the true positive vs the false positive rate of our system

4 Conclusions

We have applied a large margin nearest neighbor (LMNN) approach to viewpoint invariant pedestrian recognition problem. Also, we proposed a new variant of LMNN called large margin nearest neighbors classification with rejection (LMNN-R) to obtain a classifier with the option of rejecting unfamiliar matches. Using only color histograms as features, these methods achieved significant improvement over previously reported results on a benchmark dataset. Experimental results suggest that our LMNN-R formulation to metric learning is able to achieve improved results over LMNN. Color correction improved the matching accuracy when Euclidean distance is used to compare images (i.e. no learning). However, this was not the case for LMNN and LMNN-R which suggests that these supervised learning approaches are more robust in handling illumination changes than color correction alone.

References

1. Gray, D., Tao, H.: Viewpoint invariant pedestrian recognition with an ensemble of localized features. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 262–275. Springer, Heidelberg (2008)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, p. 886 (2005)

3. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 1713–1727 (2008)
4. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. International Journal of Computer Vision 80, 3–15 (2008)
5. Park, U., Jain, A., Kitahara, I., Kogure, K., Hagita, N.: Vise: visual search engine using multiple networked cameras. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006), vol. 3, pp. 1204–1207 (2006)
6. Hu, W., Hu, M., Zhou, X., Tan, T., Lou, J., Maybank, S.: Principal axis-based correspondence between multiple cameras for people tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 663–671 (2006)
7. Gheissari, N., Sebastian, T., Hartley, R.: Person reidentification using spatiotemporal appearance. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1528–1535 (2006)
8. Yu, J., Amores, J., Sebe, N., Radeva, P., Tian, Q.: Distance learning for similarity estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 451–462 (2008)
9. Weinberger, K., Saul, L.: Distance metric learning for large margin nearest neighbor classification. The Journal of Machine Learning Research 10, 207–244 (2009)
10. Rennie, J.D.M., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the 22nd International Conference on Machine Learning, ICML 2005, pp. 713–719 (2005)
11. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning with application to clustering with side-information. In: Advances in Neural Information Processing Systems, pp. 521–528 (2003)
12. Gray, D., Brennan, S., Tao, H.: Evaluating appearance models for recognition, reacquisition, and tracking. In: Proc. IEEE Int'l Workshop on Performance Evaluation for Tracking and Surveillance, PETS (2007)