Target Domain Adaptation for Face Detection in A Smart Camera Network with Peer-to-Peer Communications

Shuixian Chen[1,2], Xiang Lu[1,2], Limin Sun[1,2], Shiming Ge[1,2]
[1]Institute of Information Engineering, CAS, Beijing, China
[2]2Beijing Key Laboratory of IOT Information Security Technology, IIE, CAS, Beijing, China
{chenshuixian, luxiang, sunlimin, geshiming}@iie.ac.cn

***Abstract*—With the fast advance of mobile chips technologies, a node in a smart camera network can afford sophisticated processing via on-board multicore CPUs and GPUs, e.g., face detection. The performance of a general purpose face detector, however, may degrade seriously under specific situations with unexpected challenges such as facial coverage or bad illumination. This degradation is due to the difference of probability distributions between training data and testing data, known as source data domain and target data domain, respectively. To better adapt a smart camera network to a specific situation, some form of target domain adaptation is needed, which usually requires both the source domain data and as much as possible target domain data at each node, which may strain storage capacity and bandwidth. In this paper, we propose an adaptation method which fuses the source specific hypotheses (SSHs) and target specific hypotheses (TSHs) — requiring only a pre-trained face detector and a few target data to be shared by peer-to-peer communications, thus relieving the storage and bandwidth constraints. The method uses the "accuracy-regularization" objective as the adaptation model, to fuse SSHs and TSHs, and tries to minimize the misclassification error on target data. With an existing frontal face detector, we conduct experiments to verify our algorithm, covering cases of video surveillance, extreme pose challenge, and different illumination spectra. Significant performance gains are observed with only dozens of target data in all the experiments, demonstrating the effectiveness of the proposed adaptation model. Therefore, the proposed adaption can be applied to a smart camera network with peer-to-peer communications to improve the network's overall face detection performance.**

I. Introduction

In a smart camera network, each node needs to have sufficient computational power for image/video capturing, encoding, and processing, and is capable of peer-to-peer (P2P) communications. This only becomes practical with the unprecedented advancement of mobile chips technology in recent years. Today, a smart phone, with its camera, multicore CPUs, GPUs, and Digital Signal Processors (DSPs), and with its communications modules (e.g., cellular, WiFi, Bluetooth), is powerful enough to act as a node in a smart camera network. It's also possible to build dedicated smart camera nodes around mobile chip sets with additional features like weather-proof and controllable orientation.

This work was supported in part by Special Forerunner Research Projects under grant XDA06040101, and Science and Technologies Projects of Xinjiang Autonomous Region under grant Y3V0021402. (*Corresponding author: Xiang Lu*)

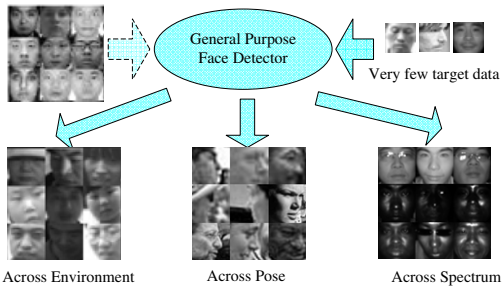

Fig. 1. Adaptation of a face detector. An existing face detector trained on general frontal faces are adapted across different data domains, with only few target data available. In this paper, the pre-trained detector is adapted across environment, across pose and across spectrum.

Such platforms allow sophisticated image/video processings [1], [2], [3]. Face detection, for example, is one of the processings. It's a crucial procedure in face biometrics, the accuracy of which significantly affects subsequent operations, such as face recognition. Nowadays the dominating method is the Viola-Jones framework [4], which adopts a cascade structure of boosting classifiers to achieve robust and real-time performance. Based on the framework, many variants have been proposed to further advance the literature, such as [5], [6], [7], [8], [9], [10], [11], [12]. These remarkable improvements result in today's high face detection performance, even when faces exhibit serious pose or rotation challenges [12]. Please refer to [13] for a complete survey in this field.

There still exists, however, expensive and time-consuming manual work for face image collecting and labeling. Generally speaking, a good face detector requires thousands or even tens of thousands of images for training [9], [4]. Such a huge number will cause tremendous workload not only in data preparation but also in training computation. The case may become even worse when we try to detect faces in real world applications where the in-site faces may exhibit distinctive appearances from training data. If so, the pre-trained face detector may not work well, and numerous new data have to be collected for another training. In Fig.1 we give several cases about the data domain difference, which will be discussed in this paper.

A node in a smart camera network may have enough computational power for face detection. But storing all the pre-training data and sharing new data among nodes may well

978-1-4799-5952-5/15/$31.00 ©2015 IEEE

beyond the storage and communications capacity of nodes. Therefore, it is highly desirable if a good detector can be trained with only a few data available from target domain. In the machine learning literature, this is known as transfer learning (TL [14], [15]) and domain adaptation (DA[16], [17], [18]). In TL/DA techniques, a large amount of existing source data are combined with a small amount of target data, so that knowledge can be leveraged from source data domain to assist classifier training in target domain. The basic principle of TL and DA techniques is to minimize domain difference or to explore domain relations, such as methods in [19], [20], [21]. Please refer to [22], [23] for a thorough survey in the field.

These TL/DA techniques generally require source data to work, which is, however, very often unavailable. When a pre-trained face detector is to be distributed, for example, we seldom distribute its training data as well. C. Huang[24] proposed an elegant incremental learning algorithm to relax the requirement: Rather than keeping source data, the algorithm keeps the intermediate optimization information when the face detector was trained. When a few target data are used to adapt the pre-trained detector, the preserved information can simulate the existence of the source data. C. Zhang explores a similar idea [25], where the intermediate information is preserved via Taylor expansion.

Additional information, however, is usually absent from face detectors, making their method inapplicable in most cases. So the question is, can we adapt a detector without source data or any information of source data? G. Chen et al. propose such an algorithm in [26]: Cross-entropy is minimized followed by a minmax formation to avoid overfitting. In [27], Gaussian Process Regression (GPR) is used to describe the similarity between face candidates so that a lower threshold is set for face-like candidate. Both methods demand nothing besides a common pre-trained detector, and therefore can be directly applied to any existing detectors.

In this paper, we propose a novel adaptation method for face detectors, by fusing source domain specific and target domain specific hypotheses. Similar to [26], [27], our method handles a common pre-trained detector, and requires no source data. But unlike the previous methods, our method explicitly explores the target domain specific hypotheses. We observed that sometimes test data deviates significantly from the source specific hypotheses, which significantly impairs detection performance. Hence target domain specific hypotheses are necessary for effective adaptation. We further design an adaptation model formulated as an "accuracy-regularization" objective, so that high accuracy can be achieved on the few target data meanwhile preserving good generalization ability on unseen data. Extensive experiments have shown the superiority of our algorithm.

The paper is organized as follows. In Section II the motivation and the basic principle of our algorithm is explained; in Section III we present our algorithm in details; in Section IV the experiments on several data sets are provided and finally in Section V.

II. Motivation

Typically, to train a general-purpose face detector, we need sufficient and representative samples along with their labels of being face or not. These samples and their labels are in the source data domain, denoted as $D^S = \{(x_1^S, y_1^S), \ldots, (x_n^S, y_n^S)\}$, where $x_i^S \in \mathbf{R}^d$ is a training sample as a d-dimensional vector, and $y_i^S \in \{-1, +1\}$ is the corresponding label. After training, we have a general-purpose face detector Q, which judges a test sample x with a label or score $Q(x)$. If well-trained, the detector should have good performance, i.e., high verification rate (VR) and low false alarm (FA) rate, but only on test samples similar to those in D^S. But under a specific environment or setting, e.g., infrared camera images, the detector may encounter samples that are far distinct from those in D^S. These field samples and their labels, denoted as $D^T = \{(x_1^T, y_1^T), \ldots, (x_m^T, y_m^T)\}$, are called target data domain. Usually, target samples are far fewer than source samples, but unavailable for the training. We want to upgrade Q in such a way that the performance on target samples is improved while the performance on general samples is untouched.

Training of the detector Q can be abstracted as minimizing a loss function L of the source data over some set $\mathcal{S}$ of candidate detectors:

$$Q = \arg\min_{Q \in \mathcal{S}} \sum_{i=1}^{n} L(y_i^S, Q(x_i^S)). \quad (1)$$

But (1) does not guarantee good performance on the target data domain D^T. Adaptation of Q to the target samples is needed. Direct applying (1) to D^T, however, is prone to overfitting due to the very limited amount of target data.

Instead, we relax (1) so that both source data and target data are considered:

$$F = \arg\min_{F \in \mathcal{S}} \sum_{i=1}^{n} L(y_i^S, F(x_i^S)) + \lambda \sum_{j=1}^{m} L(y_j^T, F(x_j^T)), \quad (2)$$

where λ weights the relative importance between D^S and D^T. The formulation of (2) frequently appears in the literature of transfer learning [15], and has been shown very effective in many real world applications [14]. But keeping the large amount of source data (typically 10^4 to 10^6 for face detector training) is not convenient and even impossible sometimes. It is more practical if we can train a good detector F with only the pre-trained Q and the small amount of the target data.

We propose a two-step procedure to solve the above problem:

- First, using the pre-trained detector Q as a prior to train a target-specific detector G over the few target data. The target data can be divided into two subsets by Q, those that are corrected detected and those incorrect detected. Larger weights should be put to the latter when training G. Details will be presented in Section III-A. But since the source data are not accessible to the training of G, without further processing, this G may overfit the few target data and not generalize well to unseen samples. So we need another step as follows.
- Second, fusing the pre-trained detector Q with the target-specific detector G to form an enhanced detector F. The general problem of fusing two detectors optimally can be very nonlinear and complicated. But we

can make a safe assumption about the face detectors: They have the boosting based structure [4], which is the most successful structure to date and widely used in research and practice. Under this assumption, the enhanced face detector F will be a linear combination of simple hypotheses (as do Q and G):

$$F(x) = \sum_{i=1}^{k} w_i h_i(x), \tag{3}$$

where w_i is a weighting coefficient, h_i is a hypothesis based on features, e.g., Haar-like features in the Viola-Jones face. Then the problem of fusing Q and G is simplified to assigning optimal weighting coefficients for all the hypotheses from Q and G to form the detector F We will present the details of fusing in Section III-B.

III. Domain Adaptation via Hypotheses Fusing

In this section we will introduce our domain adaptation method using a pre-trained face detector, according to the procedures mentioned above. To simplify analysis, features are assumed to be linear functions of samples (viewed as vectors in $\mathbb{R}^N$). This is not a restriction in practice, most features used in face detection are linear, including Haar-like features [4] and granular features [28]. Also, the features are scaled to $[0, 1]$ over the sample space (e.g., all 24×24 blocks of pixels with gray scale between 0 and 1). Real AdaBoost [29] is used to learn a strong classifier from weak classifiers. It usually performs better than the original discrete AdaBoost, i.e., faster convergence, lower training and test errors [29].

A. Learning Target-Specific Hypotheses G under Source-Specific Prior Q

We want to learn new weak classifiers that complements Q. Then these weak classifiers can be boosted to a strong classifier G using any of the AdaBoost family of algorithms. But the size of the target data is usually small, so the learned weak classifiers, without other constraints, might overfit the data. We use manifold regularization to address the overfitting problem, i.e, under some appropriate conditions, we try to minimize regularization loss

$$R_{\text{loss}}(G) = \sum_{j_1, j_2} S(x_{j_1}^T, x_{j_2}^T)\big[G(x_{j_1}^T) - G(x_{j_2}^T)\big]^2, \tag{4}$$

where $S(x_i^T, x_j^T) \in [0, 1]$ is a similarity measure of x_i^T and x_j^T. A common choice of $S(x_i, x_j)$ is K-Nearest Neighbor (K-NN) similarity measure:

$$S(x_i^T, x_j^T) = \begin{cases} 1, & \text{if } x_j^T \in N_K(x_j^T), \\ 0, & \text{else.} \end{cases} \tag{5}$$

Notice that K-NN is not necessary symmetric, i.e., it's possible that $x_i^T \in N_K(x_j^T)$ but $x_j^T \notin N_K(x_i^T)$. Manifold regularization implies that for each neighborhood $N_K(x_i^T)$, $G(N_K(x_i^T))$ should be as consistent as possible.

In each iteration of AdaBoost, a new weak classifier $g(x)$ is learned only to minimize weighted exponential loss:

$$E_{\text{loss}}(g) = \sum_j w_j \exp(-y_j^T g(x_j^T)), \tag{6}$$

where w_j is a normalized weight under the current strong classifier G.

Here, we need to incorporate regularization loss (4) into the learning of weak classifier to avoid overfitting. Every weak classifier corresponds to a feature, be it a Haar-like feature [4] or a granular feature [28]. Given a feature $f(x)$, we can have a partition of data space based on feature values: $\cup_k D_k = \cup_k f^{-1}(B_k)$ where $\cup B_k$ is a partition of a continuous interval B in $\mathbb{R}$. Usually each B_k is a continuous interval and the total number of domains if finite. Then the corresponding weak classifier $g(x)$; given $f(x)$ and B_k), under the framework of real AdaBoost [29], is essentially a mapping of partition domains D_k to some prediction values $\alpha_k \in \mathbb{R}$:

$$\begin{aligned} &g_f(x) = \alpha_k, \text{if } x \in D_k, \\ \Leftrightarrow &g_f(x) = \alpha_k, \text{if } f(x) \in B_k. \end{aligned} \tag{7}$$

According to real AdaBoost, the optimal prediction value α_k is uniquely determined by the current sample weights and the number of positive samples and negative samples within each D_k:

$$\alpha_k = \frac{1}{2} \log \frac{w_k^+ + \epsilon}{w_k^- + \epsilon}, \tag{8}$$

where $w_k^{\pm} = \sum_{x_j^T \in D_k, y_j^T = \pm 1} w_j$ and ϵ is a noise floor, usually on the order of $1/|D^T|$. Since all features are assumed to have the same range $B = [0, 1]$, we can evenly partition it into M intervals. But different features will have different sample space partition, thus are different classifiers. Note that a weak classifier $g(x)$ constructed by (7) from a feature $f(x)$ will not increase the exponential loss with an arbitrary partition, as shown in [29]. In most cases, with proper selection of features, we can reduce the exponential loss at each iteration.

Finer partition generally leads to lower training error, upper bounded by the exponential loss (6), but also tends to overfit. This can be seen from the perspective of the regularization loss (4): Finer part ion will more often split $N_K(x_j^T)$, thus increases regularization loss. So we need to limit the number of intervals M. On the other hand, given M, for each feature $f(x)$, there is a unique weak classifier is determined by (7) and (8). We then try to find out a feature from $\mathcal{S}$ that minimizes a combined loss function:

$$f = \arg\min_{f \in \mathcal{S}} R_{\text{loss}}(G + g_f) + \beta E_{\text{loss}}(g_f), \tag{9}$$

in which β is a parameter to control the relative weighting of the regularization loss and exponential loss. If the feature set $\mathcal{S}$ is small, e.g, the Haar-like feature set used in the Viola-Jones face detection [4], we may exhaustively search the $\mathcal{S}$ and find the optimal feature (since the target data is usually small, this is still practical). If the feature set $\mathcal{S}$ is very large, e.g., the granular feature set [28], we may randomly sample a number of features from $\mathcal{S}$ and find the best feature among the sampled features. The complete target-specific learning algorithm is listed in 1.

B. Fusing Source- and Target-Specific Hypotheses

So far we have learned source and target specific classifier Q and G, now we will need a proper fusion scheme to combine them together. As mentioned above, it is difficult to combine

Algorithm 1: Target-specific real AdaBoost with manifold regularization

Input:
Target data: $D^T = \{(x_1^T, y_1^T), \ldots, (X_m^T, y_m^T)\}$
Source-specific prior: Q
Normalized feature set: $\mathcal{S}$
Number of partitions: M
Number of iterations: N_{iter}
Relative weight coefficient: β
Pre-weight bias coefficient: μ

Initialization:
weight: $w_j^1 = 1/(1 + \exp(-\mu y_j^T Q(x_j^T)))$
hypothesis: $G_1 = Q$

for $t = 1, 2, \ldots, N_{\text{iter}}$ **do**
- Normalization: $w_j^t = w_j^t / \sum_j w_j^t$
- Searching a feature f_t by (9);
- Constructing g_t based on f_t by (7) and (8)
- Weight update: $w_j^{t+1} = w_j^t \exp(-y_j^T g_t(x_j^T))$
- Hypothesis update: $G_{t+1} = G_t + g_t$

end
Output: Target-specific hypothesis $G = G_{T+1}$

G and S as basic elements to approximate F because of the latent nonlinearity. When we take the hypotheses in G and S as basic elements, luckily we can fuse G and S in a more flexible manner.

Let $Q = \sum_{j=1}^n z_i h_i$ and $G = \sum_{i=1}^m w_i g_i$. We now treat z_i and w_i as adaptation parameters, and $h_i(x)$ and $g_i(x)$ as features. Obviously this hypotheses-as-element style can provide sufficient nonlinearity in adaptation than classifier-as-element style. It is also noticeable that we only take weight as parameters but keep the hypotheses constant. These hypotheses, especially h_i trained on sufficient x^S, represent underlying data distribution. The few target data x^T is not sufficient to adjust them properly, and any change may be unstable and harmful.

We set $C = \{c_i, i = 1 : m + n\}$ as the adaptation weight, and we set $V = \{v_i, i = 1 : m + n\}$ as the feature values. Obviously $\{v_i\}_{i=1}^{m+n} = \{h_1, .., h_n, g_1, .., g_m\}$. We define the final classifier F as the linear combinations of C and V

$$F = C' \cdot V = \sum_{i=1}^{m+n} c_i v_i(x) \tag{10}$$

Therefore, the fusion problem can be formulated as an "accuracy-regularization" objective from which optimal C is obtained

$$C = \arg\min_C \sum_{x_i^T} e^{-y(x_i^T) C' \cdot V(x_i^T)} + \lambda \|C - C^0\|_2 \tag{11}$$

This objective, denoted as L_f, finds the optimal weight C so that the fused F in (10) can achieve the minimum loss on x^T. Meanwhile, C should not deviate from initial C^0 too much, so that the final fused F can preserve good generalization ability on unseen D^T data. In this case, $C^0 = \{z_1, .., z_n, 0_1, ..0_m\}$ which means the initial F equals

Algorithm 2: Adaptation of Cascade Face Detector

Input:
Cascade face detector: $\mathcal{Q} = \{Q_1, .., Q_L\}$
Target data: $D^T = \{(x_1^T, y_1^T), \ldots, (x_m^T, y_m^T)\}$
Negative sample pool X^N
Layer VR.

Initialization:
$\mathcal{F} = \emptyset$.

for $l = 1, 2, \ldots, L$ **do**
- Use $\mathcal{F}$ to sample negative data from X^N
- Learn G on x^T and x^N by algorithm 1
- Adapt Q_l and G via (11) to generate F_l
- Adjust threshold of F_l to meet layer VR
- Push F_l into $\mathcal{F}$

end
Output: Adapted face detector $\mathcal{F} = \{F_1, ..., F_L\}$.

Q. Therefore, the adaptation process gradually fuses Q and G so that F evolves from Q to the final existence.

We use the Newton-Raphson method to solve (11) by calculating the gradient vector $\triangledown L_f(C)$

$$\frac{\partial L_f}{\partial C_p} = \sum_{x_i^T} (-y V_p) e^{-y C' \cdot V} + 2\lambda (C_p - C_p^0),$$

and Hessian matrix $H_{L_f}(C)$

$$\frac{\partial^2 L_f}{\partial C_p \partial C_q} = \sum_{x_i^T} V_p V_q e^{-y C' \cdot V} + 2\lambda I.$$

So C can be iteratively updated as

$$C^{t+1} = C^t - H_{L_f}^{-1}(C^t) \cdot \triangledown L_f(C^t).$$

The iteration may stop when the Newton decrement is smaller than a certain threshold. For simplicity, in practice we turn to a simpler stopping criteria which stops iteration when the change in C is smaller than a threshold ξ

$$\frac{1}{m+n} \sum_{k=1}^{m+n} \|C^{t+1}(k) - C^t(k)\|_2 < \xi.$$

C. Cascade Adaptation

The above explanation only deals with a single strong classifier, yet an applicable face detector adopts cascade structure to achieve high accuracy and real-time speed. Here we extend the above adaptation process into cascade structure, as is listed in Algorithm 2.

In Algorithm 2, a pre-trained cascade face detector Q is given, as well as few target domain data x^T and a negative sample pool x^N. The negative sample pool may contain hundreds or thousands of images without human face, and during adaptation negative data will be extracted as image patches from it. Then the target specific G will be learned, and adapted F_l will be generated by fusing G and Q_l. We also expect the verification rate (VR) of each layer to be determined, so that the layer's threshold can be adjusted. We do

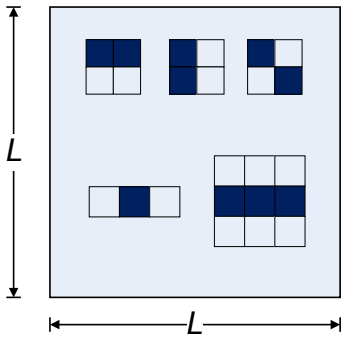

Fig. 2. Examples of the Haar-like features.

not demand too much on false acceptance rate (FAR), however, due to the algorithm structure. Data in target domain may come from a specific real practice, so that the false positive problem is not as serious as a general-purposed detector. We will show more details in the following experiments.

IV. Experiments

We trained a general purpose face detector as the source domain detector, which contains 14 strong hypotheses (layers) and 1257 weak hypotheses. The positive training set contains more than $15,000$ gray images (out of total $16,000$, the rest was used for cross check of the fused detector) of front faces, size of 20×20. The negative training set is randomly sampled from 1000 photos (larger than 800×600) without faces. We use the Haar-like features (Fig. 2). Each weak hypothesis is constructed from one of the Haar-like features by a confidence lookup table with 32 entries (8). Real AdaBoost [29], which generally has higher performance and faster convergence rate than the discrete AdaBoost, is used to train strong hypotheses from the weak hypotheses. Over the source training set, each strong hypothesis (layer) maintains a verification rate no less than 99.7% and false alarm rate no more than 30%. The 18 strong hypotheses are cascaded together as the general face detector.

Three types of target face images were collected: 1) partially covered; 2) wearing glasses; and 3) under extreme illumination. For each of the three types, there are 500 samples, totally 1500 samples. Some of these samples (randomly ordered) are shown in Fig. 3.

Using the general purpose face detector, a large number of faces in the test data are missed, mainly due to the lacking of similar faces in the source training samples (Fig. 4). To adapt this detector to the target data, we feed a few missed faces of each type into Algorithm 1 to train target-specific hypotheses and then use 2 to fuse the source detector with the target-specific detector. In training target-specific detectors,the pre-weight coefficient μ is adjusted to have the sum of pre-weights of verified faces equal to the sum of missed faces. Different relative weight β are tried with $k = 1$ for the K-NN similarity measure (5). It was found that optimal β depends mainly on target training set size. The feature set is still the Haar-like feature set, with the same real AdaBoost with confidence lookup tables of the same size 32 as in the training of source domain hypotheses. In fusing the target-specific detectors and the source-specific detectors, a range of the weight λ in (11)

Fig. 3. Collected face samples that are partially covered, wearing glasses, or under extreme illumination.

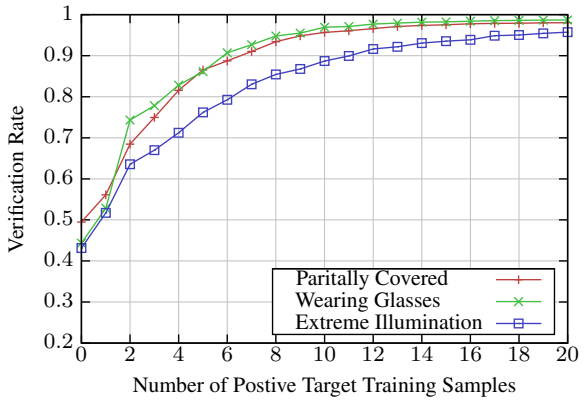

Fig. 4. Detection rates of the fused detector on the three types of target data: With no more than 20 missed faces as target positive training set, almost all missed face in the target data are detected by the fused detector for all the three types.

were tested over different target training set size and optimal ones are chosen. This new face detector combines the power of general purpose detector and target-specific detector. On the target data, verification rate for all three types of faces are significantly improved, compared to the general face detector. Moreover, the number of new train samples need not be very large: Usually fewer than 20 missed faces are able to achieve very high verification rate.

We cross checked the fused face detector on source data. The test data contains $1,000$ gray images of faces and $100,000$ randomly cropped non-face gray images. For appropriate β and λ, the fused face detector (trained with 20 missed partially covered faces in the target domain), maintains similar verification rate as the general purpose detector, and is significantly higher the target-specific detector (Fig. 5). This demonstrates that the new face detector, by fusing source- and target-specific hypotheses, reaches the higher performance of the two with only limited manual intervention. On the other hand, some combinations of β and λ will drag down verification rate significantly, also shown in Fig. 5. Therefore, we need explore various combinations of β and λ to have good performance at both the source and target data.

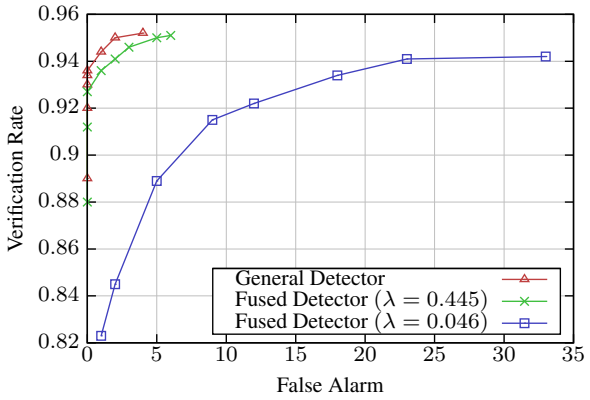

Fig. 5. ROC of the general detector and fused detectors on source domain data ($\beta = 1.891$).

Face detection, due to its cascaded structure, is fast enough for real time processing even on embedded systems. Target domain adaptation is more complex. But on smart cameras, especially those for surveillance running 24 hours a day, we may use low load time to do target domain adaptation.

V. Conclusions

In a smart camera network, target domain adaption can improve face detection performance, but requires limited target data sharing among nodes. We have shown that, sharing only a few dozens of target face data without any source domain data, the performance of a general face detector can be improved significantly by fusing source specific hypotheses and target specific hypotheses, Thus our method can be applied to a smart camera network with P2P communications to effectively addresses a typical problem of real-world face detection: Faces in the field, e.g., covered by scarves, with glasses, or under extreme illumination, may have statistical differences to the general training faces, thus drag down detection rates. Two key ingredients to our methods are "manifold regularization," which avoids overfitting the small set of the annotated target faces by the target specific hypotheses, and "accuracy-regularization," which combines the effectiveness of the source and target specific hypotheses. In the future, we will study how to extend this method to other object detection tasks, e.g., vehicle detection and pedestrian detection in a smart camera network.

References

[1] M. Wittke, M. Hoffmann, J. Hahner, and C. Muller-Schloer, "Midsca: Towards a smart camera architecture of mobile internet devices," in *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, Sept 2008, pp. 1–10.

[2] Y. Wang, S. Velipasalar, and M. Casares, "Detection of composite events spanning multiple camera views with wireless embedded smart cameras," in *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, Aug 2009, pp. 1–8.

[3] ——, "Cooperative object tracking and composite event detection with wireless embedded smart cameras," *Image Processing, IEEE Transactions on*, vol. 19, no. 10, pp. 2614–2633, Oct 2010.

[4] P. Viola and M. Jones, "Robust real-time face detection," in *Intl. Journal of Computer Vision*, 2004.

[5] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Intl. Conf. on Image Processing*, 2002.

[6] C. Huang, H. Ai, Y. Li, and S. Lao, "Learning sparse features in granular space for multi-view face detection," in *Intl. Conf. on Automatic Face and Gesture Recognition*, 2006.

[7] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block lbp representation," in *IAPR/IEEE Conf. on Biometrics*, 2007.

[8] B. Wu, H. Ai, C. Huang, and S. Lao, "Fast rotation invariant multi-view face detection based on real adaboost," in *Intl. Conf. on Automatic Face and Gesture Recognition*, 2004.

[9] C. Huang, H. Ai, Y. Li, and S. Lao, "Vector boosting for rotation invariant multi-view face detection," in *Intl. Conf. on Computer Vision*, 2005.

[10] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg, "Fast asymmetric learning for cascade face detection," in *IEEE Trans on PAMI*, 2008.

[11] P. Viola, J. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *Advances in Neural Information Processing Systems*, 2006.

[12] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant mutiview face detection," in *IEEE Trans. on PAMI*, 2007.

[13] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," in *MSR-TR-2010-66*, 2010.

[14] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[15] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 759–766.

[16] H. Daumé III, "Frustratingly easy domain adaptation," in *ACL*, vol. 1785, no. 1786, 2007, p. 1787.

[17] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the 2006 conference on empirical methods in natural language processing*, 2006, pp. 120–128.

[18] H. Daumé III and D. Marcu, "Domain adaptation for statistical classifiers." *J. Artif. Intell. Res.(JAIR)*, vol. 26, pp. 101–126, 2006.

[19] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representation for domain adaptation," in *Advances in Neural Information Processing Systems*, 2006.

[20] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Conf. Empirical Methods in Natural Language Processing*, 2006.

[21] H. D. III and D. Marcu, "Domain adaptation for statistical classifiers," *Journal of Artificial Intelligence Research*, 2006.

[22] S. J. Pan and Q. Yang, "A survey on transfer learning," in *IEEE Trans on Knowledge and Data Engineering*, 2009.

[23] J. Jiang, "A literature survey on domain adaptation of statistical classifiers," Tech. Rep., 2008. [Online]. Available: http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey

[24] C. Huang, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Incremental learning of boosted face detector," in *Intl. Conf. on Computer Vision*, 2007.

[25] C. Zhang, R. Hamid, and Z. Zhang, "Taylor expansion based classifier adaptation: Application to person detection," in *Intl. Conf. on Computer Vision and Pattern Recognition*, 2008.

[26] G. Chen, T. X. Han, and S. Lao, "Adapting an object detector by considering the worst case: a conservative approach," in *Intl. Conf. on Computer Vision and Pattern Recognition*, 2011.

[27] V. Jain and E. Learned-Miller, "Online domain adaptation of a pre-trained cascade of classifiers," in *Intl. Conf. on Computer Vision and Pattern Recognition*, 2011.

[28] C. Huang, H. Ai, Y. Li, and S. Lao, "Learning sparse features in granular space for multi-view face detection," in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*. IEEE, 2006, pp. 401–406.

[29] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine learning*, vol. 37, no. 3, pp. 297–336, 1999.