Boosted Multi-Task Learning for Face Verification With Applications to Web Image and Video Search

Xiaogang Wang[2] Cha Zhang[1] Zhengyou Zhang[1]
[1]Microsoft Research, One Microsoft Way, Redmond, WA 98052
[2]Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139

Abstract

Face verification has many potential applications including filtering and ranking image/video search results on celebrities. Since these images/videos are taken under uncontrolled environments, the problem is very challenging due to dramatic lighting and pose variations, low resolutions, compression artifacts, etc. In addition, the available number of training images for each celebrity may be limited, hence learning individual classifiers for each person may cause overfitting. In this paper, we propose two ideas to meet the above challenges. First, we propose to use individual bins, instead of whole histograms, of Local Binary Patterns (LBP) as features for learning, which yields significant performance improvements and computation reduction in our experiments. Second, we present a novel Multi-Task Learning (MTL) framework, called Boosted MTL*, for face verification with limited training data. It jointly learns classifiers for multiple people by sharing a few boosting classifiers in order to avoid overfitting. The effectiveness of Boosted MTL and LBP bin features is verified with a large number of celebrity images/videos from the web.*

1. Introduction

One of the most frequent types of image/video search queries on the web are about people (celebrities). Current search engines mainly rely on the text information nearby the images/videos for such tasks. Although the top few returned examples are often satisfactory, the precision drops quickly when more lower ranked examples are included. Apparently, text information alone is not sufficient to achieve very high accuracy. For instance, if "Eva Longoria" is used as the keyword to query from YouTube, the returned video ranked as No. 9 actually shows how to fake a bob with the hair style of Eva Longoria, where Eva does not show up at all. In addition, for celebrity videos, it would be helpful to know when the celebrity appears in a video and the frequency of appearance, which is an indicator of how much a video is relevant. For example, people may be more interested in an MTV of Eva Longoria than a video of a long news program where she just shows up once. In these scenarios, face recognition technologies may greatly improve the search results.

Figure 1. Examples of faces of celebrities from our data sets for experiments. (a) Face images of Bill Gates. (b) Faces of Bill Gates from videos. (c) Faces images of three other celebrities. Images in the same row belong to the same celebrity.

In this paper, we address the following problem: Given a small set of face images of a celebrity (e.g., the top query results from a text-based search engine), verify whether the celebrity is in other images and videos that are returned by the search engine. In the past, face recognition has achieved significant progress under controlled conditions. However,

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

web-based face verification is a much harder problem since the web is an open environment [6, 16] where pose, lighting, expression, age and makeup variations are more complicated. In addition, face images and videos on the web often have very low resolution and exhibit severe compression artifacts. Some examples are shown in Figure 1.

Our contribution is two-fold. First, we propose a framework for face verification using Local Binary Pattern (LBP) [11] features and boosting classifiers. LBP-based face recognizer [1] outperformed many existing popular face recognition approaches such as Eigenface [15], Fisherface [2], Bayesian face [10] and Gabor features [17] on public databases due to its high discrimination power and robustness to lighting and misalignment. We show that by selecting discriminative features from a large set of LBP features using boosting, the verification accuracy can be further improved. Moreover, these LBP features can be computed very efficiently through early rejection, which results in significant computation reduction compared with [22].

Second, a boosted Multi-Task Learning (Boosted MTL) algorithm is presented. Since the positive training examples are often retrieved from top query results from text-based search engines, it is necessary to limit the size of the training data set. Consequently, a typical machine learning algorithm may easily overfit on the training data set. We propose to use Boosted MTL to address the overfitting issue. In our approach, K boosting classifiers are learned *jointly* for M celebrities, where $M \geq K$. Every celebrity then composes his/her own combined classifier, which is a mixture of the K classifiers. The key observation is that celebrities who have commonality in their features should be explained by the same boosting classifiers, while those who do not have commonality should have a different mixture of the K boosting classifiers. Boosted MTL effectively overcomes overfitting and achieves better results than training individual classifiers for each person or training a single classifier that simply tells if two images are from the same person or not, as shown in our experimental results.

2. Related work

Web-based face recognition is an emerging research topic. Stone et al. [13] utilized the social network context provided by Facebook to autotag personal photographs on the web. Yagnik and Islam [19] used the text-image co-occurrence as a weak signal to learn a set of consistent face models from a very large and noisy training set of face images of celebrities on the web. Face images of a celebrity are clustered, and the outliers are removed. This consistency learning framework requires a large initial data set and high computational cost. In contrast, our work assumes a small and noise-free training set for face verification.

Most face recognition work was evaluated on databases collected in controlled environments where the variations of poses, lightings and expressions are relatively limited. In recent years, there has been an increasing interest in studying face recognition in unconstrained environments [16, 6, 18, 5]. Huang and Learned-Miller et al. built a database, Labeled Faces in the Wild (LFW), which contains more than $13,000$ images of faces from web [7]. It was shown that the performance of many existing face recognition approaches dropped significantly in such unconstrained environments.

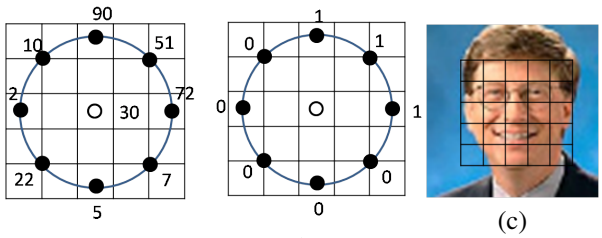
Boosting was applied to face recognition using PCA, LDA and Gabor features to build weak classifiers [4, 8, 20]. The work most relevant to ours is [22], where histograms of LBP inside local regions are used to build the weak classifiers in an AdaBoost framework. Although it was better than many other features, the improvement compared with using LBP directly in [1] is marginal. In our work, counts of bins instead of local histograms are used to build weak classifiers. Compared with [22], we have a much larger pool of more independent features, and the weak classifiers are simpler. Our experimental results show that this leads to a surprisingly big improvement on face verification accuracy and makes verification much faster compared with [22].

Most existing boosting approaches for face recognition train classifiers for different individuals separately [4, 8] or a single classifier applying to all the faces [10, 20, 22]. The first approach may easily lead to overfitting, since our training examples for each individual are limited. The second approach fails to consider variations among different celebrities, and therefore its performance is suboptimal. In the past, people proposed different schemes such as feature sharing [14] for combatting overfitting issues. In this paper, we propose a novel algorithm called *Boosted MTL*, which is rooted from multi-task learning [12]. Multi-task learning is a machine learning approach that learns a problem together with other related problems at the same time using a shared representation. It often leads to a better model for a single task than learning it independently, because it allows the learner to use the commonality among tasks. To our best knowledge, our proposed algorithm is the first boosting framework that inherently performs multi-task learning.

3. LBP Features for Boosting

3.1. Introduction of LBP

LBP is a powerful texture descriptor introduced by Ojala et al. [11]. As shown in Figure 2 (a), it defines the neighborhood of a pixel i by uniformly sampling P pixels along the circle centered at i with radius R. If the pixels in the neighborhood do not fall exactly on the grid, their values are estimated using bilinear interpolation. The pixels in the neighborhood are assigned with binary numbers by thresholding against the value of pixel i, as shown in Figure 2 (b). These binary numbers are then assembled into a decimal number, which is used as the label or the local binary



(a) (b) (c)

Figure 2. LBP operator. See details in text.

pattern of pixel i. For instance, in Figure 2 (b), the centered pixel will be labeled as $11100000 = 224$. A local binary pattern is called *uniform* if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. For example, 00000000, 00011110 and 10000011 are uniform patterns, and 00010100 is not.

One extension of LBP is to keep only uniform patterns and map all the non-uniform patterns to a single label. As observed from the experiments in [11], uniform patterns appear much more frequently than non-uniform patterns. It has been shown that such an extension of LBP can increase the robustness to noise and improve the computational efficiency (since the size of the codebook is significantly reduced). In some sense, uniform patterns characterize edges with particular directions, length and scales. In the following, we denote the operator of uniform patterns as $LBP^{u2}_{P,R}$.

In [1], the face region is first divided into local regions, as shown in Figure 2 (c). The histograms of uniform patterns inside the local regions are used as features for face recognition. This approach outperformed many popular face recognition approaches. LBP has several advantages for face recognition. First, it has high discrimination power by characterizing a very large set of edges. If there are P pixels uniformly sampled along the circle, there will be $3P(P-1)$ uniform patterns. When choosing different P and radius R, different uniform patterns can be computed. Second, LBP is more robust to lighting variations. Local binary patterns do not change, if the values of centered pixels and their neighborhoods are under the same monotonous transformation. Third, since histograms are used as features, they are more robust to misalignment and pose variations.

The approach proposed in [1] simply added the distances of local histograms. It included all the uniform patterns, some of which are redundant and may deteriorate the accuracy. The computational efficiency is also low. In the following, we propose to use boosting to select a small set of uniform patterns best for face verification.

3.2. LBP for Boosting

We apply the well-known AdaBoost algorithm [3] for face verification, as shown in Algorithm 1. Given a set of positive examples $A = \{x^a_1, \ldots, x^a_{N_a}\}$ of celebrity C and a set of negative examples $B = \{x^b_1, \ldots, x^b_{N_b}\}$ of other celebrities excluding C, we learn a similarity score $h(x_1, x_2)$, which is large if both x_1 and x_2 belong to C, and small otherwise. From A and B, we compose a training data set, where positive examples are pairs of examples that are both from set A, i.e., $\{(x^a_i, x^a_j)\}$, and negative examples are pairs of examples that are from both set A and B, i.e., $\{(x^a_i, x^b_j)\}$. The similarity between a testing image x and A, computed as

$$h_{\max} = \max_{x^a_i \in A} h(x^a_i, x), \tag{1}$$

Algorithm 1 Adaboost Learning with LBP Features

1: **Input** an image training set $A = \{x^a_1, \ldots, x^a_{N_a}\}$ of celebrity C, an image training set $B = \{x^b_1, \ldots, x^b_{N_b}\}$ of other celebrities excluding C, feature candidates $F = \{f_1, \ldots, f_L\}$, and distance measure of features $d(f(x_1), f(x_2))$.

2: **Output** $h(x_1, x_2)$, a similarity score of whether the two images x_1 and x_2 are from celebrity C or not.

3: Initialize weights $w^a_{1,i,j} = \frac{2}{N_a(N_a-1)}$ with $1 \le i \le j < N_a$, and $w^b_{1,i,j} = \frac{1}{N_a N_b}$ with $1 \le i \le N_a, 1 \le i \le N_b$.

4: **for** t = 1, ..., T **do**

5: Normalize the weights,

$$w^a_{t,i,j} \leftarrow \frac{w^a_{t,i,j}}{\sum_{i',j'} w^a_{t,i',j'} + \sum_{i',j'} w^b_{t,i',j'}}$$

$$w^b_{t,i,j} \leftarrow \frac{w^b_{t,i,j}}{\sum_{i',j'} w^a_{t,i',j'} + \sum_{i',j'} w^b_{t,i',j'}}$$

6: For each f_k train a threshold ρ_k, minimizing error,

$$\epsilon_k = \frac{1}{2}\Big\{ \sum_{i,j} w^a_{t,i,j} \left(1 - S\left(d\left(f_k\left(x^a_i\right), f_k\left(x^a_j\right)\right), \rho_k\right)\right) + \sum_{i,j} w^b_{t,i,j} \left(1 + S\left(d\left(f_k\left(x^a_i\right), f_k\left(x^b_j\right)\right), \rho_k\right)\right) \Big\},$$

where $S(z, \rho) = 1$ if $z \le \rho$ and -1, otherwise.

7: Choose the feature f_t with the lowest error ϵ_t.

8: Update the weights:

$$w^a_{t+1,i,j} = w^a_{t,i,j} \beta_t^{1-e^a_{i,j}}, w^b_{t+1,i,j} = w^b_{t,i,j} \beta_t^{1-e^b_{i,j}}$$

where $e^a_{i,j} = 0$ if $d\left(f_t\left(x^a_i\right), f_t\left(x^a_j\right)\right) \le \rho_t$, and 1, otherwise; $e^b_{i,j} = 0$ if $d\left(f_t\left(x^a_i\right), f_t\left(x^b_j\right)\right) > \rho_t$, and 1, otherwise; and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

9: **end for**

10: $h(x_1, x_2) = \sum_{t=1}^{T} \alpha_t S\left(d\left(f_t\left(x_1\right), f_t\left(x_2\right)\right), \rho_t\right)$, where $\alpha_t = log \frac{1}{\beta_t}$.

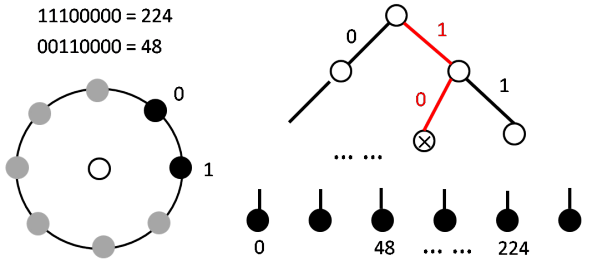


Figure 3. An example of fast computation of LBP features. For a local region, only two local binary patterns 11100000 and 000110000 are selected by Adaboosting. For a pixel inside this region, after estimating the values of the first two pixels, which have binary number '10', in its neighborhood, we can decide that this pixel does not belong to any of the two local binary patterns and stop estimating other pixels in its neighborhood. This procedure can be easily implemented by a binary search try on the right.

is used as a measure to verify whether x belongs to celebrity C or not. In other words, we compare the test image with all the example images of celebrity C and take the highest score to make the decision.

In Algorithm 1, $F = \{f_1, \ldots, f_L\}$ is a large pool of candidate features. $f_l = LBP^{u2}_{P,R}(E, k)$ is the count of the k^{th} bin of the histogram of uniform patterns inside local region E. The features in F are computed using different LBP operators by choosing different P and R values, and using different sizes of local regions. For instance, in the experiments presented in Section 5, by varying P, R and region sizes, a pool of $38,150$ LBP bin features can be constructed, among which a few hundred will be selected by the AdaBoost learning algorithm. Note in our implementation, the distance between two bin features is computed as

$$d(f(x_1), f(x_2)) = \frac{(f(x_1) - f(x_2))^2}{f(x_1) + f(x_2)}. \quad (2)$$

3.3. Fast Computation of LBP Bin Features

Efficiency is very important for web based face verification, especially when handling videos. Our LBP bin feature based AdaBoost algorithm dramatically improves the efficiency at the verification stage compared with the approaches in [1] and [22] in two aspects. First, computing the distance between histograms as in [1, 22] requires to add the differences of all the bins. In the experiments presented in Section 5, there are in total $38,150$ bins for all the local histograms. Even after selecting some of the histograms as in [22], the number of bins is still large. Our algorithm only needs to compute and sum the difference of a few hundred bins, which are far fewer than the total number of bins.

Furthermore, the LBP bin features can be computed very efficiently through early termination. Note that the most time consuming part of computing LBP features is to estimate the values of pixels on the circle using bilinear interpolation. The bins selected by our AdaBoost algorithm distribute among many different local regions. On average, we only need to compute the counts of one or two bins inside a local region. Instead of labeling every pixel with a local binary pattern, we only need to verify whether a pixel belongs to the local binary patterns of interest or not. Many pixels will be discarded after examining the first few pixels in their neighborhood without requiring to estimate all the pixels in the neighborhood pixels. This procedure can be easily implemented using a binary tree. An example is shown in Figure 3. Our approach can speed up the computation of LBP bin features by three to four times.

4. Boosted Multi-Task Learning

4.1. Boosted Multi-Task Learning

As mentioned earlier, we assume that for each celebrity, a small number of training examples are available for learning. If individual classifiers are learned for each celebrity, overfitting is inevitable. An alternative approach is to train a generic classifier which classifies that any two examples are from the same person or not. A positive training set $\{\xi_i^+ = (x_{i_1}, x_{i_2})\}$, in which (x_{i_1}, x_{i_2}) are a pair of examples from the same person, and a negative set $\{\xi_i^- = (x_{i_1}, x_{i_2})\}$, where two examples in a pair are from two different persons, are built to training a binary classifier. This classifier is used to verify any person. Many approaches such as Bayesianface [10] and AdaBoost face recognition in [20, 22] used this scheme. In certain scenarios, this approach can effectively reduce the chance of overfitting, since the positive and negative training sets can be very large. However, since only a single classifier is built to recognize all the faces, the recognition performance is usually not satisfactory, as shown in our experiments in Section 5. In this paper, we propose a novel algorithm we call Boosted Multi-Task Learning (MTL) to solve these problems.

Multi-task learning [12] is a machine learning approach that learns a problem together with other related problems at the same time using a shared representation. It often leads to a better model for a single task than learning it independently, because it allows the learner to use the commonality among tasks. In our case, the tasks are the verification of multiple celebrities. Assuming there are M celebrities to be verified. A celebrity m has N_{a_m} training examples $A_m = \{x_1^{a_m}, \ldots, x_{N_{a_m}}^{a_m}\}$. There is another set $B = \{x_n^b, \ldots, x_{N_b}^b\}$ which includes training examples of other people excluding these M celebrities. For each celebrity m, a training set $\{(\xi_{m,n}, y_{m,n})\}$ is built, where $\xi_{m,n} = (x_{m,n,1}, x_{m,n,2})$ is a pair of image examples, $y_{m,n} = 1$ if both $x_{m,n,1}$ and $x_{m,n,2}$ are in A_m, and $y_{m,n} = 0$ if $x_{m,n,1} \in A_m$ and $x_{m,n,2} \in B$ [1].

[1] We can choose $x_{m,n,2} \in \bigcup_{l \neq m} A_l \bigcup B$ to expand the training set.

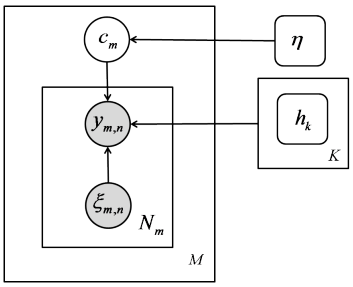


Figure 4. The graphical model of Boosted Multi-Task Learning.

We use a graphical model to represent the structure of Boosted MTL, as shown in Figure 4. In our model, there are K boosting classifiers $\{h_k\}$ to be learned, where $K \leq M$. The boosting classifiers are in the following form:

$$h_k(\xi_{m,n}) = \sum_{t=1}^{T} \alpha_{k,t} h_{k,t}(\xi_{m,n}), \tag{3}$$

$$h_{k,t}(\xi_{m,n}) = S(d\left(f_{k,t}(x_{m,n,1}, x_{m,n,2})\right), \rho_{k,t}), \tag{4}$$

where $S(z,\rho) = 1$ if $z < \rho$ and -1, otherwise. η is a multinomial hyperparameter. For a given celebrity m, the model samples a boosting classifier indexed as $c_m \in \{1, \cdots, K\}$ based on the conditional probability $p(c_m|\eta)$, and uses h_{c_m} to predict $y_{m,n}$ given $\xi_{m,n}$. The joint distribution is

$$\begin{aligned} & p(\{y_{m,n}\}|\{\xi_{m,n}\}, \{h_k\}, \eta) \\ = & \prod_m \sum_{c_m} \prod_n p(y_{m,n}|\xi_{m,n}, h_{c_m}) p(c_m|\eta) \end{aligned} \tag{5}$$

where

$$p(y|\xi, h_k) = \left(\frac{1}{1 + exp\left(-h_k(\xi)\right)}\right)^y \left(\frac{exp\left(-h_k(\xi)\right)}{1 + exp\left(-h_k(\xi)\right)}\right)^{1-y} \tag{6}$$

We use an EM algorithm to learn $\{h_k\}$ and η.

E-step:

$$\begin{aligned} q_{m,k}^{(t)} &= p(c_m = k|\{y_{m,n}\}, \{\xi_{m,n}\}, \{h_k^{(t)}\}, \eta^{(t)}) \\ &= \frac{\eta_k^{(t)} \prod_n P(y_{m,n}|\xi_{m,n}, h_k^{(t)})}{\sum_{k'=1}^{K} \eta_{k'}^{(t)} \prod_n p(y_{m,n}|\xi_{m,n}, h_{k'}^{(t)})}. \end{aligned} \tag{7}$$

M-step:

$$\eta_k^{(t+1)} \propto \sum_m q_{m,k}^{(t)}. \tag{8}$$

$$h_k^{(t+1)} = \arg\max_{h_k} \sum_{m,n} q_{m,k}^{(t)} \log\left[p(y_{m,n}|\xi_{m,n}, h_k)\right] . \tag{9}$$

To solve Eq. (9), the object function of boosting is

$$\begin{aligned} C_k^{(t+1)} = \sum_{m,n} q_{m,k}^{(t)} \Bigg[& y_{m,n} \log \frac{1}{1 + \exp\left(-h_k(\xi_{m,n})\right)} \\ & + (1 - y_{m,n}) \log \frac{\exp\left(-h_k(\xi_{m,n})\right)}{1 + \exp\left(-h_k(\xi_{m,n})\right)} \Bigg] \end{aligned} \tag{10}$$

Let $h_k^{(t+1)} = h_k^{(t)} + \alpha_{k,t+1} h_{k,t+1}$. Following the AnyBoost approach [9], the weight on each example is given as the derivative of the cost function with respect to a change in the score of the example. Then, if $y_{m,n} = 1$,

$$\frac{\partial C_k^{(t+1)}}{\partial h_k(\xi_{m,n})} = w_{k,m,n} = q_{m,k}^{(t)} \frac{\exp\left(-h_k^{(t)}(\xi_{m,n})\right)}{1 + \exp\left(-h_k^{(t)}(\xi_{m,n})\right)}. \tag{11}$$

If $y_{m,n} = 0$,

$$\frac{\partial C_k^{(t+1)}}{\partial h_k(\xi_{m,n})} = w_{k,m,n} = -q_{m,k}^{(t)} \frac{1}{1 + \exp\left(-h_k^{(t)}(\xi_{m,n})\right)}. \tag{12}$$

We find $h_{k,t+1}$ by maximizing $\sum_{m,n} w_{k,m,n} h_{k,t+1}(\xi_{m,n})$ and $\alpha_{k,t+1}$ by maximizing $C_k^{(t+1)}$.

After $\{h_k\}$ and η have been learnt by EM, the classifier of celebrity m is given by

$$\begin{aligned} & p(y_{new}|\xi_{new}, \{(\xi_{m,n}, y_{m,n})\}, \{h_k\}, \eta) \\ = & \sum_{k=1}^{K} p(y_{new}|\xi_{new}, h_k) p(h_k|\{(\xi_{m,n}, y_{m,n})\}, \{h_k\}, \eta) \\ = & \sum_{k=1}^{K} q_{m,k} p(y_{new}|\xi_{new}, h_k). \end{aligned} \tag{13}$$

The algorithm is summarized in Algorithm 2.

4.2. Discussions

In Boosted MTL, celebrities that have commonality in feature selection are clustered and share training data. The posterior $q_{m,k}$ indicates how well a boosting classifier h_k fits the training data of celebrity m. From Eq. (10), (11) and (12), if h_k cannot explain the training data of celebrity m well, the training data of m has less contribution to the learning of h_k since the weight of each example is multiplied by $q_{m,k}$. Instead of training M boosting classifiers for M, in our approach only K boosting classifiers are learnt, so it is less likely to overfit. On the other hand, as shown in Eq. (13), the training data of each celebrity can be well explained by properly linearly combining the K boosting classifiers. Instead of requiring all the celebrities to share training data as in training a single generic boosting classifier, in Boosted MTL, a set of celebrities share training data only when their training data can be well explained by the same boosting classifier. If K is smaller, the trained classifiers are less likely to overfit. We can choose the smallest K which leads to the accuracy on the training data above an expected threshold. Thus Boosted MTL provides a way to maximize the generalization ability while guaranteeing certain accuracy on the training data.

Algorithm 2 Boosted Multi-Tasking Learning

Input training sets of M individuals, $\{\Delta_1, \ldots, \Delta_M\}$ where $\Delta_m = \{(\xi_{m,n}, y_{m,n})\}$, candidate features $F = \{f_1, \ldots, f_L\}$, and the number of components K in the mixture.
Output M binary classifiers of m celebrities. A classifier classifies whether two examples x_1 and x_2 are from the same celebrity m or not.
Randomly assign a value from 1 to K to c_m. $q_{m,k} = 1$ if $c_m = k$, and 0, otherwise.
Initialize weights. If c_m=k, $w_{k,m,n} = \frac{1}{N_m^+}$ if $y_{m,n}$=1, and $-\frac{1}{N_m^-}$, otherwise. N_m^+ and N_m^- are the numbers of $y_{m,n}$=1 and $y_{m,n}$=0 in Δ_m. If $c_m \neq k$, $w_{k,m,m} = 0$.
for t = 1, ..., T **do**
  for k = 1, ..., K **do**
    normalize weights such that $\sum_{m,n} |w_{k,m,n}| = 1$.
  end for
  for k=1, ..., K **do**
    Search for a best feature $f_{k,t}$ and its optimal threshold $\rho_{k,t}$ to maximize $\sum_{m,n} w_{k,m,n} h_{k,t+1}(\xi_{m,n})$, where $h_{k,t+1}$ is defined in Eq. (4).
    Search for $\alpha_{k,t}$ maximizing $C_k^{(t+1)}$ in Eq. (10).
  end for
  Update η_k using Eq. (8).
  Update $q_{m,k}$ using Eq. (7).
  Update $w_{k,m,n}$ using Eq. (11) and (12).
end for
$\{h_k\}$ are given by Eq. (3).
The classifier of celebrity m is given by Eq. (13).

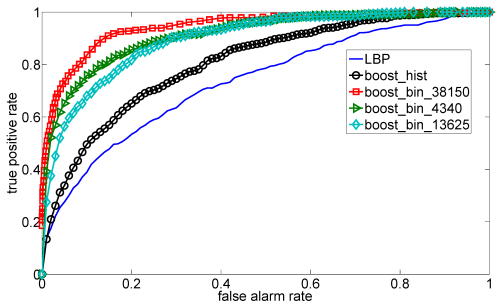


Figure 5. ROC curves of verifying George Bush. *LBP*: directly comparing local histograms of LBP as in [1]. *Boost_bin_38150*: AdaBoosting LBP as described in Section 3.2. The counts of individual bins are used as candidate features. *Boost_hist*: AdaBoosting using local histograms as features as in [22]. All these three approaches above include three types of LBP operators, four difference sizes of local regions, and in total $38,150$ bins of all the local histograms. $Boost\_bin\_4340$ and $Boost\_bin\_13625$ are also AdaBoosting LBP and use counts of individual bins as features. However, $Boost\_bin\_4340$ uses only one type of LBP operator ($LBP^{u2}_{P=8,R=2}$) but four different sizes of local regions. $Boost\_bin\_13625$ uses three types of LBP operators but only one size of local regions (10×10).

Table 1. The true positive rates of face identification on an image set of George Bush when the false alarm rate is fixed at 0.1. The abbreviations of approaches are the same as in Figure 5.

LBP	$boost\_hist$	$boost\_bin\_38150$
0.3910	0.5086	0.8319
$boost\_bin\_4340$	$boost\_bin\_13625$	
0.7610	0.7055	

5. Experimental Results

5.1. LBP Features for Boosting

In the first set of experiments, we compare the proposed LBP bin features with direct LBP based recognition [1] and LBP histogram feature based AdaBoost [22]. Following the notations in Section 3.2, the training data set A has 73 images of George Bush from our database, training set B has 8861 images of other celebrities from the LFW database [7], a positive testing set has 523 images of George Bush from the LFW database, and a negative testing test has 4000 images of other people from the LFW database. After running a face detector [21], face regions are normalized to 50×50 pixels. Features are computed using three types of LBP operators ($LBP^{u2}_{P=8,R=2}$, $LBP^{u2}_{P=16,R=3}$, and $LBP^{u2}_{P=16,R=4}$), and four different sizes of local regions (10×10, 15×15, 20×20, 25×25). Local regions of the same size may have overlap. There are in total $38,150$ candidate features, among which 150 features are selected by the AdaBoost algorithm.

The ROC curves are shown in Figure 5. Table 1 shows the true positive rates when the false alarm rate is fixed as 0.1, which is an intersection of Figure 5. Our LBP bin feature based AdaBoost approach significantly outperforms the approach which compares the distances of local histograms of LBP as described in [1]. It also outperforms the similarly trained AdaBoost classifier based on LBP histograms of local regions [22]. As explained in Section 3.1, local binary patterns characterize edges with different orientation, length and scale. The approach in [22] kept all the edges and only selected local regions best for face verification while our approach selects both edges and regions. Thus our feature pool is much larger and the weak classifiers built from our features are simpler. Experimental results show that when using local histograms as features AdaBoosting can marginally improve the performance compared with directly using LBP, but it is much worse than using counts of individual bins as features. We also compare the performance when features are computed only using a single type of LBP operator (curve $Boost\_bin\_4340$) or only a fixed local region size (curve $Boost\_bin\_13625$). It shows that

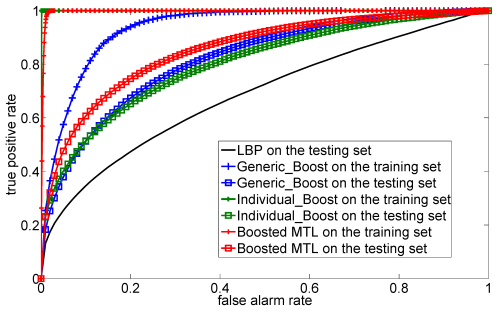


Figure 6. ROC curves of verifying images of 101 celebrities.

the performance is better when both different types of LBP operators and different local region sizes are used.

5.2. Boosted MTL

We next compare the performance of Boosted MTL with two traditionally approaches: training boost classifiers for each individual separately and training a generic boosting classifier for all the celebrities. A total of 101 celebrities are selected for this experiment. The training set has 10 examples of each celebrities from our database, and $8,861$ examples of other people from the LFW database [7]. The testing set has 50 different examples of each celebrity from our database and $4,000$ examples of other people from the LFW database. Some examples are shown in Figure 1 (c). This data set is very challenging since faces of celebrities have very large variations caused by factors such as makeup and aging. Figure 6 and Table 2 shows the performance of four different approaches on both the training data and the testing data: (1) directly using LBP features for verification (LBP as in [1]), (2) training a different boosting classifier for each celebrity separately ($Individual\_Boost$), (3) training a generic boosting classifier to recognize all the celebrities ($Generic\_Boost$), and (4) Boosted MTL with $K = 7$. $Individual\_Boosting$ explains the the training data perfectly but performs poorly on the testing data because of overfitting. $Generic\_Boost$ cannot explain the training data well and its performance is also low on the testing data. Boosted MTL can explain the training data very well and also has better performance than the other three methods on the testing data.

5.3. Face Verification in Videos

We dedicate a short subsection on the topic of face verification in videos here, because we found the problem of face verification in web videos can be more challenging. Faces in videos are often of lower resolutions, with more compression artifacts, with larger pose variations, and under more dramatic lighting conditions than faces in images. Since our training examples are collected from images, there is a mismatch between the training and the testing data set. We

Table 2. The true positive rates of face identification from images of 101 celebrities using three different boosting algorithms when the false alarm rate is fixed at 0.2.

	LBP	*Individual_Boost*
Training	N/A	1.000
Testing	0.3465	0.5174
	Generic_Boost	*Boosted MTL*
Training	0.8067	1.000
Testing	0.5150	0.6098

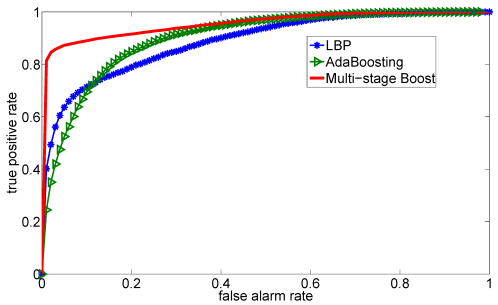


Figure 7. ROC curves of verifying faces from videos given images.

designed a simple multi-stage boosting algorithm to bridge the gap between faces in images and in videos. It first finds a small set of face examples in videos with high confidence and then includes more examples from videos through face tracking, and those face examples from the videos are used to retrain the boosting classifier. The algorithm is summarized as follows.

1. A is a set of training images of a celebrity. B is a negative training set of faces excluding the celebrity. B is easy to get without much labeling effort and can be used to train the classifiers of different celebrities. Q is a set of faces from videos to be verified. Train a boosting classifier using A and B.
2. Identify a small set of a positive examples F from Q using the trained Boosting classifier with a very low false alarm rate τ.
3. Expand F to a larger set E through tracking. E include more variations of poses, resolutions and blurring in videos but not found in the image training set.
4. Retrain the boosting classifier using A, B and E.
5. Repeat step $2 - 4$ for a few times.

In this experiment, there are 63 positive training images of Bill Gates and 12661 negative training examples. The task is to verify the faces of Bill Gates from 15 videos downloaded from YouTube. We manually labeled 263 positive tracks of faces including $8,152$ examples and 606 negative tracks of faces including $18,085$ examples in the testing videos. Some face examples of Bill Gates from images and videos are shown in Figure 1 (a) and (b). Figure 7 shows

the the ROC curves of three approaches: (1) directly comparing LBP features as in [1], (2) LBP bin feature based AdaBoost only using images of Bill Gates for training, and (3) Multi-stage Boosting as described above. The true positive rates and false alarm rates are computed in the units of tracks. A face track is detected as positive if one of its face instances is detected as positive. LBP bin feature based AdaBoost is not much better than LBP because of the mismatch between the training images and the testing videos. Multi-stage Boosting, on the other hand, significantly outperforms the other two approaches.

6. Conclusions and Future Work

We have proposed a framework using LBP bin features and boosting classifiers to verify the faces of celebrities from images and videos on the web. It significantly outperforms the approach that directly uses LBP for face verification in terms of both accuracy and speed. We have also proposed a Boosted Multi-Task Learning algorithm, where classifiers of multiple celebrities are jointly learnt and share training data. To our best knowledge, our proposed algorithm is the first boosting framework that inherently conducts multi-task learning to overcome potential overfitting issues due to the lack of training data.

There are rooms for improvement in the proposed methods. For instance, currently, in order to verify if a given face image is a certain celebrity, the image needs to be compared against a number of positive training examples before the maximum score is used to make a decision. This process can be slow, in particular for video applications. For Boosted MTL, we still do not have a simple scheme to determine the value of K, which is the number of boosting classifiers. Currently we determine it in a trial-and-error manner, and choose the smallest K that yields an accuracy on the training data above a threshold. This could be expensive since the training process typically takes hours to run.

References

[1] T. Ahonen, A. Hadid, and M. Peitikainen. Face recognition with local binary patterns. In *Proc. of ECCV*, 2004.

[2] P. N. Belhumeur, J. Hespanda, and D. Kiregeman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. 19:711–720, 1997.

[3] A. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of European Conference on Computational Learning Theory*, 1995.

[4] G. Guo, H. Zhang, and S. Z. Li. Pairwise face recognition. In *Proc. of ICCV*, 2001.

[5] G. B. Huang, M. J. Jones, and E. Learned-Miller. Lfw results using a combined nowak plus merl recognizer. In *Proc. of ECCV Workshop on Faces in Real-Life Images*, 2008.

[6] G. B. Huang, M. Narayana, and E. Learned-Miller. Towards unconstrained face recognition. In *Proc. of IEEE Computer Society Workshop on Perceptual Organization in Computer Vision IEEE CVPR*, 2008.

[7] G. B. Huang, R. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst, Technical Report 07-49, 2007.

[8] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, and S. Z. Li. Ensemble-based discriminant learning with boosting for face recognition. *IEEE Trans. on Neural Networks*, 17:166–178, 2006.

[9] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *Proc. of NIPS*, 1999.

[10] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. 33:1771–1782, 2000.

[11] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on PAMI*, 24:971–987, 2002.

[12] C. Rich. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[13] Z. Stone, T. Zickler, and T. Darrell. Autotagging facebook: Social network context improves photo annotation. In *Proceedings of CVPR Workshop on Internet Vision*, 2008.

[14] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. on PAMI*, 29:854–869, 2007.

[15] M. Turk and A. Pentland. Eigenface for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[16] R. Verschae, J. Ruiz-del Solar, and M. Correa. Face recognition in unconstrained environments: A comparative study. In *Proc. of ECCV Workshop on Faces in Real-Life Images*, 2008.

[17] L. Wiskott, M. Fellous, N. Krger, and C. Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. on PAMI*, 19:775–779, 1997.

[18] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Proc. of ECCV Workshop on Faces in Real-Life Images*, 2008.

[19] J. Yagnik and A. Islam. Learning people annotation from the web via consistency learning. In *Proc. of the International Workshop on Multimedia Information Retrieval*, 2007.

[20] P. Yang, S. Shan, W. Gao, S. Z. Li, and D. Zhang. Face recognition using ada-boosted gabor features. In *Proc. of International Conference on Face and Gesture Recognition*, 2004.

[21] C. Zhang and P. Viola. Multiple-instance pruning for learning efficient cascade detectors. In *Proc. of NIPS*, 2007.

[22] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu. Boosting local binary pattern (lbp)-based face recognition. In *Advances in biometric person authentication*, 2004.