

Improving Alignment of Faces for Recognition

Md. Kamrul Hasan

Département de génie informatique et génie logiciel
École Polytechnique de Montréal,
Québec, Canada
md-kamrul.hasan@polymtl.ca

Christopher J. Pal

Département de génie informatique et génie logiciel
École Polytechnique de Montréal,
Québec, Canada
christopher.pal@polymtl.ca

Abstract—Face recognition systems for uncontrolled environments often work through an alignment, feature extraction, and recognition pipeline. Effective alignment of faces is thus crucial as can be an entry point in the process and poor alignments can greatly affect recognition performance. The task of alignment is particularly difficult when a face comes from highly unconstrained environments or so called *faces in the wild*. A lot of recent research activity has focused on faces in the wild and even simple similarity or affine transformations have proven both effective and essential to achieving state of the art performance. In this paper we explore a straightforward, fast and effective approach to aligning faces based on detecting facial landmarks using Haar-like image features and a cascade of boosted classifiers. Our approach is reminiscent of widely used face detection approaches, but focused on much more detailed features of a face such eye centres, the nose tip and corners of the mouth. This process generates multiple candidates for each landmark and we present a fast and effective filtering strategy allowing us to find sets of landmarks that are consistent. Our experiments show that this approach can outperform contemporary methods and easily fits into popular processing pipelines for faces in the wild.

I. INTRODUCTION

The human face is an important source of biometric information. Face recognition is a well studied problem with many practical applications including: document control (digital chip in passports, drivers' licenses), transactional authentication (credit cards, ATMs), physical access control (smart doors), law enforcement and police investigations (identifying suspects in security camera video) and security (user access verification). Given a query or test face (from a still image or a video frame) and a database of stored faces, the face recognition problem can be formulated in for two different scenarios:

- Identification: Identify the person (if they are present) within a database of faces, or
- Verification: For a second face image (from inside or outside of a database), decide whether these two faces represent the same person or not.

The experiments we present in Section III focus on verification, the second scenario above; however, our approach should be well suited to improve identification results as well.

In the long history of face recognition research, numerous evaluation benchmarks have been produced - some of the most important ones are compiled in table I. Of particular note is the Face Recognition Vendor Tests (FRVT) [1], which were a set of independent evaluations of commercially available

TABLE I
IMPORTANT FACE DATABASES

| Name | no of subjects and images | Variations |
|-------------------------|---------------------------|--------------|
| FERET | 1199; 14126 | p,il,e,i,o,t |
| AR Face Database | 126; 4000 | il,e,o,t |
| CMU-PIE | 68; 61,368 | p,il,e |
| FRGC Database | 466; 50,000 | il,e,i |
| Caltech 10000 web faces | -; 10524 | natural |
| LFW | 5749; 13,233 | natural |
| PubFig | 200; 59,476 | natural |

p: pose, il: illumination, e: expression, i: indoor, o: outdoor, t: time, -: unknown

TABLE II
THE REDUCTION OF ERROR RATE FOR FERET, AND THE FRVT [1]

| Year of Evaluation | FRR at FAR=0.001 |
|---|------------------|
| 1993 (Turk and Pentland [2](Partially automatic)) | 0.79 |
| 1997 (FERET 1996) | 0.54 |
| 2002 (FRVT 2002) | 0.20 |
| 2006 (FRVT 2006) | 0.01 |

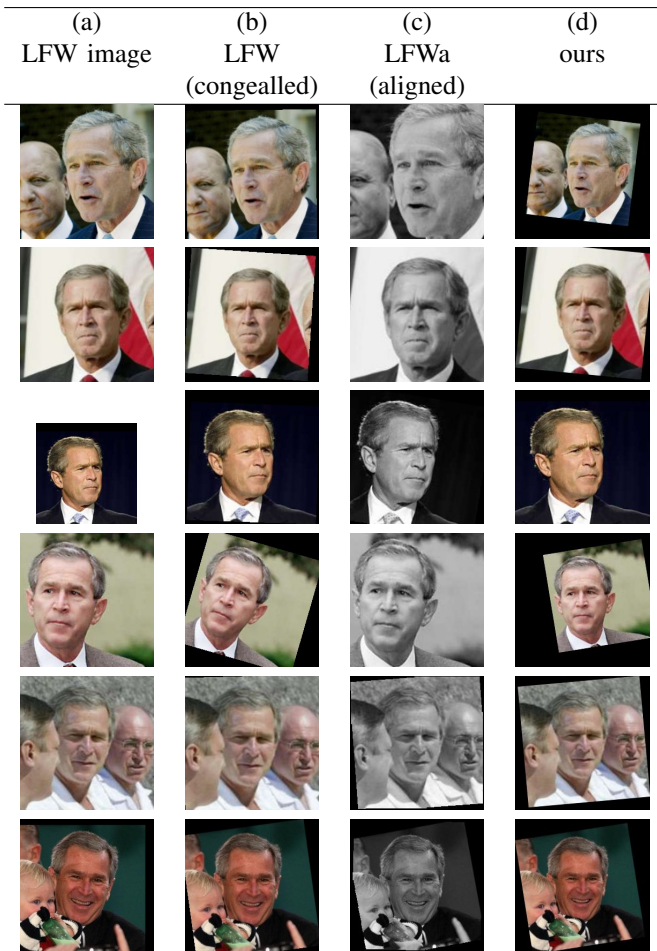
and prototype face recognition technologies conducted by the US government in the years 2000, 2002 and 2006. The FRVT is considered to be a very important milestone in face recognition evaluations; it also includes three previous face recognition evaluations – the FERET evaluations of 1994, 1995 and 1996. Table II quantifies the improvement at four key milestones, where, for each milestone, the FRR (False Rejection Rate) at a FAR (False Acceptance Rate) of 0.001 (1 in 1000) is given for a representative state-of-the-art algorithm [1]. The lowest error rate of 0.01 was achieved by the NV1-norm algorithm (Neven Vision Corporation) on the very high-resolution still images and by V-3D-n algorithm (L-1 Identity Solutions) on the 3D images among. These results were impressive, however, the problem is that the FRVT benchmarks were generated from controlled environments, and hence are limited in their applicability to natural environments.

One might imagine two different strategies for obtaining a more realistic assessment of the performance of a face recognition system: (i) Test the models independently on different subtasks like pose, illumination, expression recognitions, and deduce some collective performance metric as a sum ; or (ii) Build a dataset that accumulates the possible maximal variability of (pose, expression, aging, illumination, race, occlusion etc.) in the same benchmark, and performance evaluations are done on it. Labeled Faces in the Wild (LFW)[3] is a dataset that was constructed from the second view point. Table III(a) shows a set of LFW George W. Bush images from this benchmark. The inherent natural variability of faces can be verified from the examples there.

Generally, a face recognition pipeline for the *in the wild* setting [4], [5], [6] works through an (1) alignment, (2) feature extraction, and (3) recognition pipeline. Table III (b,c) shows the aligned images of table III (a) with two popular face aligner: (b) the congealing and funneling [4], and (c) a fiducial point based commercial aligner (LFWa dataset) by L. Wolf [7]. It can be verified from the LFW results site [8] that the best recognition results were achieved on the aligned dataset, compared to the raw images.

The feature extraction step of a faces in the wild recognition pipeline typically selects (i) the whole image, or (ii) the face detection bounding box area (table IV(a)), or (iii) a smaller central patch cut our of aligned images (table IV (b) and (c)). Features are then computed using one of these underlying representations. Examples of the best face verification results reported so far include Descriptor Based Methods in the Wild (DBMW) [9], Cosine Similarity Metric Learning (CSML)[5], and Probabilistic LDA [6]. Most of these methods use the center patch strategy. Top performing methods on the LFW benchmark have used simple image transformations based on affine or similarity transforms. For example, the LFW benchmark website itself provides faces that have been aligned using Learned-Miller’s congealing algorithm [10]. Most groups posting results to the LFW evaluation have been using faces that have been aligned using congealing as it produces higher performance results. Some recent work such as [7], has replace the congealing based alignment step with a commercial system and other work such as the CSML approach in [5] use this commercially aligned variant of the LFW data known as LFWa. Irrespective of the alignment method it is common to then simply cut out a central rectangle from the aligned image form a rectangular patch used to build feature descriptors. In particular, the DBMW strategy takes LFW-funneled (or congealed) images and crops out a 110×115 pixel from the center of the image. The CSML strategy crops a 80×150 pixel image from the LFWa commercially aligned imagery. Some example patches for the DBWM and CSML setup are shown in table IV (b,c). We also show the face bounding box produced by the OpenCV 2.1 face detector in table IV (a) and some results using our complete method in (c). It is evident

TABLE III
A SET OF LFW ORIGINAL, CONGEALLED, AND ALIGNED IMAGES



























from these examples that the center patches sometime miss important face information; one can think of this as noise manifesting as translation and scaling artifacts. The concrete impact of these errors in the alignment is that the central patch of the image can be poorly centred on the face. In this paper we test our hypothesis that if the alignment step prior to cutting out a central rectangle could be improved, a better feature descriptor could be computed, and thus the recognition performance could be improved. We also explore what happens we simply do a better job of cutting out the final rectangle.

The remainder of this paper is structured as follows. In section II, we will briefly describe our proposed model. Section III is a compilation of some experiments and the corresponding results. Finally, section IV concludes the paper with some future research ideas.

II. OUR APPROACH

Our approach begins by detecting a set of landmarks or fiducial points on a face image using a Viola-Jones style boosted cascade of Haar-like image features. This procedure

TABLE IV
IMAGE PATCHES, USE BY DIFFERENT ALGORITHMS

| (a) fbbox | (b) DBMW | (c) CSML | (d) ours |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

will produce multiple candidates for each landmark. We thus need to filter detections and identifying a set of landmarks that are consistent with a face. We describe our procedure for doing this in more detail below. Once we have a set of consistent landmarks we can transform or register the face into a common coordinate system using a simple affine, similarity or simpler transformation such that when a central patch is cut out of a larger image, this patch is well centred on the face.

When processing new images we use the Viola-Jones face detector found in OpenCV to find faces. In our experiments here multiple face detections are filtered by selecting the largest face. A border of approximately one third the height of

the detected face is then defined and used to crop the face out of the original image into a slightly smaller image. We then search within this smaller image for five facial landmarks : the left eye, the right eye, the nose tip and the two corners of the mouth. To detect these landmarks we have trained five different boosted cascade based classifiers using Haar-like features (again using the Viola-Jones inspired implementation found in OpenCV) using two data sets of labelled fiducial points: the BioID database and the PUT [11] database. Previous work using Haar cascade classifiers trained on PUT was able to properly detect and localize eyes 94% of the time with a false positive rate of 13% using a heuristic procedure for finding eye pairs [12]. In our approach as well the Haar cascade classifiers produce a number of candidates for each of our five different landmarks. We filter these candidate landmarks and identify a set of two to three geometrically consistent facial landmarks using the following procedure.

A. Fiducial point filtering

The filtering pipeline works in two steps. First, a list of easy false positives are removed through a heuristic rule filter. A set of simple rules, for example: (i) points within the border area of certain width are discarded, (ii) the nose must not be within the upper 1/3 area of the face, (iii) The left eye must be within a certain region in the upper left corner, right eye in the top right, (iv) The mouth should be in the lower half of the face region are used a rules for this filtering step.

The points that make it past the heuristic filtering become the candidate points that which are evaluated under simple probabilistic model for the spatial positions of all points. We estimate the parameters of a diagonal covariance Gaussian model in 2D for the spatial fiducial point distributions of points in the PUT database. Figure 1 illustrates the positions of our five landmarks within the PUT database. Images were scaled to a size of 250×250 pixels following the LFW face benchmarking process[13].

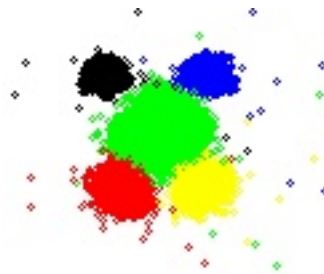


Fig. 1. The spatial distribution of the five landmarks used here within the faces of the PUT database. The left eye, right eye, nose tip, left mouth corner and right mouth corner x,y coordinates are shown as black, blue, green, red and yellow markers respectively.

Let, the shape of a face, $x_f^{(j)}$ be defined through a set of points $\{x_i, y_i\}_{i=1}^n$, where (x_i, y_i) is the location of a fiducial point, and n is the number of points defining the face. Let, a subset of m points from those n points constitute a spatial configuration, $(A_1 A_2 \dots A_m)$. The probability of observing

this configuration is

$$P(A_1 A_2 \dots A_m) = \prod_{i=1}^m P(A_i), \quad (1)$$

where, $P(A_i) = N(\boldsymbol{\mu}_i, \Sigma_i)$. If only one category of landmark is detected, our system does nothing. If we detect candidates for four or fewer categories of landmark, we enumerate all candidates and find the configuration with maximal probability under our model. If candidates are detected for 5 categories of landmarks, we use a procedure of randomly selecting three different landmark categories and enumerate all possible combinations. For each combination of points in the configuration we compute its probability under our model. The process is repeated a number of times depending on our time constraints (for our experiments here we randomly identify 10 combinations of three points and enumerate all possible candidates), and identify the most probable combination. Table V shows the noisy output points, while table VI shows the points that passed the rule filter, and the white dots are the model outputs.

B. Similarity Transformation

We use a similarity transformation to register faces and we thus need at least two landmarks to register a face to a common coordinate frame. Let, $[x \ y]^T$ be the coordinates of a fiducial point detected on a face, and $[u \ v]^T$ be the corresponding reference point position on a reference face that we have computed by taking the average for each landmarks coordinates over the BioID database. A similarity transformation can be represented as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & -m_2 \\ m_2 & m_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (2)$$

where, $[t_x \ t_y]^T$ is the translation parameter vector, $m_1 = s \cos \theta$ and $m_2 = s \sin \theta$, are two other parameters which contain the traditional parameters of rotation, θ and scale, s . This defines the transformation parameter vector, $T = [m_1, m_2, t_x, t_y]^T$. To solve for the transformation parameters the problem can be re-formulated as a system of linear equations

$$\begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \dots & \dots & & \\ \dots & \dots & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \dots \\ \dots \end{bmatrix} \quad (3)$$

where, (x_i, y_i) is a feature point on the observed face, $\{x_i, y_i\}_{i=1}^n$, while (u_i, v_i) is the corresponding target point on the latent face $\{u_i, v_i\}_{i=1}^n$. This re-formulation of a similarity transformation is similar to the commonly used reformulation of an affine transformation as discussed in [14]. We can write the system in matrix notation as $\mathbf{Ax} = \mathbf{b}$, such that $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. This approach provides the transformation parameters for a correspondence fiducial point set between an

observed face and our average or latent face. These parameters are used to register the unaligned face to a common coordinate frame. For two reference points placed in correspondence with a reference face one can solve for \mathbf{x} exactly; however, for three points (or more), one can obtain a least squares estimate through computing a pseudo inverse, $\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$. After transforming the image the final pixel values in the aligned image are calculated through bi-linear interpolation. Table III (d) shows the aligned George W. Bush images through this methodology.

C. Cutting out the central patch

This is the final step of the pipeline where, first, the face detector is run again on the aligned image. The eye pair locations are used to estimate a reference point, (x_c, y_c) . Let, (x_{le}, y_{le}) and (x_{re}, y_{re}) be the left and the right eye locations, and W and H be the width and height of the face window, returned by the face detector. Then, the face patch, selected by the model is: $\{(x_c - r_w * W, y_c - r_{hu} * H), (x_c + r_w * W, y_c + r_{hb} * H)\}$, where, $x_c = (x_{re} - x_{le})/2$ is the x axis location of the reference point and $y_c = (y_{re} - y_{le})/2$ is the y axis value, r_w is the width factor and r_{hu} and r_{hb} are the height factors. Finally, the selected area is rescaled to a constant size, say, $m \times n$.

III. EXPERIMENTS AND RESULTS

We used Bio-id [15] and PUT[11] datasets for local fiducial point classifiers learning, while spatial point distributions were learned from the PUT[11] dataset. The models were tested for the face verification task on the challenging LFW dataset[13]. In addition, the fiducial point localization accuracies were tested on an unseen Bio-id dataset.

A. Dataset

1) *Bio-Id*:: This dataset consists of 1521 gray level images of resolution 384×286 for 23 different subjects. The dataset also includes 20 manually marked fiducial points on each face.

2) *LFW*: LFW dataset comes with three different versions:

- LFW (unfunneled): The raw images, as collected.
- LFW (funneled) : The images were aligned through a pipeline called congealing and funnelling [4].
- LFWa : A dataset by L. Wold [7], aligned through a commercial aligner.

Each version has two settings:

View 1: This data set is for model development, and it comes with a precise *training – test* split. The training data contains 1100 positive pairs and 1100 negative pairs, while the test data is with 500 positive and 500 negative pairs. Training and test data are mutually exclusive. Although provided in this strict setting, this data could be used as one likes, however the performance reporting should strictly follow the following *view – 2* set up.

View 2: This data is is compiled for any performance reporting. The data contains 10 folded splits, each with 300 positive and 300 negative pairs.

TABLE V
INITIALLY DETECTED LANDMARKS

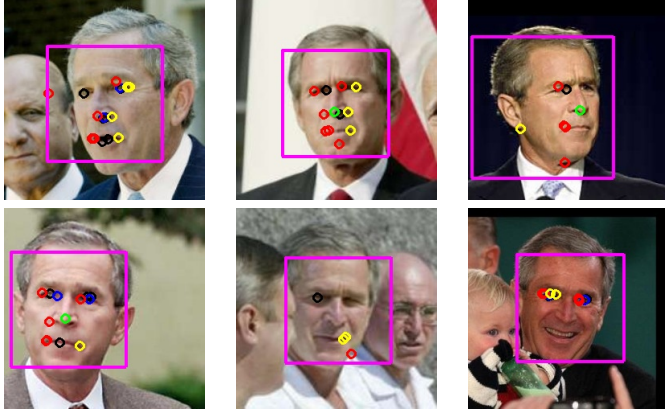
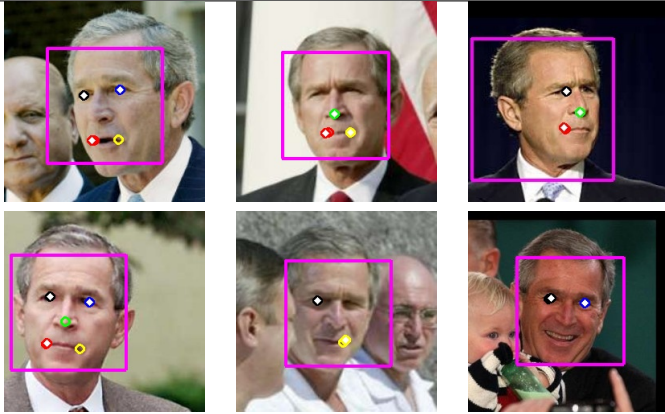


TABLE VI
DETECTED LANDMARKS AFTER FILTERING



B. Fiducial point localization

For each of the five fiducial points, 1000 positive patches from the Bio-id dataset and 9000 from the PUT dataset (centered on each point) were extracted. For each training image, three negative patches from random locations, exclusive to the positive patch area were extracted from the same training set. For each of the cases the patch size was set to 15×15 . The fiducial point detection model was tested on an unseen dataset of 200 Bio-id images. The euclidean distance between the model output and the actual point normalized by the eye pair distance (referred to as normalized distance) was used as the efficiency measure, and the results are compiled in table VII.

For table III(a) George W. Bush images table VI shows the fiducial point outputs (white dots) of the model. The output patches, computed from these points are shown in table IV(d). For a patch selection, the parameters were used as: $\{r_w = W/4, r_{hu} = W/4, r_{hb} = W/1.75\}$.

C. Face verification task

For the face verification task, we implemented the Cosine Similarity Metric Learning Model (CSML)[5]. The basic

TABLE VII
LOCALIZATION ACCURACIES

| point | accuracy |
|--------------------|----------------|
| Light eye | $.081 \pm .04$ |
| Right eye | $.086 \pm .05$ |
| Nose tip | $.104 \pm .06$ |
| Left mouth corner | $.28 \pm .2$ |
| Right mouth corner | $.28 \pm .2$ |

idea of the CSML model is to calculate the cosine distance (equation 4) between a given pair of faces (x_f, y_f) by first projecting them to a lower dimensional space through a linear map A , and using a pre-learned threshold, θ to decide whether they are a positive pair or not.

$$CS(x_f, y_f, A) = \frac{(Ax_f)^T(Ay_f)}{\|Ax_f\| \|Ay_f\|} \quad (4)$$

To learn A , from n labeled examples, $\{x_f^{(i)}, y_f^{(i)}, l_i\}_{i=1}^n$, where $(x_f^{(i)}, y_f^{(i)})$ is data instance with label $l_i \in \{+1, -1\}$, CSML formulated it as a maximization problem, $arg_A \max f(A)$, and the objective function $f(A)$ is defined as equation (5). The basic idea is to push the positive and negative samples towards the direction $+1$ and -1 respectively, and maximize the class distance in the Cosine space. The model also added a quadratic regularizer $\|A - A_0\|^2$ to control the over fitting artifacts.

$$f(A) = \sum_{i \in Pos} CS(x_f^{(i)}, y_f^{(i)}, A) - \alpha \sum_{i \in Neg} CS(x_f^{(i)}, y_f^{(i)}, A) - \beta \|A - A_0\|^2 \quad (5)$$

Table VIII compiles the verification results for four different patch selection strategies: DBMW, the selection strategy from [9]; CSML, the selection strategy from [5]; our-LFWa approach, which corresponds to applying our final central patch cutting strategy on the commercially aligned LFWa images; and ours-aligned, which corresponds to using our complete processing pipeline for alignment and final patch selection. The DBMW strategy takes LFW-funneled (or congealed) images and crops out a 110×115 pixel from the center of the image. The CSML strategy crops a 80×150 pixel image from the LFWa commercially aligned imagery. The first two columns are for two different A_0 initializations $\{PCA, \text{and } WPCA\}$ as used in [5], while the third column is for the final CSML model. We used stochastic gradient ascent with a mini batch size of 20 for A learning, while the parameter θ was chosen to be at $p(\theta|C_+) = p(\theta|C_-)$, where C_+ and C_- are the positive (the same) and negative (not same) classes, modeled through two Gaussian distributions. The test results are for the LFW view-2 test-set, and we used Local Binary Patterns (LBP) as the feature definition for a face patch. The LBP descriptors were extracted for a non overlapping block of size 10×10 , and concatenated to form a global LBP descriptor

TABLE VIII
VERIFICATION RESULTS FOR THREE PATCH SELECTIONS

| | PCA | WPCA | CSML |
|--------------|-------------|-------------|-------------|
| DBMW | .622 ± .005 | .660 ± .006 | .711 ± .007 |
| CSML | .693 ± .004 | .720 ± .004 | .786 ± .005 |
| ours-LFWa | .700 ± .004 | .752 ± .005 | .819 ± .005 |
| ours-aligned | .688 ± .006 | .745 ± .006 | .802 ± .007 |

vector. The LBP features were first reduced to a dimension of 500 by Principal Component Analysis(PCA), before putting into the CSML framework.

From table VIII we see that our fiducial point based patch selection methodology worked better than the conventional centralized patch based face verifiers. Qualitatively the face alignment pipeline had close performance to the commercially available aligner. However, our approach and implementation here failed to align 10% of the images due to 0 or 1 local point class detections. This may have caused the slightly poorer performance of ours-aligned vs. ours-LFWa in table VIII . We hypothesize that an improvement in the local classifier and/or defining more fiducial point classes may solve this problem, and thus might boost alignment pipeline.

Generally, Haar-cascades output a window containing an object of interest (for example, a face). For fiducial point localizations, we hypothesized that the center of a output patch is the output point. We believe that this resulted in some localizations (for example, the mouth corners) to be more error prone than others (for example, the eyes) that have distinctive localization details. These results can be verified from table VII.

IV. CONCLUSIONS

In this paper we have presented a model to localize a consistent set of landmarks or fiducial points on a face image. The primary goal of these localizations was to then align faces to a common coordinate frame and reduce translation, rotation and scaling artifacts that impact the quality of subsequent descriptor based representations of the face. Accordingly, five Haar-cascades classifiers were trained and a simple probabilistic model was used to clean away false positives and determine a similarity transformation based alignment.

Based on our analysis we believe a promising direction for future research would be to increase the number of landmarks or fiducial points that could be detected in the first phase of our framework. However, as more landmark classes are added, the task of searching for a set of consistent landmarks becomes more time consuming. Thus a more sophisticated search strategy would also be needed. Further, for more precise spatial localization of such landmarks we believe some sort of spatially localized search within the Haar-cascade’s output region could be a promising direction. Finally, we would like to improve the filtering pipeline by a more sophisticated probabilistic model.

ACKNOWLEDGMENT

This work was supported in part by grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada and through a Google Research award to CJP. The authors would like to thank David Rim for his fruitful suggestions and discussions.

REFERENCES

- [1] “Face recognition vendor test (frvt),” 2007, <http://www.frvt.org/>.
- [2] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [3] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.
- [4] G. B. Huang, V. Jain, and E. Learned-Miller., “Unsupervised joint alignment of complex images,” in *International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil, 2007, pp. 153–160.
- [5] H. V. Nguyen and L. Bai, “Cosine similarity metric learning for face verification,” in *ACCV (2)*, 2010, pp. 709–720.
- [6] S. Prince, P. Li, Y. Fu, U. Mohammed, and J. Elder, “Probabilistic models for inference about identity,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, no. PrePrints, 2011.
- [7] L. Wolf, T. Hassner, and Y. Taigman, “Similarity scores based on background samples,” in *Proc. of ACCV*, 2009.
- [8] “Labeled faces in the wild (lfw) <http://vis-www.cs.umass.edu/lfw/>,” 2011.
- [9] L. Wolf, T. Hassner, and Y. Taigman, “Descriptor based methods in the wild,” in *Real-Life Images workshop at the European Conference on Computer Vision (ECCV)*, Marseille, France, October 2008. [Online]. Available: <http://www.openu.ac.il/home/hassner/projects/Patchlbp>
- [10] E. Learned-Miller, “Data driven image models through continuous joint alignment,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 28, no. 2, pp. 236–250, 2005.
- [11] A. Kasinski, A. Florek, and A. Schmidt, “The put face database,” *Image Processing and Communications*, vol. 13, no. 3, pp. 59–64, 2008.
- [12] A. Kasinski and A. Schmidt, “The architecture and performance of the face and eyes detection system based on the haar cascade classifiers,” *Pattern Analysis and Applications*, vol. 13, pp. 197–211, 2010.
- [13] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.
- [14] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [15] “The bioid database,” 2010, <http://www.bioid.com/>.