Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments

Gary B. Huang,Manu Ramesh, Tamara Berg, and Erik Learned-Miller

Abstract— **Face recognition has benefitted greatly from the many databases that have been produced to study it. Most of these databases have been created under controlled conditions to facilitate the study of specific parameters on the face recognition problem. These parameters include such variables as position, pose, lighting, expression, background, camera quality, occlusion, age, and gender.**

While there are many applications for face recognition technology in which one can control the parameters of image acquisition, there are also many applications in which the practitioner has little or no control over such parameters. This database is provided as an aid in studying the latter, unconstrained, face recognition problem. The database represents an initial attempt to provide a set of labeled face photographs spanning the range of conditions typically encountered by people in their everyday lives. The database exhibits "natural" variability in pose, lighting, focus, resolution, facial expression, age, gender, race, accessories, make-up, occlusions, background, and photographic quality. Despite this variability, the images in the database are presented in a simple and consistent format for maximum ease of use.

In addition to describing the details of the database and its acquisition, we provide specific experimental paradigms for which the database is suitable. This is done in an effort to make research performed with the database as consistent and comparable as possible.

I. Introduction

This report describes a database of human face images designed as an aid in studying the problem of *unconstrained face recognition*.[1] The database can be viewed and downloaded at the following web address: `http://vis-www.cs.umass.edu/lfw/`.

Face recognition is the problem of identifying a specific individual, rather than merely detecting the presence of a human face, which is often called *face detection*. The general term "face recognition" can refer to a number of different problems including, but not limited to, the following.

1) Given two pictures, each of which contains a face, decide whether the two people pictured represent the same individual.
2) Given a picture of a person's face, decide whether it is an example of a particular individual. This may be done by comparing the face to a model for that individual or to other pictures of the individual.
3) Given a picture of a face, decide which person from among a set of people the picture represents, if any. (This is often referred to as the *face verification* paradigm.)

Our database, which we call Labeled Faces in the Wild (LFW), is designed to address the first of these problems, although it can be used to address the others if desired. We shall refer to this problem as the *pair matching* problem.

The main motivation for the database, which is discussed in more detail below, is to provide a large set of relatively unconstrained face images. By unconstrained, we mean faces that show a large range of the variation seen in everyday life. This includes variation in pose, lighting, expression, background, race, ethnicity, age, gender, clothing, hairstyles, camera quality, color saturation, focus, and other parameters. Figures 1 and 2 show some examples of the database images. The reason we are interested in natural variation is that we are interested in the problem of pair matching given a pair of *pre-existing face images*, i.e., images whose composition we had no control over. We view this problem of *unconstrained pair matching* as one of the most general and fundamental face recognition problems.

Before proceeding with the details of the database, we present some summary statistics and properties of the database.

- The database contains 13,233 target face images. Some images contain more than one face, but it is the face that contains the central pixel of the image which is considered the defining face for the image. Faces other than the target face should be ignored as "background".
- The name of the person pictured in the center of the image is given. Each person is given a unique name ("George_W_Bush" is the current U.S. president while "George_HW_Bush" is the previous U.S. president), so no name should correspond to more than one person, and each individual should appear under no more than one name (unless there are unknown errors in the database).
- The database contains images of 5749 different individuals. Of these, 1680 people have two or more images in the database. The remaining 4069 people have just a single image in the database.
- The images are available as 250 by 250 pixel JPEG images. Most images are in color, although a few are grayscale only.
- All of the images are the result of detections by the

[1] We note that for more general classes of objects such as cars or dogs, the term "recognition" often refers to the problem of recognizing a *member of the larger class*, rather than a specific instance. That is, when one "recognizes" a cat (in the context of computer vision research), it is meant that one has identified a particular object as a cat, rather than that one has identified a particular cat. In the context of more general objects, we prefer the term *identification* to refer to the problem of recognizing a specific instance of a class (such as Bob's Toyota). For example, see the work by Ferencz et al. to see examples of this usage [7], [8], [15]. However, in the literature on human faces, the term *recognition* is typically used to refer to the identification of a particular individual, not just a human being. Since this report is about faces, we adopt this latter terminology here.

Fig. 1. **Matched pairs**. These are the first six matching pairs in the database under View 1, as specified in the file `pairsDevTrain.txt`. These pairs show a number of properties of the database. A person may appear in more than one training pair (first two rows). An image may have been cropped to center the face (3rd row, right image) according to the Viola-Jones detector, but the image has been padded with zeros to make it the same size as other images.

Viola-Jones face detector [35], but have been rescaled and cropped to a fixed size (see Section VI for details). After running the Viola-Jones detector on a large database of images, false positive face detections were manually eliminated, along with images for whom the name of the individual could not be identified.

- We define two "Views" of the database, one for algorithm development, and one for performance reporting. By using View 1 for algorithm development, the experimenter may avoid inappropriately fitting a classifier to the final test data. See Section III for details.

Additional details are given in the remainder of the report, which is organized as follows. In Section II, we discuss other databases, and the origins of Labeled Faces in the Wild. In Section III, we describe the structure of the database and its intended use for the unconstrained pair matching problem. We focus particular attention on the proper use of the two database Views, which is critical for accurate measurement of classifier generalization. In Section IV, we discuss two paradigms for using training data, the image-restricted paradigm, and the unrestricted paradigm. Experimenters should be careful to report which method is used when results are published. In Section V, we discuss the role of LFW in the Detection-Alignment-Recognition pipeline. In Section VI, we describe the construction of the database and details about resolution, cropping, removal of duplicate images, and other properties.

II. Related Databases

There are a large number of face databases available to researchers in face recognition. A non-exhaustive list can be found in Figure 3. These databases range in size, scope and purpose. The photographs in many of these databases were acquired by small teams of researchers specifically for the purpose of studying face recognition. Acquisition of a face database over a short time and in a particular location has significant advantages for certain types of research. Such an acquisition gives the experimenter direct control over the parameters of variability in the database.

On the other hand, in order to study more general face recognition problems, in which faces are drawn from a very broad distribution, one may wish to train and test face recognition algorithms on highly diverse sets of faces. While it is possible to manipulate a large number of variables in the laboratory in an attempt to make such a database, there are two drawbacks to this approach. The first is that it is extremely labor intensive. The second is that it is difficult to gauge exactly which distributions of various parameters one should use in order to make the most useful database. What percentage of subjects should wear sunglasses? What percentage should have beards? How many should be smiling? How many backgrounds should contain cars, boats, grass, deserts, or basketball courts?

One possible solution to this problem is simply to measure a "natural" distribution of faces. Of course, no single canonical distribution of faces can capture a natural distribution of faces that is valid across all possible application domains. Our database uses a set of images that was originally gathered from news articles on the web. This set clearly has its own biases. For example, there are not many images which occur under extreme lighting conditions, or very low lighting conditions. Also, because we use the Viola-Jones detector as a filter for the database, there are a limited number of side views of faces, and few views from above or below. But the range and diversity of pictures present is very large. We believe such a database

Fig. 2. **Mismatched pairs**. These are the first six *mismatched* pairs in the database under View 1, as specified in the file `pairsDevTrain.txt`.

will be an important tool in studying the unconstrained pair matching problem.

While some other databases (such as the Caltech 10000 Web Faces [1]) also present highly diverse image sets, these databases are not designed for face recognition, but rather for face detection. We now discuss the origin for Labeled Faces in the Wild and a number of related databases.

Faces in the Wild. The impetus for the Labeled Faces in the Wild database grew out of work at Berkeley by Tamara Berg, David Forsyth, and the computer vision group at UC Berkeley [3], [4]. In this work, it was shown that a large, partially labeled, database of face images could be built by using imperfect data gathered from the web. In particular, the Berg database of faces was built by jointly analyzing pictures and their associated captions to cluster images by identity. The resulting data set, which achieved a labelling accuracy of 77% [3], was informally referred to as the "Faces in the Wild" data set.

However, since the database was not originally intended to act as training and test data for new experiments, it contained a high percentage of label errors and a high percentage of duplicated images. As a result, various researchers derived ad hoc subsets of the database for new research projects [14], [15], [25], [27]. It seemed that there would be sufficient interest in a clean version of the data set to warrant doing the job thoroughly and publishing a new database.

Before addressing the details of LFW, we discuss some of the databases most closely related to it. While these databases share some features with LFW, we believe that LFW represents an important contribution to existing databases, especially for studying the problem of unconstrained face recognition.

The Face Recognition Grand Challenge Databases [28]. The Face Recognition Grand Challenge (FRGC) was not just a set of databases, but a carefully planned scientific program designed to promote rigorous scientific analysis of face recognition, fair comparison of face recognition technologies, and advances in face recognition research [28]. It represents the most comprehensive and scientifically rigorous study of face recognition to date. We applaud the organizers and implementers of the FRGC, and believe that the FRGC, along with earlier vendor tests, have been important motivators and reality checks for the face recognition community. The FRGC was successful in stimulating researchers (in both the private sector and academia) to achieve certain milestones in face recognition.

The goals of our research, and hence of our database, are somewhat different from the goals of the FRGC. One of the key differences is that the organizers of the FRGC wished to study the effect of new, richer data types on the face recognition problem. The databases for the FRGC thus include high resolution data, three-dimensional scans, and image sequences of each individual. (The databases contain more than 50,000 total recordings, including 3D scans and images.) Each of these data types is potentially more informative than the simple and moderate resolution images of our database. While one of the major goals of the FRGC was to study how higher fidelity data can help make face recognition more accurate, the goal of Labeled Faces in the Wild is to help study the problem of face recognition using *previously existing images*, that is, images that were not taken for the special purpose of face recognition by machine. Thus, from the beginning we decided to build our database from previously existing photographs that were taken for other purposes.

Another important difference between the data sets associated with the FRGC and our data set is the general variety of images. For example, while there are large numbers of images with uncontrolled lighting in the FRGC data sets, these images contain a great deal less natural variation than the LFW images. For example, the FRGC outdoor uncontrolled

Database	# of people	Total images	Highlights	References
AR Face Database, Purdue University, USA	126	4000	frontal pose, expression, illumination, occlusions, eye glasses, scarves	[21]
AT&T Database (formerly ORL Database)	40	400	variation of time, lighting, facial expression, eye glasses	[29]
BioID Face Database	23	1521	real world conditions, gray scale, background, lighting, expression, eye positions given	[17]
Caltech Faces	27	450	lighting, expression, background	
Caltech 10000 Web Faces	$\approx$ 10000	10000	wide variability, facial features annotated	[1]
CAS-PEAL Face Database	1040	99,594	very large, expression, accessories, lighting, simultaneous capture of multiple poses, Chinese	[10]
Cohn-Kanade AU-Coded Facial Expression Database	100	500 sequences	dynamic sequences of facial expressions	[6]
EQUINOX HID Face Database	?	?	non-visible light modalities	
Face Video Database of the Max Planck Institute for Biological Cybernetics	?	246 video sequences	6 simulataneous viewpoints, carefully synchronized, video data	[18]
Facial Actions and Expressions	24	$\approx$ 7000	expression, color, grayscale	
Face Recognition Grand Challenge Databases	>466	>50,000 images and 3D scans	very large, lighting, expression, background, 3D, sequences	[28]
FERET Database, Color	1199	14126	color images, changes in appearance through time, controlled pose variation, facial expression	[23]
Georgia Tech Face Database	50	750	expression, illumination, scale, orientation	[26]
Indian Face Database	40	$>$ 440	frontal, Indian subjects	[16]
Japanese Female Facial Expression (JAFFE) Database	10	213	rated for emotional content, female, Japanese	[19]
MIT-CBCL Face Recognition Database	10	$>$ 2000	synthetic images from 3D models, illumination, pose, background	[37]
M2VTS Multimodel Face Database (Release 1.00)	37	185	large pose changes, speaking subjects, eye glasses, time change	[30]
M2VTS, Extended, Univ. of Surrey, UK	295	1180 videos	rotating head, speaking subjects, 3D models, high quality images	[22]
NIST Mugshot ID	1573	3248	front and side views	[36]
NLPR Face Database	$\approx$ 22	450	lighting, expression, backgrounds	[24]
PIE Database, CMU	68	41368	very large database, pose, illumination, expression	[33]
Psychological Image Collection at Stirling (PICS)	?	?	targeted at psychology experiments	[13]
UCD Colour Face Image Database for Face Detection	$\approx$ 299	299	targeted at detection applications, highly varied, color	[32]
UMIST Face Database	20	564	pose, gender, race, grayscale	[12]
University of Essex, UK	395	7900	racial diversity, eye glasses, beards, college age	[34]
University of Oulu Physics-Based Face Database	125	$>$ 2000	highly varied illumination, eye glasses	[20]
VALID Database	106	530	highly variable office conditions	[9]
VidTIMIT Database	43	multiple videos per person	video, audio, reading, head rotation	[31]
Yale Face Database	15	165	expressions, eye glasses, lighting	[2]
Yale Face Database B	10	5760	pose, illumination	[11]

Fig. 3. Face databases. This table shows some of the face databases available at the time of writing. This list is not meant to be exhaustive, nor to describe the databases in detail, but merely to provide a sampling of the types of databases that are available. Where possible, a peer-reviewed paper or technical report was cited, and otherwise a citation referring to the web page for the database is given when available. Much of the information on this page was gathered with the help of the excellent "Face Recognition Homepage," maintained by Mislav Grgic and Kresimir Delac (`http://www.face-rec.org/`).

lighting images contain two images of each subject, one smiling and one with a neutral expression. The LFW images, in contrast contain arbitrary expressions. Variation in clothing, pose, background, and other variables is much greater in LFW than in the FRGC databases. One may sum up the differences as *controlled variation* (FRGC) versus *natural* or *random* variation (LFW).

We believe that the FRGC served a very important role in advancing the state of the art in face recognition, especially the specific problem of face recognition under the assumption that certain types of data can be acquired. We believe that our database fills a complementary need for a large data set of labeled images in studying the unconstrained face recognition problem.

The BioID Face Database [17]. Another database which shares important properties with LFW is the BioID Face Database. This database consists of 1521 gray level images with a resolution of 384 by 286 pixels. Each image shows a frontal view of the face of one out of 23 different test persons. The most important property shared by the BioID Face Database and Labeled Faces in the Wild is that both databases strive to capture realistic settings, with significant variability in pose, lighting, and expression. BioID backgrounds include what appear to be realistic office or home settings for their pictures, and these backgrounds vary simultaneously with subject pose, expression, and other parameters. Since one of the main goals of LFW is to provide realistic images, this is a significant similarity.

Despite this important similarity, BioID is quite different from LFW. Important differences include the following.

- While BioID and LFW both strive to capture a set of realistic images, the distributions they capture are significantly different. The distribution of images in BioID is focussed on a small number of office and home environments. For each individual, most pictures are taken in the same setting, but from a slightly different point of view. LFW pictures of the same individual, in contrast, are often taken in completely different settings, and at different times. For example, the same athlete may be photographed during a sporting event and at a news conference.
- According to the database web site, it appears that BioID is targeted more at the face detection problem. LFW is targeted at face recognition, or identification.
- BioID has relatively low variability with respect to race, with the large majority of images being of caucasians. LFW has a broad distribution of people from different parts of the world, different races, and different ethnicities.
- BioID has manually marked eye positions in each image. LFW has no such markings. The only positional information given for LFW is that the image is the immediate output (up to a fixed rescaling and recropping, described in Section VI) of the Viola-Jones face detector. Thus, the face is usually (but not always) centered and usually (but not always) at a similar scale.
- LFW includes color images. BioID does not.
- BioID has a relatively large number of images per person (66.13), with a relatively small number of people (23). LFW has a much smaller average number of images per person (2.30), with a much larger number of people (5749).

Overall, BioID is an interesting database of face images which may be useful for a number of purposes such as face detection in indoor enviroments. We believe that LFW, on the other hand, will be useful for solving more general and difficult face *recognition* problems with large populations in highly variable environments.

Caltech 10000 Web Faces [1]. The Caltech 10000 Web Faces database is interesting in that it also provides a very broad distribution of faces. The distribution of faces included in the Caltech collection is similar to the distribution of faces in LFW. In particular, the faces in each database show a broad mixture of ages, expression, hairstyles, lighting effects, race, and gender. The backgrounds are highly varied in both data sets, although the Caltech data set includes significantly more background area.

However, the Caltech database is again geared more toward face detection and alignment rather than face recognition. It provides the position of four facial features, but does not give the identity of individuals. Thus, it is not particularly suitable for face recognition experiments.

In summary, there are a great number of face databases available, and while each has a role in the problems of face recognition or face detection, we believe LFW fills an important gap for the problem of unconstrained face recognition.

III. Intended Uses

As mentioned in the introduction, this database is aimed at studying the problem of pair matching. That is, given a pair of face images, we wish to decide whether the images are of the same person. By outputting a probability of match or mismatch rather than a hard decision, one can easily create a Receiver Operating Characteristic, or ROC curve, that gives the minimium cost decisions for given relative error costs (false match or false mismatch).

Even within what we call the pair matching paradigm, there are a number of subtly, but importantly different recognition problems. Some of these differences concern the specific organization of training and testing subsets of the database. **A critical aspect of our database is that for any given training-testing split, the people in each subset are mutually exclusive.** In other words, for any pair of images in the training set, neither of the people pictured in those images is in any of the test set pairs. Similarly, no test image appears in a corresponding training set.

Thus, at training time, it is essentially impossible to build a model for any person in the test set. This differs substantially from paradigms in which there is a fixed gallery of test subjects for whom training images are available, and the goal is to find matches of so-called probe images to members of the gallery. (Such fixed gallery paradigms are often referred to as *face verification*.) In particular, for LFW, since the people in test images have never been seen before, there is no opportunity to build models for such individuals, except to do this at test time from a single image.

Instead, this paradigm is meant to focus on the generic problem of differentiating *any two individuals* that have never been seen before. Thus, a different type of learning is suggested– learning to discriminate among any pair of faces, rather than learning to find exemplars of a small (or even large) gallery of people as in face verification. There are numerous examples of this kind of face recognition research [7], [15], [25].

A. Pair Matching and Learning from One Example

We shall refer to the specific pair matching problem, in which neither of the individuals pictured in a test pair has been seen during training, as the *unseen pair match* problem. This is closely related to the problem of *learning from one example*, in which a single training image of a person is provided, and the goal is to determine whether a new image represents the individual for whom one training image was provided.

In particular, the unseen pair match problem can be viewed as a specific instance of the problem of learning from one example. Specifically, given a pair of images and the question of whether they are the same, one of the images can be considered to define the "model", and the other can be considered to be an instance of the person defined by the model or not. But there are important differences between the classical problem of learning from one example, as discussed for example in the paper of Beymer et al. [5], and the unseen pair match problem (see for example [7]). The main differences are as follows.

- In learning from one example (per person), training examples are given at training time. Whereas in the unseen pair match problem, the single model image is not available until test time. If processing speed is an important constraint, then it may be advantageous to have a training example ahead of time, as in the learning from one example paradigm.
- Another important difference is that in learning from one example, at test time, the objective is usually to determine which, if any, of the models the test image corresponds to. One would not normally identify the test image with more than one model, and so a winner-take-all or maximum likelihood approach for selecting a match would be reasonable. On the other hand, in the unseen pair match problem, the objective is to make a binary decision about whether a given single image matches another image. If a test set contains multiple pairings of a single image B, i.e., a group of pairs of images of the form $(A_i, B), 1 \leq i \leq n$, there is no mechanism for deciding that the image B should match only one of the images A_i. In other words, each pairwise decision is made independently. This rules out the winner-take-all or maximum likelihood style approaches.

In summary then, LFW is intended for the unseen pair matching paradigm, which is characterized by the conditions that

- no images of test subjects are available at training time, and
- the decisions for all test pairs are made independently.

Conformance to this second condition disallows techniques such as semi-supervised learning in which examples are used from across an entire test set. For each test pair, any algorithm should behave as if these are the only two test images. Put another way, an algorithm should not use more than a single pair of test images at a time.

B. Training, Validation, and Testing

Proper use of training, validation, and testing sets is crucial for the accurate comparison of face recognition algorithms. In describing the Face Recognition Grand Challenge [28], the authors note that using sequestered test sets, i.e. test sets not publicly available to researchers, is the best way to ensure that algorithm developers do not unfairly fit the parameters of their algorithms to the test data. Allowing the experimenter to choose the parameters of an algorithm that work best on a test set, or equivalently, allowing the experimenter to choose the best *algorithm* for a given test set, biases upward the estimate of accuracy such an algorithm would produce on a sequestered test set. While we fully support this point of view, we have decided for practical reasons not to use a sequestered test set, but to include the test data in the public database. We hope that by providing clear guidelines for the use of this data, that "fitting to the test data" will be minimized. Also, the size and difficulty of the data set may mitigate the degree to which unintended overfitting problems may occur.

We organize our data into two "Views", or groups of indices. View 1 is for algorithm development and general experimentation, prior to formal evaluation. This might also be called a model selection or validation view. View 2, for performance reporting, should be used only for the final evaluation of a method. The goal of this methodology is to use the final test sets as seldom as possible before reporting. Ideally, of course, each test set should only be used once. We now describe the two views in more detail.

View 1: Model selection and algorithm development. This view of the data consists of two subsets of the database, one for training (`pairsDevTrain.txt`), and one for testing (`pairsDevTest.txt`). The training set consists of 1100 pairs of matched images and 1100 pairs of mismatched images. The test set consists of 500 pairs of matched and 500 pairs of mismatched images. In order to support the unseen pair match paradigm, the people who appear in the training and testing sets are mutually exclusive.

The main purpose of this view of the data is so that researchers can freely experiment with algorithms and parameter settings without worrying about overusing test data. For example, if one is using support vector machines and trying to decide upon which kernel to use, it would be appropriate to test various kernels (linear, polynomial, radial basis function, etc.) on View 1 of the database.

To use this view, simply train an algorithm on the training set and test on the test set. This may be repeated as often as desired without significantly biasing final results. (See caveats below.)

View 2: Performance reporting. The second view of the data should be used sparingly, and only for performance reporting. Ideally, it should only be used once, as choosing the best performer from multiple algorithms, or multiple parameter settings, will bias results toward artificially high accuracy.

The second view of the data consists of ten subsets of the database. Once a model or algorithm has been selected (using View 1 of the database if desired), the performance of that algorithm can be measured using View 2. To report accuracy results on View 2, the experimenter should report the aggregate performance of a classifier on 10 separate experiments in a leave-one-out cross validation scheme. In each experiment, nine of the subsets should be combined to form a training set, with the tenth subset used for testing. For example, the first experiment would use subsets $(2, 3, 4, 5, 6, 7, 8, 9, 10)$ for training and subset 1 for testing. The fourth experiment would use subsets $(1, 2, 3, 5, 6, 7, 8, 9, 10)$ for training and subset 4 for testing.

It is critical for accuracy performance reporting that the final parameters of the classifier under each experiment be set **using only the training data for that experiment.** In other words, an algorithm may not, during performance reporting, set its parameters to maximize the combined accuracy across all 10 training sets. The reason for this is that training and testing sets overlap across experiments, and optimizing a classifier simultaneously using all training sets is essentially fitting to the test data, since the training set for one experiment is the testing data for another. In other words, for performance reporting, each of the 10 experiments (both the training and testing phases) should be run completely independently of the others, resulting in 10 separate classifiers (one for each test set).

While there are many methods for reporting the final performance of a classifier, including ROC curves and Precision-Recall curves, we ask that each experimenter, at a minimum, report the **estimated mean accuracy** and the **standard error of the mean** for View 2 of the database.

In particular, the **estimated mean accuracy** $\hat{\mu}$ is given by

$$\hat{\mu} = \frac{\sum_{i=1}^{10} p_i}{10},$$

where p_i is the percentage of correct classifications on View 2, using subset i for testing. It is important to note that accuracy should be computed with parameters and thresholds chosen independently of the test data, ruling out, for instance, simply choosing the point on a Precision-Recall curve giving the highest accuracy.

The **standard error of the mean** is given as

$$S_E = \frac{\hat{\sigma}}{\sqrt{10}},$$

where $\hat{\sigma}$ is the estimate of the standard deviation, given by

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{10} (p_i - \hat{\mu})^2}{9}}.$$

Because the *training sets* in View 2 overlap, the standard error may be biased downward somewhat relative to what would be obtained with fully independent training sets and test sets. However, because the test sets of View 2 are independent, we believe this quantity will be valuable in assessing the significance of the difference among algorithms.[2]

Discussion of data splits. The multiple-view approach described above has been used, rather than a traditional training-validation-testing split of the database, in order to maximize the amount of data available for training and testing. Ideally, one would have enough images in a database so that training, validation, and testing sets could be non-overlapping. However, in order to maximize the size of our training and testing sets, we have allowed reuse of the data between View 1 of the database and View 2 of the database. While this introduces some bias into the results, we believe the bias will be very small in most cases, and is outweighed by the benefit of the resulting larger training and test set sizes.

Given our multiple-view organization of the database, it is possible to "cheat" and produce a classifier which shows artificially good results on the final test set. In particular, during the model selection phase, using View 1 of the database, one could build a classifier which simply stores all of the training data in a file, and declare that this file is now part of the classifier. During performance reporting, using View 2 of the database, examples in each test set could be compared against the stored examples from View 1, and since many of them are the same, performance would be artificially high.

While we trust that no researcher would use such a scheme intentionally, it is possible that similar schemes might be implemented unintentionally by giving the classifier a large store of memory in which to memorize features of the View 1 training set, and then to reuse these memorized features during performance reporting. The reason we believe that this scenario would not arise accidentally is that such a scheme would do very poorly on the testing portion of View 1, since the training and testing for View 1 do not overlap. That is, there should be no performance benefit during View 1 testing from memorizing large sets of features or parts of images. If the classifier is built using View 1 in order to minimize generalization error, then the memorization scheme described above would not be expected to work well. In other words, if experimenters legitimately strive to maximize performance on the testing data in View 1, and then run experiments on View 2 without modifying the inherent form of their classifiers, we believe our database organization will successfully measure the generalization ability of classifiers, which is our goal.

Summary of usage recommendations. In summary, for proper use of the database, researchers should proceed roughly according to the following procedure.

1) Algorithm development or model selection.
   a) Use View 1 of the database to train and test as many models, with as many parameter settings, as desired.
   b) Retain model M^* which has best performance on test set.
2) Performance reporting.
   a) Use View 2 of the database.

[2]We remind the reader that for two algorithms whose standard errors overlap, one may conclude that they their difference is not statistically significant at the 0.05 level. However, one *may not conclude,* in general, that algorithms whose standard errors do not overlap are statistically different at the 0.05 level.

b) For $i = 1$ to 10
   i) Form training set for experiment i by combining all subsets from View 2 except subset i.
   ii) Set parameters of model M^* using training set, producing classifier i.
   iii) Use subset i of View 2 as a test set.
   iv) Record results of classifier i on test set.

c) Use results from all 10 classifiers to compute the estimated mean classification accuracy $\hat{\mu}$ and the standard error of the mean S_E as described above.

d) Finally, make sure to report which training method (image-restricted or unrestricted) was used, as described in Section IV.

IV. Transitivity and the Image-Restricted and Unrestricted Use of Training Data

Whenever one works with matched and mismatched data pairs such as those described in `pairsDevTrain.txt`, the issue of creating auxiliary training examples arises by using the transitivity of equality.

For example, in a training set, if one matched pair consists of the 10th and 12th images of George_W_Bush, and another pair consists of the 42nd and 50th images of George_W_Bush, then it might seem reasonable to add other image pairs, such as (10, 42), (10, 50), (12, 42) and (12, 50), to the training data using an automatic procedure. One could argue that such pairs are *implicitly present* in the original training data, given that the images have been labeled with the name George_W_Bush. Auxiliary examples could be added to the mismatched pairs using a similar method.

Rather than disallowing such augmentation on the one hand, or penalizing researchers who do not wish to add many thousands of extra pairs of images to their training sets on the other, we describe two separate methods for using training data. When reporting results, the experimenter should state explicitly whether the *image-restricted* or the *unrestricted* training method was used to generate results. These two methods of training are described next.

A. Image-Restricted Training

The idea behind the image-restricted paradigm is that the experimenter should *not* use the name of a person to infer the equivalence or non-equivalence of two face images that are not explicitly given in the training set. Under the image-restricted training paradigm, the experimenter should discard the actual names associated with a pair of training images, and retain only the information about whether a pair of images is matched or mismatched. Thus, if the pairs (10,12) and (42,50) of George_W_Bush are both given explicitly in a training set, then under the image-restricted training paradigm, there would be no simple way of inferring that the 10th and 42nd images of George_W_Bush were the same person, and thus this image pair should not be added to the training set.

Note that under this paradigm, it is still possible to augment the training data set by comparing *image similarity*, as opposed to name equivalence. For example, if the 1st and 2nd images of a person form one matched training pair, while the 2nd and 3rd images of the same person form another matched training pair, one could infer from the *equivalence of images* in the two pairs that the 1st and 3rd images came from the same person, and add this pair to the training set as a matched pair. Such image-based augmentation is allowed under the image-restricted training paradigm.

Both Views of the database support the image-restricted training paradigm. In View 1 of the database, the file `pairsDevTrain.txt` is intended to support the image-restricted use of training data, and `pairsDevTest.txt` contains test pairs. In View 2 of the database, the file `pairs.txt` supports the image-restricted use of training data. Formats of all such files are given in Section VI-F.

B. Unrestricted Training

The idea behind the unrestricted training paradigm is that one may form as many pairs of matched and mismatched pairs as desired from a set of images labeled with individuals' names. To support this use of the database, we defined subsets of *people*, rather than image pairs, that can be used as a basis for forming arbitrary pairs of matched and mismatched images.

In View 1 of the database, the files `peopleDevTrain.txt` and `peopleDevTest.txt` can be used to create arbitrary pairs of training and testing images. For example, to create mismatched training pairs, choose any two people from `peopleDevTrain.txt`, choose one image of each person, and add the pair to the data set. Pairs should *not* be constructed using mixtures of images from training and testing sets.

In View 2 of the database, the file `people.txt` supports the unrestricted training paradigm. Training pairs should be formed only using pairs of images from the same subsets. Thus, to form a training pair of mismatched images, choose two people from the same subset of people, choose an image of each person, and add the pair to the training set. *Note that in View 2 of the database, which is intended only for performance reporting, the test data is fully specified by the file* `pairs.txt`*, and should not be constructed using the unrestricted paradigm.* The unrestricted paradigm is only for use in creating *training* data.

Due to the added complexity of using the unrestricted paradigm, we suggest that users start with the image-restricted paradigm by using the pairs described in `pairsDevTrain.txt`, `pairsDevTest.txt`, and, for performance reporting, `pairs.txt`. Later, if the experimenters believes that that their algorithm may benefit significantly from larger amounts of training data, they may wish to consider using the unrestricted paradigm. In either case, it should be made clear in any publications which training paradigm was used to train classifiers for a given test result.

V. The Detection-Alignment-Recognition Pipeline

Many real world applications wish to automatically detect, align, *and* recognize faces in a larger still image, or in a video of a larger scene. Thus, face recognition is often naturally described as part of a Detection-Alignment-Recognition (DAR) pipeline, as illustrated in Figure 4.

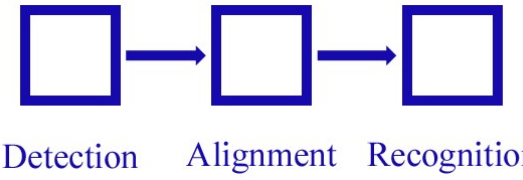

Fig. 4. The Detection-Alignment-Recognition (DAR) pipeline. The images of the Labeled Faces in the Wild database represent the output of the Viola-Jones detector. By working with such a database, the developer of alignment and recognition algorithms know that their methods will fit easily into the DAR pipeline.

To complete this pipeline, we need automatic algorithms for each stage of the pipeline. In addition, each stage of the pipeline must either accept images from, or prepare images for, the next stage of the pipeline. To facilitate this process, we have purposefully designed our database to represent the output of the detection process.

In particular, every face image in our database is the output of the Viola-Jones face detection algorithm [35]. The motivation for this is as follows. If one can develop a face alignment algorithm (and subsequent recognition algorithm) that works directly on LFW, then it is likely to also work well in an end-to-end system that uses the Viola-Jones detector as a first step.

This alleviates the need for each researcher to worry about the process of detection, on the one hand, and to worry about the possibility that a manually aligned database does not adequately represent the true variability seen in the world. In other words, it allows the experimenter to focus on the problems of alignment and recognition rather than the problem of detection. The specific details of how the database was constructed are given in the next section.

VI. Construction and Composition Details

The process of building the database can be broken into the following steps:

1) gathering raw images,
2) running a face detector and manually eliminating false positives,
3) eliminating duplicate images,
4) labeling (naming) the detected people,
5) cropping and rescaling the detected faces, and
6) forming pairs of training and testing pairs for View 1 and View 2 of the database.

We describe each of these steps in the following subsections.

A. Gathering raw images

As a starting point, we used the raw images from the Faces in the Wild database collected by Tamara Berg at Berkeley. Details of this set of images can be found in the following publication [4].

B. Detecting faces

A version of the Viola-Jones face detector [35] was run on each image. Specifically, we used the code in OpenCV, version 1.0.0, release 1. Faces were detected using the function `cvHaarDetectObjects`, using the provided Haar classifier cascade `haarcascade_frontalface_default.xml`, with scale_factor set to 1.2, min_neighbors set to 2, and the flag set to `CV_HAAR_DO_CANNY_PRUNING`.

For each positive detection (if any), the following procedure was performed:

1) If the highlighted region was determined by the operator to be a non-face, it was omitted from the database.
2) If the name of the person of a detected face from the previous step could not be identified, either from general knowledge or by inferring the name from the associated caption, then the face was omitted from the database.
3) If the same picture of the same face was already included in the database, the face was omitted from the database. More details are given below about eliminating duplicates from the database.
4) Finally, if all of these criteria were met, the face was recropped and rescaled (as described below) and saved as a separate JPEG file.

C. Eliminating duplicate face photos

A good deal of effort was expended in removing duplicates from the database. While we considered including duplicates, since it could be argued that humans may often encounter the exact same picture of a face in advertisements or in other venues, ultimately it was decided that they would prove to be a nuisance during training in which they might cause overfitting of certain algorithms. In addition, any researcher who chooses may easily add duplicates for his or her own purposes, but removing them is somewhat more tedious.

Definition of duplicate images. Before removing duplicates, it is necessary to define exactly what they are. While the simplest definition, that two pictures are duplicates if and only if the images are numerically equivalent at each pixel, is somewhat appealing, it fails to capture large numbers of images that are indistinguishable to the human eye. We found that the unfiltered database contained large numbers of images that had been subtly recropped, rescaled, renormalized, or variably compressed, producing pairs of images which were visually nearly equivalent, but differed significantly numerically.

We chose to define duplicates as images which were judged to have a common original source photograph, irrespective of the processing they had undergone. While we attempted to remove all duplicates as defined above from the database, there may exist some remaining duplicates that were not found. We believe the number of these is small enough so that they will not significantly impact research.

In addition, there remain a number of pairs of pictures which are extremely similar, but clearly distinct. For example, there appeared to be pictures of celebrities taken nearly simultaneously by different photographers from only slightly different angles. Whenever there was evidence that a photograph was distinct from another, and not merely a processed version of another, it was maintained as an example in the database.

D. Labeling the faces

Each person in the database was named using a manual procedure that used the caption associated with a photograph as an aid in naming the person. It is possible that certain people have been given incorrect names, especially if the original news caption was incorrect.

Significant efforts were made to combine all photographs of a single person into the same group under a single name. This was at times challenging, since some people showed up in the original captions under multiple names, such as "Bob McNamara" and "Robert McNamara". When there were multiple possibilities for a person's name, we strove to use the most commonly seen name for that person. For Chinese and some other Asian names, we maintained the common Chinese ordering (family name followed by given name), as in "Hu Jintao". Note that there are some people in the database with just a single name, such as "Abdullah" or "Madonna".

E. Cropping and rescaling

For each labeled face, the final image to place in the database was created using the following procedure. The region returned by the face detector for the given face was expanded by 2.2 in each dimension. If this expanded region would fall outside the original image area, then a new image of size equal to the desired expanded region was created, containing the corresponding portion of the original image but padded with black pixels to fill in the area outside the original image. The expanded region was then resized to 250 by 250 pixels using the function `cvResize`, in conjuction with `cvSetImageROI` as necessary. The images were then saved in the JPEG 2.0 format.

F. Forming training and testing sets

Forming sets and pairs for View 1 and View 2 was done using the following process. First, each specific person in the database was randomly assigned to a set. In the case of View 1, each person had a 0.7 probability of being placed into the training set, and in the case of View 2, each person had a uniform probability of being placed into each set.

The people in each set are given in `peopleDevTrain.txt` and `peopleDevTest.txt` for View 1 and `people.txt` for View 2. The first line of `peopleDevTrain.txt` and `peopleDevTest.txt` gives the total number of people in the set, and each subsequent line contains the name of a person followed by the number of images of that person in the database. `people.txt` is formatted similarly, except that the first line gives the number of sets. The next line gives the number of people in the first set, followed by the names and number of images of people in the first set, then the number of people in the second set, and so on for all ten sets.

Matched pairs were formed as follows. First, from the set of *people* with at least two images, a person was chosen uniformly at random (people with more images were given the same probability of being chosen as people with fewer images). Next, two images were drawn uniformly at random from among the images of the given person. If the two images were identical or if the pair of images of the specific person was already chosen previously as a matched pair, then the whole process was repeated. Otherwise the pair was added to the set of matched pairs.

Mismatched pairs were formed as follows. First, from the set of *people* in the set, two people were chosen uniformly at random (if the same person was chosen twice then the process was repeated). One image was then chosen uniformly at random from the set of images for each person. If this particular image pair was already chosen previously as a mismatched pair, then the whole process was repeated. Otherwise the pair was added to the set of mismatched pairs.

The pairs for each set are given in `pairsDevTrain.txt` and `pairsDevTest.txt` for View 1 and `pairs.txt` for View 2. The first line of `pairsDevTrain.txt` and `pairsDevTest.txt` gives the total number N of matched pairs (equal to the total number of mismatched pairs) in the set. The next N lines give the matched pairs in the format.

```
name    n1    n2
```

which means the matched pair consists of the `n1` and `n2` images for the person with the given name. For instance,

```
George_W_Bush    10    24
```

would mean that the pair consists of images `George_W_Bush_0010.jpg` and `George_W_Bush_0024.jpg`.

The following N lines give the mismatched pairs in the format

```
name1    n1    name2    n2
```

which means the mismatched pair consists of the `n1` image of person `name1` and the `n2` image of person `name2`. For instance,

```
George_W_Bush    12    John_Kerry    8
```

would mean that the pair consists of images `George_W_Bush_0012.jpg` and `John_Kery_0008.jpg`.

The file `pairs.txt` is formatted similarly, except that the first line gives the number of sets followed by the number of matched pairs N (equal to the number of mismatched pairs). The next $2N$ lines give the matched pairs and mismatched pairs for set 1 in the same format as above. This is then repeated nine more times to give the pairs for the other nine sets.

VII. Summary

We have introduced a new database, Labeled Faces in the Wild, whose primary goals are to

1) provide a large database of real world face images for the unseen pair matching problem of face recognition,
2) fit neatly into the detection-alignment-recognition pipeline, and
3) allow careful and easy comparison of face recognition algorithms.

We hope this will provide another stimulus to the vibrant research area of face recognition.

ACKNOWLEDGMENTS

First, we would like to thank David Forsyth for the original idea of building large face databases from web images. In addition to the authors of this report, the following people contributed to the construction of this database, in approximate order of their contribution: Vidit Jain, Marwan Mattar, Jerod Weinman, Andras Ferencz, Paul Dickson, David Walker Duhon, Adam Williams, Piyanuch Silapachote, Chris Pal, Allen Hanson, Dan Xie, Frank Stolle, and Lumin Zhang.

REFERENCES

[1] Anelia Angelova, Yaser Abu-Mostafa, and Pietro Perona. Pruning training sets for learning of object categories. In *CVPR*, volume 1, pages 495–501, 2005.

[2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projections. *IEEE Pattern Analysis and Machine Intelligence*, 19(7), 1997.

[3] Tamara L. Berg, Alexander C. Berg, Jaety Edwards, and David A. Forsyth. Who's in the picture. *NIPS*, 2004.

[4] Tamara L. Berg, Alexander C. Berg, Michael Maire, Ryan White, Yee Whye Teh, Erik Learned-Miller, and David A. Forsyth. Names and faces in the news. *CVPR*, 2004.

[5] David Beymer and Tomaso Poggio. Face recognition from one example view. Technical Report AIM-1536, MIT Artificial Intelligence Laboratory, 1995.

[6] Jeffrey F. Cohn, Adena J. Zlochower, James Lien, and Takeo Kanade. Automated face analysis by feature point tracking has high concurrent validity with manual FACS coding. *Psychophysiology*, 36:35–43, 1999.

[7] Andras Ferencz, Erik Learned-Miller, and Jitendra Malik. Building a classification cascade for visual identification from one example. In *ICCV*, 2005.

[8] Andras Ferencz, Erik Learned-Miller, and Jitendra Malik. Learning hyper-features for visual identification. In *NIPS*, volume 18, 2005.

[9] N. A. Fox, B. A. O'Mullane, and R. B. Reilly. The realistic multi-modal VALID database and visual speaker identification comparison experiments. In *Proceedings of the 5th International Conference on Audio- and Video-Based Biometric Person Authentication*, 2005.

[10] Wen Gao, Bo Cao, Shiguang Shan, Delong Zhou, Xiaohua Zhang, and Debin Zhao. The CAS-PEAL large-scale Chinese face database and baseline evaluations. Technical Report JDL-TR-04-FR-001, Joint Research and Development Laboratory (China), 2004.

[11] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[12] Daniel B. Graham and Nigel M. Allinson. Characterizing virtual eigensignatures for general purpose face recognition. In H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie, and T. S. Huang, editors, *Face recognition: From theory to applications, NATO ASI Series F, Computer and Systems Sciences*, volume 163, pages 446–456. 1998.

[13] Peter Hancock. Psychological image collection at stirling. http://pics.psych.stir.ac.uk/.

[14] Gary B. Huang, Vidit Jain, and Erik Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, 2007.

[15] Vidit Jain, Andras Ferencz, and Erik Learned-Miller. Discriminative training of hyper-feature models for object identification. In *BMVC*, 2006.

[16] Vidit Jain and Amitabha Mukherjee. The Indian Face Database. http://vis-www.cs.umass.edu/ vidit/IndianFaceDatabase/index.html, 2002.

[17] O. Jesorsky, K. Kirchberg, and R. Frischolz. Robust face detection using the Hausdorff distance. In J. Bigun and F. Smeraldi, editors, *Audio and Video Based Person Authentication*, pages 90–95. Springer, 2001.

[18] M. Kleiner, C. Wallraven, and H. H. Bülthoff. The MPI VideoLab - A system for high quality synchronous recording of video and audio from multiple viewpoints. Technical Report 123, Max Planck Institute for Biological Cybernetics, 2004.

[19] Michael J. Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with Gabor wavelets. In *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan*, pages 200–205, 1998.

[20] E. Marszalec, B. Martinkauppi, M. Soriano, and M. Pietikäinen. A physics-based face database for color research. *Journal of Electronic Imaging*, 9(1):32–38, 2000.

[21] A. M. Martinez and R. Benavente. The ar face database. Technical Report 24, Computer Vision Center, University of Barcelona, 1998.

[22] K Messer, J Matas, J Kittler, J Luettin, and G Maitre. Xm2vtsdb: The extended m2vts database. In *Second International Conference on Audio and Video-based Biometric Person Authentication*, March 1999.

[23] National Institute of Standards and Technology. The Color FERET Database. http://www.itl.nist.gov/iad/humanid/colorferet/home.html, 2003.

[24] Chinese Academy of Sciences National Laboratory of Pattern Recognition, Institute of Automation. Nlpr face database. http://nlpr-web.ia.ac.cn/english/irds/facedatabase.htm.

[25] Eric Nowak and Frédéric Jurie. Learning visual similarity measures for comparing never seen objects. In *CVPR*, 2007.

[26] Georgia Institute of Technology. The Georgia Tech Face Database. ftp://ftp.ee.gatech.edu/pub/users/hayes/facedb/.

[27] Derya Ozkan and Pinar Duygulu. A graph based approach for naming faces in news photos. In *CVPR*, 2006.

[28] P. Jonathon Phillips, Patrick J. Flynn, Todd Scruggs, Kevin Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, and William Worek. Overview of the Face Recognition Grand Challenge. In *CVPR*, 2005.

[29] Ferdinando Samaria and Andy Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of the Second Workshop on Applications of Computer Vision, Sarasota, Florida*, 1994.

[30] M. U. Ramos Sànchez, J. Matas, and J. Kittler. Statistical chromaticity models for lip tracking with B-splines. In *International Conference on Audio- and Video-Based Biometric Person Authentication*, 1997.

[31] C. Sanderson. *Biometric Person Recognition: Face, Speech and Fusion*. VDM_Verlag, 2008. ISBN 978-3-639-02769-3.

[32] Prag Sharma and Richard B. Reilly. A colour face image database for benchmarking of automatic face detection algorithms. In *Proceedings of EC-VIP-MC, the 4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications*, 2003.

[33] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *PAMI*, 25(12):1615–1618, 2003.

[34] Libor Spacek. University of Essex collection of facial images. http://cswww.essex.ac.uk/mv/allfaces/index.html, 1996.

[35] Paul Viola and Michael Jones. Robust real-time face detection. *IJCV*, 2004.

[36] Craig I. Watson. Nist mugshot identification database. http://www.nist.gov/srd/nistsd18.htm, 1994.

[37] B. Weyrauch, J. Huang, B. Heisele, and V. Blanz. Component-based face recognition with 3D morphable models. In *First IEEE Workshop on face processing in video*, 2004.