The Hidden Sides of Names - Face Modeling with First Name Attributes

Huizhong Chen, *Student Member, IEEE,* Andrew C. Gallagher, *Senior Member, IEEE,* and Bernd Girod, *Fellow, IEEE*

Abstract—This paper introduces the new idea of describing people using first names. We show that describing people in terms of similarity to a vector of possible first names is a powerful representation of facial appearance that can be used for a number of important applications, such as naming never-seen faces and building facial attribute classifiers. We build models for 100 common first names used in the United States and for each pair, construct a pairwise first-name classifier. These classifiers are built using training images downloaded from the internet, with no additional user interaction. This gives our approach important advantages in building practical systems that do not require additional human intervention for data labeling. The classification scores from each pairwise name classifier can be used as a set of facial attributes to describe facial appearance. We show several surprising results. Our name attributes predict the correct first names of test faces at rates far greater than chance. The name attributes are applied to gender recognition and to age classification, outperforming state-of-the-art methods with all training images automatically gathered from the internet. We also demonstrate the powerful use of our name attributes for associating faces in images with names from caption, and the important application of unconstrained face verification.

Index Terms—Facial processing, attributes learning, social contexts, multi-feature fusion.

1 Introduction

Expectant parents usually spend a great deal of time fulfilling their first official act - naming their baby. To the parents, choosing a name for the child may appear to be a choice from a near-infinite pool of possibilities. However, social context influences this decision, from the obvious factors (e.g., gender and ethnicity), to the less obvious ones (e.g., socio-economic background, popularity of names, names of relatives and friends). Consequently, first names are not distributed at random among the people in a society. As shown in Figure 1, a typical Alejandra appears to have a darker complexion and hair than a typical Heather, while Ethan mostly appears as a little boy since it is a recently popular male name. Taking these examples further, specific first names vary in prevalence even within a race. For example, though both of the following names are primarily Caucasian, the name "Anthony" has an Italian origin, and the name "Sean" has an Irish origin. We might expect different distributions of and correlations between facial shapes, complexions, and facial hair within even these two (primarily) Caucasian male first names. In a sense, each first name represents a joint distribution over a large set of facial attributes. In this work, we represent the appearance of many first names, and show that this name-based representation of facial appearance is a powerful face attribute.

This paper introduces, and then begins to answer, a new question in facial processing: Can we infer the name of a person from only a single photograph, and with no other image examples of that face? Of course, it is unrealistic to expect highly accurate performance at this task. After all, guessing the correct name for a never-seen face is an extremely challenging task. Nevertheless, such a system, even if imperfect, could have a broad range of applications in security (e.g., finding fake person identities from database) and biometrics (e.g., inferring the gender, age and ethnicity by guessing likely names of a face). In this work, we represent faces using first-name attributes and show superior performance of this modeling in various important applications, including gender and age classification, face verification, and associating faces with names. One compelling advantage of our approach is that the name models can be learned using the already name-tagged images from social media such as Flickr. Consequently, facial processing applications do not require additional human labeling to train first name attributes.

Our contributions are the following: First, we present the first treatment of first names as a facial attribute. Our model includes a novel matched face pyramid and Multi-Feature SVM representation, and has the advantage that all necessary training images and labels are mined from the internet. Second, our work is the first attempt of modeling the relation between first names and faces from a computer vision perspective, and we show that our constructed name

- *Huizhong Chen and Bernd Girod are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305. E-mail: {hchen2, bgirod}@stanford.edu*
- *Andrew Gallagher is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853. E-mail: acg226@cornell.edu*

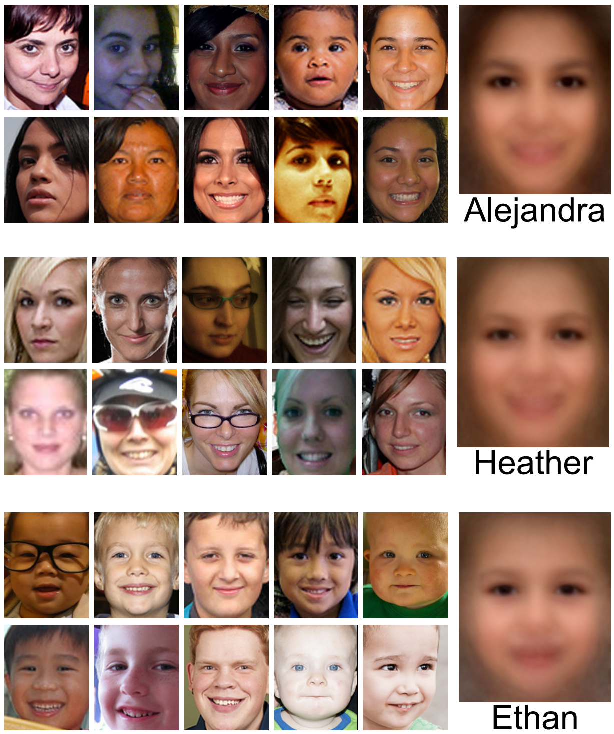

Fig. 1: Face examples of 2 female and 1 male names and their average faces computed from 280 aligned faces. Comparing the average faces, Alejandra (often Hispanic) has darker skin and hair than the average face of Heather (often Caucasian). In contrast, Ethan (a popular boy's name in recent years) has a much younger looking.

models are surprisingly accurate: guessing the correct first name at a rate greater than 4× the expected random assignment (and greater than 2× if gender is assumed to be known) from a pool of 100 choices. Third, we extend the recent work in [1], showing how our first-name attributes can be utilized on a series of important facial processing applications, such as face verification and assigning name tags to faces, often achieving state-of-the-art performance without the need for manually labeled training images. For the benefits of research community, we have released the dataset[1], the face pyramid feature extraction code, name prediction code, extracted facial features and trained first name models[2].

2 Related Work

This paper builds on recent ideas in the areas of computer vision (for learning mid-level features to represent object visual appearance) and social psychology (for investigating the social impact of first names). People's first names contain rich social contextual information, which has been extensively studied in social psychology but not explored in computer vision. In this section, we will first review the literatures that use mid-level attributes for representing images, with special discussions on those works that specifically use attributes to describe facial appearance. Then we present some social psychology studies that demonstrate the social contexts of names.

In computer vision, face detection and recognition achievements now date back around four decades [2]. Extensive studies have been made on numerically representing faces as features, from the earlier work of Eigenfaces [3], Fisherfaces [4] and Independent Component Analysis [5], to more recent developments of Local Binary Pattern [6], biologically inspired features [7], and sparse coding of low-level features [8]. Over the past few years, a thrust in computer vision concerns the representation of objects using meaningful intermediate features, i.e., attributes. Farhadi et. al. [9] learn discriminative attributes for objects and subsequently perform object categorization using the attributes representation. Torresani et. al. [10] also investigate the use of attributes for object categorization, but their attributes are emerged from the intersection of categorical properties and do not necessarily have semantic meanings. In [11] and [12], images are described by the outputs of a large number of object detectors (object bank) such as "person", "water", and "mountain" detectors. Using these object detector responses as attribute features, a simple SVM classifier is capable of achieving state-of-the-art performance on scene classification. Building from the work showing that attributes provide good descriptions of objects and scenes, several papers have shown advantages in describing facial appearance in terms of a large number of attributes [13], [14], [15] such as "male", "middle-aged", "asian". Describing people by semantic facial attributes is an intuitive and powerful technique, but in order to train a set of models for recognizing facial attributes, a large training set must be manually labeled for each attribute at high cost. Further, because the attributes are learned independently, the relationships and correlations between the attributes must also be modeled to improve performance. In this work, first names are treated as attributes of faces, and the representation implicitly jointly models age, gender, race, and other (possibly unnamed) appearance attributes associated with the people having that first name (Figure 1). Our work has a flavor similar to [16], where Berg and Belhumeur applied pairwise person classifiers to the task of face verification. Nevertheless, each of their person classifiers was trained using faces of two specific individual persons, which drastically differs from our approach that trains models on face images sampled from first names.

In [17], names from captions are matched to the faces in the image based on 2 attributes: gender and age (derived from facial analysis from images, and from records of name popularity over time). However, we have found that people's first names imply more information beyond gender and age. In this paper, we extend attributes far beyond the simple modeling of

1. Names 100 Dataset available at: http://purl.stanford.edu/tp945cq9122

2. Code, extracted features and trained models available at: http://purl.stanford.edu/fb872mg3286

faces using gender and age attributes, to an appearance model of what distinguishes first names from one another.

At a first glance, it might seem odd to expect that learning appearance models for different first names would be a fruitful strategy for facial appearance modeling. However, social psychology shows two important results regarding names. First, it shows that names matter and affect the lives of the people to whom they are assigned [18], [19], [20], [21], [22], [23]. Second, people themselves employ stereotypical models for names that even affect their perception of attractiveness and appearance [24], [25]. Building on the findings of these studies, in this work, we also demonstrate the power of first name attributes via a series of facial analysis experiments.

As shown in [20], juvenile delinquents do not have the same name distribution as the general population, even after controlling for race. Unpopular names, also correlated with a lack of education, are more common in the delinquent population. Further, [18] shows that first names associated with lower socio-economic status (e.g., names with an apostrophe, with a high "Scrabble score", or having other attributes) result in both lower standardized test scores and lower teacher expectations, even after using sibling pairs to control for race and socio-economic status.

The name a person receives at birth also affects that person's preferences and behaviors. Letters belonging to the first or last name are preferred above other letters [21]. This preference appears to transcend the laboratory and influence major life decisions. In a series of papers, Pelham, Jones, and collaborators call the effect *implicit egotism*, the gravitation towards people, places and things that resemble the self. People disproportionately choose spouses with names similar to their own [19]. For example, Eric marries Erica at a greater than the expected rate. People have careers and states of residence that are similar in sound to their names at disproportionate rates [22]. For example, Dennis is more likely to be a dentist than expected by chance, and more people with surnames beginning with Cali- live in California than expected by chance. This line of investigation is extended to towns of residence and street names in [23].

People have stereotypical ideas about names, and the appearance of people with those names. In one study [24], girls' photographs were rated for attractiveness. Those photos assigned desirable names (at the time, Kathy, Christine, or Jennifer) were rated as more attractive than those assigned less desirable names (Ethel, Harriet, or Gertrude) even though the photographs were ranked as equally attractive when no names were assigned. In another relevant study [25], subjects first used facial manipulation software to produce stereotypical face images for 15 common male names (e.g. Andy, Bob, Jason, Tim) by varying facial features. Additional subjects are able to identify the prototype names for each face at rates far above random guesses (10.4% vs. 6.7%) and for 4 of the 15 faces, the majority vote name was correct. This strong evidence provides motivation for us to learn, from actual images, visual models for first names.

3 NAMES 100 DATASET

To model the relation between names and appearance, we assembled a large dataset by sampling images and tags from Flickr. The dataset contains 100 popular first names based on the statistics from the US Social Security Administration (SSA) [26], with 800 faces tagged for each name. The 100 names were selected as follows: First, we ranked the names from the SSA database in order of the total number of times each name was used between 1940 and the present. Then, the top names for males and females were found. In turn, first names were used as a Flickr query, and names for which enough ($\geq$ 800) image examples were found were kept in the dataset. The completed dataset includes 48 male names, 48 female names, and 4 neutral (a name held by both males and females) names to model the real-world distribution of names. In Figure 11, we plot the average face of each name by taking the mean of its 800 facial images. Our Names 100 dataset covers 20.35% of U.S. persons born between 1940 and 2010. We use the name as a keyword to query Flickr and enforce the following criteria when sampling images, in an effort to sample first-name appearance space as fairly as possible: First, since name ambiguities arise when multiple people are present in an image, we run a face detector [27] and eliminate those images that contain multiple faces, and check if there exists one and only one first name in the image tag. Second, we filter out images that are tagged with any of 4717 celebrity names that could bias the sampling. Without this consideration, a query of "Brad" would return many images of the movie star "Brad Pitt", and distort the facial appearance distribution of the name "Brad". Last, no more than one image is downloaded per Flickr user. This rule is meant to prevent multiple instances of a person "David", when "David" appears in many images of a particular Flickr user. While these rules may not be sufficient to prevent all instances of either incorrectly named faces, or different images of the same person appearing more than once, they are effective at preventing many problems that more naïve strategies would encounter, and we found them to be effective. Inevitably, almost all datasets have biases, and we acknowledge several biases of this dataset. Firstly, we are limited to only the pictures taken by Flickr users so this could eliminate a large demography. Also, as we run a face detector on the images, the detected faces are mostly facing towards the camera so there are no faces from a side view. Lastly, the tags on Flickr could be noisy, for example, an image of "David's Mom" may mislead us to use "David" as the name label for this image. To study the noisy of our dataset, we sampled a subset of 10,000 images from our dataset and manually checked whether the face image agrees with the name tag, where we found 5.86% of the 10,000 images contain name tags that may not agree with the facial appearance. The sources of errors include false detected face, gender-name disagreement, drawings, and artistically manipulated faces. Assuming the dataset were clean where all faces are labeled with their actual first names, the performance of our first name attributes would be further improved as our models could be trained using clean data.

4 Describing Faces with First Name Attributes

Figure 2 shows an overview of the our system. First, the faces are normalized for scale with detected eye positions [28] and resampling the face to 150×120 pixels. We extract SIFT descriptors [29] by sampling on a dense grid with 2-pixel intervals. Each 128-dimensional SIFT descriptor is then encoded by the Locality-constrained Linear Coding (LLC) method [30] to a 1024-dimensional code. These encoded LLC codes are aggregated over a spatial pyramid [31] using max pooling, such that we have a 1024-dimensional vector at each of the 21 pyramid grid. This produces a feature vector of $21 \times 1024 = 21504$ dimensions for each face.

For each pair of first names, we then build a Support Vector Machine (SVM) [32] classifier to discriminate between that pair of names (more details on classifier construction are in Section 5). Therefore, classifying N names requires $\frac{N\times(N-1)}{2}$ pairwise classifiers. This 1-vs-1 classifier construction [33] is common for multi-class problems, and particularly relevant for distinguishing between first names. The visual features that distinguish any particular pair of individuals varies. For example, "David" and "Mary" differ in gender, but "David" and "Ethan" differ mainly in age ("Ethan" is a younger name). We also experimented with using a 1-vs-all approach for classifier training, and found the results to be inferior to the 1-vs-1 classifiers. Using these pairwise name classifiers, a test face can then be described by a vector of $\frac{N\times(N-1)}{2}$ dimensions, each being an SVM output score indicating whether the name of the face is more likely to be the first or the second in the name pair. We call this feature vector the *pairwise name attribute* representation of the face. In our case of 100 names, the pairwise name attributes is a 4950 dimensional feature vector.

The pairwise name attributes establish the link between a face and the names that best fit its appearance, which naturally leads to many interesting applications as we describe in Section 6. We show that our system accomplishes the obvious task, guessing the first name of a person, at rates far superior to random chance, even after accounting for the effects of age and gender. We then describe an application of gender classification based on our pairwise name attributes, which achieves state-of-the art performance. Further, we demonstrate that the pairwise name attributes are very effective on the task of age classification.

It is important to point out that our entire system, with the exception of training for face verification (see Sec. 6.6), requires no human labeling beyond the existing Flickr name tags. This gives our system several unique advantages. First, it is inexpensive to deploy. By not requiring any additional human labels, we do not need to pay human workers and we avoid costs associated with training workers. The labels that we do use (first names tagging the images) are freely provided on the Internet because they already provide value for searching and sharing the images. Second, because our system is driven by first names as attributes, we avoid semantic issues related to attribute tagging (e.g. ideas about what constitutes "attractive" vary between observers). Finally, our system is easily extensible. Although, for now, we explore the popular first names from the United States, extending the system to other cultures is as easy as performing additional image downloads with additional name queries as search terms.

5 Pairwise Name Classification using Multi-Feature SVM

As mentioned in the previous section, we compute a $21 \times 1024 = 21504$ dimensional feature vector for each face. Conventionally, as has been done in [30], this extremely high dimensional vector is directly fed to an SVM for classification. However, performing classification in such a high dimensional feature space is susceptible to overfitting, especially on our challenging classification task of assigning first names to faces. Therefore, instead of simply concatenating the 1024 dimensional LLC codes from all 21 pyramid bins, we regard a face as represented by 21 feature vectors, each vector coming from one pyramid bin. In this way, the 21 feature vectors can be viewed as coming from 21 feature channels that are complementary to each other, and we propose a method called Multi-Feature SVM (MFSVM) that effectively fuses the features together to achieve a better performance on the task of first name classification.

Our MFSVM follows the framework of AdaBoost [34], with the classifiers being SVMs working on different feature channels. To begin, we initialize equal weights on all training images and use feature channel 1 to perform a 5-fold cross validation using SVM. The misclassified training images with that SVM are given higher weights when training the SVM for feature channel 2. Intuitively, the SVM for feature channel 2

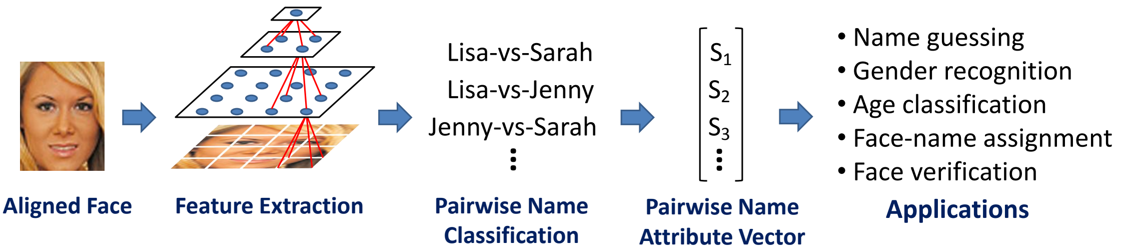

Fig. 2: Overview of our system. First, a query face is represented as a 3-level pyramid of max-pooled LLC codes, with 1 pyramid grid at the top level, 4 at the next, and 16 at the bottom level. Next, the face is classified in a 1-vs-1 fashion with a set of pairwise name classifiers. The pairwise name classifiers output confidence scores which we call pairwise name attribute vector, which can be used for many applications as we will show Section 6.

will focus on the highly weighted mis-classified images from feature 1's SVM. This procedure is repeated until we have trained an SVM on each of the feature channels.

More formally, suppose there are T feature channels and N training images, we denote the training data as $x_{t,i}$, where $t = 1, ..., T$ and $i = 1, ..., N$, meaning the t-th feature extracted from the i-th training image. Each training image is associated with a training label $y_i \in \{-1, +1\}$. For a test image, the testing data is z_t. The MFSVM is shown in Algorithm 1.

Data: Training data $x_{t,i}$, training labels $y_i \in \{-1, +1\}$, testing data z_t, where $t = 1, ..., T$ and $i = 1, ..., N$
Result: SVM classifiers $f_t(z_t)$, classifier weights α_t
Initialization: weights $D_i = 1$;
for $t = 1 : T$ **do**
(a) Using weights D, perform SVM cross validation to obtain confidence $f_t^{cv}(x_{t,i}) \in R$ and prediction $\hat{y}_{t,i}^{cv} = sign(f_t^{cv}(x_{t,i}))$, compute error $err_t = \frac{\sum_{t=1}^{N} \mathbb{I}\{\hat{y}_{t,i}^{cv} \neq y_i\}}{N}$;
(b) Train SVM f_t using D;
(c) Compute $\alpha_t = \frac{1}{2}\log(\frac{1-err_t}{err_t})$;
(d) Set $D_i = D_i \exp(-\alpha_t y_i f_t^{cv}(x_{t,i}))$, and renormalize so that $\sum_{t=1}^{N} D_i = N$;
end
Output the final classifier $f_{all}(z) = \sum_{t=1}^{N} \alpha_t f_t(z_t)$

Algorithm 1: Multi-Feature SVM

In practice, we fuse the 21 features channels from coarse to fine grids on the face image pyramid. In our experiments we find that the ordering does not have much effect on the performance. On average, the pairwise name classifiers perform quite well at distinguishing between first names as shown in Table 1. As expected, it is easier to classify between names that differ in gender. We also found, within each gender, the pairs of names that are easiest and hardest to distinguish, see Table 1. Easy-to-distinguish name pairs tend to have different ages (Figure 3). Name pairs that are hard to distinguish tend to have similar popularity patterns.

6 Applications of Name Models

In this Section, we explore the performance of our pairwise name representation for a variety of tasks. We first show that the name models are surprisingly accurate on the task of first name prediction, then propose novel applications that utilize name attributes for gender recognition, age classification, face-name assignment and face verification.

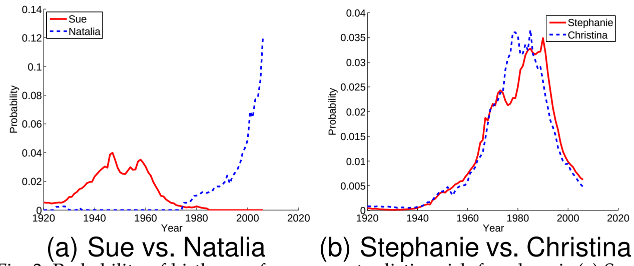

Fig. 3: Probability of birth year for an easy to distinguish female pair (a) Sue vs. Natalia: accuracy of 76.3%, and a difficult to distinguish pair (b) Stephanie vs. Christina: accuracy of 46.1%.

TABLE 1: A summary of the performance of the pairwise name classifiers. The top four rows summarize the overall performance at distinguishing between two names. The bottom four rows show the most and least accurate pairwise name classifiers when classifying between two mostly male or two mostly female names. Mike vs. Brian and Stephanie vs. Christina are indistinguishable to our classifier (which performs at the level of random chance) because the gender, age, and ethnic makeup of the samples with those name pairs are so similar. For all rows, random chance results in a 50% accuracy.

	Accuracy	STD
Overall	69.4%	11.1%
Male-Female	79.5%	4.0%
Male-Male	59.5%	6.4%
Female-Female	59.1%	5.0%
Best Male: Noah vs. Steve	79.3%	
Best Female: Sue vs. Natalia	76.3%	
Worst Male: Mike vs. Brian	45.9%	
Worst Female: Stephanie vs. Christina	46.1%	

6.1 First Name Prediction

First name predictions are derived from the pairwise name attributes as follows: Each first name is associated with $N - 1$ pairwise name classifiers. The total name margin for a particular name is produced by marginalizing over each associated pairwise name classifier. By sorting the first names according to the total name margins, a rank-ordered list of first names is produced.

We evaluate the performance of first name predictions on our Names 100 dataset by 5-fold cross validation. The dataset contains 100 names $\times$ 800 faces/name $= 80,000$ faces. In each fold we test on 16,000 faces with equal number of testing examples per name, while varying the number of training examples to study the effect of training data size on the name prediction performance.

Fig. 4: The top-3 predicted names for some face images. The correct prediction is highlighted in green, while the actual first name is shown in red if it is not ranked within the top-3 predictions. The first 2 rows give some good examples where our top-3 predictions include the actual name, and the bottom 2 rows are randomly selected from our test set. Even when our predictions are wrong, reasonable names are predicted (e.g., appropriate gender or age).

6.1.1 Low-level features

We are first interested in comparing the low-level feature of our choice, LLC coded SIFT feature [30], with Local Binary Patterns (LBP) [6] and Histogram of Orientated Gradients (HOG) [35] since these two features are very popular in the face recognition community. The details for extracting LLC feature were described in section 4. For LBP, we divide the face into blocks of 10×10 pixels and extract 59-dimensional LBP code from each block. For HOG, a 36-dimensional histogram is extracted from each cell, using the default cell size of 8×8 pixels. We compare the performance of these three features in Figure 5, by plotting the learning curves of top-1 prediction accuracy and Mean Average Precision (MAP). It can be seen that the LLC feature outperforms LBP and HOG by a large margin, thus the LLC feature is chosen as our low-level feature.

6.1.2 MFSVM Evaluation

We compare our proposed MFSVM method to the method in [30] where the LLC feature is directly used by a single SVM. The learning curves of top-1 prediction accuracy, Mean Average Precision (MAP) are shown in Figure 6. We also include the plot of Recall vs. Number of name guesses by performing the 5-fold cross validation over the 80,000 faces in our dataset. As can be seen in Figure 6, our MFSVM classifiers fuse the 21 max-pooled LLC codes from the face pyramid and offer a significant performance gain over the original LLC method in [30]. With 640 training images per name, we achieve 4.17% top-1 prediction accuracy and 0.117 MAP, which is far better than the random guess performance of 1.00% accuracy and 0.052 MAP. Table 2 shows the performance of our model for guessing first names as a function of the number of names. Some examples of first name predictions are shown in Figure 4.

TABLE 2: Performance of our approach for guessing first names given randomly selected subsets of N names.

Number Names N	5	10	40	70	100
Random Guess	20.0%	10.0%	2.50%	1.43%	1.00%
Our approach	**39.4%**	**23.5%**	**8.19%**	**5.41%**	**4.17%**

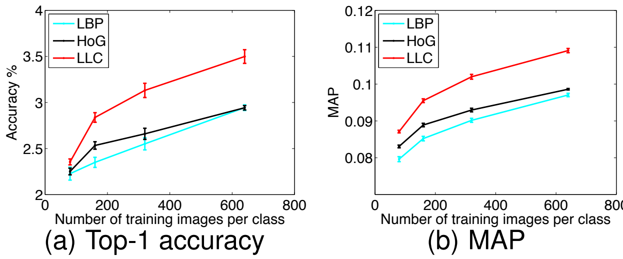

Fig. 5: Evaluation of different facial features on the Names 100 dataset. The task is to predict the first name of a previously unseen face from 100 choices. The LLC feature significantly outperforms LBP and HOG, thus the LLC feature is chosen as our low-level feature.

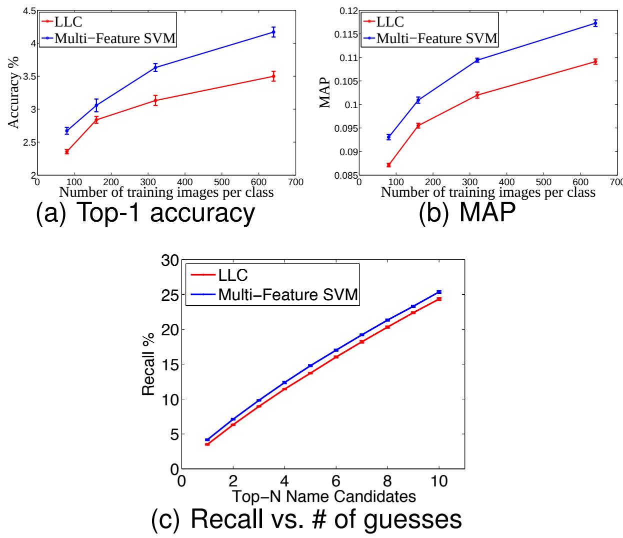

Fig. 6: Evaluation of first name prediction on the Names 100 dataset. The task is to predict the first name of a previously unseen face from 100 choices. The results of both the MFSVM classifier and the LLC method are far better than random guess (MFSVM accuracy 4.17% vs. random accuracy 1%, MFSVM MAP 0.117 vs. random MAP 0.052), with MFSVM showing improved performance over the LLC method. (c) illustrates the recall of correct name as the number of guesses changes.

6.1.3 Effects of Gender and Age on Name Prediction

How is it possible that names can be guessed more than $4\times$ better than random? It is because names are not randomly distributed across people, and many correlations exist between given names and various facial features (e.g., skin color, male-ness, facial feature size, age, and possibly even nameless attributes [36]).

To more thoroughly investigate the relationship between names and faces, we examine a baseline of estimating gender and age for the task of name prediction. In other words, how accurately can we guess a first name, given only the estimated age and gender of the face? We train gender and age classifiers using the Group Image Dataset [37], a dataset which contains a total of 5,080 images with 28,231 faces manually labeled with ground truth gender and coarse age categories (age categories include 0-2, 3-7, 8-12, 13-19, 20-36, 37-65, 66+). We construct the gender and age classifiers in the exact same manner as we train the name models, by first extracting max-pooled LLC codes on the face pyramid, then passing the features to MFSVM classifiers and finally marginalizing the outputs from the classifiers. Having trained the gender and age classifiers, we use them to predict the gender and age of the faces in our Names 100 dataset. The gender and age predictions associated with a testing face are not independent of first name, hence considering these features offer a better performance than random guess. First names are predicted from gender and age estimates as follows: Considering estimated gender, if a test face is classified as a male, then we make a random guess among the male names. Considering estimated age category, we compute the range of predicted birth years by subtracting the predicted age from the image taken year. Since each name has a birth year probability distribution over time (see Figure 7), the first name is predicted as the name that has the maximum birth probability within the range of predicted birth years. We can also combine gender and age, by incorporating the estimated age information to make first name guess only within the subset of names selected by the estimated gender. Table 3 compares our name models trained using 640 images/name to the baseline performances achieved by considering estimated age and gender as described above. Our name models achieve superior performance (4.17%), even versus the baseline that combines both gender and age classifiers (2.33%). This observation shows the advantage of our approach that directly constructs appearance models for first names, rather than introducing an intermediate layer of variables (e.g., gender and age) to learn the relation between names and their facial appearances. In other words, our name models capture visual cues beyond just age and gender.

TABLE 3: Comparison of our approach to the methods of including gender and age effects on first name prediction. By directly modeling names and faces, we achieve much better performance even when gender and age effects are taken into account.

Method	Prediction accuracy	MAP
Our approach	**4.17%**	**0.117**
Gender $\rightarrow$ name	1.61%	0.075
Age $\rightarrow$ name	1.37%	0.063
Gender + age $\rightarrow$ name	2.33%	0.089
Random guess	1.00%	0.052

6.1.4 Human Evaluation

We additionally evaluated the human performance on guessing first names via Amazon Mechanical Turk. The test samples include 2000 male and female face images from our Names 100 dataset, and we have 3 workers work on each image. As it is unrealistic to ask human to select 1 name out of the 100 names, we show a face with 10 possible names, where the names include the correct name and 9 other random names of the same gender in random order. The human prediction accuracy is 13.7% with $\pm$0.87% margin of error for a 95% confidence interval (compared to the random baseline of 10%), compared to our method that achieves 18.2% accuracy within the 10 selected names, with margin of error being 1.4% at 95% confidence interval.

6.2 Gender Recognition From Names

Using our first name attributes, we are able to construct a state-of-the-art gender classifier by exploiting the fact that many first names have a strong association with gender. Intuitively, if a face seems more like an "Anthony" than an "Anna" then it is more likely to be the face of a male. Our gender classifier works as follows: First, we produce the pairwise name attribute vector for each test face. Next, we order the first names by their total name margins as described in Section 6.1. Finally, we classify the gender of the test face as male or female depending on the gender associated with the majority of top 5 names in the ordered list of 100 first names. A neutral name is counted as either a male or a female name based on the gender ratio of that name, which is computed with SSA database [26] statistics.

We evaluate the gender recognition performance on the Group Image Dataset [37], which contains faces with a large variation of pose, illumination and expression. We benchmark our gender-from-names algorithm against Kumar's method of [15], using the "gender" attribute predicted from their system. Kumar's facial attributes system runs their own face detector, which correctly detected 22,778 out of 28,231 faces from the Group Image Dataset, and we filtered out their falsely detected faces with the ground truth face positions. We compare the gender classification algorithms on these 22,778 test faces. As reported in Table 4, our method outperforms the result of [15], and achieves a gender classification accuracy of 90.4%, which is an impressive 29% reduction in error. It is important to again note that our gender classifier uses name models trained with names freely available on the web, and does not require *any* manually labeled gender training examples. As another comparison, we use the MFSVM approach to train gender classifier on images from [37] with human annotated gender labels. This strongly supervised classification scheme achieves 89.7% accuracy from 2-fold cross validation,

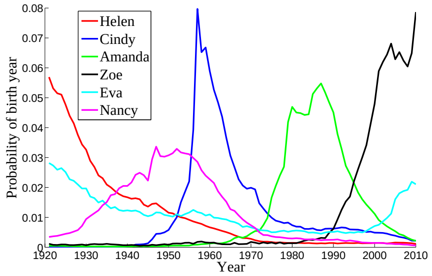

Fig. 7: The birth year probabilities of a set of names, where many names show varying popularity over the years.

still below our 90.4%, and again, our method has the benefit of not requiring manual labels.

TABLE 4: Without any gender training labels, we perform gender recognition using our name models and achieve state-of-the-art performance.

Algorithm	Gender recognition accuracy
Gender-from-names	**90.4%**
Kumar et. al. [15]	86.4%
Prior	52.4%

6.3 Age Classification From Names

Due to the evolution of culture and occurrence of significant events, the popularity of a name varies over time. We use the statistics from the SSA database to plot the birth year probabilities of several names in Figure 7, where it can be seen that the birth probabilities of names have large fluctuations over the years. If a person is named Zoe, she is likely to be young because the name Zoe became popular during the 1990s. Thus, once we are able to describe a test face with our first name models, then we can utilize the birth probability of names to predict the age of the face. The advantage of such an age-from-names approach is obvious: as with our gender classifier, we again do not require any age ground truth labels to produce a reasonable age classification.

Our age-from-names classification works by first generating a ranked list of 100 names for a test face (again following Section 6.1), using the 4950 pairwise name models trained for first name prediction. We also compute the birth year probabilities from 1921 to 2010 for these 100 names, using the SSA baby name database. Certainly, the names ranked at the top of the list should be given higher weights for the task of age classification. Therefore we assign exponentially distributed weights to the ranked 100 names, such that the i-th name is associated with a weight of $\omega_i = \lambda e^{-\lambda i}$, where $\lambda = 10$. Denoting the birth probability of the i-th ranked name in year j as $p_i(j)$, then the birth probability of the ranked 100 names are combined using weighted product:

$$p_{\text{combined}}(j) = \frac{\prod_{i=1}^{100} p_i(j)^{\omega_i}}{Z} \quad (1)$$

where $Z = \sum_j p_{\text{combined}}(j)$ is a normalization term.

Each test image contains a time stamp in its JPEG metadata, so we know the year that the image was taken. Suppose that the test image was taken in the year 2008 and we believe the face falls into the age category of 20-36, then the person should be born within the year range of 1972 to 1988. We assign the confidence score for the face belonging to the age category of 20-36 as the mean of the combined birth probability over the proportion of the years 1972 to 1988. The confidence score can be written as:

$$\text{Confidence of age } t_1 \text{ to } t_2 = \frac{\sum_{j=s-t_2}^{s-t_1} p_{\text{combined}}(j)}{t_2 - t_1 + 1} \quad (2)$$

where s is year that the image was taken, t_1 and t_2 specify the lower and the upper bound of the age category respectively.

Once again, we evaluate our age classification performance on the Group Image Dataset. Equation (2) is employed to compute the confidence scores for the 7 age categories of 0-2, 3-7, 8-12, 13-19, 20-36, 37-65, 66+, as specified in the dataset. The age category with the largest confidence score will be picked as the age prediction for the test face. We work on the same test partition that was used in [37], where there are an equal number of testing instances for each age category (1050 images total). Table 5 reports the accuracy for exact category match, as well as the accuracy when an error of one age category is allowed (e.g., a 3-7 year old classified as 8-12). We benchmark our age-from-names classifier against the performance of [37], where our system shows a significant improvement. When allowing an error of one age category, our age classifier achieves 88.0% accuracy, which is surprisingly good given the fact that we are simply utilizing the age information hidden inside the names and use no other manually labeled information. While we are pleased with this performance, we do not claim to have state-of-the-art accuracy for age estimation [38], which currently relies on manifold learning and regressing using training images for which the actual age of each face is known. We do claim to have the most accurate (and only) age estimation method for which no age labels are provided for the training images.

TABLE 5: We perform age classification using the birth probability of names over years 1921-2010. Without any age training labels, our age classification result shows significantly improved result compared to [37].

Algorithm	Accuracy for exact match	Allow error of one age category
Age-from-names	**41.4%**	**88.0%**
Gallagher & Chen [37]	38.3%	71.3%
Random Prior	14.3%	38.8%

6.4 Cross-ethnicity Experiments

Our name attributes are built on 100 popular first names in the United States. It is interesting to see whether these first name attribute models generalize well to faces from across other cultures and ethnicities. To investigate this, we apply our trained name attributes models to Asian faces to study this cross-ethnicity problem. Our goal is to perform gender and age recognition on Asian faces without any re-training of the name models.

For evaluation purposes, we need a dataset that contains Asian faces with labeled gender and age information. While there are a number of face datasets that contain Asian faces, most of them cannot be used for our task because of the limited number of human subjects, or having little variation on gender or age. We chose the FaceTracer Database [13] to perform the cross-ethnicity experiments, as it contains Asian faces across different gender and age groups taken in unconstrained settings. In the FaceTracer Database, 200 faces were labeled "Asian", but we can only collect 150 of these faces due to invalid image links. Following the dataset labeling procedure used in Gallagher's Group Image Dataset [37], we manually annotated the gender and categorical age labels (7 age categories) for each of these Asian faces. Our gender-from-names and age-from-names classifiers are then tested on the 150 labeled faces. The gender and age classification accuracies are tabulated in Table 6. Compared to the gender and age classifier performance on the Group Image Dataset (mostly Caucasian faces) in Table 4 and Table 5, we do not observe major degradation of performance by testing on Asian faces. This is encouraging and suggests that our first name attributes generalize well in the cross-ethnicity settings.

TABLE 6: Accuracies of gender-from-names and age-from-names classifiers.

Algorithm	Accuracy for exact match	Allow error of one age category
Gender-from-names	86.0%	NA
Age-from-names	43.3%	90.7%

6.5 Beyond Names 100

As indicated in Section 3, the 100 names used in our Names 100 dataset covers 20.35% of U.S. population born between 1940 and 2010. In situations where the classification task is to select the most probable name for a face from a candidate subset of the 100 names, the procedure is obvious: the pairwise SVM scores between the name candidates can be directly employed, as described in Section 6.1.

But what about the situation where the name candidates are not included in Names 100? Although we have the advantage that all training images and name tags can be crawled from the web, it is still difficult to learn face models for every existing name in the real world. It will be nice if we can handle names outside our Names 100 dataset, so that we can apply our first-name models to many more real-world situations. For example, we consider the name-face assignment problem [17] as shown in Figure 8: an image of a group of people has first name tags associated with it, but the exact mapping between the names and faces is

(a) Chris and Dana

(b) Barry, Nancy and Andrew

Fig. 8: Illustration of the face-name assignment problem. For an image with multiple faces and multiple name tags, human can work out the exact mapping between faces and names at a certain level of accuracy. In (a), the image contains Chris and Dana, both names suitable for either gender (though, in this case, Dana is actually the female). There are also more challenging cases like (b), where the names of "Barry" and "Andrew" may cause confusion for human to make the correct face-name assignments.

ambiguous. Such a scenario is extremely common in personal photo collections such as Flickr and Adobe Albums, and it is valuable to have an algorithm that associates the name tags with the faces appearing in the image. Certainly, our set of 100 names may not include all the person's names that appear in these images, thus we need a method that is capable of dealing with names outside Names 100.

6.5.1 Face-name compatibility

We propose the following method for producing a face-name compatibility score for a query face i and any given name n. First, a set of M face images having the name n are downloaded from Flickr by the method described in Section 3. Next, following the procedure of pairwise name attributes computation in Section 4, we obtain a pairwise name attribute vector $f(m)$ for each of the M downloaded face images, and the pairwise name attribute vector $f(i)$ for the query face. Finally, we define face-name compatibility score as the median of the cosine similarity scores between $f(i)$ and $f(m)$ for $m = 1, ..., M$, which can be expressed as follows:

$$\text{Face-name comp. score} = \text{median}(s_1, ..., s_M) \quad (3)$$

$$\text{where } s_m = \frac{f(i) \cdot f(m)}{\|f(i)\|_2 \|f(m)\|_2} \quad (4)$$

The design of this face-name compatibility score follows a simple intuition: we expect that if the name and query face are true matches, then the query face should have a similar pairwise name attribute representation to the face examples of that first name. Intuitively, the face-name compatibility score indicates the compatibility between a putative face-name pair. We now demonstrate such a face-name compatibility score works surprising well on the application of face-name assignment.

6.5.2 Face-name assignment

We use Gallagher's dataset [17] to evaluate the performance of face-name assignment using our pairwise name attributes. Our goal is to use our pairwise name attributes to disambiguate the tags by assigning names to people based on a single image. Gallagher's

Fig. 9: Some examples of our face-name assignments. The 3 rows of examples correspond to randomly selected images from Set A, Set B and Set C of the Gallagher dataset. For each image, the names assigned by our system correspond to the faces in the left to right order. When our face-name assignment is incorrect, the actual name is highlighted in red.

dataset has 148 images with a total number of 339 faces, where each image contains multiple people and name tags. This dataset contains 3 sets: Set A images have independent name pairs that may or may not have the same gender; Set B contains names that are less popular and may cause difficulty for human to perform the name assignment task; Set C contains all those images from Sets A and B where all people have the same gender.

Following our dataset preparation procedure in Section 3, for each of the 154 names in the Gallagher dataset, we crawled image examples from Flickr. After the image collection, 151 of these names have more than 100 face images, while 3 less popular names have fewer face images (Dolan, Graydon, and Jere have 70, 28, and 97 examples respectively). In the interest of fast computation, we use 100 training faces for each of the 151 names, while keeping all the face examples for the 3 less popular names. Next, as previously mentioned in Section 6.5, we compute the pairwise name attributes for the face images downloaded from Flickr, which gives a 4950-dimensional attribute vector per face. For a test face, we also compute its pairwise name attribute vector, and subsequently obtain the face-name compatibility score between the test face and any of the 154 names by Equation (3).

Consider the face-name assignment problem, where an image contains F people and N first name tags. There are $\max(F, N)!$ possible face-name combinations and our goal is to find the correct face-name assignment for this image. A simple strategy is to exhaustively search through all possible face-name combinations and pick the combination that maximizes the sum of face-name compatibility scores for all face-name pairs in the image. As a more computationally efficient solution, we treat this assignment problem as a bipartite graph and use the Kuhn-Munkres algorithm [39], which solves the best face-name assignment problem in $O(\max(F, N)^3)$.

In Table 7, we compare the performance of our face-name assignment algorithm against the method of [17] on the Gallagher Dataset, as well as the human performance reported in that paper. From Table 7, it is clear that our approach significantly outperforms the method of [17] on the task of face-name assignment. The reason behind this significant improvement in performance is because [17] tries to encode the relation between facial appearance and first names via 2 variables (i.e., gender and age). Compared to our work, we are directly learning the face-name relation from actual images, thus extending the face-name relation modeling far beyond the simple "gender + age" model used in [17]. Even more encouraging, as tabulated in Table 7, our algorithm is comparable to (sometimes even better than) the human performance, which demonstrates that our pairwise name attributes are extremely powerful for the application of face-name assignment. Some face-name assignment examples are shown in Figure 9.

TABLE 7: Accuracy of face-name assignment on the Gallagher dataset.

	Set A	Set B	Set C	Overall
Our algorithm	**80.1%**	78.1%	62.9%	**79.9%**
Gallagher & Chen [17]	62.2%	56.3%	61.9%	61.7%
Human subject 1	79.2%	**81.3%**	**65.7%**	79.4%
Human subject 2	78.2%	68.8%	61.0%	77.3%
Human subject 3	79.5%	43.8%	54.3%	76.1%
Human subject 4	69.1%	53.1%	41.9%	67.6%
Random prior	43.7%	43.8%	45.7%	43.7%

6.6 Unconstrained Face Verification

Face recognition is widely regarded as one of the most well studied areas in computer vision, yet recognizing faces under unconstrained environments still remains an unsolved problem. Unconstrained face recognition is challenging, due to the fact that human faces typically have large variations of poses, illuminations and expressions. In an effort to drive research forward in this field, the Labeled Faces in the Wild dataset (LFW) [40] was proposed as a standard benchmark and specifically focus on the problem of unconstrained face verification: classifying whether a given pair of faces are the same, or are different people. To evaluate face verification algorithms, 10-fold cross validation is performed on 6,000 matching and non-matching face pairs and the classification accuracy is reported.

In [15] Kumar et al. proposed the idea of using describable visual attributes to model faces. Their visual attributes include 2 sets of attributes: 1) 73 Facial attributes like "mustache" and "eyeglasses" that describe the face appearance; 2) Simile attributes that describe the similarity of a part of a person's face to the same part of reference people, e.g., "the nose of this face is similar to Brad Pitt's nose". Kumar showed that the descriptive visual attributes are effective tools for performing face verification. Motivated by their success, we are interested in investigating the effectiveness of our pairwise name attributes on unconstrained face verification.

For a given pair of faces x_1 and x_2 in the LFW dataset, we first generate the pairwise name attributes $f(x_1)$ and $f(x_2)$ for these two faces. The distance between these two name attribute vectors indicate whether this pair of faces belongs to the same person. Then, following Kumar's approach in [15], we use the absolute difference $|f(x_1) - f(x_2)|$ and element-wise product $f(x_1) \odot f(x_2)$ as distance vectors between $f(x_1)$ and $f(x_2)$. We train MFSVM (see Section 5) using these 2 distance vectors with the ground truth labels provided by LFW, and report 10-fold cross validation accuracy on face verification. As shown in Table 8, using pairwise name attributes from 100 names (i.e., a pairwise name attributes vector has 4950 dimensions), we achieve 80.43% accuracy. Encouraged by this result, we conducted an extended experiment that uses more first names to implement the name attributes. Since we have crawled extra face examples from Flickr for the face-name assignment experiment in Section 6.5.2, we expand the initial 100 names to a total of 216 names, using 100 examples per name to train our pairwise name classifiers. With the extended 216 pairwise name attributes, we can obtain an improved accuracy of 83.42%. To further improve face verification performance, we use MFSVM to combine our pairwise name attributes with the low-level LLC features that were used for name model training, and achieve 85.05% accuracy.

Compared to Kumar's visual attributes work, our name attributes have slightly inferior performance. However, our pairwise name attributes still perform quite respectably on the task of unconstrained face verification, with the following major advantages:

1) For face verification, our scheme only requires the annotations on pairs of matching and non-matching faces. The training of pairwise name attributes require no additional human labeling. In comparison, Kumar's 73 facial attributes are trained with over 10 million hand-labeled examples, while their simile attributes for the 8 face regions are learned from 60 reference people from the PubFig dataset [14] that contains a large number of face images and manual labels. While their describable visual attributes are effective, the huge cost of manual labeling introduces a major difficulty towards the reproducibility of their work. For our pairwise name attributes, as our training images and tags are automatically crawled from the web, we can bypass the expensive human labeling process.
2) Kumar's describable visual attributes are trained using 5 different types low-level features, with 3 normalization schemes and 3 aggregation methods over 10 face regions. In comparison, our name models are learned with only 1 type of feature (i.e., LLC encoded SIFT) extracted over a face pyramid, which greatly simplifies the attributes training process.

TABLE 8: LFW face verification 10-fold cross validation accuracy

Algorithm	Accuracy
Name attributes (100 names)	80.43% ± 0.56%
Name attributes (216 names)	83.42% ± 0.65%
LLC feature	80.75% ± 0.45%
Name attributes (216 names) + LLC feature	85.05% ± 0.54%
Kumar facial attributes [15]	85.25% ± 0.60%
Kumar simile attributes [15]	84.14% ± 0.41%
Kumar facial + simile attributes [15]	**85.54%** ± 0.35%
Tom-vs-Pete [16]	93.10% ± 1.35%
Tom-vs-Pete + facial attributes [16]	**93.30% ± 1.28%**

Note that the current state-of-the-art performance on the LFW dataset is achieved in [16], where Berg and Belhumeur reported an impressive 93.30% accuracy by combining their Tom-vs-Pete classifiers with Kumar's facial attributes. For their Tom-vs-Pete classifier, they selected a set of 5,000 classifiers trained using 120 labeled human subjects (20,639 labeled face images) and 11 types of low-level features. Comparing to their method, our first-name attributes do not need the hand labeled examples for training classifiers, therefore our classifiers are trained with "weak labels" harvested from the web image tags. Besides, for face alignment, Berg and Belhumeur first detected 95 facial landmarks and used a piecewise affine transform procedure, whilst we use a much simpler light-weight alignment scheme that performs similarity transform on detected eye locations. Finally, our extracted facial features are designed to work well on associating first names with facial appearance, which work well over a range of applications and are not catered towards the specific task of face verification.

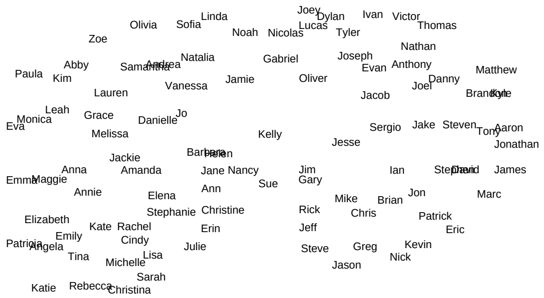

Fig. 10: By analyzing the confusion between our first name classifiers and then embedding the first names into a two-dimensional space, we see that visually similar names are placed near one another.

6.7 Visualizing Name Relations

We visually explore the similarity between first names to produce a first name embedding that represents the visual similarity of the faces having the first names in our dataset. Some pairs of first names are easier to distinguish than others. In a sense, first names that are interchangeable (i.e., pairs of names that perspective parents were deciding between) should have face populations that appear to be similar, and should be close in our embedding.

The first name embedding is accomplished as follows: A matrix indicating the accuracies of each of the pairwise name classifiers, after scaling and inverting so that the most confused name pair has distance 0, and the most distinct name pair has distance 1, is used as an affinity matrix for multi-dimensional scaling (MDS). Following MDS, we apply a force model to discourage name overlapping for facilitating viewing in this paper. Figure 10 shows our embedding. Notice that the horizontal dimension relates to gender (males on the right) and age corresponds to the vertical axis (younger names are near the top). Similar names are placed nearby one another in the embedding. This name embedding is produced solely as a by-product of our pairwise name classifiers, and is completely based on the visual similarity between faces having the given names.

7 Limitations

We propose to represent facial appearance as a set of first name attributes, and demonstrated several applications. However, our method does have some limitations which we outline as follows:

- While we showed that correlations exist between names and facial appearance, directly using these name-face relationships for predicting the name of a never-seen face cannot achieve a high accuracy. Figure 4 shows that the system makes many mistakes in predicting the first names, although we expect that all would agree that guessing the first name for a never-before seen face is an extremely challenging problem.
- In contrast to Kumar's work on describable facial attributes [15], our first name attributes do not have semantic meanings. Kumar's work describe a face using many semantic attributes such as "wear glasses" and "brown hair", while our attributes are first names and hence not semantically descriptive for attribute-based descriptions.
- For the unconstrained face verification task, our performance underperforms the state-of-the-art algorithm of Tom-vs-Pete classifiers [16]. The performance gap is due to the fact that we rely on automatically mined weakly labeled names to train the models, and does not enjoy the benefits of using strong labels as was done in [16].

8 Conclusion

In this paper, we consider facial processing by modeling the relation between first names and faces. We build models for common names and treat first names as attributes for describing the facial appearance. A novel face pyramid using Multi-Feature SVM is proposed, which offers improved performance on constructing first name attributes. We show the result that first names can be correctly inferred at rates far exceeding random chance. We have also described several practical applications of our name attributes, including gender recognition, age classification, face-name assignment, unconstrained face verification and name relations visualization. Our first name attributes representation is powerful for performing various facial analysis tasks, and has the advantage of using name labels that are freely available from the internet.

References

[1] H. Chen, A. Gallagher, and B. Girod, "What's in a name: First names as facial attributes," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2013.
[2] T. Kanade, "Picture processing by computer complex and recognition of human faces," Ph.D. dissertation, Kyoto Univ., 1973.
[3] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. of Cognitive Neurosicence*, vol. 3, no. 1, 1991.
[4] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, 1997.
[5] M. Bartlett, J. R. Movellan, and T. Sejnowski, "Face recognition by independent component analysis," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, 2002.
[6] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, 2006.
[7] N. Pinto and D. Cox, "Beyond simple features: A large-scale feature search approach to unconstrained face recognition," in *Int. Conf. Automat. Face and Gesture Recognition (FG)*, 2011.
[8] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, 2009.
[9] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2009.
[10] L. Torresani, M. Szummer, and A. Fitzgibbon, "Efficient object category recognition using classemes," in *Proc. Euro. Conf. Comput. Vision (ECCV)*, 2010.
[11] L. Li, H. Su, E. Xing, and L. Fei-fei, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2010.

Fig. 11: The average faces of all 100 names in our Names 100 Dataset. Each average face is annotated with the first name, the dominant gender (M for male, F for female, and N for Neutral), and the expected birth year ± birth year standard deviation.

[12] L. Li, H. Su, Y. Lim, and L. Fei-fei, "Objects as attributes for scene classification," in *Proc. Euro. Conf. Comput. Vision (ECCV), 1st Int. Workshop on Parts and Attributes*, 2010.
[13] N. Kumar, P. N. Belhumeur, and S. K. Nayar, "Facetracer: A search engine for large collections of images with faces," in *Proc. Euro. Conf. Comput. Vision (ECCV)*, 2008.
[14] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *IEEE Int. Conf. Comput. Vision (ICCV)*, 2009.
[15] ——, "Describable visual attributes for face verification and image search," in *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011.
[16] T. Berg and P. Belhumeur, "Tom-vs-Pete classifiers and identity-preserving alignment for face verification," in *Proc. British Machine Vision Conf. (BMVC)*, 2012.
[17] A. Gallagher and T. Chen, "Estimating age, gender, and identity using first name priors," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2008.
[18] D. Figlio, "Names, expectations and the black-white test score gap," National Bureau of Economic Research, NBER Working Papers, 2005.
[19] J. Jones, B. Pelham, M. Carvallo, and M. Mirenberg, "How do I love thee? Let me count the Js: Implicit egotism and interpersonal attraction," *J. of Personality and Social Psych.*, vol. 87, no. 5, 2004.
[20] D. Kalist and D. Lee, "First names and crime: Does unpopularity spell trouble?" in *Social Science Quarterly*, vol. 90, 2009.
[21] J. Nuttin, "Narcissism beyond gestalt and awareness - the name letter effect," *Euro. J. of Social Psych.*, vol. 15, no. 3, 1985.
[22] B. Pelham, M. Mirenberg, and J. Jones, "Why Susie sells seashells by the seashore: Implicit egotism and major life decisions," *J. of Personality and Social Psych.*, vol. 82, no. 4, 2002.
[23] B. Pelham, M. Carvallo, T. DeHart, and J. Jones, "Assessing the validity of implicit egotism: A reply to Gallucci," *J. of Personality and Social Psych.*, vol. 85, no. 5, 2003.
[24] S. Garwood, L. Cox, V. Kaplan, N. Wasserman, and J. Sulzer, "Beauty is only "name" deep: The effect of first-name on ratings of physical attraction," *J. of Applied Social Psych.*, vol. 10, no. 5, 1980.
[25] M. Lea, R. Thomas, N. Lamkin, and A. Bell, "Who do you look like? Evidence of facial stereotypes for male names," *Psychonomic Bulletin and Review*, 2007.
[26] "US social security administration baby name database," http://www.ssa.gov/oact/babynames.
[27] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2001.
[28] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models - their training and application," *Comput. Vision and Image Understanding*, vol. 61, no. 1, 1995.
[29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IEEE Int. J. of Comput. Vision*, 2004.
[30] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2010.
[31] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2006.
[32] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. and Tech.*, pp. 27:1–27:27, 2011.
[33] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, 2002.
[34] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. of Comput. and Syst. Sciences*, 1997.
[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, 2005.
[36] D. Parikh and K. Grauman, "Interactively building a discriminative vocabulary of nameable attributes," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2011.
[37] A. Gallagher and T. Chen, "Understanding images of groups of people," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2009.
[38] Y. Fu, G. Guo, and T. Huang, "Age synthesis and estimation via faces: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, 2010.
[39] J. Munkres, "Algorithms for the assignment and transportation problems," *J. of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
[40] G. B. Huang, M. Ramesh, T. Berg, and E. L. Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. of Massachusetts, Amherst, Tech. Rep., 2007.

Huizhong Chen is a Ph.D candidate in the Electrical Engineering Department of Stanford University, California. He received the B.A. and M.Eng. degrees in Electrical and Information Sciences, both from University of Cambridge, UK, in 2008 and 2009 respectively. Huizhong is a recipient of the Kodak Fellowship, the Gibbons Fellowship, and Sir Ian McFarlane Award. His current research interests are Computer Vision and Multimedia Systems.

Andrew Gallagher is a Visiting Research Scientist at Cornell University's School of Electrical and Computer Engineering. He earned the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University in 2009, advised by Prof. Tsuhan Chen. Andy worked for the Eastman Kodak Company from 1996 to 2012, initially developing image enhancement algorithms for digital photofinishing. Later, Andy developed computer vision applications such as red-eye correction and face recognition. Andy is part of a computer vision start-up, TaggPic, that recognizes landmarks in images captured across the globe. More recently, Andy's interests are in the arena of improving computer vision by incorporating context, human interactions, and unique image cues. Andy is interested in a wide variety of data analysis problems, and has developed algorithms for detecting image forgeries, assembling puzzles, and deciding what NFL teams should do on fourth down.

Bernd Girod is Professor of Electrical Engineering in the Information Systems Laboratory of Stanford University, California, since 1999. Previously, he was a Professor in the Electrical Engineering Department of the University of Erlangen-Nuremberg. His current research interests are in the area of networked media systems. He has published over 500 conference and journal papers and 6 books, receiving the EURASIP Signal Processing Best Paper Award in 2002, the IEEE Multimedia Communication Best Paper Award in 2007, the EURASIP Image Communication Best Paper Award in 2008, the EURASIP Signal Processing Most Cited Paper Award in 2008, as well as the EURASIP Technical Achievement Award in 2004 and the Technical Achievement Award of the IEEE Signal Processing Society in 2011. As an entrepreneur, Professor Girod has been involved in several startup ventures, among them Polycom, Vivo Software, 8x8, and RealNetworks. He received an Engineering Doctorate from University of Hannover, Germany, and an M.S. Degree from Georgia Institute of Technology. Prof. Girod is a Fellow of the IEEE, a EURASIP Fellow, and a member of the German National Academy of Sciences (Leopoldina). He currently serves Stanford's School of Engineering as Senior Associate Dean for Online Learning and Professional Development.