A DATA-DRIVEN APPROACH TO CLEANING LARGE FACE DATASETS

Hong-Wei Ng and Stefan Winkler

Advanced Digital Sciences Center (ADSC), University of Illinois at Urbana-Champaign, Singapore

ABSTRACT

Large face datasets are important for advancing face recognition research, but they are tedious to build, because a lot of work has to go into cleaning the huge amount of raw data. To facilitate this task, we describe an approach to building face datasets that starts with detecting faces in images returned from searches for public figures on the Internet, followed by discarding those not belonging to each queried person.

We formulate the problem of identifying the faces to be removed as a quadratic programming problem, which exploits the observations that faces of the same person should look similar, have the same gender, and normally appear at most once per image. Our results show that this method can reliably clean a large dataset, leading to a considerable reduction in the work needed to build it. Finally, we are releasing the *FaceScrub* dataset that was created using this approach. It consists of 141,130 faces of 695 public figures and can be obtained from http://vintage.winklerbros.net/facescrub.html.

Index Terms— Outlier Detection, Face Recognition

1. INTRODUCTION

The recent Big Data revolution has made it easier to obtain real world data from the Internet to build large face datasets [1,2]. This has arguably led to progress in face recognition research, but building a large dataset remains a time-consuming and tedious affair, since the noisy raw data has to be manually organized before it can be used. To help with this difficult task, researchers have increasingly turned to crowdsourcing for manpower [1, 2]. However, for crowdsourcing to work, extra effort has to go into devising schemes to ensure workers follow the instructions strictly instead of ignoring them for a quick payday. Even then, there is no guarantee that things will go as planned [3, 4].

In this paper, we propose a method to make it easier to build a large face dataset by leveraging the structure of faces to help clean it. Specifically, we consider sets of face images obtained by running a face detector on images returned from searches for public figures using a search engine. Within each set, some are false positives (i.e., non-faces) found by the imperfect face detector, and a number of them belong to other people appearing in the same image as the queried person or to people in images irrelevant to the query. We refer to these faces collectively as outliers, and faces of the queried person as inliers (see Figure 1 for a sample). Our goal is to remove the outliers among the detected faces for each queried person, so that we obtain faces belonging just to him/her and a cleaned dataset overall.

We make the following assumptions: 1) Falsely detected faces form a relatively small portion of the large set of faces; 2) A face with a different gender from the query must be an outlier; 3) Inliers for each queried person have similar appearance and are the majority; 4) The queried person typically appears at most once per image. The first observation suggests that an outlier detection classifier such as the One-Class SVM [5] trained on descriptors extracted from the detected faces can help remove some of the non-faces. If we further assume the gender of the query person to be known, then a gender classifier can also help remove incorrect ones. However, the significant variation and noise present in real-world images means that these classifiers rarely do a good job on their own. More importantly, by not enforcing visual similarity among the outliers and inliers and allowing predictions to be made independently for each face, we can get absurd results such as retaining multiple faces belonging to different people in the same image (see Figure 1). To avoid these problems, we formulate the task of identifying outliers in the set of faces detected for each query as a separate quadratic programming (QP) problem that combines the outputs from these classifiers in a suitable objective function and simultaneously enforces the conditions above to obtain better results.

2. RELATED WORK

At first glance, our problem appears similar to that of face annotation for photo albums [6–8] or TV shows [9], where the goal is to label each face with a name from a finite set of identities. However, the people in our dataset are typically unrelated to one another, unlike the people in a photo album or those appearing together on a show. This means we cannot exploit useful information such as co-occurrences to help disambiguate their identities, as was done in [6, 8, 9]. Furthermore, work in this area typically assumes that part of the

This work is supported by the research grant for ADSC's Human Sixth Sense Programme from Singapore's Agency for Science, Technology and Research (A*STAR).

dataset was already labeled properly so that face appearance models and the co-occurence statistics of individuals can be estimated [6, 7], whereas we do not.

The nature of our work is closer in spirit to that of [10], where faces detected in news images are automatically clustered with the help of names detected in the news captions associated with each image. However, we do not have hints (names in news caption) to suggest who might be in an image. Also, [10] removed the false face detections before running their algorithm, whereas we intend for our algorithm to automatically remove them.

The authors of [11] built a dataset containing faces of celebrities extracted from billions of images found on web pages. They analyzed text surrounding those images and their near duplicates on other web pages to identify the names of people likely to be in them and propagated the names (i.e., labels) on a facial similarity graph. As was the case for [10], we do not have text labels as contextual information. In a way, our work can be considered complementary to both [11] and [10], as our algorithm can be run as a post-processing step to refine the labels that they obtained by automatically removing false positives and faces with inconsistent gender.

3. DATA COLLECTION AND REPRESENTATION

To build our dataset, we first compile names of public figures (e.g., celebrities) from websites such as IMDb [12]. Concurrently, we note down their gender, which we later use to label training data for our gender classifier. Next, we search for each name in an image search engine (e.g., Google's), download the returned images, and run a face detector [13] on them to extract potential faces for each person. The faces are aligned using the code from [14] and resized to 96×96 pixels. At this stage, for the i^{th} query (i.e., the i^{th} person in our dataset), we have a set of images $\mathcal{I}^{(i)}$ matching that query, and a set $\mathcal{F}^{(i)}$ containing the faces detected in those images, some of which might not belong to this person. Table 1 summarizes the number of queries and detected faces in our *FaceScrub* dataset; Figure 1 shows a sample of the detected faces.

After detecting the faces, we crop them horizontally to a width of 48 pixels about the center to exclude the noisy background and extract two sets of descriptors for each of them: 'uniform' Local Binary Pattern (LBP) [15] (parameters $r = 1, s = 8$) and Three-Patch LBP [16] (parameters $r = 2, s = 8, w = 5, \alpha = 5$). To handle the large number of faces, we reduce their dimensions separately to 150 using Randomized PCA [17] and concatenate them to obtain a 300-dimensional descriptor for each face. We denote the final set of descriptors for faces in $\mathcal{F}^{(i)}$ as $\mathcal{X}^{(i)}$, and individual descriptors as $x^{(ij)} \in \mathcal{X}^{(i)}, j = 1, \ldots, |\mathcal{X}^{(i)}|$. $|\mathcal{X}^{(i)}|$ is the cardinality of the set $\mathcal{X}^{(i)}$. $\mathcal{X} = \bigcup_{i=1}^{N} \mathcal{X}^{(i)}$ denotes the set of descriptors from all $N = 695$ people in our dataset, and $\mathcal{X}^M$ and $\mathcal{X}^F$ the sets of descriptors from those whose queries are male and female, respectively.

	Male	Female	Total
# Queries	348	347	695
# Detected faces	91760	80809	172569
# Predicted inliers	77538	63592	141130
# Predicted outliers	14222	17217	31439

Table 1. *FaceScrub* dataset summary.

4. PROBLEM FORMULATION

We formulate the problem of identifying outliers in the descriptors $\mathcal{X}^{(i)}$ associated with the i^{th} person in our dataset as a quadratic programming problem (QP) to be minimized. Its solution is a vector of labels $y^{(i)} \in \{-1, +1\}^{|\mathcal{X}^{(i)}|}$ with $y_j^{(i)} = +1$ if the j^{th} face is an inlier and $y_j^{(i)} = -1$ if it is an outlier. This QP will try to find the optimal solution such that faces that are false positives from the face detector or have a different gender from the query are more likely to be labeled as outliers. Simultaneously, it also encourages those with similar appearances to be assigned the same label and enforces the constraint that at most one face in an image can be an inlier. We run this optimization for each set $\mathcal{X}^{(i)}$, where i indexes the people in our dataset. We describe the individual penalty terms of the objective function and its constraints in the following.

4.1. False Positives Term

One of our goals is for the QP to label faces that are likely false positives from the face detector as outliers. To do this, we need a model to assign a score for how likely that is the case for a particular face $x^{(ij)} \in \mathcal{X}^{(i)}$. We noted earlier in the introduction that these false positives form a small fraction of the detected faces, which suggests that an outlier detection classifier trained on the descriptors $\mathcal{X}$ might be useful for identifying them without us having to label any of the descriptors. Hence, we train a One-Class SVM [5] with a Gaussian kernel (parameters $\nu = 0.1$ and $\gamma = 0.01$) using $\mathcal{X}$ as training data and use the output of its decision function (i.e., the signed distance of each point in feature space to its learned hyperplane) as the 'false positive' scores for those faces. These scores are stored in a vector $b^{(i)} \in \mathcal{R}^{|\mathcal{X}^{(i)}|}$ with larger negative entries for faces more likely to be false detections and larger positive entries for those of actual faces.

A natural term to include in the objective function to encourage labeling these false positives as outliers would be $-b^{(i)T} y^{(i)}$, as $y_j^{(i)}$ is then more likely to be -1 if $b_j^{(i)}$ is a large negative number (i.e., false positive). However, this can also cause a face with descriptor $x^{(ij)}$ to be labeled an inlier ($y_j^{(i)} = +1$) just because it looks like a face (i.e., $b_j^{(i)} > 0$) even though it might be that of another person. We avoid this by setting positive entries in $b^{(i)}$ to 0 to obtain the vector $\widetilde{b}^{(i)}$

and instead use the term

$$f_{\text{false}}(y^{(i)}) = -\widetilde{b}^{(i)T} y^{(i)}. \quad (1)$$

4.2. Gender Term

We include a term in the objective function to label faces in $\mathcal{X}^{(i)}$ with a different gender from the i^{th} query as outliers. We first train a two-class linear SVM [18] to classify the gender of faces using the descriptors from the sets $\mathcal{X}^M$ and $\mathcal{X}^F$ to represent faces from the two genders. Similar to the false detection term presented in Section 4.1, we use the output of the SVM's decision function computed for a set of faces $\mathcal{X}^{(i)}$ as gender scores. The vector $g^{(i)} \in \mathcal{R}^{|\mathcal{X}^{(i)}|}$ containing these values will have larger negative entries for faces that are male and larger positive entries for female ones.

Likewise we consider adding a term $g^{(i)T} y^{(i)}$ to the objective function to prefer solutions that have -1 entries for faces in $\mathcal{X}^{(i)}$ predicted as females ($g_j^{(i)} > 0$) when the i^{th} person is male. Note that the sign of the entries in $g^{(i)}$ has to be flipped if the i^{th} person is female instead. Analogous to the case above, we do not want a face to be labeled an inlier simply because it has the same gender as the i^{th} person. Therefore we threshold positive entries in $g^{(i)}$ to 0 to obtain the vector $\widetilde{g}^{(i)}$ and use the term

$$f_{\text{gender}}(y^{(i)}) = \widetilde{g}^{(i)T} y^{(i)}. \quad (2)$$

4.3. Similarity Regularization

To encourage similar-looking faces to have the same label, we add the graph regularizer [19] to our objective function:

$$f_{\text{smooth}}(y^{(i)}) = \frac{1}{2} y^{(i)T} \widetilde{L}^{(i)} y^{(i)}. \quad (3)$$

$\widetilde{L}^{(i)} \in \mathcal{R}^{|\mathcal{X}^{(i)}| \times |\mathcal{X}^{(i)}|}$ is the normalized graph Laplacian,

$$\widetilde{L}^{(i)} = (D^{(i)})^{-\frac{1}{2}} (D^{(i)} - W^{(i)}) (D^{(i)})^{-\frac{1}{2}}.$$

$W^{(i)}$ is the affinity matrix computed from $\mathcal{X}^{(i)}$ with entries

$$W_{pq}^{(i)} = \delta_{pq} \exp\left(\frac{-\left\| x^{(ip)} - x^{(iq)} \right\|_{l_2}^2}{2\sigma} \right),$$

where $\delta_{pq} = 1$ if $x^{(ip)}, x^{(iq)} \in \mathcal{X}^{(i)}$ are k nearest neighbors, and 0 otherwise. We set $k = 7$ and estimate σ as the average distance between the points and their furthest neighbor.

The diagonal matrix $D^{(i)}$ contains the row sums of W

$$D_{pq}^{(i)} = \begin{cases} \sum_k W_{pk}^{(i)}, & \text{if } p = q \\ 0, & \text{otherwise.} \end{cases}$$

The use of the graph Laplacian to enforce label similarity is well studied in the semi-supervised machine learning literature, and we refer readers to [19] for more details.

4.4. Prior Term

To encode prior knowledge that most faces are correct, we discourage solutions with too many outliers (i.e., many -1 entries) using the term

$$f_{\text{prior}}(y^{(i)}) = \left\| \frac{1}{2} \left(\mathbf{1} - y^{(i)} \right) \right\|_{l_1}, \quad (4)$$

where $\mathbf{1}$ is a vector of all ones with dimension $|\mathcal{X}^{(i)}|$.

4.5. Unique Face Constraint

We remarked earlier that a person tends to appear at most once per image. We disregard exceptions such as when an image is a composite of multiple images of the same person, as these are rare, and impose the following set of constraints:

$$\sum_{j \in \mathcal{G}^{(ik)}} y_j^{(i)} \leq 1 - \left(|\mathcal{G}^{(ik)}| - 1 \right), k = 1, \ldots, |\mathcal{I}^{(i)}|. \quad (5)$$

$\mathcal{G}^{(ik)}$ is the set that indexes faces detected in the same image k for the i^{th} queried person, and $|\mathcal{G}^{(ik)}|$ is the number of faces in that image. The sum to the right of the inequality is the largest possible such that at most one of the labels for the faces detected in an image has a value of $+1$ and the rest -1.

4.6. QP Final Form

The constraint that entries in the solution vector $y^{(i)}$ take discrete values in $\{-1, +1\}$ leads to a difficult combinatorial optimization problem. Therefore, we relax it to allow them to be in the range $[-1, +1]$. Our final labels are obtained by thresholding non-positive entries in y to -1 and positive entries to $+1$. The full optimization problem for a set of faces $\mathcal{X}^{(i)}$ is:

$$\begin{aligned} \underset{y^{(i)}}{\text{minimize}} \quad & f_{\text{smooth}}(y^{(i)}) + \lambda_1 f_{\text{false}}(y^{(i)}) \\ & + \lambda_2 f_{\text{gender}}(y^{(i)}) + \lambda_3 f_{\text{prior}}(y^{(i)}) \\ \text{subject to} \quad & -\mathbf{1} \leq y^{(i)} \leq \mathbf{1}, \\ & \sum_{j \in \mathcal{G}^{(ik)}} y_j^{(i)} \leq 1 - \left(|\mathcal{G}^{(ik)}| - 1 \right), \\ & k = 1, \ldots, |\mathcal{I}^{(i)}|, \end{aligned}$$

which is a quadratic program. In our experiments, we fix $\lambda_1 = 1$, $\lambda_2 = 2$, and $\lambda_3 = 1$.

5. EXPERIMENTAL RESULTS

To assess our algorithm, we randomly select 10 queries for each gender in our dataset and manually label the detected faces in the images to form a test set of 5791 faces from these 20 people, with 794 of them being outliers.

For benchmarking our proposed approach, we implement two variants of our method, one without the gender term, and

Method	Precision	Recall	F1 Score	Non-faces	Inliers
Proposed (all terms)	0.530 ±0.061	**0.728** ±0.044	0.601 ±0.041	**0.944** ±0.043	0.102 ±0.023
No f_{gender}	0.503 ±0.060	0.617 ±0.058	0.540 ±0.043	0.918 ±0.056	0.094 ±0.021
No f_{false}	**0.777** ±0.057	0.614 ±0.068	**0.675** ±0.048	0.818 ±0.086	**0.032** ±0.016
Naive	0.480 ±0.068	0.676 ±0.054	0.544 ±0.049	0.889 ±0.060	0.117 ±0.029

Table 2. Performance on test set. Note that true positives here are the correctly predicted *outliers*.

one without the false positives term. In addition, we implement a naive method that classifies any face in an image as an outlier if the false positive or gender classifiers from Section 4 predict that it is a non-face or has a different gender from the query associated with that image. We then compute the precision, recall, F1 scores, fraction of non-faces and inliers predicted as outliers, based on each method's predicted labels for the test set. The averages and standard deviations of these metrics over the queries are reported in Table 2.

We argue that for this particular application, we should prioritize achieving high recall (of outlier faces) over the other metrics, as we have plenty of data and can afford to mislabel some of them, if only to find more outliers. Hence, using recall as the main metric for assessment, our method's recall of 0.728 makes it the best for removing outliers among the different methods that we implemented. This is over 10% higher than when either the gender or false positives term is omitted, suggesting that these two factors have complementary effects in this task. It is also higher than the naive method's recall, even though the naive method also uses both classifiers, which implies that combining them in our optimization framework improves results. This reasoning is further supported by the fact that our method is able to remove more of the non-faces than the naive method, which simply treats all faces predicted by the false positive classifier as outliers (94% vs. 89%). Presumably, the graph regularizer and constraints help our method identify outliers that the individual classifiers miss, based on their visual similarity to outliers that the method was able to classify confidently.

Our method's precision comes second after the case when the false positives term is not used, because the unsupervised One-Class SVM probably misclassifies many inliers as non-faces. But the higher precision for this approach comes with a much lower recall (0.614), which is a big drawback. Nevertheless, this suggests that a better classifier for non-faces could help improve our results.

Finally, to illustrate the advantage of our method over the naive approach, we present two concrete examples in Figure 1 (top row). In both examples, the naive approach fails, because there are two faces (the single inlier and an outlier) in each of them that have the correct gender and are proper faces, leading it to accept them as inliers even though one of them (marked by the arrow) is an outlier. Because our QP constrains the maximum number of inliers in an image to one and forces similar-looking faces to have the same label, it is able to remove these extra 'inliers' and produce the correct result.

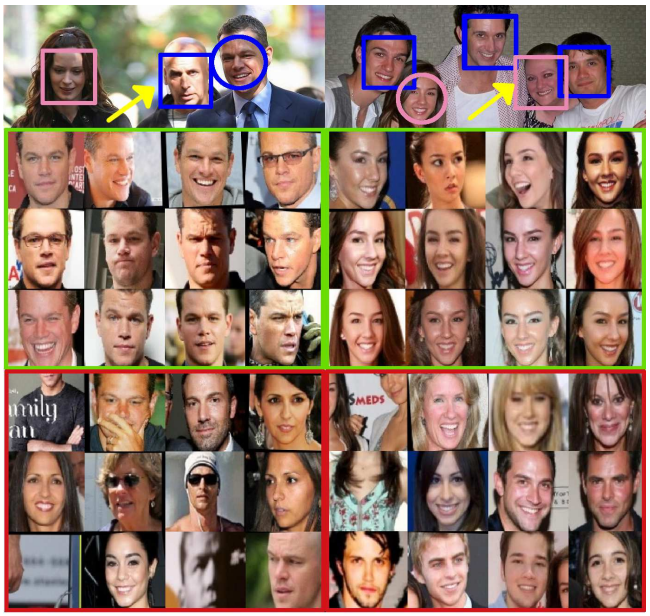

Fig. 1. Sample results for two people in our dataset: Matt Damon (left) and Lexi Ainsworth (right). Top row: Predictions made by our method for one image of each of the two celebrities. Predicted inliers are marked with circles, outliers with rectangles. Colors indicate the predicted gender: Pink for female, blue for male. Arrows mark the faces that our method correctly predicts as outliers, whereas the naive approach does not. More examples of inliers and outliers for both persons are shown in the boxes below with green and red outlines, respectively. Best viewed in color.

6. CONCLUSION

We have described a method for automatically removing outliers from a set of faces, where the majority is assumed to belong to a particular individual. Central to our proposed approach is a quadratic program (QP) that combines the outputs of an outlier detection classifier and a gender classifier, enforces visual similarity among the outliers and inliers, while simultaneously constraining at most one face per image to be an inlier. Our results show that the QP can leverage these conditions to effectively clean the raw data, thereby greatly reducing the manual workload required for building face datasets. The resulting *FaceScrub* dataset is available at http://vintage.winklerbros.net/facescrub.html.

7. REFERENCES

[1] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[2] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. International Conference on Computer Vision (ICCV)*, 2009.

[3] Alexander Sorokin and David Forsyth, "Utility data annotation with Amazon Mechanical Turk," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2008.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li, "ImageNet: A large-scale hierarchical image database," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[5] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[6] Andrew C. Gallagher and Tsuhan Chen, "Using group prior to identify people in consumer images," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[7] Andrew C. Gallagher and Tsuhan Chen, "Clothing cosegmentation for recognizing people," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[8] Zak Stone, Todd Zickler, and Trevor Darrell, "Toward large-scale face recognition using social network context," *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1408–1415, 2010.

[9] Timothée Cour, Benjamin Sapp, Chris Jordan, and Benjamin Taskar, "Learning from ambiguously labeled images," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 919–926.

[10] Tamara L. Berg, Alexander C. Berg, Jaety Edwards, Michael Maire, Ryan White, Yee Whye Teh, Erik G. Learned-Miller, and David A. Forsyth, "Names and faces in the news," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004, pp. 848–854.

[11] Xiao Zhang, Lei Zhang, Xin-Jing Wang, and Heung-Yeung Shum, "Finding celebrities in billions of web images," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 995–1007, 2012.

[12] IMDb, "Internet Movie Database," http://www.imdb.com.

[13] Paul Viola and Michael J. Jones, "Robust real-time face detection," *International Journal on Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[14] Gary B. Huang, Vidit Jain, and Erik G. Learned-Miller, "Unsupervised joint alignment of complex images.," in *Proc. International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.

[15] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[16] Lior Wolf, Tal Hassner, and Yaniv Taigman, "Descriptor based methods in the wild," in *Proc. European Conference on Computer Vision (ECCV) Workshop on Faces in Real-Life Images Workshop*, 2008.

[17] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.

[18] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008, software available at http://www.csie.ntu.edu.tw/~cjlin/liblinear/.

[19] Xiaojin Zhu and Andrew B. Goldberg, *Introduction to Semi-Supervised Learning*, vol. 3 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Morgan & Claypool Publishers, 2009.