Album-Oriented Face Recognition For Online Social Networks

Zhongkai Han, Syed Zain Masood, Jason Hochreiter, Spencer Fonte and Marshall Tappen

Abstract— In this paper, we propose an album-oriented face-recognition model that exploits the album structure for face recognition in online social networks. Albums, usually associated with pictures of a small group of people at a certain event or occasion, provide vital information that can be used to effectively reduce the possible list of candidate labels. We show how this intuition can be formalized into a model that expresses a prior on how albums tend to have many pictures of a small number of people. We also show how it can be extended to include other information available in a social network. Using two real-world datasets independently drawn from Facebook, we show that this model is broadly applicable and can significantly improve recognition rates.

I. INTRODUCTION

Traditional face recognition systems have relied on image features to identify the individual the photographs. The advent of popular social networks, such as Facebook, which host photo albums and make it possible to tag photos with user identities, adds a new dimension of data that can be used to help identify faces.

In Facebook, as in most photo management services, photos are grouped into albums. These albums are a rich source of information because they often correspond to trips, events, or specific groups of people. In this paper, we show how the organizational structure of photos into albums can be used to significantly increase recognition accuracy. In addition, as will be shown in Section III, our model based on using album information can be applied to significantly more pictures than other models that exploit co-occurrence in photos, such as [18].

Much of the contribution of this work lies in modeling how individuals tend to co-occur in photo albums. The basic model is constructed with the underlying idea that albums tend to contain multiple photos of a small number of people, such as an album containing photographs from a trip. An album may contain many photos, but it is likely that the individuals pictured in the album will be dominated by the small number of people that participated in the event.

After introducing this basic model, we will then show how it can be improved and extended by considering other factors such as previous co-occurrence in an album, friendship information, and the identity of the person who uploaded the photo to the social network.

The rest of the paper follows this rough outline:

- Section II illustrates the difficulties of working with data from publicly available social networks.
- Section III shows that a CRF model based on limiting the number of individuals appearing in an album is useful for a significant portion of photos on Facebook. This section will show how this model can also be

applied more widely than just modeling co-occurrence in photographs. Following this, Section IV discusses related work.
- Sections V through VII describe the design of the model, inference with the model, and the training procedure for the model.
- Section VIII describes experiments showing how this model significantly increases recognition performance. Most importantly, for the two datasets we test, this approach increases accuracy by around 20% over a baseline classifier.

II. REPRODUCIBLE RESEARCH ON SOCIAL NETWORK DATA

As will be discussed in Section VIII, we validate our methods on real data gathered from Facebook. One of the difficulties in working with social network data is the ability to share that data. Common datasets, such as the PASCAL challenge [4] and the Middlebury Stereo database [16] have facilitated significant advances in vision technology. However, sharing social network data is problematic due to privacy issues since it involves sharing the information of both users and their friends on social networks.

While it could be argued that anonymizing the type of data used for face recognition by various transformations could protect the identities of users, researchers have been effective at de-anonymizing data. In 2008, researchers de-anonymized significant portions of a dataset released by Harvard [12].

Given the constant threat of having anonymized data cracked, we have opted for the alternative approach of releasing code that will enable other researchers to experiment with our algorithms. The author's web site[1] will contain implementations of the learning and inference code, along with our current implementation for gathering data.

To more strongly validate our results, we used two different datasets downloaded from the accounts of different individuals and two implementations of our algorithms created by different members of our group. As will be reported in Section VIII, both experiments confirm the benefits of the proposed approach.

III. THE APPLICABILITY OF AN ALBUM PRIOR

A key question facing this work is whether users tend to organize albums in a way that makes this prior useful. Our study on photo albums in Facebook indicates that a prior based on the assumption that albums tend to contain multiple photos of a small number of people is much more

---

[1] http://www.cs.ucf.edu/~smasood/code/FaceRecognition.zip

applicable than a model that relies on co-occurrence inside a photo, such as [18]. To evaluate the usefulness of this prior on the occurrence of individuals in albums, we used the Facebook API to download all pictures visible to a single user's account, similar to [18]. In total, we were able to capture 8078 pictures containing 2849 different people. In all of the pictures, we only considered faces that had been tagged by a user. To ensure the accuracy of tags, we applied the OpenCV face detector to each tagged photograph we downloaded [2]; if the detector did not indicate the presence of at least one face in a photo, we discarded it.

In this collection of photographs, over 5735 photographs, or 71% of the photographs, only contained one tagged face. Presuming that, for the vast majority of photographs, all of the faces have been tagged, this means that a model based on co-occurrence inside an photograph [18] would help with recognition in roughly only 29% of photographs.

In measuring the applicability of our model, we set a high standard for its usefulness by assuming that an album prior would only be useful if there are at least twice as many photographs in an album as people occurring in that album. Despite this high threshold, we found that an album prior could still be applied to albums containing 57% of the photos in our dataset. Roughly speaking, a prior based on limiting occurrence in albums can be applied to nearly twice as many photographs as a prior just based on co-occurrence inside an photograph, such as the model in [18].

In Section VIII-D, we consider the scenario in which this high standard is not met. Training and testing on all albums – for some of these albums, the number of labels is more than half of the number of photographs – results in only a 2% performance degradation as compared to training and testing on only albums that satisfy this threshold, indicating the applicability of our approach.

Fig. 1 demonstrates why an album-based model can be applied more widely. Each row corresponds to photographs taken from a specific album. For each row, only one photograph has two people in it. Thus, a model based on photograph co-occurrence could only be applied to one of the photographs, while all three could be considered in an album-based model.

An important contribution of this work is that we show a model where the album structure can be considered in an efficient inference scheme.

IV. RELATED WORK

The most similar work on this problem is the model of co-occurrence in photographs proposed by Stone et al. [18]. In this model, photographs gathered from Facebook were used to model the frequency with which individuals appeared together in photographs.

A model like that used in [18] would be difficult to extend to album-level co-occurrence because modeling co-occurrence in photographs implicitly assumes that individuals are not repeated. However, in albums, individuals may be repeated many times. In addition, the model in [18] was based on a fully-connected graph between individuals
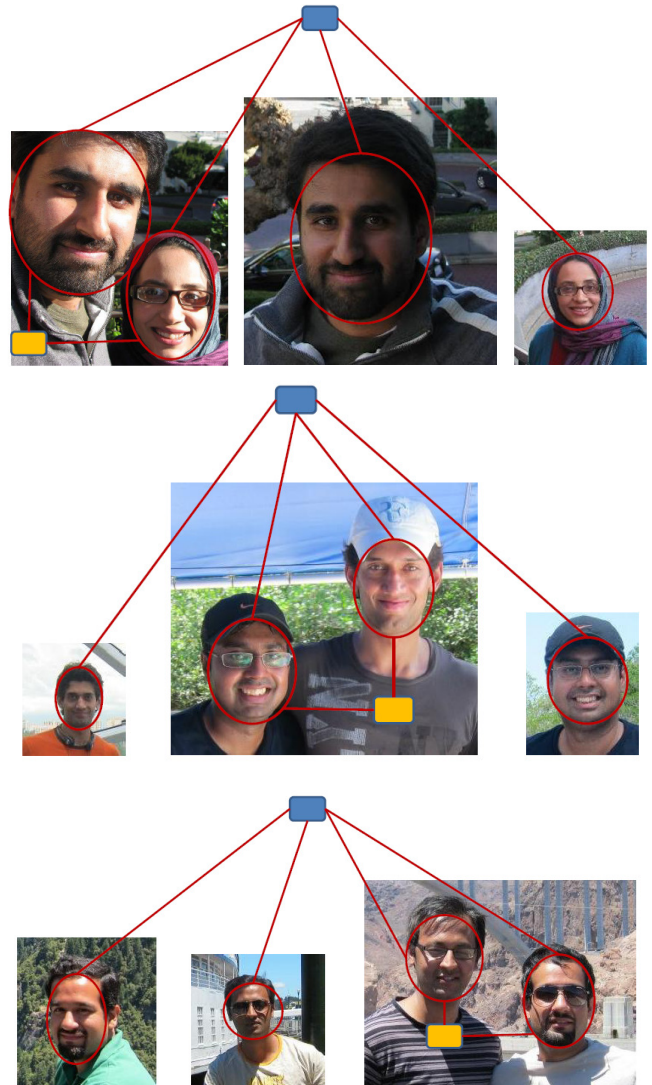

Fig. 1. The factor graph illustrating the importance of modeling how people occur in albums instead of just pictures. Each row corresponds to photographs taken from a specific album. A model based on just photograph co-occurrence, shown in yellow, is only able to work with one of the three photographs, while a model based on occurrence in an album can help recognize faces in all three pictures.

in the photograph. This was manageable because only a small number of people appeared in any photograph. Extending these fully-connected pairwise relationships between all faces appearing in an album would lead to a large, fully-connected graph representing the album. The density of the edges in the graph could pose serious challenges for inference in the model.

Instead, we propose a model based on penalizing album labelings by how many individuals appear in the album, similar to the label cost from [3]. This makes it possible to perform inference with an efficient, greedy approximation that performs well.

There have also been many studies conducted in the recent past on the problem of person recognition that use

contextual information for improved results. Predominantly, two kinds of contextual information are used: social context information and personal context information.

Personal contextual information, such as hair and clothing, can provide useful information to characterize a person, because such features do not change during a short period of time, which means the same person tends to wear the same clothes or has the same hairstyle across different pictures. Many researchers ([1], [17], [20], [19]) combine these features with face recognition results to improve recognition accuracy. However, the pictures in online communities might be captured over a long period of time and uploaded at the same time, so such personal information could be unstable.

Compared with personal contextual features, social contextual information captures the relationships between people. This is believed to be more robust because relationships change little over a long period of time. The most widely used social feature is co-occurrence ([11], [20], [7]). In [11], event and location groupings of pictures are obtained based on time and locations of photographs. Later, picture co-occurrence is used to provide a relevant short list of identities likely to appear in new photographs. Gallagher and Chen [7] use pairwise co-occurrence to calculate the grouping prior distribution, which models the probability of a group of people appearing in the same picture. Zhao and Liu [20] first cluster pictures into albums based on time and then combine picture co-occurrence and personal contextual information with face recognition results to achieve recognition accuracy for each album. These methods will narrow down the candidates to a small group of people, such as family members ([20], [19]) or the people going on the same trip [17]. For an online community, however, the candidates are all the people in the online community related to the uploader, and effectively narrowing down the list of candidates is the main concern in this paper.

The embedded social network in online communities provides useful information for recognition tasks. In [18], social network information such as friendship, pairwise co-occurrence and face recognition score are incorporated into a CRF model. Specifically, each detected face in one picture is regarded as a node in the graph model, and the edges between nodes are the social relationships. In this way, the total energy is the weighted sum of all the potentials in the graph.

V. ALBUM-ORIENTED FACE RECOGNITION

In this section, we describe how to construct a model based on occurrence in albums. This model is based on introducing a label cost, similar to the label cost described in [3]. Essentially, a cost is assigned to each label present in the album, regardless of whether the label appears more than once, effectively limiting the number of people appearing in an album. In this section, we first present a framework for incorporating data cost and label cost; then, we introduce the specific data cost used. Section V-C will introduce the inference strategy used.

A. Face Recognition Using Label Cost

The central goal is to correctly label the faces in an album F. Thus, the album F can be thought of as a set of face images. Following the notation used in [9], the vector $\vec{y}$ will denote the labelings of each face image in one album, making its fth entry y_f the label of the fth facial image in album F. A traditional face recognition system can be described by the energy function

$$E(\vec{y}) = D(\vec{y}; \vec{x}) = \sum_{f \in F} D_f(y_f; \vec{x}_f), \qquad (1)$$

where D_f is the data cost of assigning label y_f to face $f \in F$. The vector $\vec{x}_f$ is the vector of features gathered from the image corresponding to the facial image f.

Our model builds on this by adding costs based on who is in a particular labeling $\vec{y}$, ignoring how many times that label appears in $\vec{y}$. With a slight abuse of notation, we will use the notation $l \in \vec{y}$ to denote that the label l appears at least once in the label vector $\vec{y}$. Formally, we express this as

$$l \in \vec{y} \Leftrightarrow \exists f \text{ s.t. } y_f = l. \qquad (2)$$

Following [3], we will refer to costs based on this type of membership as label costs. Suppose there are only N candidates $\{l_1, \ldots, l_N\} \in \mathcal{L}$ for each label y_f. The basic system in (1) will be extended as

$$E(\vec{y}) = \sum_{f \in F} D_f(y_f; \vec{x}_f) + C(\vec{y}), \qquad (3)$$

where $C(\vec{y})$ is the label cost of album labeling $\vec{y}$. This label cost combines several types of social network information. The specific form of this cost will be introduced in Section VI.

This model is interesting in that recognition is simultaneously performed on all images in the album.

B. Data Cost

The data cost can be thought of as the result from a baseline, image-only, face recognition system. In this paper, the data cost is implemented using the high-dimensional V1-like features proposed in [13], [14], [15], who showed that excellent face-recognition results could be achieved by linear classification of very high-dimensional feature vectors extracted from the image.

Assuming that $\mathcal{L}$ denotes the set of possible labels for each face and that there are N labels in $\mathcal{L}$, the data cost is computed using the negative log of a soft-max function:

$$D(\vec{y}; \vec{x}) = \sum_{f \in F} -\log \frac{e^{-V_{y_f}(\vec{x}_f)}}{\sum_{l=1}^{N} e^{-V_l(\vec{x}_f)}}, \qquad (4)$$

where $\vec{x}_f$ is vector of features extracted from an image f. The outer summation is computed over all images in the album F. The functions $V_1(\vec{x}), \ldots, V_N(\vec{x})$ are linear combinations of the feature vector $\vec{x}$ with weight vectors, as is standard in linear classifiers.

Because the length of $\vec{x}$ is huge, containing 86400 entries, we use a multi-class generalization of the LogitBoost

and GentleBoost algorithms [6], combined with regression stumps, to greedily select a subset of features. In our experiments, we found that a classifier using only 400 of the 86400 features performed nearly as well as linear classification using the entire feature vector.

C. Inference

Inference in (3) is an NP-hard problem. However, this model is similar to the well studied *uncapacitated facility location (UFL)* problem [10], so it is possible to use a greedy algorithm which would yield a $O(\log |\mathcal{L}|)$-approximation ([8], [10]) in this case. As we will show, our problem is not a standard *UFL* problem as the label costs will depend on each other when we introduce the social label cost. Although there is no theoretical guarantee that a greedy algorithm will give a good approximation, a greedy method worked well in our experiments.

This greedy algorithm, which is described in detail in Algorithm 1, operates by adding one label to the album that maximizes the energy function into (3) at each iteration. For each iteration, the greedy method scans all the available candidates and selects the best one. It stops when adding new candidate labels does not result in further improvement.

Algorithm 1 Greedy approximation

1: Define:
2:   Q: queue containing all the candidate labels in $\mathcal{L}$
3:   l_i: ith label in Q
4:   L: the set of all labels appearing in the album
5:   $\vec{w}$: learned weight vector (Section VII-B)
6:   $\Psi(\vec{x}, \vec{y})$: feature vector depicting input/output relation (see Sections VII-A and VII-B for details)
7:
8: Initialize:
9:   $L^* = \emptyset$
10:   $E^* = -\infty$
11:
12: **while** $Q \neq \emptyset$ **do**
13:   **for** $i = 1$ *to* $|Q|$ **do**
14:     $L = L^* \cup l_i$
15:     $E_i = \max_{\vec{y}} \vec{w}^T \Psi(\vec{x}, \vec{y})$
16:   **end for**
17:   $\hat{E} = \max_i E_i$
18:   $\hat{l} = l_i$
19:   **if** $E^* < \hat{E}$ **then**
20:     $E^* = E_i$
21:     $L^* = L^* \cup \hat{l}$
22:     Eject(Q, $\hat{l}$)
23:   **else**
24:     break;
25:   **end if**
26: **end while**
27: return: L^*

VI. LABEL COST

The central idea of this paper is to use the label cost to effectively constrain the number of labels in an album, so it is important to assign an appropriate label cost to different labels. In this section, we first present how to construct a label cost; after that, we will illustrate each component in detail.

The total cost for adding an individual to the album is the combination of two costs: a personal label cost and a social label cost. The basic idea behind this strategy is that every label should pay its personal cost to enter into the album. The social label cost represents the compatibility of different labels in the album. Formally, we express the total cost for a label as

$$C(\vec{y}) = C_{personal}(\vec{y}) + C_{social}(\vec{y}), \quad (5)$$

where $C_{personal}(\vec{y})$ is the personal label cost for each label and $C_{social}(\vec{y})$ represents the social cost for incorporating label l. The costs are described in the following sections.

A. Personal Label Cost

The personal label cost expresses the idea that a limited number of people should appear in an album. We define this cost as

$$C_{personal}(\vec{y}) = \sum_{l \in \mathcal{L}} \lambda \mathrm{I}(l, \vec{y}), \quad (6)$$

where $\mathrm{I}(l, \vec{y})$ is an indicator function defined as

$$\mathrm{I}(l, \vec{y}) = \begin{cases} 1 & \text{if } l \in \vec{y} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

This definition ensures that every label entering into the album pays its cost, which is the weight λ learned (see Section VII).

B. Social Label Cost

The social label cost function is similar to the personal cost formulation but is based on information from the social network. Given a particular labeling $\vec{y}$, the social cost of including a label l in the labeling is computed by summing costs representing the interaction between label l and all other labels in the labeling $\vec{y}$, which are formally expressed as

$$C_{social}(\vec{y}) = \sum_{l \in \mathcal{L}} S(l, \vec{y}) \mathrm{I}(l, \vec{y}), \quad (8)$$

where

$$S(l, \vec{y}) = \sum_{j \in \mathcal{L}} (\alpha_f C_f(l, j) + \alpha_c C_{co}(l, j)) \mathrm{I}(l, \vec{y}) + \alpha_u C_u(l). \quad (9)$$

The three components introduced below are the social costs derived from the social network:

- **Friendship Cost**. This cost measures whether individuals co-occurring in the album are friends:

$$C_f(i,j) = \begin{cases} 0 & i \text{ and } j \text{ are friends} \\ 1 & \text{otherwise.} \end{cases} \quad (10)$$

- **Co-Occurrence Cost**. This cost is similar to the friendship cost but measures whether the individuals in album have ever co-occurred in albums in the training data:

$$C_{co}(i,j) = \begin{cases} 0 & i \text{ and } j \text{ have co-occurred} \\ 1 & \text{otherwise.} \end{cases} \quad (11)$$

- **Uploader Cost**. The uploader cost uses the history contained in an uploader's previous photos. Like the previous two costs, it is based on an indicator function. Here, we define a potential which captures the relationship between an individual and the owner of the photo album[2]:

$$C_u(i) = \begin{cases} 0 & \text{if } i \text{ has appeared in images} \\ & \text{uploaded by the owner of } F \\ 1 & \text{otherwise.} \end{cases} \quad (12)$$

VII. TRAINING

Training the parameters for the structural model is a challenging problem. We use the Structural SVM (SSVM) because it can optimize parameter values even if inference can only return approximate solutions, as is the case in this model [5]. In this section, we will first introduce the SSVM method; after that, we demonstrate how to transform the energy in (3) into the SSVM expression.

A. Structural SVM

The SSVM deals with the general problem of learning a mapping from the inputs $x \in X$ to discrete outputs $y \in Y$ based on a training sample set $S = \{(x_i, y_i) | (x_i, y_i) \in X \times Y\}$. The label here is in a general form, which could be a numbered label in the case of multiclass classification or a parsing tree for a sentence in natural language parsing.

During training, the SSVM tries to learn a discriminant function over input/output pairs of the form

$$f(x, y, w) = w^T \Psi(x, y), \quad (13)$$

where Ψ generates a feature vector which depicts relations between inputs and outputs and w are model parameters that need to be learned.

A prediction is made by maximizing f over the response variable for a specific given input x. Formally, this can be expressed as

$$h(x) = \arg\max_{y \in Y} f(x, y). \quad (14)$$

To learn this map, SSVMs solve the following quadratic program

$$\min_{w, \xi \geq 0} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i,$$

[2]This model assumes that all photos in an album have been uploaded by a single user.

$$\forall i, \forall y \in Y \backslash y_i : w^T \Psi(x_i, y_i) \geq w^T \Psi(x_i, y) + \Delta(y_i, y) - \xi_i,$$

where $\Delta(y_i, y)$ is the loss function indicating how far $h(x_i)$ is from the true output y_i.

Introducing a constraint for every wrong output is typically intractable, and Joachims proposed a cutting plane algorithm which defines a separation oracle and finds the most violated label, constructs a sufficient subset of the constraints using these labels and iteratively solves the QP only over this subset; this guarantees polynomial time runtime and correctness [9].

B. Learning the Weight Vector

The parameters in our model that need to be learned are the personal label cost weight λ and the three weights α_f, α_{co} and α_u for the social costs. These parameters can be optimized using the SSVM training procedure by expressing the energy in (5) as the inner product of the weight vector $\vec{w}$ and feature vector $\Psi(\vec{x}, \vec{y})$. This can implemented by representing $\vec{w}$ as

$$\vec{w} = [1, \lambda, \alpha_f, \alpha_{co}, \alpha_u]. \quad (15)$$

The vector $\Psi(\vec{x}, \vec{y})$ is defined as

$$\Psi(\vec{x}, \vec{y}) = [D(\vec{y}; \vec{x}), C_{person}(\vec{y}), Z_f(\vec{y}), Z_{co}(\vec{y}), Z_u(\vec{y})]^T, \quad (16)$$

where

$$Z_f(\vec{y}) = \sum_{l \in \vec{y}} \sum_{j \in \vec{y}} C_f(l, j) \quad (17)$$

$$Z_{co}(\vec{y}) = \sum_{l \in \vec{y}} \sum_{j \in \vec{y}} C_{co}(l, j) \quad (18)$$

$$Z_u(\vec{y}) = \sum_{l \in \vec{y}} C_u(l) \quad (19)$$

For the small number of parameters in this model, we also had success with a randomized coordinate ascent approach, though the structural SVM solution was much faster.

VIII. EXPERIMENTS

In this section, we present experimental results conducted on two datasets collected from Facebook. In the following section, we will first give a detailed description about the datasets; then, we compare our results with a baseline system. After that, we combine our method with the model proposed in [18].

A. Datasets

As described in Section I, we replicated our experiments on two different datasets, using different implementations of our algorithms. These experiments were conducted separately on each dataset.

Both datasets were gathered from the Facebook accounts of volunteers, using a downloading application similar to that used in [18]. The datasets differed in the users used to capture the photographs. In the first dataset, the photographs were gathered from all of the albums visible to one user. The second dataset was gathered from a larger set of volunteers

that agreed to let us access photographs in their Facebook accounts, using the same permissions mechanism that any Facebook application can use to access personal data. We accessed all photos in albums available to our downloading application. We also stored available social network information, including friendship relationships, the identity of the uploader of each picture, and the way in which people have co-occurred in tagged photographs.

Not surprisingly, although these applications gave us access to photographs of hundreds of individuals, most individuals only had a handful of photographs. To ensure that we had enough data to properly perform both training and evaluation, we culled the dataset to include only individuals with a large number of samples available. This resulted in one dataset with 1951 facial images of 25 people and a second dataset with 1994 facial images of 15 people.

Each dataset is then partitioned into three parts:

1) A set of albums and images to be used to train the weights for the data cost. These images are used to train the linear classification weights for the image-only training.
2) A second set of albums and images that will be used to train the weights for the personal label and social costs. It is necessary to use a second training set because the training process used to learn the linear weights can separate the training data perfectly. If the training data in the first partition is used, then the data cost will be unrealistically accurate, and not enough weight will be given to the personal label and social costs.
3) A third set that will be used as a testing set. In our experiments, this test set included 454 images of faces in our first dataset and 487 images of faces in our second dataset.

The albums are divided chronologically to simulate how photographs would arrive in a social network: the first partition contains the oldest albums, while the third contains the newest.

B. Results

Fig. 2 summarizes our results for both datasets. The baseline bar in the graph shows the accuracy achieved using only the data cost, which is computed using a linear classifier and the image features extracted from each face image.

Implementing a personal label cost, as in (3), leads to an improvement in recognition accuracy for both datasets. This cost is based on the number of labels present in the album. The accuracy for the first dataset is improved from 38.1% (baseline) to 45.4%, and the accuracy for the second dataset is improved from 66.8% (baseline) to 75.0%.

The remaining bars in Fig. 2 show the accuracies of different combinations of social costs from Section VI-B. Most importantly, for both datasets, incorporating some form of a social label cost improves recognition performance considerably. Out of all of the social costs we explored, the uploader cost, which penalizes a potential candidate in a photograph if he or she has never appeared in any album uploaded by the owner of the photograph, seems to be the
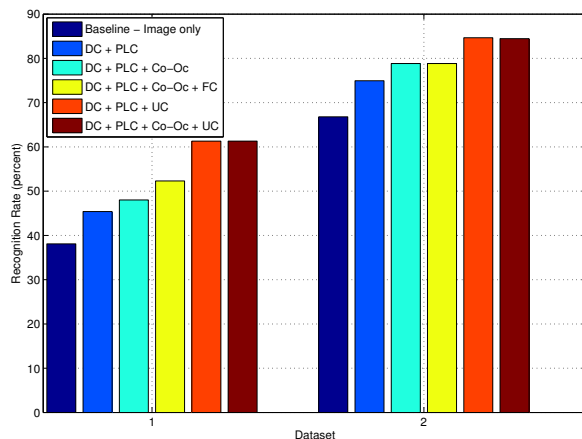

Fig. 2. This figure shows a visualization of the improvement provided by incorporating label-based priors into recognition. DC refers to the Data Cost, PLC refers to the Personal Label Cost, Co-Oc refers to the Co-Occurrence Cost, FC refers to the Friendship Cost, and UC refers to the Uploader Cost. In both datasets, adding a personal label cost significantly increases recognition performance over the image-only baseline face recognition system (described in Section V-B). Adding additional social costs also leads to further improvements. In both of the datasets, over 400 photographs were tested.

most important. The combination of the data cost, personal label cost, co-occurrence cost, and uploader cost improves accuracy on the first dataset from 38.1% to 61.3% and improves the accuracy on the second dataset from 66.8% to 84.5%.

On our first dataset, we also explored the use of a per-person label cost, tuning a specific weight for each person (e.g. learning λ_l for each l in $\mathcal{L}$); this increased accuracy by a further 5% on this dataset. However, we feel that incorporating a label cost specific to each individual in a large-scale social network would be infeasible, so we did not pursue this further.

C. Consistency of Results Between Datasets

For both datasets, incorporating the label-based priors yields consistent improvements: around 7% for the first and 8% for the second. Likewise, the inclusion of additional social costs further improves recognition performance on the two sets, up to a maximum gain of 23% for the first and 18% for the second.

We observe, though, that simply using the image-based data cost produces varied results across the datasets. This variation is most likely due to the differing number of people present in each of these datasets and not a result of our learning methodology. As expected, the dataset featuring more people – in this case, the first – achieves a lower accuracy. However, comparable performance gains achieved on both datasets when utilizing the personal and social label cost indicate the usefulness of our system across various scenarios.

D. Albums With Many People

While the results in the previous section demonstrate that the addition of a label cost can significantly improve

recognition results, this cost is designed for albums where the number of photos outnumbers the number of individuals. While we have shown that this behavior is common, we cannot expect it to hold across all albums.

To investigate this issue, we characterized each album with ratio measuring the number of individuals in the album with respect to the number of facial images; we will refer to this value as the *IdentityRatio*:

$$\text{IdentityRatio} = \frac{|\mathcal{L}_F|}{|F|}, \qquad (20)$$

where $|\mathcal{L}_F|$ is number of individuals appearing in album F and $|F|$ is the number of facial images in F. Because the set $\mathcal{L}$ is used to denote all possible labels, $\mathcal{L}_F$ denotes the labels in album F.

For the purpose of experimentation, we manually choose a threshold of 0.5 for the IdentityRatio and so only apply the label cost to albums having a value ≤ 0.5. The threshold is selected based on the observation that the label cost can be applied to the simplest possible album, which contains two facial images belonging to only one person.

Using this ratio, we can split our training sets into one set of albums well-suited for the label cost (i.e. those albums with IdentityRatio ≤ 0.5) and another set containing all albums, some of which could be problematic (i.e. some albums may have IdentityRatio > 0.5). The resulting performance difference is tiny: when training and testing on "good" albums, which are consistent with our label cost prior, we see only a very slight improvement – approximately 2% – over training and testing on the complete dataset. This suggests that incorporating the label cost does not induce instability into the recognition system.

E. Comparison with Image Co-Occurrence Model

Our use of a CRF model makes it convenient to use it to aid a model based on co-occurrence in photographs, similar to [18]. Accordingly, we culled our dataset to find photographs containing two individuals in our set[3]. This was difficult because a number of photos contained multiple individuals, but our tests are limited to a subset of individuals. Because of the restriction in finding photographs with two people, our training set was limited to 33 pictures. The testing set contained 32 for the same reason. While this set is too small to make statistically significant results, our preliminary experiments have shown that when these models can be combined, recognition rates improve by around 5%.

IX. CONCLUSIONS AND FUTURE WORK

The album storage structure for photos in a social networks provides a strong source of information regarding the identities of the people in those photographs. This paper has introduced a structural SVM-based system that is able

[3]The number of people per photograph was limited to two, as in [18], to make brute-force inference possible during training.

to exploit this information. As shown in the experiments, utilizing this information leads to significant improvements in recognition accuracy.

In future work, we hope to better integrate models like the work of Stone *et al.* [18] that model co-occurrence inside pictures.

REFERENCES

[1] D. Anguelov, K. chih Lee, S. Gokturk, and B. Sumengen. Contextual identity recognition in personal photo albums. pages 1 –7, June 2007.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2173 –2180, June 2010.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[5] T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *International Conference on Machine Learning (ICML)*, pages 304–311, 2008.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Additive Logistic Regression: a Statistical View of Boosting. *The Annals of Statistics*, 38(2), 2000.

[7] A. Gallagher and T. Chen. Using group prior to identify people in consumer images. pages 1 –8, June 2007.

[8] D. S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22:148–162, 1982.

[9] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.

[10] A. A. Kuehn and M. J. Hamburger. A heuristic program for locating warehouses. *Management Science*, 9(4):pp. 643–666, 1963.

[11] M. Naaman, H. Garcia-Molina, A. Paepcke, and R. B. Yeh. Leveraging context to resolve identity in photo albums. Technical Report 2005-2, Stanford InfoLab, January 2005.

[12] M. Parry. Harvard's Privacy Meltdown - Harvard Researchers Accused of Breaching Students' Privacy. In *The Chronicle of Higher Education*, July 2011.

[13] N. Pinto, J. DiCarlo, and D. Cox. How far can you get with a modern face recognition test set using only simple features? *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:2591–2598, 2009.

[14] N. Pinto, J. J. Dicarlo, and D. D. Cox. Establishing Good Benchmarks and Baselines for Face Recognition. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, Marseille France, 2008. Erik Learned-Miller and Andras Ferencz and Frédéric Jurie.

[15] N. Pinto, D. Doukhan, J. J. DiCarlo, and D. D. Cox. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS Computational Biology*, 5(11), 11 2009.

[16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. 2002.

[17] J. Sivic, C. L. Zitnick, and R. Szeliski. Finding people in repeated shots of the same scene. In *Proceedings of the British Machine Vision Conference*, 2006.

[18] Z. Stone, T. Zickler, and T. Darrell. Autotagging Facebook: Social network context improves photo annotation. IEEE, 2008.

[19] L. Zhang, L. Chen, M. Li, and H. Zhang. Automated annotation of human faces in family albums. In *In MULTIMEDIA 03: Proceedings of the Eleventh ACM International Conference on Multimedia*, pages 355–358. ACM Press, 2003.

[20] M. Zhao and S. Liu. Automatic person annotation of family photo album. In *Proc. International Conf. on Image and Video Retrieval*, pages 163–172, 2006.