User-based Ranking of Facial Attractiveness

Abhineet Jain and Anurag Gupta

***Abstract*— In this paper, we propose and implement a user-based relative ranking of facial attractiveness. Given training data of faces sorted based on a users personal preference, we learn how to rank novel faces according to the user [1]. Using GIST feature descriptor and Ranking SVM classifier our system achieves an average accuracy of 79.6% on pairwise comparisons of novel faces. We examine the effectiveness of our method using GIST, DSIFT and HOG feature descriptors, and learn to rank with classifiers like RankSVM and RankRLS.**

I. INTRODUCTION

”Beauty lies in the eyes of the beholder.” Facial attractiveness is highly subjective and based on personal taste of the user.

In today’s era, many people use web-based dating services, that use textual information for matching interests, and ranking user profiles. Services employing use of profile pictures for matching user’s beauty preferences are not known as of now.

We define our goal as: given a set of faces ordered by preference, learn how to rank novel faces using a criteria similar to that of the individual who presented that ordering.

This paper focuses on differentiating between any two faces, in terms of which face is more attractive based upon an individuals preference. We stress that in our work, we express relationships over the attractiveness of faces at the fine granularity of pairs of faces, and not at some coarse granularity of a categorical fashion.

We introduce a system that predicts the relative ranking of facial attractiveness based on different individuals tastes. The average accuracy achieved by our system is 79.6%, achieved by the number of correctly ordered pairs among all possible pairs. This accuracy metric is based on the Kendall Tau rank similarity measure This accuracy was achieved when GIST feature descriptor was used with Ranking SVM classifier.

A. Dataset

We used the widely known Color FERET dataset [9] and searched online, to extract 150 female faces that were cropped and oriented into frontal faces, with minimal extra features outside the face. Each image is 256x256 pixels in size. Figure 1 shows some of images from the dataset.

B. Organization

This paper is organized in the following manner. Section 3 describes the approach of this paper in tackling this problem. Section 4 discusses our experiments and their results. Section 5 presents the current future direction of this work. Section 6 concludes.

Fig. 1. Dataset

II. PROCEDURE

In this section we discuss how we obtain the relative orders sorting of faces (Section 2.A), and the main learning techniques used (Section 2.B).

A. Sorting and Relative Order

We use a sorting method that includes binning. Sorting is split into two phases. In phase 1, images are placed in bins, where placing a face in a bin, say A, established that it is better than all faces in another bin, say B. For concreteness, the binned-sort method establishes that:

$$\forall F_i \in A, F_j \in B \rightarrow F_i \succ F_j$$

where A and B are bins of faces. We used ten bins to emulate a score from 1 to 10. However, we emphasize that this scoring emulation has no connection to the actual ranking, it is only meant to reduce the number of sorting operations. The second phase of the binned-sort begins by showing the user all the faces in each bin, and the user is asked to arrange them in order. At the end of this phase, we obtain the ordered tuple $O = \langle F'_1, F'_2, ..., F'_n \rangle$.

B. Learning to Rank Relatively

The faces in the dataset are represented as a set $F = \langle F_1, F_2, ..., F_n \rangle$. The sorted faces list is given by a subject as an ordered tuple $O = \langle F'_1, F'_2, ..., F'_n \rangle$ and is interpreted in the following way - $F'_1 \succ F'_1 ... \succ F'_1$, where ”$F'_i \succ F'_j$” denotes that F'_i is more attractive than F'_j

Let x_i represent the feature vector of Fi. Our goal is to learn the function:

$$g(x_i) = w^T x_i$$

subject to the constraints:

$$\forall F'_i, F'_j, i \neq j, F'_i \succ F'_j \rightarrow g(x_i) > g(x_j)$$

The problem can be solved using two methods -

1) Ranking SVM: We present in this section the primal training algorithm of a linear SVM classifier along the lines of (Keerthi and DeCoste, 2005; Chapelle, 2007b) [2]. Given a training set of n examples $(x_1, y_1), ..., (x_n, y_n)$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, 1\}$, the aim is to find the solution of the following optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i(w^T x_i + b))^2$$

A pair-wise preference matrix is created containing n rows and n columns. Each row represents a pair. The index that has value 1 has higher preference over index having value -1. All other n-2 elements in the row are zero. The preference matrix is generated for every user and is input to the algorithm. This implementation is based on RankSVM code [2] by O.Chapelle.

2) Ranking RLS: Following (Pahikkala et al., 2007) [3] we consider the following type of least-squares based ranking cost function -

$$l(f(X), Y) = \frac{1}{2}\sum_{i,j=1}^{m}((y_i - y_j) - (f(x_i) - f(x_j)))^2$$

The cost function is the sum of the squares of differences between the predicted and correct magnitudes of all the pair-wise preferences in the training set. We define the Laplacian Matrix as follows -

$$L = \begin{cases} -1 & i \neq j \\ m-1 & i = j \end{cases}$$

Accordingly, the cost function can be re-written as

$$l(f(X), Y) = (Y - Ka)^T L(Y - Ka),$$

and the objective function to be minimized becomes -

$$J(a) = (Y - Ka)^T L(Y - Ka) + \lambda(a^T Ka),$$

where K is the linear kernel, and a is the weight vector to be learned.

Taking derivative of J(a) with respect to a and setting it to zero, we can determine the value of the coefficient vector a that determines a minimizer of J(a) for a training set S:

$$a = (KLK + \lambda K)^{-1} KLY$$

Leave-Pair-Out Cross Validation is applied on the dataset to find how the classifier performs, which is averaged out over all features.

III. Experiments

A. Features

To accomplish our ranking goal, we have extracted the features of each image using three different descriptors - GIST, HOG and Dense-SIFT + PCA.

1) Dense-SIFT + PCA: This was one of the descriptors used[5]. Dense grid of SIFT features was computed at each pixel with a spatial bin of 3x3 pixels. On the 241x241x128 feature vector generated, PCA was applied using fast Singular Value Decomposition [7]. K was chosen to be 20, and each face was thus represented by a 2560-dimensional vector storing local gradient information.

2) GIST: It provides a rough description of the image, by summarizing the gradient information for different parts [6]. We convolved image with 32 Gabor filters at 4 scales and 8 orientations, producing 32 feature maps. Each feature map was divided into 16 regions, and mean calculated within within each region. It resulted in a 512-dimensional vector.

3) HOG: Histogram of Oriented Gradients counts occurrences of gradient orientation in localized portions of an image [8]. The image was divided into small connected cells (31x31). For the 8x8 pixels within each cell, the histogram was compiled and contrast-normalized across the block of 4 cells. The result was a vector of size 34596.

B. Measuring Accuracy

To measure the accuracy of our methods, we used ranked order comparing tools like Kendall Tau [4]. The Kendall Tau measures the number of pairwise inversions between two ordered lists L_1, L_2 as follows:

$$\tau(L_1, L_2) = \sum_{\forall (i,j) \in L_1} I((j, i) \in L_2)$$

Based on the Kendall Tau we construct our accuracy measurement to account for correct pairs divided by the total number of pairs. If N is the total number of pairs, then our accuracy measurement for a list L_1 matching L_2 is:

$$\alpha(L_1, L_2) = 1 - \frac{\tau(L_1, L_2)}{N}$$

C. Facial Attractiveness Ranking

After collecting 150 facial images, we computed GIST, DSIFT and HOG feature descriptors for the dataset of images. Thus, three feature vectors are obtained for each image. The dimensions for each of them was 512, 241x241x128, and 34596 respectively. Hence, it was necessary to implement PCA to reduce dimensions of the DSIFT vector, which was reduced to a 2560-dimensional vector. We used Fast Singular Value Decomposition (fsvd) to implement PCA over the DSIFT descriptor.

Five users volunteered to take part in the two-phase sorting survey. A portal was developed, wherein they had to first place the images in bins numbered from 1 to 10, and then give relative ranking to the faces in each bin, based on how attractive they found the faces. The rankings provided by each user were saved in a separate file.

Based on these rankings, a vector containing indices in order of rank, and the respective feature descriptors, were input to the RankSVM classifier. The preference matrix (pair-wise) was also calculated. The classifier implemented 5-fold

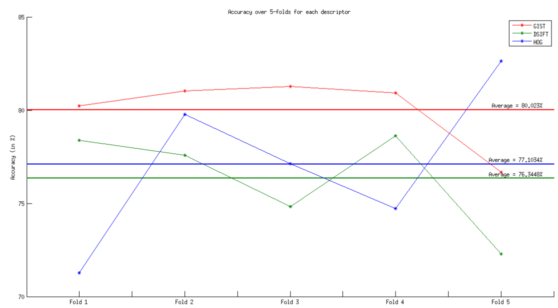

Fig. 2. Held-out estimation for User 1, 3 descriptors, RankSVM

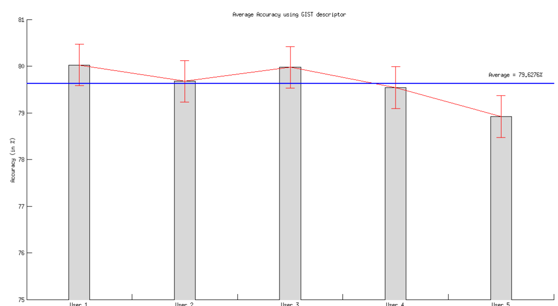

Fig. 3. User-wise accuracy for GIST using RankSVM

cross validation over the 150 image dataset, as shown in Figure 2.

The average accuracy for each feature descriptor per user was stored and observed. Refer to Figure 3, 4, and 5 for the results.

To explore the method of Ranking RLS, a vector containing rank of each index, the respective feature descriptors, were input to the RankRLS classifier. The classifier implemented 5-fold cross validation over the 150 image dataset.

The average accuracy for each feature descriptor per user was stored and observed. Refer to Figure 3, 4, and 5 for the results.

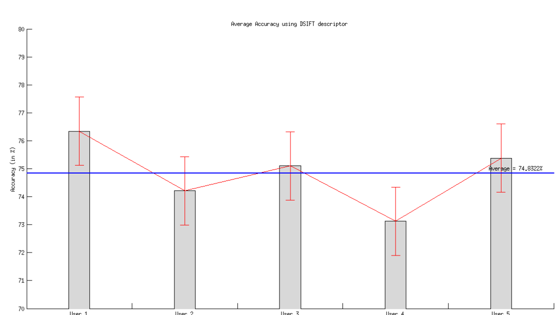

Fig. 4. User-wise accuracy for DSIFT using RankSVM

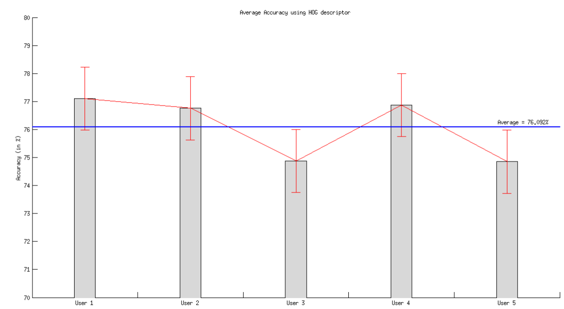

Fig. 5. User-wise accuracy for HOG using RankSVM

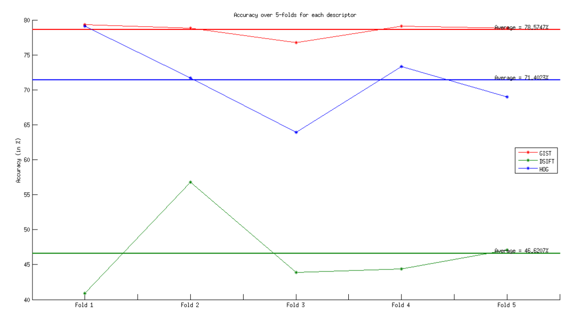

Fig. 6. Held-out estimation for User 1, 3 descriptors, RankRLS

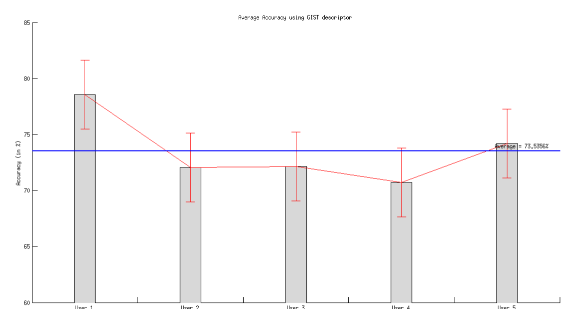

Fig. 7. User-wise accuracy for GIST using RankRLS

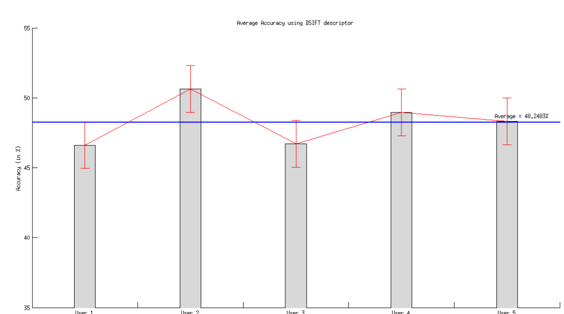

Fig. 8. User-wise accuracy for DSIFT using RankRLS

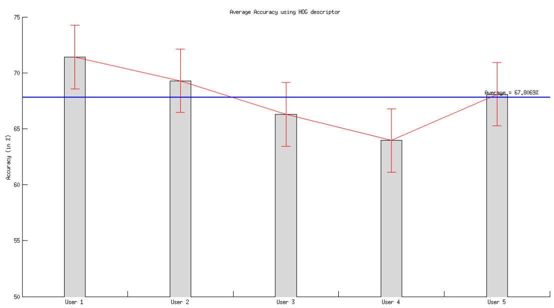

Fig. 9. User-wise accuracy for HOG using RankRLS

TABLE I
COMPARISON OF RESULTS

Feature	Average Accuracy RankSVM	Average Accuracy RankRLS
GIST	79.627	73.536
DSIFT	74.832	48.248
HOG	76.092	67.807

IV. FUTURE WORK

The current application of the idea can be improved by using better feature descriptors, including Facial Geometry, L*a*b color histograms. A combination of features may also be used to describe each image.

The use of the concept of kernelization may be employed while using classifiers like RankSVM, RankRLS. Kernels like polynomial, or RBF, may improve the accuracy of our method.

The possible application of this solution lies in suggesting matches to a user based on his preferences in facial attractiveness. In the world of online profile matching, it would be very useful.

V. CONCLUSION

We learned the ranking of facial images by users to learn what elements of beauty they saw in a face. The classifiers learned were then able to rank novel faces according to the personalized preferences, showing an average accuracy of 79% in case of GIST descriptor with RankSVM classifier. This clearly shows that our system was able to identify certain beauty trends which subjects revealed in their preferences as a form of consistency. On the other hand, random preference permutation failed to deliver such consistencies.

Also, out of the two classifiers applied, RankSVM was clearly better and more accurate than RankRLS for each descriptor.

ACKNOWLEDGMENT

We would like to express our gratitude towards our professor Dr. Avinash Sharma, who mentored us through the project and pushed us to do move out of our comfort zone and excel. We are also grateful to our Teaching Assistants for their valuable comments and feedback throughout the duration of the project, as well as the volunteers who took part in our extensive survey. Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office.

REFERENCES

[1] Altwaijry, H., & Belongie, S. (2013, January). Relative ranking of facial attractiveness. In Applications of Computer Vision (WACV), 2013 IEEE Workshop on (pp. 117-124). IEEE.

[2] O. Chapelle. Training a support vector machine in the primal. Neural Computing, 19:11551178, May 2007.

[3] Pahikkala, T., Airola, A., Boberg, J., & Salakoski, T. (2008). Exact and efficient leave-pair-out cross-validation for ranking RLS. Proceedings of AKRR, 2008, 1-8.

[4] R. Kumar and S. Vassilvitskii. Generalized distances between rankings. In Proceedings of WWW 10, pages 571580, New York, NY, USA, 2010. ACM.

[5] A. Vedaldi and B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008, http://www.vlfeat.org

[6] Aude Oliva, Antonio Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, International Journal of Computer Vision, Vol. 42(3): 145-175, 2001.

[7] Halko, N., Martinsson, P. G., Shkolnisky, Y., & Tygert, M. (2010), An algorithm for the principal component analysis of large data sets, Arxiv preprint arXiv:1007.5510, 0526

[8] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886-893). IEEE.

[9] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, ”The FERET Evaluation Methodology for Face Recognition Algorithms,” IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 22, pp. 1090-1104, 2000.