# User-based Ranking of Facial Attractiveness

Final Presentation

Group 6
Abhineet Jain and Anurag Gupta

# Problem Statement

- A personalized facial attractiveness ranking system
- Given training data of faces sorted based on the subject’s personal taste, we learn how to rank novel faces for automated beauty filtering based on user preferences

- Related Work –
  Relative Ranking of Facial Attractiveness, *Hani Altwaijry and Serge Belongie*
  IEEE Workshop on Applications of Computer Vision (WACV), 2013

# Basic Pipeline

# Step I - Dataset

- 150 female faces extracted from publicly available image datasets
- Requested dataset from Color FERET, plus searched online
- Each image, cropped and oriented, ensured straight face with minimal extra features outside the face
- Resized to 256x256 pixels for uniformity

# Step II - Two Phase Sorting

- Developed a portal for survey
- After login, user directed to instructions
- Phase I – User placed each face in a bin ranging from 1 to 10

Phase I - Bins range from 1 to 10, 1 being the lowest, and 10 being the highest in beauty.

Showing 81 - 85 of 150 images

7

7

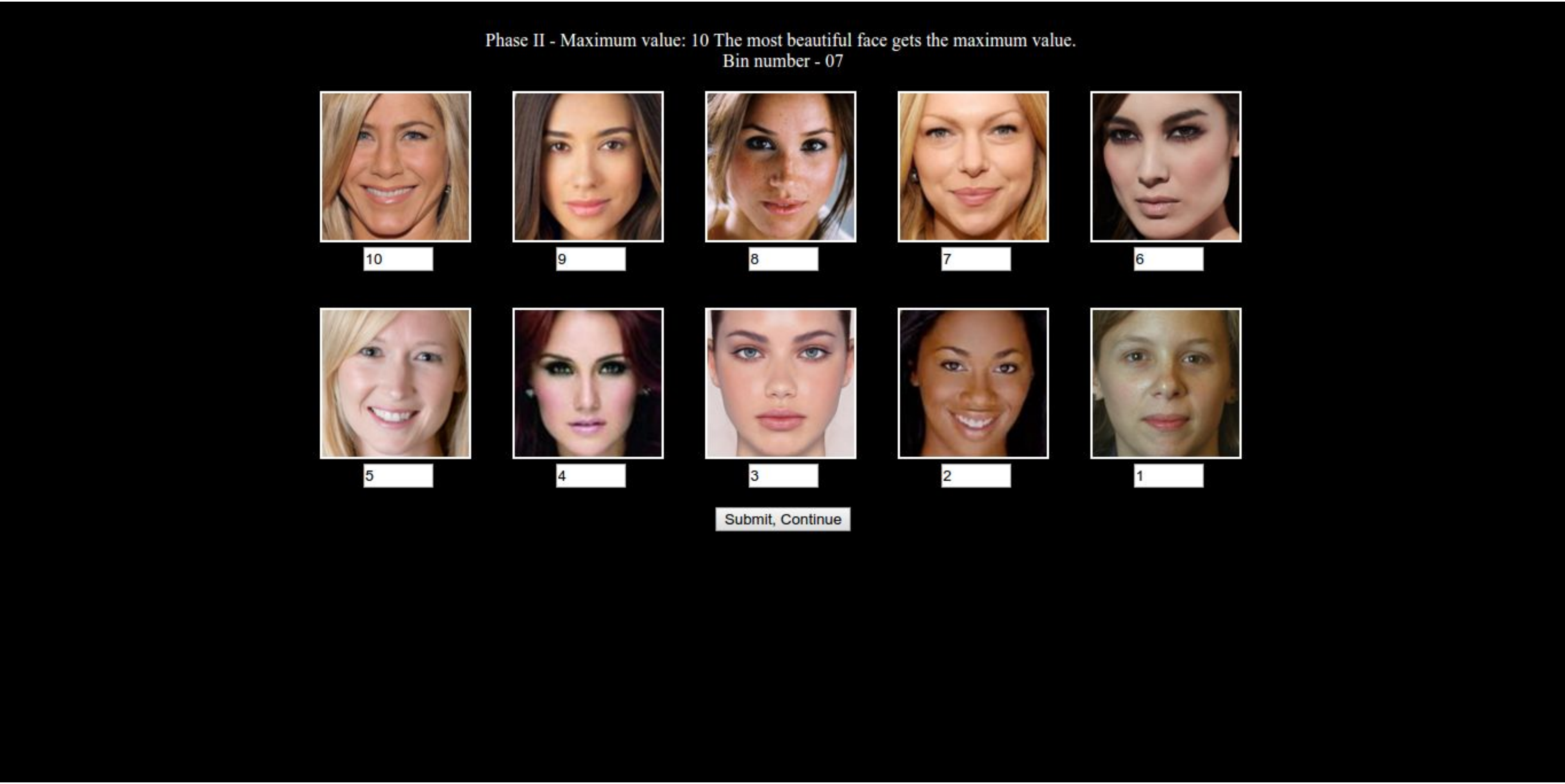
10

7

7

# Step II - Two Phase Sorting

- Developed a portal for survey
- After login, user directed to instructions
- Phase I – User placed each face in a bin ranging from 1 to 10
- Phase II – User sorted faces in each bin according to their attractiveness
- Ground truth obtained for each user, will apply held-out estimation to determine error


Phase II - Maximum value: 10 The most beautiful face gets the maximum value.
Bin number - 07
10
9
8
7
6
5
4
3
2
1
Submit, Continue

# Step III - Feature Descriptors

- On each image, implement three descriptors
  - GIST
  - Dense SIFT + PCA
  - HOG

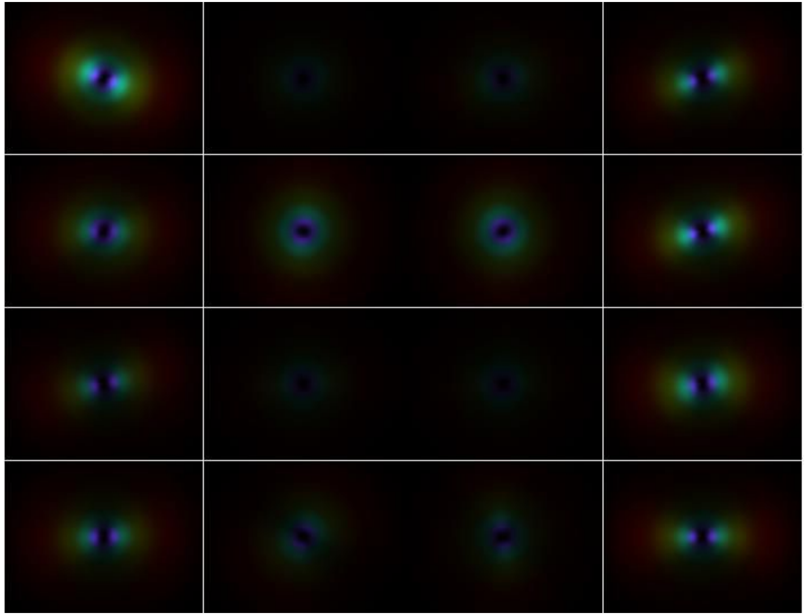
# GIST

Summarizes the gradient information for different parts of image, providing rough description.

- Convolve image with 32 Gabor filters at 4 scales, 8 orientations, producing 32 feature maps
- Divide each feature map into 16 regions, take mean of feature values within each region
- Concatenate mean values of all 32 feature maps
- Result - GIST descriptor of size 512 (16X32)

# GIST

- We have implemented GIST descriptor for each image in the dataset.



# Dense SIFT + PCA

Dense Scale Invariant Feature Invariant –

- Local descriptor for all keypoints (location, scale, orientation) w.r.t. gradient feature for each pixel
- Compute a dense grid of SIFT descriptors at each pixel
    - Spatial-bin size of 3x3 pixels
    - Resulting in 241x241x128 dimensions for each image
- Apply PCA using FSVD with k=20
    - Each 241x241 slice is projected onto a 20-dimensional vector
- Result - DSIFT descriptor of size 2560 (20x128)

# Histogram of Oriented Gradients

The technique counts occurrences of gradient orientation in localized portions of an image.

- Image is divided into small connected regions – cells (31x31)
- For pixels within each cell (8x8 pixels), a histogram of gradient directions is compiled
- Local histograms are contrast-normalized –
    - Calculate measure of intensity across a larger region – block (composed of 4 cells)
    - Use this value to normalize all cells within the block
    - Results in invariance to changes in illumination and shadowing
- Result - HOG descriptor of size 34596

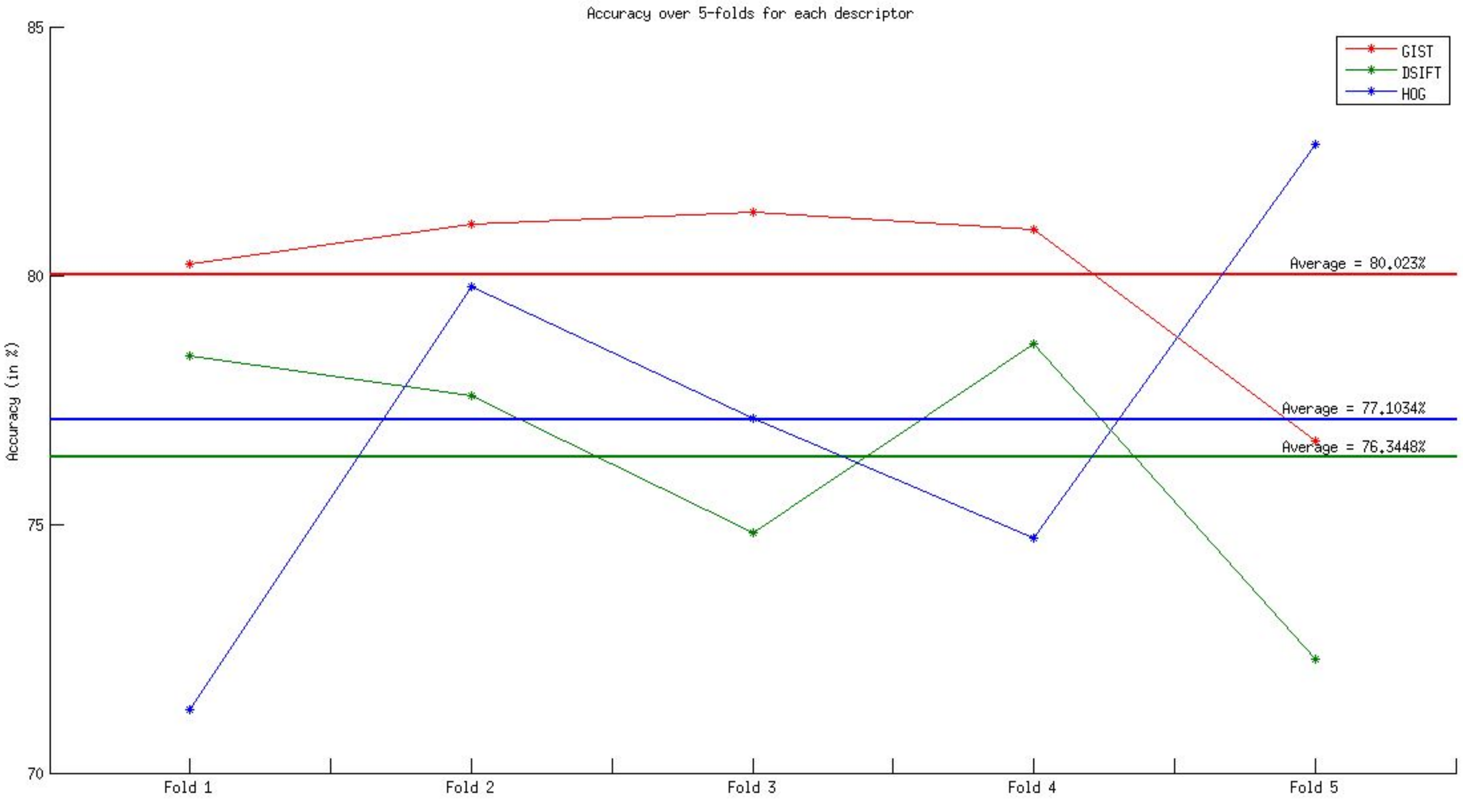
# Step IV - Learning to Rank

- On each descriptor, implement two ranking classifiers
    - Ranking SVM
    - Ranking Regularized Least Squares

# Ranking SVM

- If $x_i$ is feature vector of $F_i$ , our goal is to learn the function:
  $$g(x_i) = w^T x_i$$
  subject to the constraint: $\forall\ F_i , F_j , i \neq j, F_i > F_j \rightarrow g(x_i) > g(x_j)$
- Given a training set of n examples $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_i \in R^d$ and $y_i \in \{-1, 1\}$, the aim is to find the solution of the following optimization problem -
  $$\min_{\mathbf{w},b} \ \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))^2.$$
- Preference matrix - pairwise preference of images (n x n) is computed and input to Ranking SVM, along with respective feature descriptors

# Ranking SVM - Held-out Estimation

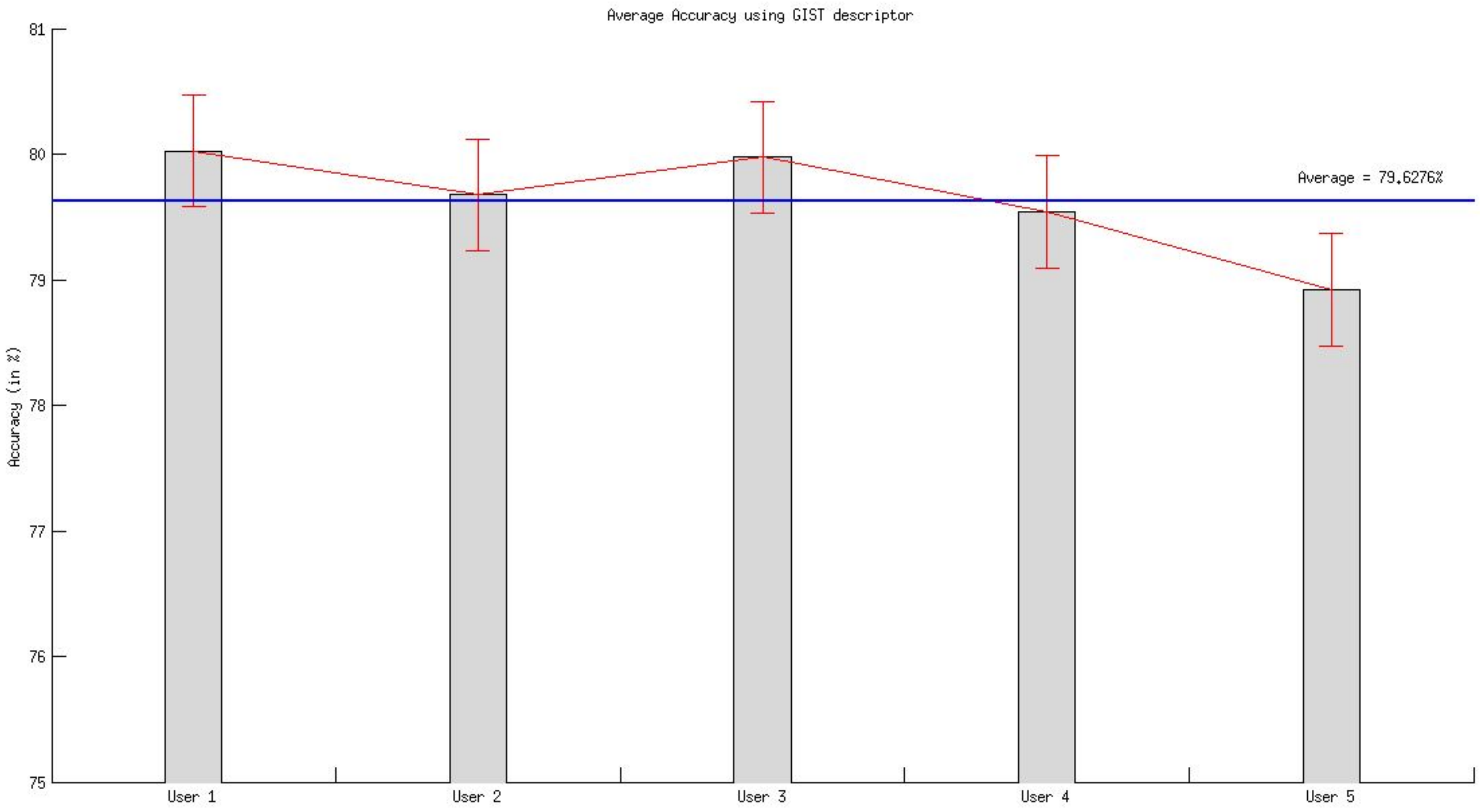


# Ranking SVM - Held-out Average

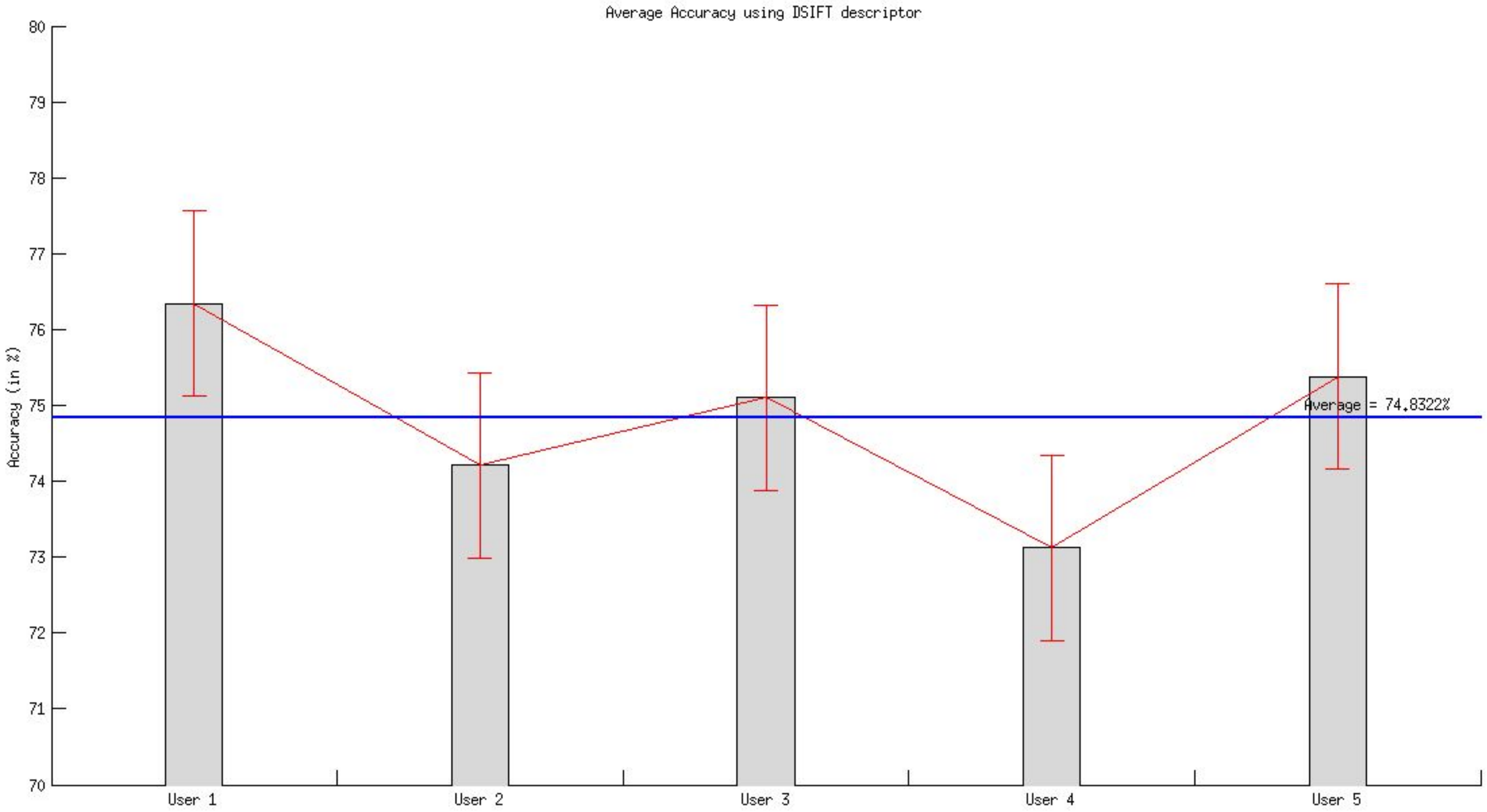
Accuracy for each feature per user averaged over 5-fold held-out estimation

| Features | User 1 | User 2 | User 3 | User 4 | User 5 |
|---|---|---|---|---|---|
| **GIST** | 80.023 | 79.678 | 79.977 | 79.540 | 78.919 |
| **DSIFT** | 76.345 | 74.206 | 75.103 | 73.126 | 75.379 |
| **HOG** | 77.103 | 76.758 | 74.873 | 76.873 | 74.850 |

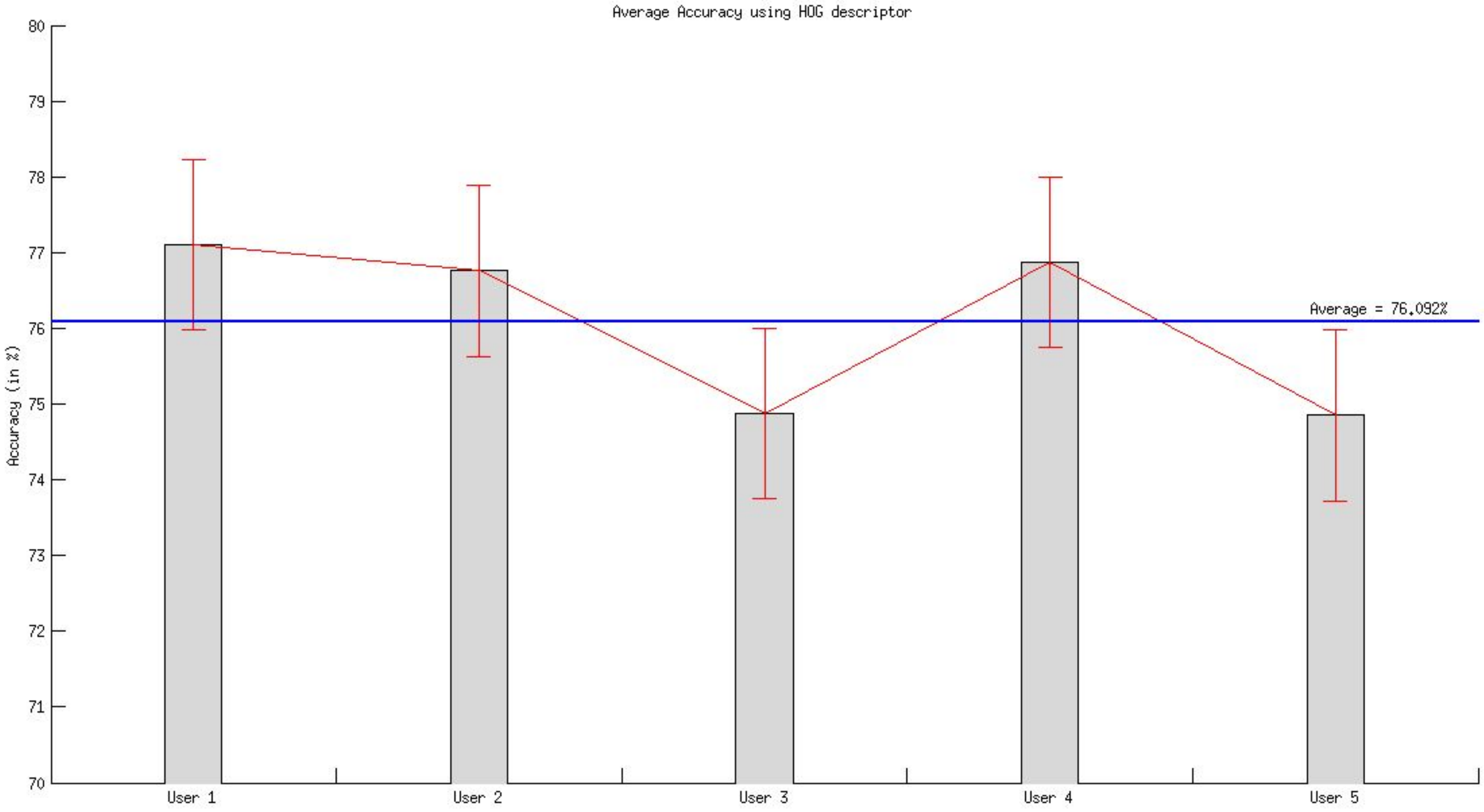
# Ranking SVM - GIST



# Ranking SVM - DSIFT



# Ranking SVM - HOG



# Ranking RLS

- Ranking cost function is based on least squares -

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

- Expressing it as the square of differences of $r_i$ and $r_j$ $\forall$ i, j, we get -

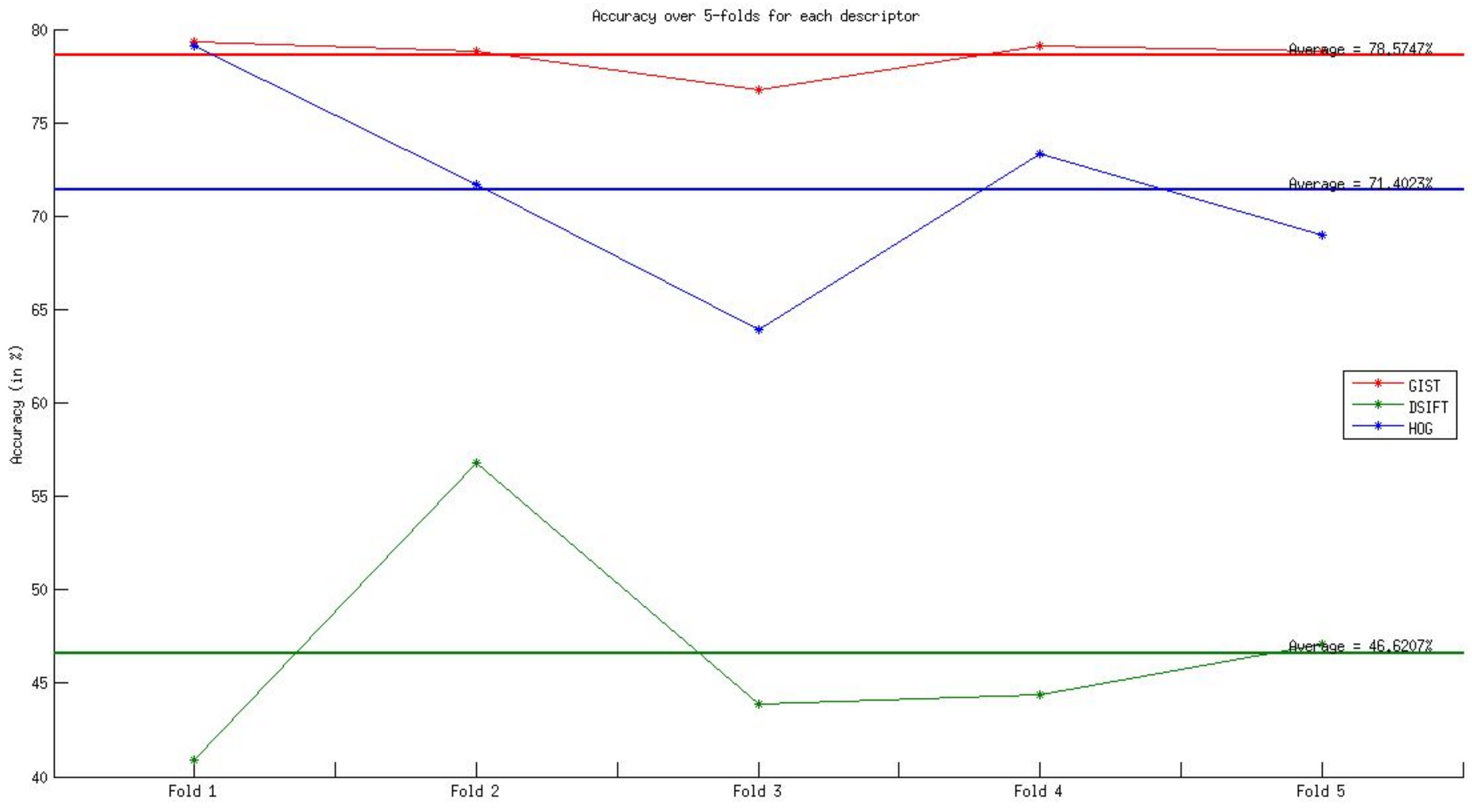
$$J(\vec{a}) = (Y - K\vec{a})^{\mathrm{T}} L (Y - K\vec{a}) + \lambda \vec{a}^{\mathrm{T}} K \vec{a}.$$

  where K is the linear kernel, a is the weight vector, and L is the Laplacian matrix given by

$$L_{i,j} = \begin{cases} -1 & \text{if } i \neq j \\ m-1 & \text{if } i = j \end{cases}$$

- Leave-Pair-Out Cross Validation is applied over trained weight vector.

# Ranking RLS - Held-out Estimation

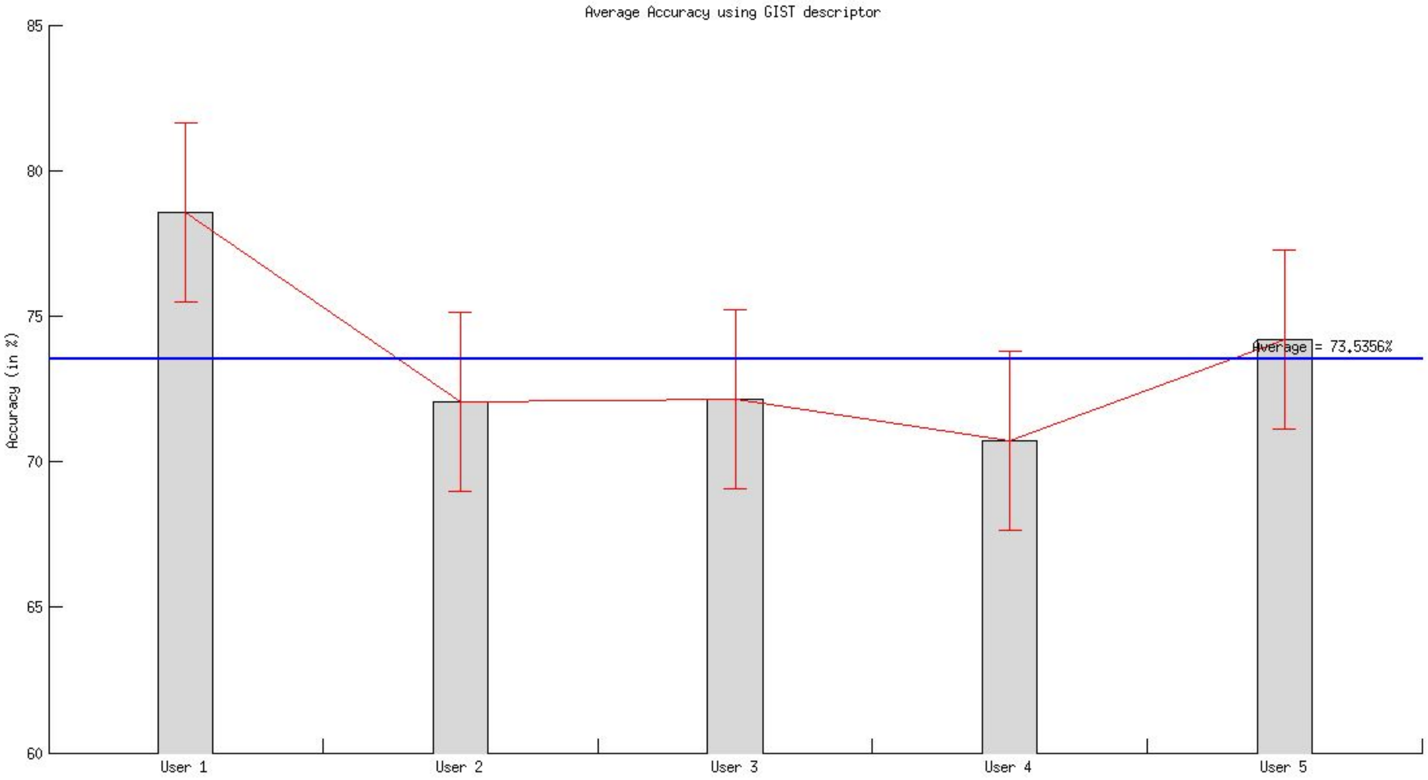


# Ranking RLS - Held-out Average

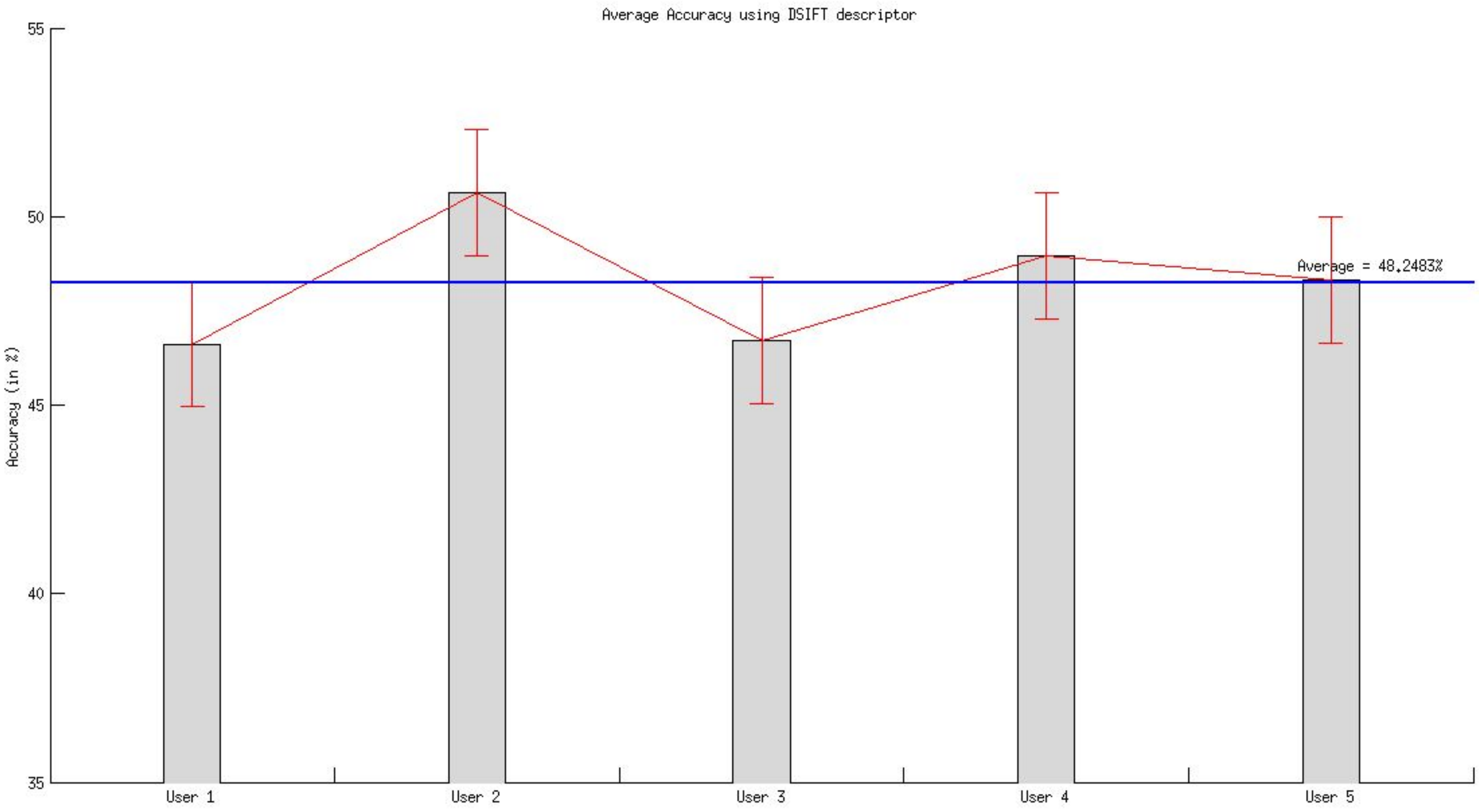
Accuracy for each feature per user averaged over 5-fold held-out estimation

| Features | User 1 | User 2 | User 3 | User 4 | User 5 |
|---|---|---|---|---|---|
| **GIST** | 78.575 | 72.046 | 72.138 | 70.712 | 74.207 |
| **DSIFT** | 46.620 | 50.620 | 46.712 | 48.965 | 48.322 |
| **HOG** | 71.402 | 69.287 | 66.299 | 63.954 | 68.092 |

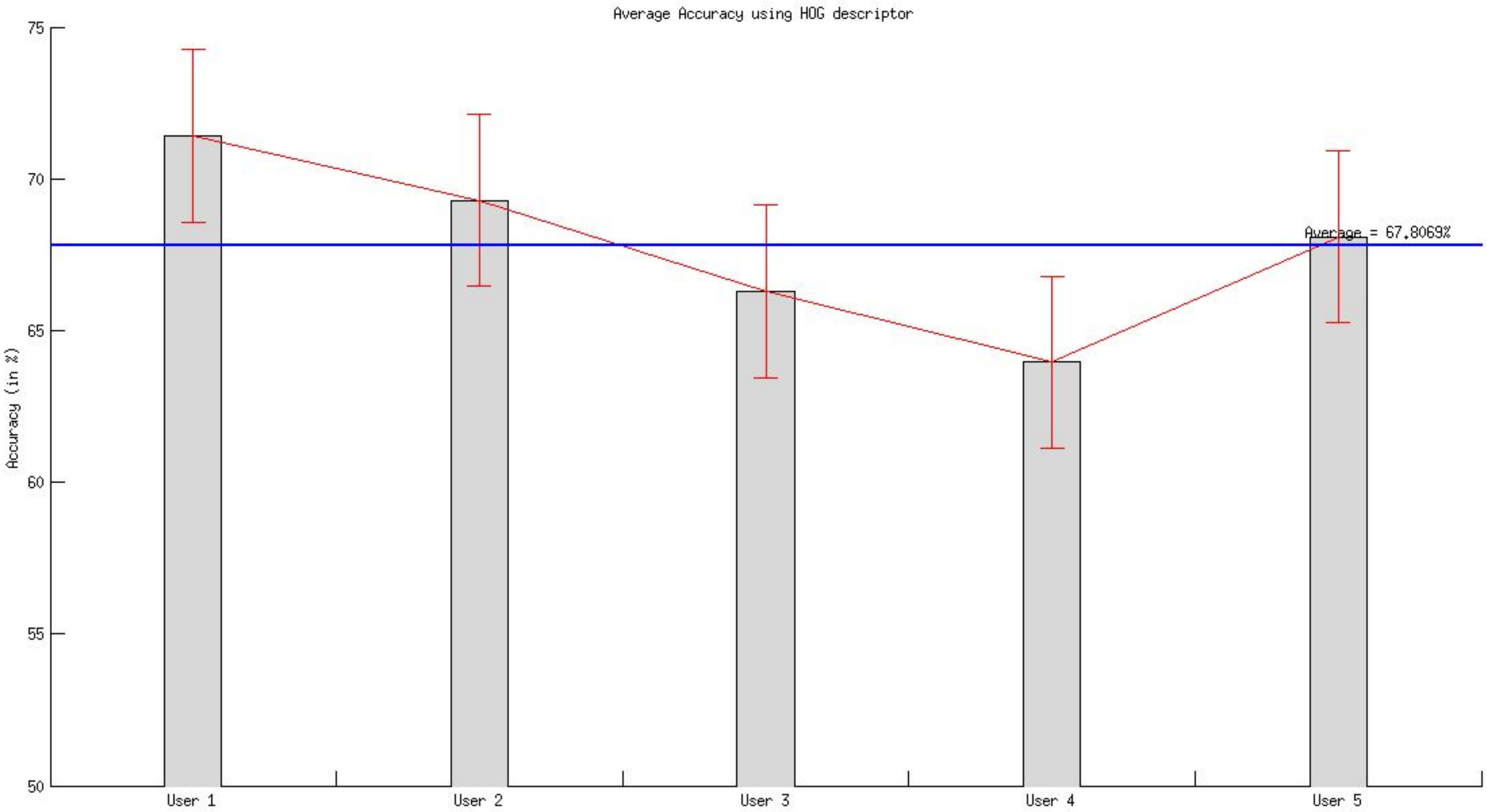
# Ranking RLS - GIST



# Ranking RLS - DSIFT



# Ranking RLS - HOG



# Results - Comparison

Accuracy of each feature averaged over all users for both classifiers

| Feature | Average Accuracy Ranking SVM | Average Accuracy Ranking RLS |
|---|---|---|
| **GIST** | 79.627 | 73.536 |
| **DSIFT** | 74.832 | 48.248 |
| **HOG** | 76.092 | 67.807 |

# Observations

- Ranking SVM gives much better accuracy than Ranking RLS
- DSIFT performs the worst and GIST performs the best with both classifiers

# 79.62%

Best accuracy is achieved using GIST descriptor with Ranking SVM