Abstract—**Face recognition is an active research area that is widely applied in surveillance, entertainment and human computer interaction applications. In this project, We use histogram features to classify if the person in the CMU face image dataset is wearing a pair of sunglasses or not. Using the SMO algorithm with a polynomial kernel, the classification result shows a accuracy of 0.6223 (kappa = 0.2438), which is a statistical significant improvement over the baseline model.**

I. Introduction

A. Background

Computer vision and face recognition are both areas that involve machine learning techniques to do image classification. Comparing with other data mining tasks, image classification requires more advance methods to generate image features/descriptors. Using raw pixels to understand the context of the image usually cannot give reliable results, because they are noisy and affected a lot by the background, the light condition and the distortion of the camera [1] [2]. Scale Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) are both techniques used in image feature generation [3] [4]. However, they are both heavy-weight in computation.

The requirement of real-time face recognition is growing with the increasing capability of robots and mobile consumer devices. In this project, we try to explore a light-weight feature engineering method to do real-time face recognition. There are existing researches in which people use light-weight features such as Local Binary Pattern to do face recognition [5] [6]. In this project, we try to explore even simpler approach. Instead of HOG, SIFT or local patterns, we want to do image classification only with the statistical distribution of the pixel greyscale values.

We use the greyscale face images donated by Tom Mitchell [7]. This dataset is already used in a variety of image classification researches. Niyogi et al. use this dataset to study Locality Preserving Projections, a projection method that can reduce the dimension of image data features [8]. Meila et al. use this dataset to test the mix-of-tree model they developed to do image classification and other general purpose classification tasks [9].

We limit the scope of this project within classifying if the person in picture is wearing a pair of sunglasses. We used a open source machine learning software - LightSide to build, test and evaluate our model [10]. We also use the machine learning toolset Waikato Environment for Knowledge Analysis (WEKA) to implement machine learning algorithms [11]. We first generate the histogram features for each images. Then we add geometric information in the features space by dividing the image into 9 regions and generating histogram features for each region. Total 303 features are generated and used to build the model in a support vector machine with a polynomial kernel. The final test result shows an accuracy value of 0.6223 and a kappa = 0.2438.

The rest of the paper is organized as follows. Section 2 introduces the description and the preparation of the dataset. Section 3 describes the baseline algorithm in this project. Section 4 gives the detail steps of improving the classification algorithm of this project. We will describe our error analysis, the improved model, the regional histogram features and the feature selection tuning process in that section. Section 5 gives the final test results of the model. Section 6 provides the conclusion.

II. Data Description

CMU face image dataset contains 640 grey-scale images in .pgm format. They were collected from 20 volunteers. 32 different pictures were taken from each volunteer. During one image collection experiment, the volunteer was requested to turn the head to four different positions: up, left, right and bottom. The volunteer was also asked to express four different facial expressions: neutral, happy, sad, and angry. For each combination of a position and a expression, the volunteer was asked to take two pictures, one with a pair of sunglasses and one without. The user id, position, facial expression and open/sunglasses were all labeled in the title of the picture files. A sample picture is shown in Fig. 1.

Each picture is stored in three formats: a full-resolution (128×120), a half-resolution image (64×60), and a quarter-resolution image (32×30). Each pixel is represented by an 8-bit unsigned integer number for its greyscale value. Images are in 20 different folders named after the volunteer name. In this project, we choose to classify the full-resolution images. This is because the more pixels in one image, the more information we can extract from the image.

We divide the dataset into 1) a development dataset, on which we do feature space design and error analysis, 2) a cross-validation dataset, on which we explore the algorithms and do parameter tuning, and 3) a test dataset, on which we do the final test of our final model. Since this face image dataset is collected from different people, and each person contributes multiple images to the dataset. If we randomly make division on the dataset, the images collected from a same

Fig. 1. Example face image in the dataset

person will be distributed in multiple datasets. Then there will be a subpopulation problem. The model we built may overfit images for a specific person, because the data we use to do error analysis and build our model are from a same person. The final testing result will also be inflated, due to that the data we use to build the model and test the build may come from a same person.

To prevent the subpopulation problem, we make our the dataset division according to person names. The development dataset contains 64 images from 2 people. The cross-validation dataset contains 373 images from 12 people. The test dataset contains 188 images from 6 people.

III. Baseline Algorithm

An intuitive solution to this sunglasses classification task is comparing the average greyscale values of each image. When the people in the image is wearing a pair of sunglasses, the mean and median values of the pixel greyscale values will be lower than without sunglasses. This is because the sunglasses in the image are presented as black pixels. The images with sunglasses will have more black pixels than those without sunglasses. According to this assumption, we calculate the mean value, the median value and the standard deviation value of each image as our features for the baseline test. Table I shows the detail descriptions of the features.

TABLE I
FEATURE TABLE OF BASELINE ALGORITHM.

No	Feature	Type	Description
1	Name	Nominal	People ID to identify the source of the image
2	Mean	Numerial	Mean value of the pixel greyscale values
3	Median	Numerial	Median value of the pixel greyscale values
4	Std	Numerial	Standard deviation value of pixel greyscale values
5	Glasses	Nominal	0: without sunglasses; 1: with sunglasses

The **Glasses** feature in the table is the label of the image that we want classify. The gold standard for labeling the data is provided by the creator of the dataset. The **Name** feature is not related to the image classification, therefore it will not be used for building models. We generate this feature just for performing automatic leave-one-out cross-validation in LightSide. We use leave-one-out cross-validation instead of default stratified cross-validation to prevent the subpopulation problem. In the baseline test, we only have three features to do the classification. We choose J48 decision tree to build a simple model. The setup of the LightSide software is shown in Fig. 2.

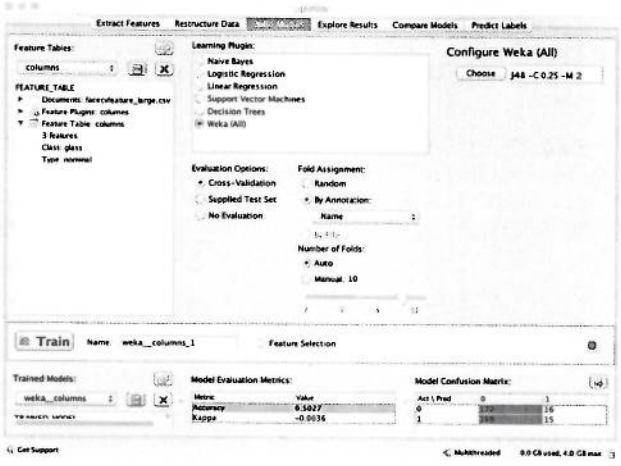

Fig. 2. LightSide setup for the baseline algorithm

The classification accuracy in the baseline test is 0.5027, and the Kappa value is -0.0036.

IV. Model Building

A. Error Analysis

We start to explore our model from doing an error analysis on the baseline test result. To prevent any overfitting when building the model, I test the baseline model and do the error analysis on the development dataset. The baseline model predicts all the images in the development dataset are without sunglasses. The accuracy value is 0.5 and the kappa value is 0. Table II shows the confusion matrix of the baseline test on the development dataset.

TABLE II
CONFUSION MATRIX OF BASELINE TEST ON DEVELOPMENT DATA

Act/Pred	0	1
0	32	0
1	32	0

We used the *Explore Result* function to analysis the confusion matrix. We focus on the **Mean** feature since the assumption we made is that the mean values of the pixel greyscale values in images with sunglasses are lower than those without sunglasses. In LightSide, we find the vertical difference of **Mean** features between two actual classes is 3.566. The average value of **Mean** features without sunglasses is higher than those with sunglasses. This proves that our assumption is correct. However, the horizontal difference of **Mean** features

is 71.9831, which is much larger than the vertical difference. This means that though the average values of **Mean** features in two categories are discrimitive, the **Mean** values of images in each category are distributed in a large range. The distribution of the **Mean** features in 64 development dataset images is shown in Fig. 3.

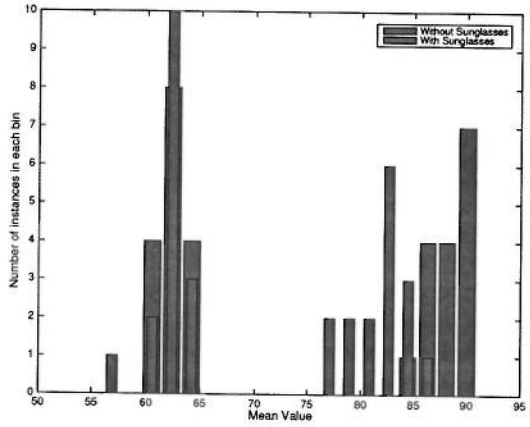

Fig. 3. Mean Value distribution in the development dataset.

The distribution figure is naturally divided into two parts. This is due to the development dataset consists of images from two people. Images counted in the left part comes from ID *an2i* while those counted in the right part come from ID *at33*. From the distribution of each individual part, the **Mean** value of the images without sunglasses is higher. However, this difference is very small comparing with the difference between the two parts. As shown in Fig. 3, the images from *at33* with sunglasses all have larger **Mean** value than the images from ID *an2i* without sunglasses. From looking into two sample images in the development dataset from ID *at33* and ID *an2i*, we can see that images from different people will have very different distribution of greyscale values (Fig. 4). Using the mean value solely for classification will neglect the difference of the greyscale distributions in the images.

B. Improvement

1) Feature Engineering: From the error analysis, we know that the single mean value of pixels greyscale values is not discriminative enough for this classification task. It cannot reflect the greyscale distribution of the whole image. Therefore, we generate histogram features to reflect the distribution of pixel values in each image. We mainly focus on the greyscale value ranging from 0 to 120. This is because the pixels with greyscale over 120 are mostly overexposure pixels that cannot provide any information. We divide the greyscale range into 30 bins and count the pixels in each bin. We normalized the counts by dividing them with the total amount of pixels. The normalized values reflect the percentage of pixels in each histogram bin. We add these 30 features into the baseline

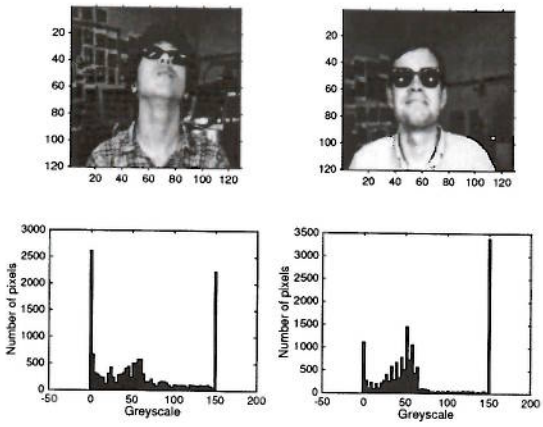

Fig. 4. Different greyscale distribution of two images in development dataset.

feature space. The new feature space table is shown in Table III.

TABLE III
FEATURE TABLE OF THE IMPROVED ALGORITHM

No	Feature	Type	Description
1	Name	Nominal	People ID to identify the source of the image
2	Mean	Numerial	Mean value of the pixel greyscales
3	Median	Numerial	Median value of the pixel greyscales
4	Std	Numerial	Standard deviation value of pixel greyscales
5 – 34	Bin{4, 8, 12, 16 ...120}	Numerial	Percentage of pixels in each histogram bin. (eg. Bin4 is the percentage of pixels greyscale values ranging within 0 - 4)
35	Glasses	Nominal	0: without sunglasses; 1: with sunglasses

Again, the feature **Name** will not be used in classification but is added for performing leave-one-out cross-validation in LightSide. The column **Glasses** is the label of the dataset which is generated by parsing the image filenames. There are total 33 features including the 3 features from the baseline algorithm and 30 histogram features.

2) Algorithm Selection: Since the number of features in our feature spaces is large, the decision tree algorithm may build a very complex model. Therefore, we change our algorithm to the SMO (a optimized version of support vector machine) in the WEKA algorithm plugins to build the model. We choose a support vector machine algorithm because image classification tasks usually require a nonlinear model to make robust prediction. We use a polynomial kernel with exponential value 1.0 in SVM.

3) Evaluation: We use the new feature space and the SMO algorithm to build our model again in LightSide. The leave-one-out cross-validation shows an accuracy value of 0.5484 and a kappa value of 0.0956. Using the model comparison function in LightSide, we find the model we built with the

new feature space is insignificant improved over the baseline model (p = 0.141, shown in Fig. 5).

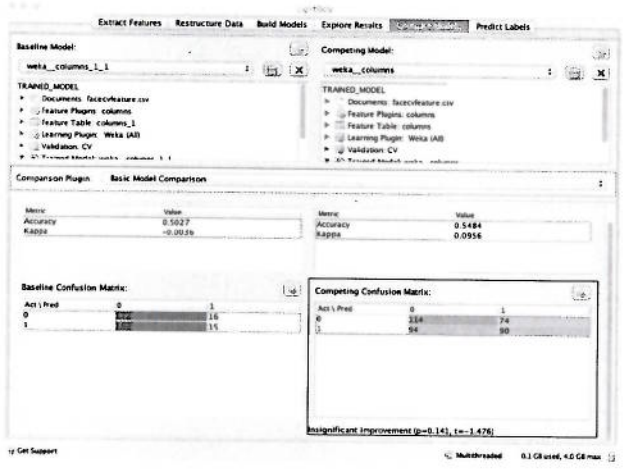

Fig. 5. Comparison of the improved model and the baseline model.

We use the development dataset to do the error analysis again. The accuracy of the model when testing on development dataset is 0.5781 (kappa = 0.1562). The confusion matrix is shown in the Table IV.

TABLE IV
CONFUSION MATRIX OF THE IMPROVED MODEL ON DEVELOPMENT DATASET

Act/Pred	0	1
0	27	5
1	22	10

By exploring the features of the error cases in the development dataset, the **Bin60** and **Bin4** features have high weights and high horizontal differences. We filter out the pixels counted in these two bins in the development dataset images and shown them in Fig. 6 and Fig. 7.

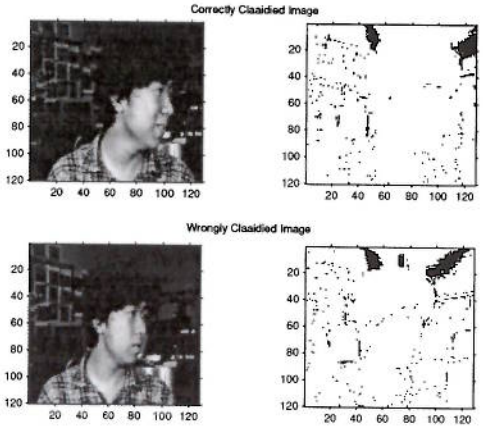

Fig. 6. Comparison of bin60 pixels in two images.

Fig. 6 shows the distribution of pixels counted by bin60 in a correctly classified image and a wrongly classified image. These two images are slightly different from each other in the person's facial expressions. However the bin60 feature in the wrongly classified image has a much higher value than that in the correctly classified image. This is due to that the change of light condition or camera exposure creates more shady area in the background.

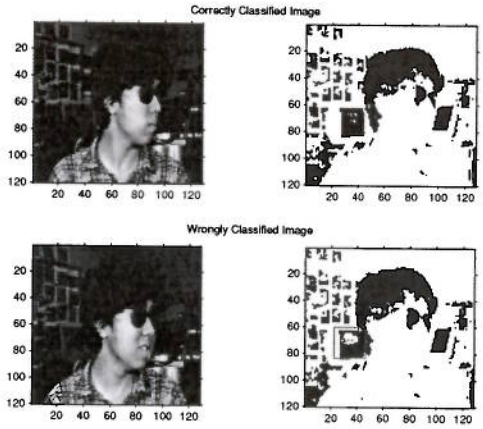

Fig. 7. Comparison of bin4 pixels in two images.

Fig. 7 shows the distribution of pixels counted by bin4 in a correctly classified image and a wrongly classified image. Pixels of the sunglasses is usually counted in this bin (greyscale values ranged from 0 to 4). In the comparison, we notice that the distribution of the pixels in the face area is not largely changed. However, there is one area of pixels missed in the wrongly classified image in the background (marked in the red rectangular). This miss part is due to the change of light condition or camera exposure as well.

From these two comparison, we know that the pixel distribution in the background is very sensitive to the light condition and the camera exposure. However, the area of people face is not that sensitive to these environmental factors.

4) Regional Histogram Features: The previous error analysis shows the distribution of greyscale values in the background area can be affected by the light condition or the camera exposure. It will generate noise in the features and makes the classification inaccurate. To compensate the noise, inspired by the research of Local Binary Pattern features, we design regional histogram features in our feature space.

We divide the image into 9 different zones evenly and generate histogram features for all nine regions. We label them zone 1 to zone 9 from the left top sub-image to the right bottom sub-image. As shown in Fig. 8, the center region (zone 5) has the pixels of sunglasses, which should be focused in our classification model. We generate the histogram features for each regions of the image. By doing so, we indirectly adds geometric information in our feature space. We expect the geometric information will be able to compensate the noise

Fig. 8. Nine divided regions of the image.

introduced by the background pixels. Table V (in the next page) shows our feature table after adding regional histogram features.

There are 303 features in our feature space. We use the SMO algorithm to build the new model on the cross-validation dataset.The leave-one-out cross-validation shows an accuracy value of 0.629 and a kappa value of 0.2581. Using the student t-test function in the *Compare Model* panel, we compare the new model and the baseline model. It shows a high significant improvement of the new model over the baseline model (p = 0.001, shown in Fig. 9).

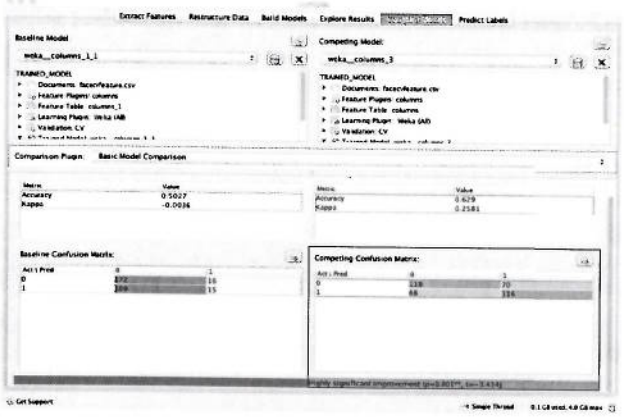

Fig. 9. Model comparison of the baseline model and the new model.

C. Feature Selection Tuning

The current feature table has 303 features. We want to use feature selection technique to remove the redundant features and irrelevant features in the feature space. Considering the goal of the this project is realizing robust image classification with a light-weight feature space, we want to control the feature number within 60. We choose three feature numbers - 20, 40 and 60. We do leave-one-out cross validations with these three feature selection settings. The results are shown in Table VI.

TABLE VI
RESULTS OF FEATURE SELECTION.

Feature Number	Baseline	F=20	F=40	F=60
Accuracy	0.629	0.7016	0.6747	0.6425
Kappa	0.2581	0.4034	0.3488	0.2832

To validate the feature selection tuning method, we further divide the cross-validation dataset into 6 training/testing pairs. Each pair contains a training dataset of images from 10 persons and a testing dataset of images from 2 person. We do leave-one-out cross-validations on each training set with three feature selection settings as well as with the default setting (no feature selection). We then select the best model built in each fold and test it using the test dataset. We use the kappa value to evaluate our result. The whole process is documented in Table VII.

TABLE VII
FEATURE SELECTION TUNING VALIDATION.

Fold	Default	F=20	F=40	F=60	Selection	Test Result
1	0.1033	0.1755	0.1499	0.1179	20 features	0.25
2	0.2398	0.3181	0.2078	0.2279	20 features	0.2812
3	0.2652	0.3043	0.3286	0.3361	60 features	0.073
4	0.2217	0.3436	0.2917	0.3502	60 features	0.3752
5	0.198	0.3384	0.4231	0.266	40 features	0.2773
6	0.2038	0.2675	0.2739	0.1146	40 features	0.5218
Average	0.2053	0.2912	0.2792	0.2355		0.2964

Though the average kappa of the testing results is higher than that of the default setting, the student t test shows the difference is not significant (p = 0.2460). So we will keep the default setting to build our final model.

V. FINAL MODEL BUILDING AND TESTING

We build our final model with the feature space shown in Table V on the cross-validation dataset. We use the SMO algorithm with a polynomial kernel. The final model shows a high significant improvement over the baseline model (shown in Fig. 9). We use the test dataset to test the model. The final test result shows an accuracy value of 0.6223 (kappa = 0.2438).

VI. CONCLUSION

In this project, we explored a histogram feature space for a image classification task. The feature space includes 30 histogram features of the whole image, and 270 (9×30) histogram features of 9 different regions of the image. Adding the mean value, the median value and the standard deviation value, there are total 303 features in the feature space. We trained the model using support vector machine with a polynomial kernel. The final model shows a significant improvement over the baseline model. The final test result shows an accuracy value of 0.6223 and a kappa value of 0.2438.

There are three conclusions that can be learnt from this project.

TABLE V
FEATURE TABLE WITH REGIONAL HISTOGRAM FEATURES

No	Feature	Type	Description
1	Name	Nominal	People ID to identify the source of the image
2	Mean	Numerial	Mean value of the pixel greyscales
3	Median	Numerial	Median value of the pixel greyscales
4	Std	Numerial	Standard deviation value of pixel greyscales
5 – 34	Bin{4, 8, ...120}	Numerial	Percentage of pixels in each histogram bin of the whole image.
35 – 64	Bin_1_{4, 8, ...120}	Numerial	Percentage of pixels in each histogram bin of zone 1.
65 – 94	Bin_2_{4, 8, ...120}	Numerial	Percentage of pixels in each histogram bin of zone 2.
95 – 124	Bin_3_{4, 8, ...120}	Numerial	Percentage of pixels in each histogram bin of zone 3.
125 – 154	Bin_4_{4, 8, ...120}	Numerial	Percentage of pixels in each histogram bin of zone 4.
155 – 184	Bin_5_{4, 8, ...120}	Numerial	Percentage of pixels in each histogram bin of zone 5.
185 – 214	Bin_6_{4, 8, ...120}	Numerial	Percentage of pixels in each histogram bin of zone 6.
215 – 244	Bin_7_{4, 8, ...120}	Numerial	Percentage of pixels in each histogram bin of zone 7.
245 – 274	Bin_8_{4, 8, ...120}	Numerial	Percentage of pixels in each histogram bin of zone 8.
275 – 304	Bin_9_{4, 8, ...120}	Numerial	Percentage of pixels in each histogram bin of zone 9.
305	Glasses	Nominal	0: without sunglasses; 1: with sunglasses

1) Human face dataset suffers from the subpopulation problem. In the error analysis of the baseline algorithm, we found in images from different people, the distributions of the mean values are very different. We can infer that if we do a stratified cross validation instead of the a leave-one-out cross validation, the model evaluation result will be inflated a lot.
2) Image data are sensitive to the environmental factors such as light condition especially in the background area. The distribution of the pixel greyscale values may be completely different in two images even when the subjects are exactly the same. Due to that fact, adding geometric information into the histogram feature space is useful. In this project, we use regional histogram features to add the geometric information indirectly. Obviously, using more advanced algorithm to separate the background will give more robust prediction results. However, this is out of the scope of this project.
3) Feature selection may not help getting a higher prediction accuracy in image classification tasks. Again, histogram features of image data are affected by the environmental factors a lot. Doing feature selection may give a better result for a subset of data, but it may make the model unstable. This effect is shown in the feature selection tuning validation process in this project. The kappa value for each fold can be as high as 0.5218 and as low as 0.073.

A future improvement for this project is calculating the regional histogram features in a finer granularity. Fig. 10 shows one way to divide the image. The center part in the image is divided into more parts. This is because in a webcam image, the human face is usually placed in the center region. A finer division will generate more geometric information into the image and will probably improve the accuracy of the classification.

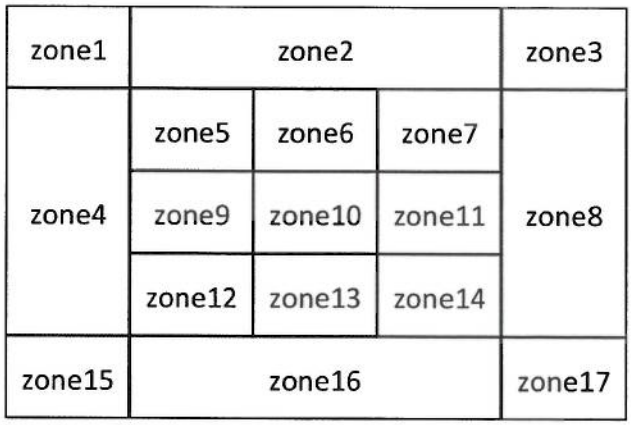

Fig. 10. 17 zones devision of a image.

REFERENCES

[1] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 10, pp. 1090–1104, 2000.

[2] P. J. Phillips, P. Grother, R. Micheals, D. M. Blackburn, E. Tabassi, and M. Bone, "Face recognition vendor test 2002," in *Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003. IEEE International Workshop on*. IEEE, 2003, p. 44.

[3] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "Hoggles: Visualizing object detection features," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–8.

[4] J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and B.-L. Lu, "Person-specific sift features for face recognition," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2. IEEE, 2007, pp. II–593.

[5] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 2037–2041, 2006.

[6] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 786–791.

[7] T. Mitchelll. (1999) Cmu face images data set. [Online]. Available: hhttp://archive.ics.uci.edu/ml/datasets/CMU+Face+Images

[8] X. Niyogi, "Locality preserving projections," in *Neural information processing systems*, vol. 16, 2004, p. 153.

[9] M. Meila and M. I. Jordan, "Learning with mixtures of trees," *The Journal of Machine Learning Research*, vol. 1, pp. 1–48, 2001.

[10] E. Mayfield and C. Rosé, "Lightside: Open source machine learning

for text accessible to non-experts," *Invited chapter in the Handbook of Automated Essay Grading*, 2012.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.