

EmoDetect – Smart Emotion Detection from Facial Expressions

Rishabh Animesh
ra523

Skand Hurkat
sh953

Abhinandan Majumdar
am2352

Aayush Saxena
as2825

Abstract

Human emotion detection is important to improve the interaction between humans and computers, by allowing computers to tailor their behavior according to the mood of the human operating the computer. In this project, we present a study of various feature extraction methods — *Gabor features*, *Histogram of Gradients*, *Haar-like features*, *Moments*) coupled with different machine learning algorithms — *Support Vector Machines (SVM)*, *Random Trees* (aka Random ForestsTM), and *Artificial Neural Networks (ANN)* to recognize and identify human emotion using facial expressions. We train the classifier on a database of around 10,000 images of 14 subjects emoting *Anger*, *Disgust*, *Fear*, *Happiness*, *Neutral*, *Sadness*, and *Surprise*. We train and cross validate over these extractor-learner combinations to find the best parameters; and then compare their relative performances. We also conduct experiments to test how accurately humans identify emotion. Our results show that the best performance is obtained using Gabor features coupled with a linear SVM with an accuracy of around 63%. However, we find that humans show an accuracy of around 74% and outperform every extractor-learner combination that we implemented.

1 Introduction

The past years have seen computers come into every aspect of our lives. An active area of research is in improving the interactions between humans and computers. Our project aims to improve the human computer interaction by providing techniques for a computer to identify human emotion, and to tailor its behavior accordingly. Detection of human emotion can improve interactions with machines in everyday life. For instance, a personal robot can detect the emotions of its user, and respond accordingly. Smart houses can detect the mood of the residents, and adjust parameters like lighting, air conditioning, power usage of personal equipment etc. accordingly. A smart car can detect when the driver is incensed, and automatically pull over and stop; thereby preventing accidents caused due to road rage. Some existing applications of emotion detection are nViso[12] which captures and analyzes the emotional response and visual attention of consumers for applications such as market research and brand management. Samsung researchers have developed a smart phone that can infer the user’s emotional state based on how the user operates the phone[8]. This shows a huge potential of emotion detection applications in the commercial market today.

In this project, we develop a system capable of detecting human emotion from static images. We detect the emotions *Anger*, *Disgust*, *Fear*, *Happiness*, *Neutral*, *Sadness*, and *Surprise*. We use 10,000 images comprising 14 subjects from the MUG[2] facial expression dataset and apply the following feature detection algorithms: *Gabor Wavelets*, *Histogram of Gradients*, *Haar-like features*, and *Moments*. We train on these extracted features using the following machine learning techniques: *Support Vector Machine (SVM)*, *Random Trees*, and *Artificial Neural Networks (ANN)*. We perform two different cross validation schemes on all extractor-learner combinations to get optimal learning parameters. We use these parameters to statistically compare relative performances of these extractor-learner combinations. The first scheme is to randomly shuffle feature data into training and validation data, and perform 20-fold cross validation. The second scheme splits the feature data by subjects and performs 14-fold cross validation, in which we train on 13 subjects, and validate on the remaining subject.

We find that the best results are obtained when using Gabor features in conjunction with a linear SVM. We report a prediction error rate of $37.1 \pm 8.9\%$ on a random untrained subject. We also experimentally evaluate the ability of humans to recognize emotion, by showing 50 random images from a dataset of 355 images comprising 52 subjects to 14 humans, and record their responses. We note that humans have a prediction error rate of $26.4 \pm 4.7\%$, only 11% better than the best extractor-learner combination.

The strength of our approach is its generalization beyond the subjects in the training set. Our method of cross-validation by separating the data-set by subjects, as compared to usual random shuffling enforces the generalization of our approach for a real application. We validate this hypothesis with a demonstration during the poster presentation, which showed that Gabor+SVM is quite effective in recognizing emotion from live images taken during the demonstration. Furthermore, by comparing the accuracy of our approach with humans who have over 20 years of experience identifying emotion, we are comparing our approach with the best available benchmark.

This document is organized as follows. Section 2 defines the problem statement. Section 3 presents a background on the feature extraction and machine learning algorithms used. Section 4 evaluates different extractor-learner combinations and presents a statistical analysis. Section 5 presents related work in this area, and Section 6 describes limitations of our approach and possible avenues for future work.

2 Problem Definition

In this project, we attempt to develop a system to identify human emotion from images and predict one of seven facial expressions, namely, *Anger*, *Disgust*, *Fear*, *Happiness*, *Neutral*, *Sadness*, and *Surprise*. Our work experiments with different feature extractors and machine learning algorithms, and proposes the best combination for the task. The questions we are trying to answer are:

- Which feature extractor is the best?
- Which learning algorithm gives best results in conjunction with the feature extractors?
- What are the optimal learning parameters?
- How well does our approach generalize beyond subjects in the training set?
- How well does it compare against emotion recognition by humans?

3 Our Method

3.1 System Overview

A high level system overview is shown in Figure 1. An image of a person whose facial expression is to be classified is captured. The face of the person is then extracted from the image using Viola and Jones method for object detection[13]. Feature extraction algorithms then extract relevant features from the cropped facial image. Trained classifiers then map these features to the appropriate emotion. The system uses the best extractor-learner combination.

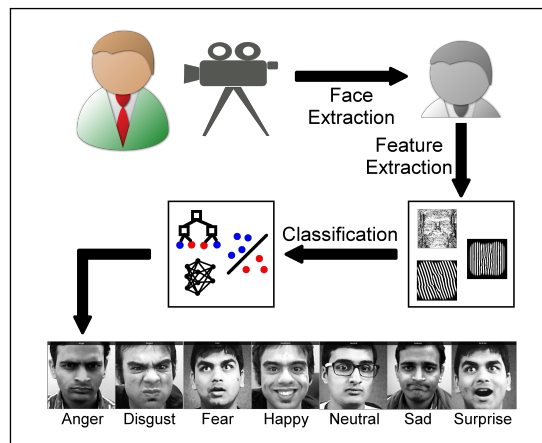


Figure 1: System Overview

We implement our code using OpenCV[1], an open source computer vision library. We have made the source code for our project available at [3].

3.2 Data Set

We use a facial expression database from MUG[2], the Multimedia Understanding Group at the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki. We signed an agreement with MUG to get access to the database with the conditions that we cite their work[2], that we do not publish the images without express consent of the subjects, and that we are using their database for non-commercial purposes.

The MUG database consists of image sequences of 52 subjects performing facial expressions. The background is a blue screen and well illuminated. The size of this database is approximately 38 GB of image sequences. The subjects were asked to emote expressions of anger, disgust, fear, happiness,

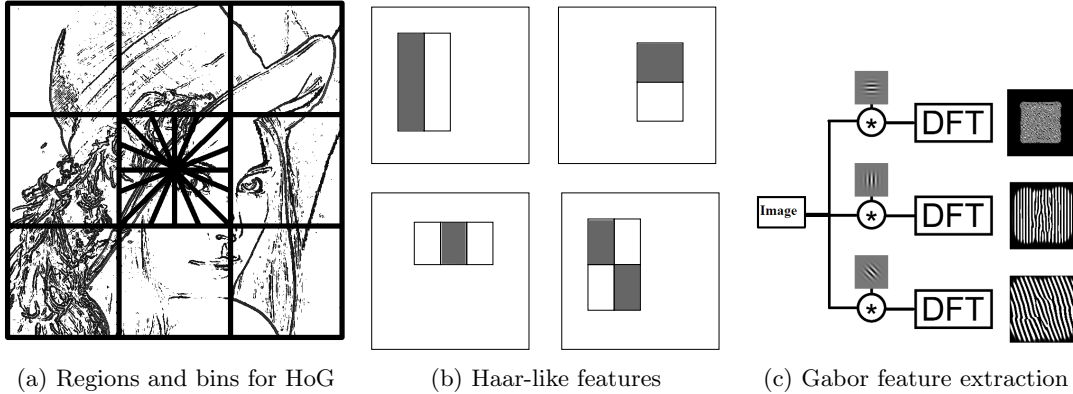


Figure 2: Feature Extraction Methods

neutral, sadness, and surprise. The image sequences start from a neutral expression, and return to a neutral expression. The database has around 3–5 image sequences of each subject, and each sequence consists of 50–160 images. Of these, we use a subset of images which correspond to the apex of the emotion, and discard the images which show neutral emotion.

We explored other options such as the CMU face images data set[10]. However, this dataset was limited in the training data, and the data was limited in the number of facial expressions. Other options had very restrictive and/or expensive licensing terms.

3.3 Feature Extraction Methods

This section presents various feature extraction methods which we have explored in our project. The feature extraction methods used in our project are *Moments*, *Histogram of Gradients*, *Haar-like Features* and *Gabor Features*. Each of these feature extractors are described in detail below.

Gabor Features We use a set of Gabor filters with different tuning parameters like *frequencies*, and *orientations* which allow different features to be extracted from the given image. A Gabor filter is defined as below:

$$g(x, y, \lambda, \Theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

The filter has the real and the imaginary component in orthogonal directions. The components can be used individually as well. The real component is defined below:

$$g_{real}(x, y, \lambda, \Theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

where $x' = x \cos \Theta + y \sin \Theta$, $y' = -x \sin \Theta + y \cos \Theta$, λ represents the wavelength of sinusoidal factor, Θ represents the orientation of the normal to the parallel stripes of Gabor function, ψ is the phase offset, σ is the sigma and γ is the aspect ratio for the function.

For our implementation, we have considered a filter bank of 40 different kernels which represent Gabor filters with different frequencies, orientations, and scale. The given image is convolved with all the filter kernels and the features are extracted using a linear transformation of the discrete Fourier transformation (DFT) of the filtered images. This process is shown in Figure 2c.

We use an open source implementation of Gabor feature extractors by Zhu et al.[17, 16].

Histogram of Gradients (HoG) Histogram of Gradients is a technique used for object detection. This method calculates the relative weighted frequencies of occurrence of gradient orientation in a particular section of the image. The implementation is done by dividing the image into smaller tiles and for each tile computing the histogram of gradients. We divide the image into 9 tiles and bin the gradients in each tile into 16 bins based on the orientation of gradients. This method is shown in Figure 2a. The feature vector is a concatenation of histogram of gradients in each tile.

Haar-like Features Haar-like features are digital image features, similar to Haar wavelets, which encode the existence of oriented contrasts between regions in the image. A set of these features can be used to encode the contrasts which are prominent in faces and spatial relationships of similar objects. Haar-like features were used for object detection by Viola and Jones[13]. A rectangular *Haar-like Feature* can be defined as the difference of the sum of the pixels which lie inside the rectangle which can be at any position and scale within the given image. We have shown some sample Haar-like features in Figure 2b.

Moments An image moment is a certain weighted average of the image pixels' intensities or a function of other such moments. All the moments up to the third order of the rasterized image, including spatial moments, central moments and normalized central moments are calculated. The OpenCV implementation of moments is used in our project. The feature vector is a set of all the three types of moments which are considered for a given image.

3.4 Learning Algorithms

We experiment with three learning algorithms, namely, SVM, Random Trees (aka Random ForestsTM, hereafter referred to as Random Trees (as named in OpenCV) to prevent violations), and ANN. All the above learning algorithms have been implemented in OpenCV.

Support Vector Machine (SVM) We use an SVM with two kernels, linear and radial. As we find out, linear SVM performed well, whereas radial performed poorly.

Random Trees (RT) For preliminary testing, we use OpenCV's default parameters in the RT implementation. However, we performed detailed analysis using cross-validation to evaluate the best maximum depth parameter for the Random Trees implementation.

Artificial Neural Network (ANN) We experimented with a 3 layer MLP back-propagation neural network, with the middle layer size chosen to be 128. We also tried implementing a neural network with the middle layer size set to the sum of the first and third layer sizes, but we quickly found out that large neural networks take up a lot of memory and time to train and store.

4 Experimental Evaluation

In this section, we evaluate the best feature extractor and learning algorithm combination. We first describe the methodology of our testing and present preliminary results for different extractor-learner combination. Then we cross validate for the best learning algorithm parameters and statistically compare all extractor-learner combinations. Lastly, we compare the learning algorithms against human recognition of emotions.

4.1 Methodology

In order to examine how well our method generalizes for images of subjects that are part of the training set, and for subjects beyond the training set, our analysis includes following two cases:

1. **RandomCV:** Randomly shuffle the feature data and perform 20-fold cross validation on the data, training on 95% of the data and validating on the remaining 5%. This method shows generalization of the classifier to subjects in the training data set.
2. **PerSubjectCV:** Split the feature data by subject. Perform 14-fold cross validation using 13 subjects for training and the remaining 1 subject for validation. This method serves as an indicator of prediction error for subjects not in the training set.

4.2 Preliminary Results

Figures 3a and 3b show the preliminary results across different extractor-learner combinations across 10,000 images of 14 subjects. Figure 3a shows the validation error rate on 10% data of a classifier that is trained on the remaining 90% data. In Figure 3b, the training error shows the error for a subject part of the training data, and testing error is for a subject not used for training. We train the SVM using `CvSVM::train_auto` function in OpenCV, which optimizes over all the SVM parameters including `C` values. For Random Trees, we perform a 10-fold cross validation and report the error

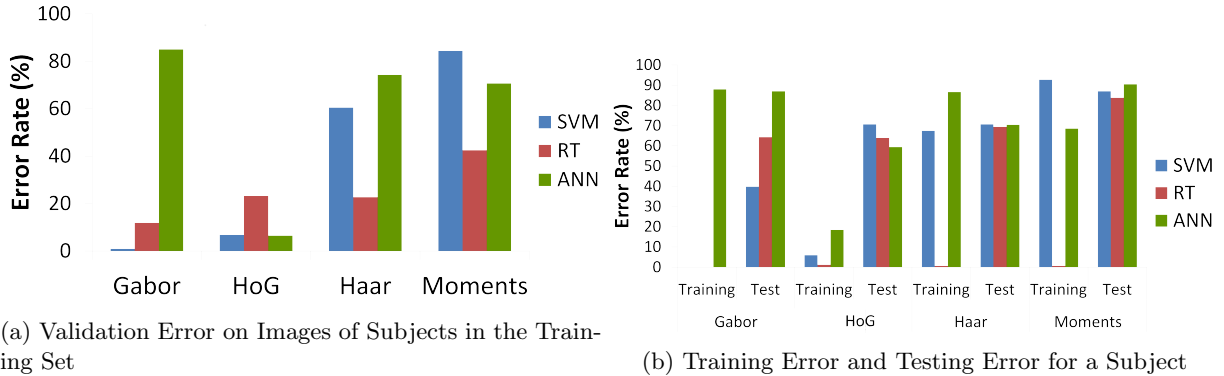


Figure 3: Preliminary Results

rates for the optimal `max_depth` parameter. For ANN, we used a 3-layer MLP backpropagation algorithm with the size of the middle layer set to the 128.

Comparing the error rates across various combinations, we observe Gabor in conjunction with SVM gives the least error rate with an accuracy of around 60% on an untrained subject. HoG+RT also gives good accuracy for random subjects. Moments and Haar, in general, show poor results. ANNs show consistently poor accuracy. We experimented with a different middle layer size of sum of number of inputs and outputs. This configuration of ANNs with moments (with least number of features) for 300 images required a training time of more than 1.5 hours and the error rate was still more than 90%. With this basis, we estimate ANN to take roughly 40 hours to train on 10,000 images using moments, while training time with a larger feature space like Haar features and Gabor features would be significantly higher. With such a large training time, we cannot perform cross validation and statistical analysis across many test sets. Therefore we do not evaluate ANN further just because of limited time and the fact that ANNs provide little theoretical insight into the problem.

4.3 Cross Validation for Best Parameters

Figure 4 shows the cross validation for different feature extractor and learning algorithm combinations. For SVM, we vary the `C` values, whereas for Random Trees, we change the `max_depth`.

Haar and Moments do not extract the necessary features required for emotion detection. Therefore they have little effect on `C` value of SVM. Hence, there is no change in the error rate as shown in Figures 4c and 4d. For other features with very small `C` value, the margin is extremely soft. Therefore the training as well as validation error is expected to be high. As `C` increases, the margin becomes hard and thus the error rate increases. For Random Trees, smaller `max_depth` leads to under-fitting whereas larger `max_depth` increases the validation error rate due to over-fitting. Therefore an optimal `max_depth` leads to least validation error. We report the optimum `C` and `max_depth` values for each feature extractor in Figure 4 and use these values for the statistical analysis of the relative performances of all extractor-learner combinations.

4.4 Performance Analysis

In this section, we use the optimal parameters from Figure 4 and run two cases - `RandomCV` and `PerSubjectCV`. For `RandomCV`, we do a 20-fold cross validation where the classifier is trained on 95% of the shuffled data and validate on the remaining 5% data set. For `PerSubjectCV`, we train the classifier on 13 subjects and validate on the remaining subject. We also evaluate the micro-averaged precision and recall and compute the F_1 -Score for each emotion. To compare it against human recognition of emotions, we conduct an experiment where 14 people guessed the true emotion of 50 randomly selected images from a data set of 355 images comprising 52 subjects, and validate their responses against the labeled emotion. In the following sections, we discuss the results and analyze the relative performances.

4.4.1 Extractor-Learner Performance

Table 1 shows the sample error rate for `RandomCV`. The sample error rates are small for all combinations except SVM+Haar and SVM+Moments. Table 2 shows F_1 -Score for each emotion. F_1 -Score is

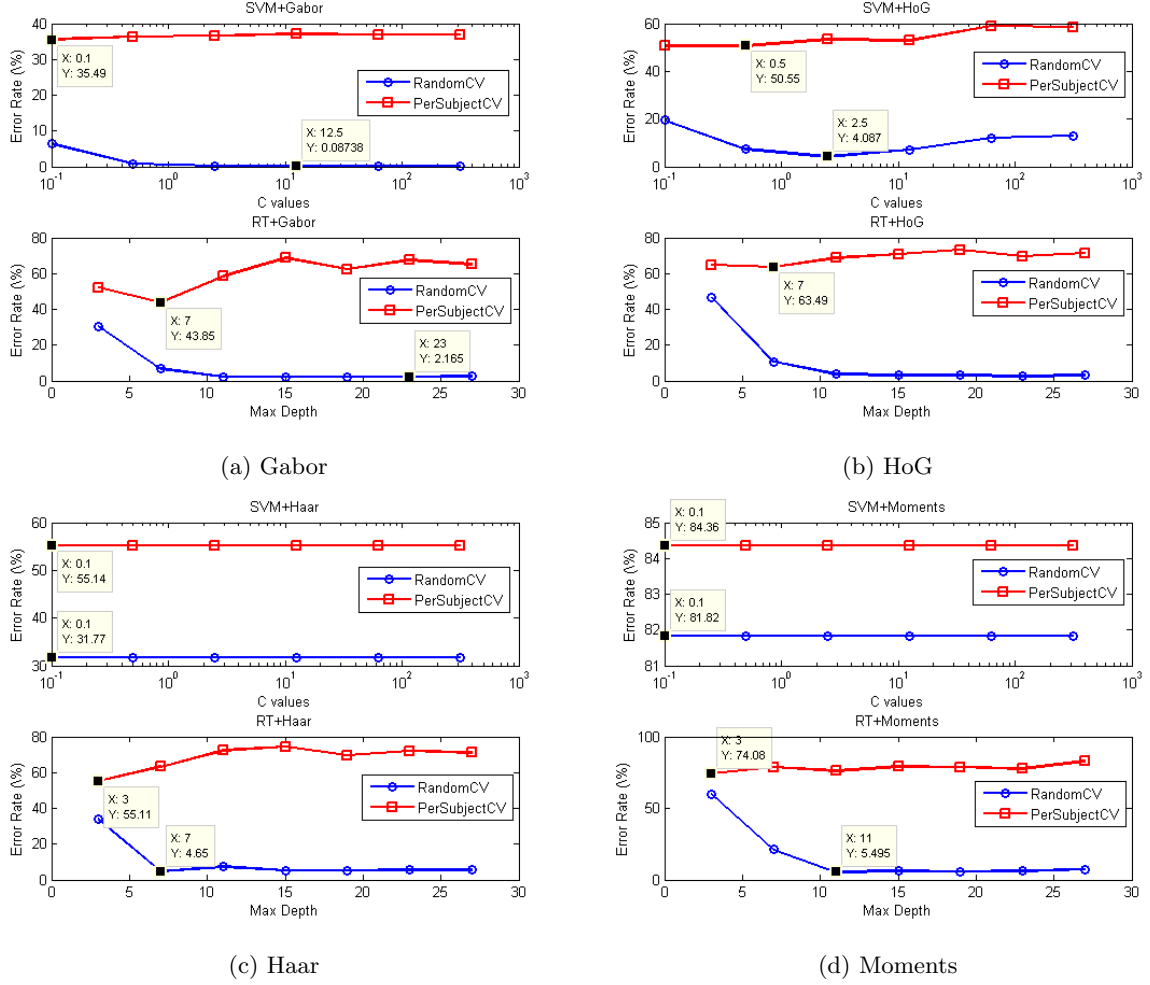


Figure 4: Cross Validation for Best Parameters

obtained by computing the harmonic mean of per-emotion micro-averaged precision and recall. The F_1 score lies in the range $[0, 1]$. F_1 scores close to 1 show that the algorithm shows good precision and recall. We see that all algorithms except SVM+Moments have good F_1 scores.

Table 3 shows the sample error rate for **PerSubjectCV** for all extractor-learner combination. We also show the sample error rate for human emotion recognition averaged over 14 humans who participated in the experiment. Gabor gives the highest accuracy of 63% and 54% with SVM and RT respectively. It is supported by the F_1 -score from Table 4. HoG is the second best, where as Haar and Moments do not show good results.

4.4.2 Statistical Analysis (Welch's t-Test)

In order to evaluate the relative performances of different extractor-learner combinations, and to compare their performance with human recognition, we used Welch's one-tailed t-test [14]. Welch's t-test is a generalization of Student's t-test for the case when several different population variances are involved, as is the case with different learning algorithms. As we have no reason to believe that different learning algorithms have the same variance for their error rates, we use Welch's t-test over Student's t-test.

Welch's t-test defines the test statistic 't' as

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{N_1} + \frac{\sigma_2^2}{N_2}}}$$

Table 1: Mean and Standard Deviation of Sample Error Rate (in %) for RandomCV

	SVM				RT			
	Gabor	HoG	Haar	Moments	Gabor	HoG	Haar	Moments
Mean	0.05	4.26	34.30	81.81	2.41	2.81	4.99	5.58
Standard Deviation	0.11	1.00	3.96	2.43	0.81	0.63	0.95	0.97

Table 2: Micro-averaged F_1 -Score for RandomCV

	SVM				RT			
	Gabor	HoG	Haar	Moments	Gabor	HoG	Haar	Moments
Anger	0.999	0.974	0.914	0.258	0.980	0.975	0.949	0.9376
Disgust	1.000	0.990	0.780	0.301	0.967	0.971	0.970	0.9583
Fear	1.000	0.953	0.539	0.155	0.971	0.967	0.914	0.9593
Happiness	1.000	0.996	0.896	0.296	0.978	0.979	0.984	0.964
Neutral	0.999	0.948	0.568	0.011	0.976	0.975	0.959	0.9778
Sadness	0.999	0.982	0.931	0.379	0.981	0.980	0.983	0.9758
Surprise	0.999	0.979	0.818	0.341	0.977	0.970	0.971	0.9573

Table 3: Mean and Standard Deviation of Sample Error Rate (in %) for PerSubjectCV

	Human	SVM				RT			
		Gabor	HoG	Haar	Moments	Gabor	HoG	Haar	Moments
Mean	26.43	37.07	52.12	55.65	83.76	46.05	64.51	57.97	77.38
Standard Deviation	7.77	14.89	19.66	13.61	7.53	12.11	17.74	13.56	10.88

Table 4: Micro-averaged F_1 -Score for PerSubjectCV

	Human	SVM				RT			
		Gabor	HoG	Haar	Moments	Gabor	HoG	Haar	Moments
Anger	0.5598	0.6644	0.6041	0.6814	0.1413	0.5324	0.3599	0.4240	0.2015
Disgust	0.7109	0.7816	0.5745	0.3949	0.1254	0.6830	0.4010	0.5013	0.0468
Fear	NaN	0.5582	0.4303	0.2172	0.0866	0.3868	0.3521	0.0000	0.0440
Happiness	0.9879	0.8964	0.7941	0.6783	0.3318	0.8168	0.6846	0.7179	0.4066
Neutral	0.8102	0.0042	0.0042	0.0039	0.0000	NaN	NaN	NaN	NaN
Sadness	0.8035	0.3715	0.3972	0.4439	0.1468	0.4016	0.4622	0.3656	0.2915
Surprise	0.7329	0.6017	0.4852	0.4633	0.0192	0.5743	0.4250	0.4850	0.3353

Table 5: P-values (Welch’s t-test). Values highlighted in cyan indicate that the algorithm on the row outperforms the algorithm on the column

		Human	SVM				RT			
			Gabor	HoG	Haar	Moments	Gabor	HoG	Haar	Moments
Human			0.014	0.000	0.000	0.000	0.000	0.000	0.000	
SVM	Gabor	0.986		0.016	0.001	0.000	0.046	0.000	0.000	
	HoG	1.000	0.984		0.293	0.000	0.832	0.046	0.185	
	Haar	1.000	0.999	0.707		0.000	0.970	0.076	0.328	
	Moments	1.000	1.000	1.000	1.000		1.000	0.999	1.000	
RT	Gabor	1.000	0.954	0.168	0.030	0.000		0.002	0.011	
	HoG	1.000	1.000	0.954	0.924	0.001	0.998		0.858	
	Haar	1.000	1.000	0.815	0.672	0.000	0.989	0.142		
	Moments	1.000	1.000	1.000	1.000	0.042	1.000	0.985	1.000	

and the degrees of freedom are approximated as

$$\nu = \frac{\left(\frac{\sigma_1^2}{N_1} + \frac{\sigma_2^2}{N_2}\right)^2}{\frac{\sigma_1^4}{N_1^2(N_1-1)} + \frac{\sigma_2^4}{N_2^2(N_2-1)}}$$

Table 5 shows the P-values¹ for each pair-wise extractor-learner combination. The null hypothesis assumes that the two algorithms are equal, while the alternative hypothesis proposes that the row algorithms have lower error mean than the column algorithms. The colored portions of the Table 5 indicates the pairs that have P-values smaller than $\alpha = 0.05$. For such pairs, we reject the null hypothesis and accept the alternative hypothesis. For the uncolored pairs, we cannot reject null hypothesis with 95% confidence.

From the results of Welch’s t-test, we conclude (with 95% confidence) that

- Humans perform better than every machine learning approach that we tried.
- Gabor features in conjunction with a linear SVM perform the best amongst all our machine classifiers.
- HoG is better only with respect to Moments. We cannot conclude the best among HoG and Haar with 95% confidence.
- Moments are universally bad for emotion detection.

4.4.3 True Error Rate

Using k -fold cross validation, where $k = 14$, we estimate the true error with $N = 95\%$ confidence using the formula below.

We use the following formulae to determine the true classification error.

$$\bar{Y} = \frac{1}{k} \sum_{i=1}^k Y_i \quad (1)$$

$$\mu = \bar{Y} \pm t_{N,k-1} s_{\bar{Y}} \quad (2)$$

$$s_{\bar{Y}} = \sqrt{\frac{1}{k(k-1)} \sum_{i=1}^k (Y_i - \bar{Y})^2} \quad (3)$$

where Y_i is the observed set of i.i.d. variables, \bar{Y} is the sample mean, the product $s_{\bar{Y}}$ is the estimated standard deviation, μ is the true classification error, and the constant $t_{N,k-1} = 2.16$ for $N = 95\%$ confidence level[11]. Table 6 shows the true error rate for all extractor-learner combinations with 95% confidence.

¹P-values are the smallest values that could be assigned to α and still reject null hypothesis

Table 6: True Error Rate (in %) with 95% confidence

	Human	SVM				RT			
		Gabor	HoG	Haar	Moments	Gabor	HoG	Haar	Moments
Mean	26.43	37.07	52.12	55.65	83.76	46.05	64.51	57.97	77.38
Estimated Standard Error Rate	± 4.66	± 8.92	± 11.78	± 8.15	± 4.51	± 7.26	± 10.63	± 8.12	± 6.52

5 Related Work

Relevant work has been done in the field of human emotion detection. Black et al.[6] and Cohn et al.[7] use optical flow tracking to track the movement of facial features in videos or image sequences to detect facial activity, and infer emotion using that information. Bartlett et al.[5, 4] systematically compares a number of methods (including SVM, Adaboost, Linear Discriminant Analysis, PCA for feature selection, and Adaboost for feature selection) to detect and classify human emotion, and conclude that a Gabor filter with Adaboost and using an SVM is the best approach. Hussain et al.[9] describe the way emotion detection is adapted to get the state of the users before offering personalized services to customers. They use sensor and auxiliary data as input to statistical algorithms to form emotion classifiers. Zhou et al.[15] use SVM and naïve Bayes classifiers to determine the implicit association between user emotions and music to enrich the user experience of music information retrieval. Most of the work described above uses machine learning techniques to solve problems in different domains associated with emotion detection. However, they do not experiment with as comprehensive a bouquet of feature extractors as our method.

6 Limitations And Future Work

In this project, we have presented a variety of feature extraction and machine learning algorithms to identify human emotion from static facial images. We have evaluated the performance of these algorithms and conclude that Gabor features in conjunction with an SVM give the best results. We also evaluate the accuracy of our approach against the accuracy with which humans classify the same images, and find that humans perform better, but only by around 11%.

This project is limited to only 14 subjects, and could be potentially extended to 52 subjects already part of the data set. Training on a larger number of subjects may increase the accuracy of our machine classifier. Comparing our machine classifier to humans is not exactly an apples-to-apples comparison, as humans have had years of experience identifying emotion over possibly thousands of people. This project is intended primarily as a proof-of-concept, and our results seem promising.

At the same time, we have not tried methods to identify strong features to detect emotion from facial expressions, like boosting or PCA. We have also not experimented extensively with ANNs mainly because the use of ANNs with large feature spaces will require large networks, which require memory and time to train. Considering that all the work on this project was done on regular laptop computers, time and memory became a bottleneck to train a large ANN. Another consideration was that the ANN will take up a lot of memory to run and test new images, and this approach will not be feasible for the applications listed in Section 1, many of which will require a real time implementation as we showed in our project demonstration.

Further, our method to evaluate the accuracy of human subjects is limited, and is meant to serve just as an indicator. We tested humans by presenting 50 random images from a set of 355 images, so that images from the same image sequence in our data set were never shown to the same subject. We restricted the experiment to 50 images because it is not feasible to ask a human to evaluate a larger number of images, and because it does not make sense to ask a human to detect emotion from images belonging to the same sequence. While not a serious drawback, the limited set of images could mean that the central limit theorem may not apply, and hence the assumption of normal distributions made by Welch’s t-test may not hold.

Future work could address these issues, training machine classifiers on a larger data set, and conducting more accurate experiments to evaluate the performance of machine classification vs. humans. We can also experiment with other approaches using image sequences, or 3-D models of facial features.

7 Conclusions

From our experiments, we conclude (with 95% confidence) that Gabor features, in conjunction with an SVM show the best results for emotion detection using facial expressions in static images. We also conclude (with 95% confidence) that humans perform better than our machine classifiers, only by 11%. While our approach can be improved by training on a larger data set, it serves as a good proof-of-concept that machines can detect human emotion. Other techniques may involve further sophistication by using image sequences, and/or reconstructing 3-D models of facial features.

References

- [1] Opencv. <http://opencv.org>.
- [2] N. Aifanti, C. Papachristou, and A. Delopoulos. The MUG facial expression database. In *11th International Workshop on Image Analysis for Multimedia Interactive Services*, pages 1–4, April 2010.
- [3] R. Animesh, S. Hurkat, A. Majumdar, and A. Saxena. Emodetect. <http://github.com/skandhurkat/EmoDetect>.
- [4] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 568–573, June 2005.
- [5] M. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, and J. Movellan. Machine learning methods for fully automatic recognition of facial expressions and facial actions. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 592–597, October 2004.
- [6] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25:23–48, 1997.
- [7] J. F. Cohn, J. J. Lien, and A. J. Zlochower. Feature-point tracking by optical flow discriminates subtle differences in facial expression. In *In Proceedings International Conference on Automatic Face and Gesture Recognition*, pages 396–401, 1998.
- [8] D. Graham-Rowe. A smart phone that knows you’re angry. <http://m.technologyreview.com/computing/39434>.
- [9] S. S. Hussain, C. Peter, and G. Bieber. Emotion recognition on the go: Providing personalized services based on emotional states. In *MobileHCI*, volume 1, 2009.
- [10] T. Mitchell. Cmu face images data set. <http://archive.ics.uci.edu/ml/datasets/CMU+Face+Images>, 1997.
- [11] T. Mitchell. *Machine Learning*, chapter 5, page 148. McGraw-Hill Companies Inc., 1997.
- [12] nViso. 3D facial imagine technology. <http://www.nviso.ch>.
- [13] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.
- [14] B. L. Welch. The Generalization of ‘Student’s’ Problem when Several Different Population Variances are Involved. *Biometrika*, 34(1/2):28–35, 1947.
- [15] L. Zhou, H. Lin, and W. Liu. Enriching music information retrieval using emotion detection. In *SIGIR 2011 Workshop on Enriching Information Retrieval*, volume 1, July 2011.
- [16] J. Zhu. gaborextract: Gabor wavelets feature extraction. <http://code.google.com/p/gaborextract>.
- [17] J. Zhu, M. Vai, and P. Mak. A new enhanced nearest feature space (enfs) classifier for gabor wavelets features-based face recognition. In D. Zhang and A. Jain, editors, *Biometric Authentication*, volume 3072 of *Lecture Notes in Computer Science*, pages 124–130. Springer Berlin Heidelberg, 2004.