From Emotion Recognition to Website Customizations

O.B. Efremides
School of Web Media
Bahrain Polytechnic
Isa Town, Kingdom of Bahrain

Abstract—**A computer vision system that recognizes the emotions of a website's user and customizes the context and the presentation of this website accordingly is presented herein. A logistic regression classifiers is trained over the Extended Cohn-Kanade dataset in order to recognize the emotions. The Scale-Invariant Feature Transform algorithm over two different part of an image, the face and the eyes without any special pixel intensities preprocessing, is used to describe each emotion. The testing phase shows a significant improvement in the classification results. A toy web site, as a proof of concept, is also developed.**

***Keywords*—Emotion recognition; classification; computer vision; web interfaces**

I. Introduction

The development of emotion aware systems is the next step in creating effective, trustworthy and persuasive web applications and websites. Websites capable of *sensing* and reacting to user's emotional state by adjusting their context and their *look & feel (L&F)* can be used from e-companies as powerful recommendation and advertisement tools or by website designers as a medium to increase user satisfaction.

Automatic detection of human emotions from digital images and videos is an active research area attracting a lot of attention in recent years. Interdisciplinary in nature it combines image processing, computer vision and machine learning and can be applied in large number of application especially in the area of human-computer-interaction (HCI).

Recognizing emotions with a high accuracy is a difficult task though. The emotions being communicated by human facial expressions are complex and vary constantly. Variations related to camera/face pose, occlusions of main facial components (e.g., eyes and nose), features as glasses and beards along with illumination conditions and camera technical characteristics make the problem even harder.

In this paper a system that automatically recognizes user emotions and customizes the context and the L&F of a website is presented. The representation of emotions by Scale-Invariant Feature Transform (SIFT) [4] descriptor is investigated. The descriptor applied on a dense grid of keypoints on two images; the face and the eyes of the user as they captured by a web camera. A logistic regression model is used to label the emotion and the context of the website along with its presentation are customized accordingly.

This works is organized as follows. Section II briefly presents the related work. The overview of the system and the description of its parts are given in section III. Settings, experiments and their acquired results are presented in section IV. Finally conclusions are drawn and further work is given in section V.

II. Related Work

Automatic emotion recognition consists of two key factors: the emotion representation and the development of a classifier. A set of features which effectively represents the emotion must be derived and from these features a model must be learned.

Pantic et al [7] proposed a ruled-based classifier in order to recognize facial actions based on contours changes. A multistage model is used to extract and encode features. Initially face, face profile and facial components detectors are used to locate contours. Then a number of fiducial points are extracted and selected defining a mid-level feature parameters used for the final encoding of the actions. They reported 86% accuracy.

Recognition accuracy 81.4% on Cohn-Kanade (CK) [5] dataset reported by Buciu et al. [1]. They used Principal Component Analysis (PCA) as baseline and proposed a non-negative matrix factorization and a local non-negative matrix factorization technique for recognizing six facial expressions.

Shan et al. [8] worked with Local Binary Patterns (LBP). To extract the features and recognize the emotions they proposed a Support Vector Machine (SVM - RBF kernel) classifier with Boosted-LBP features. They used a fixed distance between the two eyes to normalized the faces and they manually labeled the eyes location, to evaluate LBP features in conditions without face registration errors. They reported a 91.4% accuracy.

Combining PCA with an SVM based classifier Vretos et al. [10] achieved 90% accuracy on the Cohn-Kanade dataset. They worked using analysis of vertices on Candide model (a parameterized face mask developed for model-based coding of human faces).

Kalita et al. [3] used an eigenvector based method. Their images are cropped to produce five different eigen-spaces and Euclidean distance is used for classification. They achieved 95% recognition rate.

Donia et al. [2] used histogram of oriented gradients (HOG) to extract features and trained an SVM classifier. The face is cropped for five regions to be created and HOG is calculated

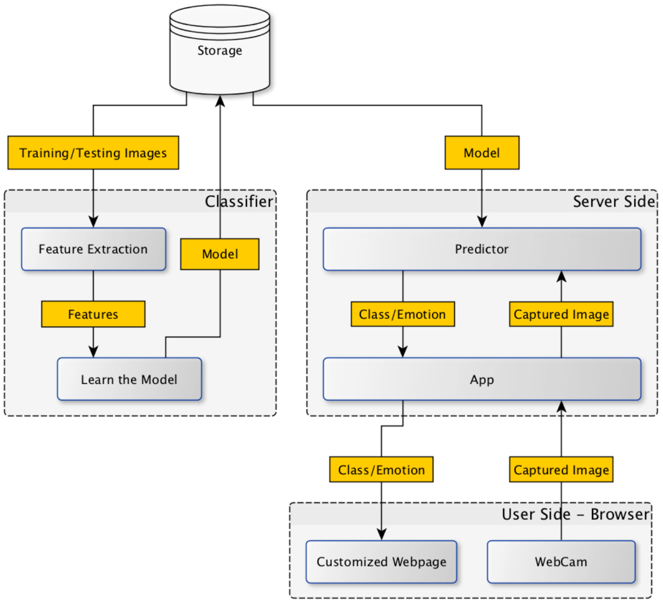

Fig. 1: System Overview

for each region. Using a linear kernel for their SVM they achieved 95% recognition accuracy on static images.

Scale-Invariant Feature Transform (SIFT) [4] has been extensively used for face but not for emotion recognition. In a recent study Neeru [6] reported 89.67% accuracy on the JAFEE dataset using the algorithm and also proposed an modified version of SIFT achieving 97.65% on the same dataset.

III. The System

The system is divided into two different modules as shown in Figure 1. The first one deals with the training of a classifier and results to a model. This model is used from the web part of the system (second module) for predictions. Based on these prediction the website is customized.

A. *Features Extraction and Classification*

Initially the original images of the dataset are loaded and for each one of them the face is detected using Viola-Jones [9] algorithm. The detected face is cropped, resized and saved to the disk (this is not an essential part of the process but it can significantly improve the speed later (e.g., during cross-validation phase); the entire process can be done on-the-fly (as it happens to the web part of the system)). A second Haar cascade detector [9] is used then to detect the eyes on this new image of the face. The detected eyes are also cropped and resized. Since the entire area of the face and the eyes must be clearly described a dense grid of keypoints is applied to both of them.

Every keypoint on the two grids is described by a SIFT descriptor. SIFT features are characterized by a high distinctiveness power and they are invariant to minor affine distortions, noise and illumination changes. In this work we will use only the description (and not the detection) part of the algorithm. For each keypoint a set of orientation histograms is created ($4x4$ pixel neighborhoods, 8 bins each). These histograms are computed by sampling the gradient magnitude and orientation values around the keypoint ($16x16$ region). A weight is assigned to the magnitude of each sample drawn by a weighted Gaussian function with σ equal to one half the width of the descriptor window. The values of the histograms forms the vector of the descriptors which is normalized to unit length, thresholded (less than 0.2 threshold value is given on the original paper) and normalized again. The method produces a feature vector with 128 elements for each keypoint.

The resulted feature vectors for the face and the eyes are concatenated to form the final feature vector and the process repeats for the next image. The results is a NxD-dimensional array (where N is number of examples and D is the number of features) (see Table II) which is used for the training of the classifier.

A logistic regression classifier (softmax) is trained in this work. In this model, the probabilities of the possible outcome label for a single example are modeled using a logistic function (Eq. 2). The implementation used herein can fit a multi-class logistic regression with L2 regularization (Eq. 1) by following the *one-vs-rest* technique. That mean that a single binary classifier is trained per class.

$$L = -\sum_{i=}^{n} log\, g(y_i z_i) + \frac{C}{2} \sum_{k}^{l} w_k^2 \qquad (1)$$

where g is the logistic fiction:

$$g(z) = \frac{1}{1 + e^{-z}} \qquad (2)$$

and $z_i = \sum_k w_k x_{ik}$, with $w_k, k \in \{0, \ldots, l\}$ the weight for the k^{th} feature and l the number of the features, w_0 the bias weight and C balances the tradeoff between the two terms.

The learnt model is store to the disk in order to be available to the web part of the system.

The proposed approach requires no preprocessing corresponding to pixel intensities. The images are used as they captured by the camera (just cropped and slightly resized). Both the eyes together are detected and handled as a separated single image. Previous approaches (see section II) depended on preprocessing or on facial landmarks in order to work with different parts of the face (e.g, the eyes). In some cases these landmarks should be manual registered. This preprocessing time can negatively affect the total elapsed time of the system when it is finally deployed. It should be noted that even though this is not a critical-time system it remains a real time system.

B. *Website*

Concerning the client-end of the system the functionality is simple enough. At predefined intervals (and after user's permission) a web camera captures frames from the users while they are visiting the website. Each captured image is sent to the server for processing. As soon as the result (class number - emotion) comes back from the server the interface is changed. On the server-end the learnt model is loaded from the disk and the web application is ready to process the images.

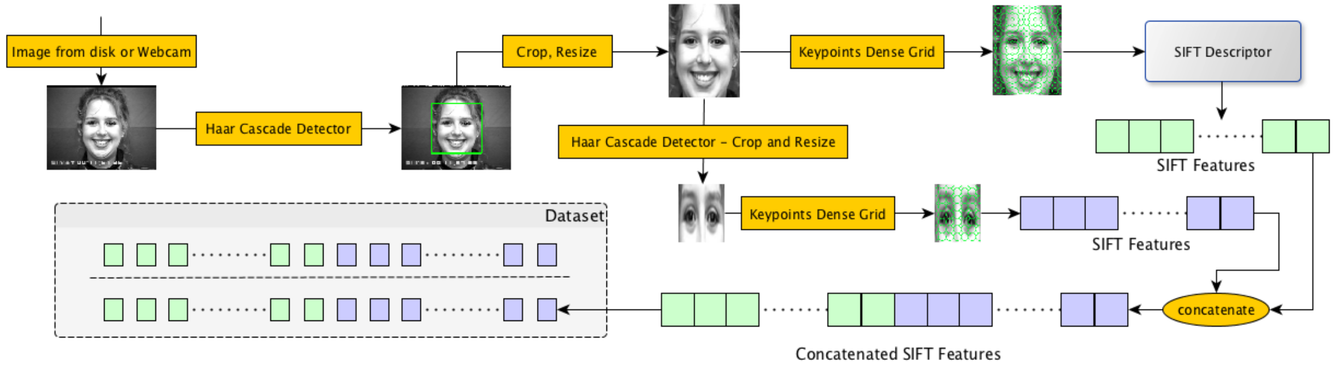

Fig. 2: Feature Extraction from a Single Image

When an incoming image arrives a process similar to the one used for the training of the classifier is applied. The face and the eyes are detected resized and cropped, a dense grid of keypoints is applied, the SIFT descriptors are produced and concatenated (see Figure 2), and the prediction is made. The resulted emotion is sent back to the user-end of the application.

A toy website is built as a proof of concept. This site is only capable of recognizing 3 different emotions (neutral, happy, sadness). These three have been chosen as their facial expressions can last more and they are highly likely to be presented while a user is surfing on the internet. Of course there might be moments that the user is surprised or feels disgust but these emotions change the facial characteristics for a very small period of time which is not justify the change of the environment. A model is also trained to recognize only these three emotions.

IV. Experiments and Results

The system developed using python as programming language and the experiments were conducted on a 2.3 GHz Intel Core i5 mini-Mac system with 8GB main memory.

The Extended Cohn-Kanade (CK+) dataset [5] is used in this work. Currently, the set is one of the most commonly used datasets for facial emotion recognition. Facial behavior of 210 adults from 18 to 50 years of age belonging to different gender and racial groups is shown. 23 facial displays (began and ended in a neutral face) are performed by each participant, Image sequences are digitized into 640x490 pixel arrays with 8- bit gray-scale or 24-bit color values. The emotions included in this dataset are *neutral*, *anger*, *contempt*, *disgust*, *fear*, *happiness*, *sadness* and *surprise*.

A. Classification

Four different models of the haar cascade detectors for the face and two for eyes where checked. A class creating the dense grid is developed and the SIFT descriptor is used to provide the feature vector for each image. The number of the feature produced are 33792 per image.

A number of linear, non-linear classifiers and ensembles are initially checked to find those who might perform well on the data under investigation. For this phase 80% of the data were used as training set and the rest 20% for testing. Since the dataset is small a 10-fold cross-validation resampling process was applied for the hyper-parameters tuning.

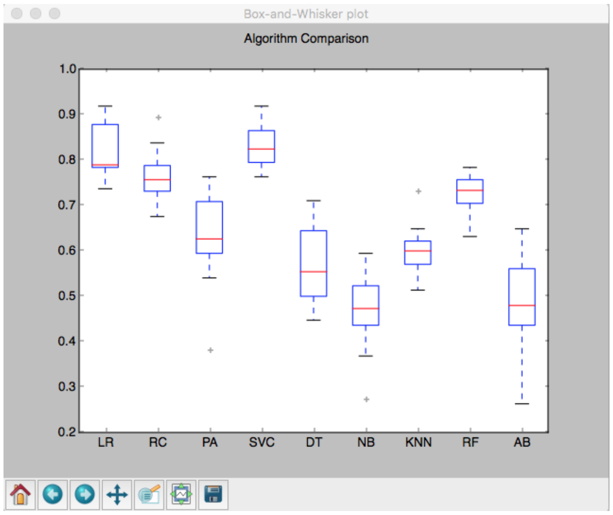

Fig. 3: Comparing the Classifiers (LR: Logistic Regression, PA: Passive Aggressive, SVM: Support Vector Machine, DT: Decision Tree, NB: Gaussian Naive Bayes, KNN: k-Nearest Neighbors, RF: Random Forest, AB: Adaboost)

Numerical results (Figure 3, Table I) suggested that logistic regression and linear SVM classifiers were promising for good results and further investigation. Their performance was more or less the same as expected. To improve the results further an exhausted grid search with cross-validation approach were taken. For almost all trials the logistic regression performed slightly better compared to SVM with linear kernel for this specific dataset. To boost even more the performance two ensembles were tested. A voting ensemble combining both classifiers did not improve the results. Then a boosting ensemble for the logistic regression was built and tested but again without any improvement of the results. The logistic regression was finally selected as the appropriate classifier for the dataset.

TABLE I: Initial Result - Comparing the Classifiers

	SIFT Features	
	Mean (%)	**St. Dev (%)**
Logistic Regression	81.81	5.98
Passive Aggressive	62.93	11.02
SVM (linear kernel)	83.12	4.56
Dessision Trees	57.08	8.76
Gaussian Naive Bayes	46.66	9.08
k-Nearest Neighbors	60.34	5.59
Random Trees	72.68	4.57
Adaboost	48.32	11.29

TABLE II: Classification Results

	Full Emotions Set	**Reduced Emotions Set**
	Parameters	
Total Images	467	237
Features per Example (D)	33792	33792
Training Examples (N)	373	189
Testing Examples (N)	94	48
Validation k-fold	10	10
	Classification Report (average)	
Precision	0.90	0.94
Recall	0.89	0.94
F1-score	0.89	0.94
Support	94 (total)	48 (total)
	Accuracy	
Training/Validation	87.10 (+/- 4.35)	93.65 (+/- 3.93)
Testing	89.36	93.75

After tuning the parameters, the classifier was trained and evaluated using the test set. The numerical results are presented in Table II and the corresponding confusion matrices are depicted in Figures 4, 5. As it is mentioned before, two different models are trained. The first is trained to recognize the full range of emotions in our dataset while the second a reduced set of emotions (in order to be used as an example in the toy website). Thus, two different set of results are presented. As it is shown the method improves the initial accuracy results from 81.81% too 87.10% while reduces the standard deviation from 5.98% to 4.35%. The testing time on the experimentaion system is $0.119sec$ for 94 images.

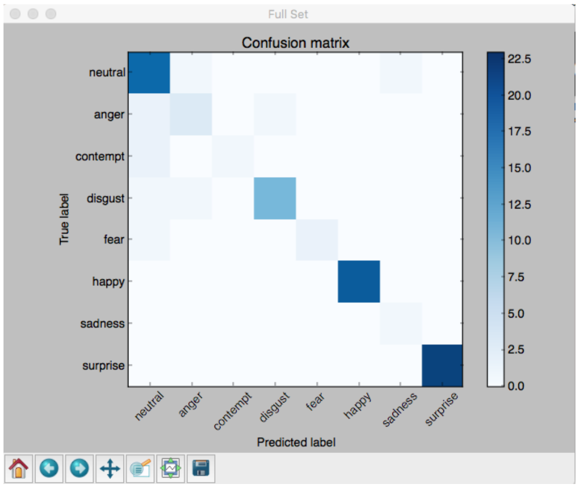

Fig. 4: Confusion Matrix - Full Emotions Set - Testing Phase

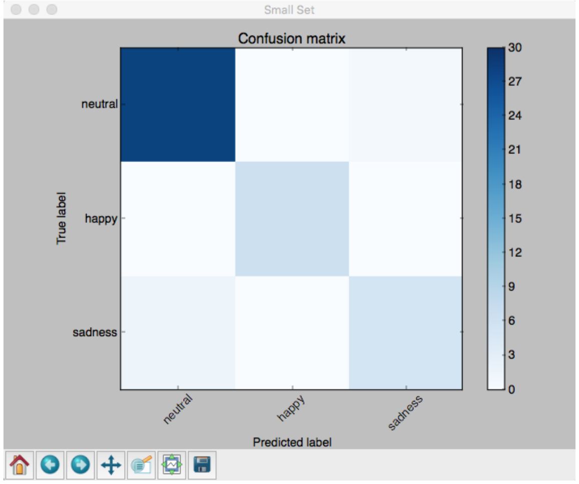

Fig. 5: Confusion Matrix - Reduced Emotions Set - Testing Phase

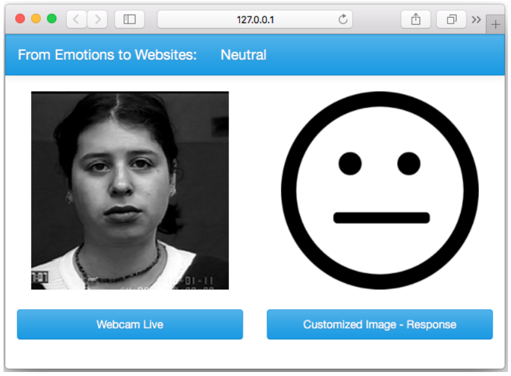
127.0.0.1
From Emotions to Websites: Neutral
Webcam Live
Customized Image - Response

Fig. 6: Toy Website Customized according to User Emotion - Neutral

B. Website

The client-side of the website is developed using HTML for the context part of the page and the Bootstrap framework for CSS styling. The most important component is the WebcamJS javascript library (and open source MIT licensed library) which provides all the necessary functions for an image to be captured and sent to the server. It is an AJAX based communication and a callback function accepts the server response. The JQuery library is used for accessing and altering the Document Object Model (DOM) of the page. Customizations are i) the bootstrap theme is changed and b) a different images are shown to the user (each wiht the recognized emotion). The server-side part is developed using the Flask framework (BSD licensed) and python as programming language. Figures 6, 7 and 8 show three different snapshots of the site. It presents the image captured and the adjustment made to the context and to the look & feel of the site.

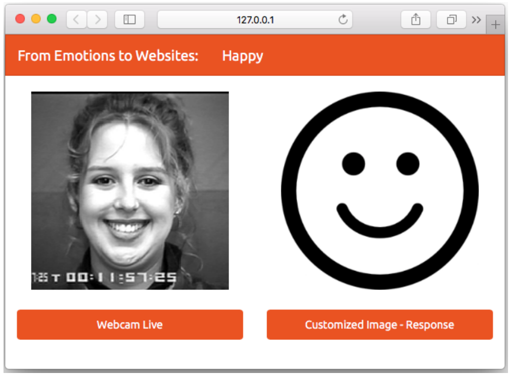

Fig. 7: Toy Website Customized according to User Emotion - Happy

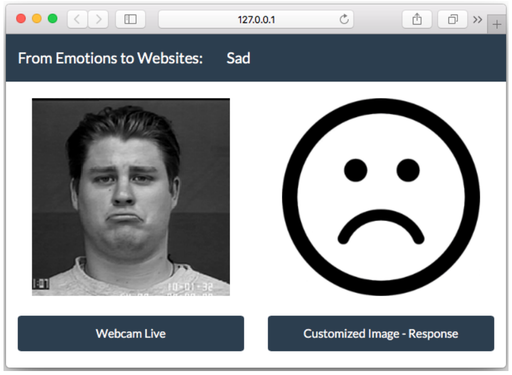

Fig. 8: Toy Website Customized according to User Emotion - Sad

V. Conclusions

In this work a system capable of customizing a web site according to its user emotions is presented. Initially, the SIFT algorithm was used to extract descriptors from a dense grid applied on two different images: the face and the eyes. Using these features a logistic regression (softmax) classifier was trained to recognize emotions. No preprocessing to pixels intensities is applied. The images are used as they are captured by the camera in term of illumination, subject pose, etc. After that, the learnt model is loaded on a website application which accepts images of a website's user through a web camera. Three different emotions are used as an example. The toy site reacts to the emotion recognized by changing its L&F as well as the images presented to the user.

The classifier is the key factor in this application. For the Cohn-Kanade Extented (CK+) dataset which used herein a logistic regression classifier has been chosen and tuned. Cross-validation shows an estimated training/validation accuracy of 87.10% and a testing accuracy of 89.36%. The recognition accuracy goes up to 93% concerning the three emotions the website reacts to.

Further work can be done to improve the classification results for the entire range of emotions included in this dataset. Techniques as Convolutional Neural Networks (CNNs) for this and other bigger datasets can used for models to be trained and tested. Concerning the website, proper usability tests and experiments must be contacted in order determine parameters like the time intervals between the changes of the website interface without frustrating the user or the set of the emotions that must be recognized along with the system's reactions them.

References

[1] I. Buciu and I. Pitas. Application of Non-Negative and Local Non Negative Matrix Factorization to Facial Expression Recognition. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, volume 1, pages 288–291, 2004.

[2] M.M.F. Donia, A.A.A. Youssif, and A. Hashad. Spontaneous Facial Expression Recognition Based on Histogram of Oriented Gradients Descriptor. *Computer and Information Science*, 7(3):31–37, 2014.

[3] J. Kalita and K. Das. Recognition of Facial Expression Using Eigenvector Based Distributed Features and Euclidean Distance Based Decision Making Technique. 4(2):196–202, 2013.

[4] D. G Lowe. Distinctive image features from scale invariant keypoints. *Int'l Journal of Computer Vision*, 60(2):91–110, 2004.

[5] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, (July):94–101, 2010.

[6] N. Neeru. Modified SIFT Descriptors for Face Recognition under Different Emotions. 2016:1–19, 2016.

[7] M. Pantic and L.J.M. Rothkrantz. Facial Action Recognition for Facial Expression Analysis From Static Face Images. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 34(3):1449–1461, 2004.

[8] Caifeng Shan, Shaogang Gong, and Peter W. McOwan. Facial expression recognition based on Local Binary Patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.

[9] P Viola and M Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition (CVPR)*, 1:511–518, 2001.

[10] N. Vretos, N. Nikolaidis, and I. Pitas. A model-based facial expression recognition algorithm using Principal Components Analysis. *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3301–3304, 2009.