CHONNAM NATIONAL UNIVERSITY

School of Electronics & Computer Engineering

Artificial Intelligence Theory

2019-06 Term Project Report

Facial Emotion Recognition in the wild

Using HOG, LBP Features
with Supper Vector Machine Classification

Student:
Tran Nguyen Quynh Tram 198507

Professor:
이 칠 우

June 10, 2019

1 Introduction

In recent year, Artificial Intelligence (AI), more specifically, machine learning is emerging as a proof of the fourth industrial revolution. Machine learning is a sub-field of AI, in other words, machine learning is a small branch of computer science, it has self-study ability from input data without specific programming. Recently, computing power of computers has achieved to new level and hug data has been collected by big technology firm. Machine learning has gone a step further, and a new sector appeared which was called deep learning. Deep learning helped computer to perform things that seemed impossible several years ago: classify many objects in thousands of images, recognize and mimic humans voice, communicate with humans, etc. Human-computer interaction is one of highlight sectors which supplies for human serving abilities of computer, in which facial emotion recognition plays an important role with wider applications in health-care, customer marketing, and emotionally intelligent robotic interfaces.

Facial expression is also one of ways that people transmit information for interpersonal interaction. In 1997, P. Ekman et al. suggested six basic emotions recognized through facial expression [1]. Their significant research changed most of complex emotion classification studies into one of seven categories (anger, disgust, fear, happiness, sadness, surprise and neutral). In facial expression recognition applications, computer automatically determines or recognizes emotion of somebody from digital images or a frame video from video to. The problem requires to detect seven facial expressions with universal meaning: anger, disgust, fear, happiness, sadness, surprise and neutral . There has been important research interest in computer vision field to recognize face emotion recognition. But it is also a challenging problem due to complex and dynamic properties in emotion. In past years, problems of object detection were solved by simple algorithms having fast computing speech, but accuracy was not as good as using deep learning. Although, studying traditional methods will be basis to understand better the idea of more complex algorithms.

In this final-term project, we exploit Facial Emotion Recognition in the wild based on RAF-DB dataset [2]. The dataset contains the image receiving Internet with manual ground-truth for emotion and facial alignment. We extract HOG [3], LPB [4] features on the aligned faces. After that, we apply supper vector machine (SVM) [5] on features to classify seven basic emotions.

2 Related Research and Key Technology

Several recent studies have proposed modern technologies, but the purpose of the report is to study the basic recognition methods on RAF-DB dataset for basic emotion recognition.

2.1 RAF-DB dataset

Real-world Affective Faces Database (RAF-DB) [2] as Fig.1 is a hug database containing around thirty thousand of various facial expression images labeled via crowdsourced annotation. RAF-DB is a diverse image set of age, gender, ethnicity, head posture, light, enclosed area, activity post-processing etc. RAF-DB has a large number of images in real world. Each image is represented by a 7-dimensional vector, and noted 5 exactly landmark positions, 37 automatic landmark positions, bounding box, race, gender, age range . It has two subsets including single-label subset (7 classes of basic emotion) and two-tab subset (12 classes of compound emotion). The database has been divided into 80 percent of a training set and 20 percent of testing set.

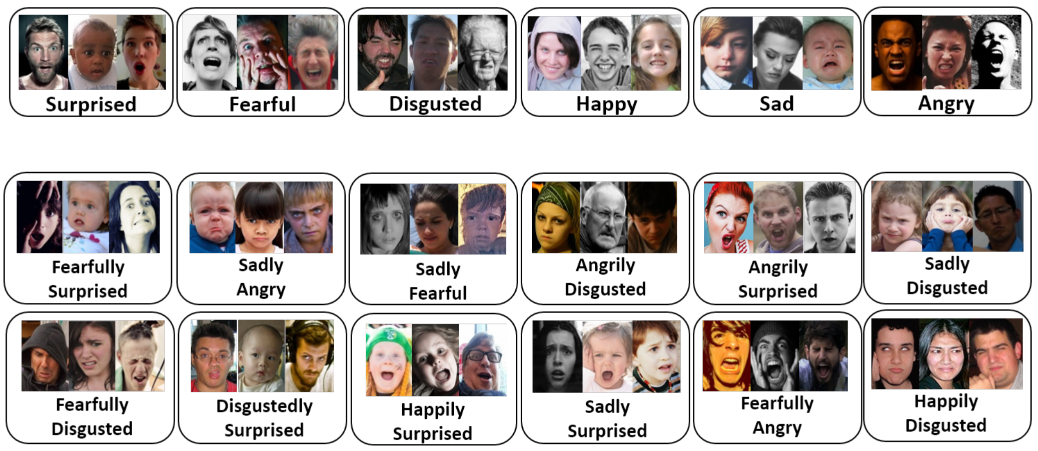


Figure 1: Sample images in RAF-DB [2]

2.2 Histogram of oriented gradients (HOG)

Feature extraction is a process of transforming complex input data into a more simple ways to represent data which is more suitable for machine learning. In the extracting process, the data is eliminated redundancy, and kept useful data for the problem.

HOG [3] is a feature descriptor which simplifies image by extracting the object features and throwing away non-useful information. HOG uses a sliding window detector over an image to compute HOG descriptor for each position. Typically, HOG descriptor converts an image with size 64 x 128 x 3(width x height x channels) and output feature vector is of length 3780. Thus, HOG is mainly used to describe shape and appearance of an object in image as Fig.2.

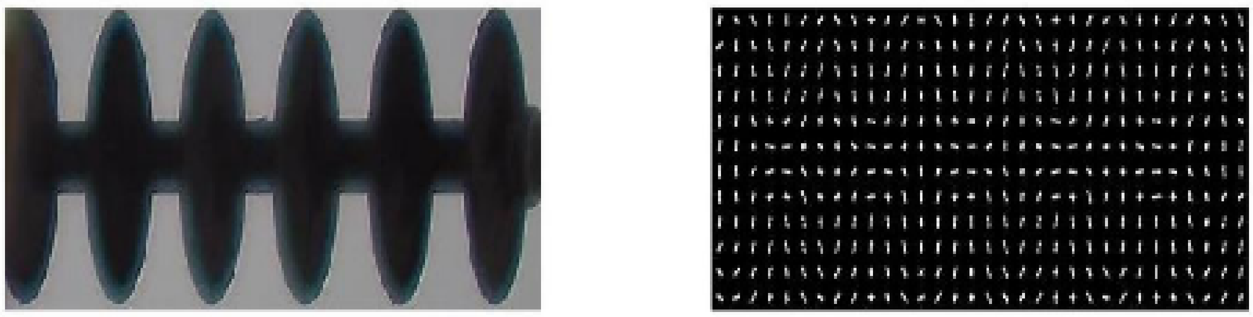


Figure 2: original image and extracted HOG feature visualization

Essence of HOG method is using population information of gradient magnitudes or edge directions to trace local objects in image. HOG operators is set by splitting an image into sub-regions called as cells. Points inside every cell is be computed a histogram of gradient or edge orientations, and the original image is presented by combining all histograms. To increase recognizing efficiency, local histograms may be standardized contrast by calculating an intensity threshold in areas larger than cells which are called blocks. Thereafter, all cells of blocks are be normalize by using the threshold. The result after normalization is a feature vector having higher invariant to resistant small changes in other lighting conditions.Generally, There are five stage to create HOG descriptors: (1) normalizing the image prior to description, (2)computing gradients in both the x and y directions (3) obtaining weighted votes in spatial and orientation cells (4) contrast normalizing overlapping spatial cells (5) collecting all histograms of oriented gradients to form the final feature vector. Fig.3 shows the process to calculate HOG features in the specific image.

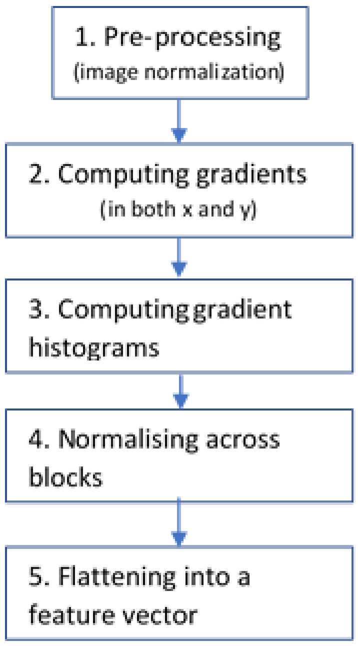

1. Pre-processing
(image normalization)
2. Computing gradients
(in both x and y)
3. Computing gradient histograms
4. Normalising across blocks
5. Flattening into a feature vector

Figure 3: The basic steps build a HOG descriptor

2.3 Local Binary Pattern (LBP)

Local Binary Patterns, or LBPs for short, are a texture descriptor made popular by the work of Ojala et al [4]. LBPs compute a local representation of texture. This local representation is constructed by comparing each pixel with its surrounding neighborhood of pixels.

The first step in constructing the LBP texture descriptor is to convert the image to grayscale. For each pixel in the grayscale image, we select a neighborhood of size r surrounding the center pixel. A LBP value is then calculated for this center pixel and stored in the output 2D array with the same width and height as the input image as in Fig.4.

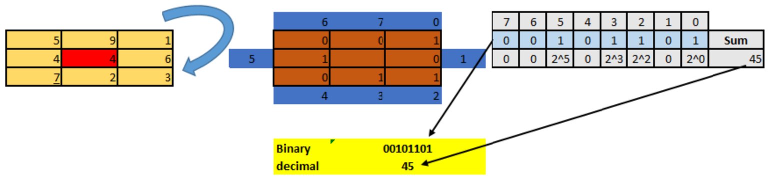


Figure 4: The example of computing LBP

This process of threshold, accumulating binary strings, and storing the output decimal value in the LBP array is then repeated for each pixel in the input image. The last step is to compute a histogram over the output LBP array. Since a 3 x 3 neighborhood has $2^8 = 256$ possible patterns, LBP 2D array thus has a minimum value of 0 and a maximum value of 255, allowing to construct a 256-bin histogram of LBP codes as final feature vector. The result in extraction LBP features is shown in Fig.5.

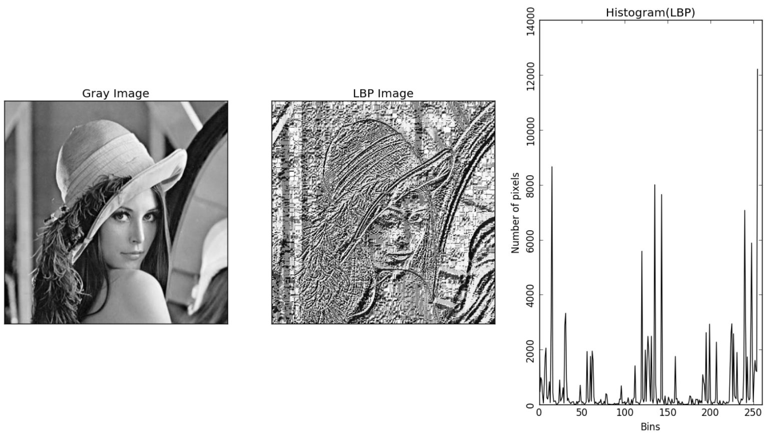


Figure 5: The example of computing LBP

A primary benefit of this original LBP implementation is that we can capture extremely fine-grained details in the image. However, being able to capture details at such a small scale is also the biggest drawback to the algorithm. To handle this, an extension to the original LBP implementation was proposed by Ojala et al. to handle variable neighborhood sizes. To account for variable neighborhood sizes, two parameters were introduced as in Fig.6: (1) The number of points p in a circularly symmetric neighborhood to consider (thus removing relying on a square neighborhood) (2) The radius of the circle r, which allows us to account for different scales.

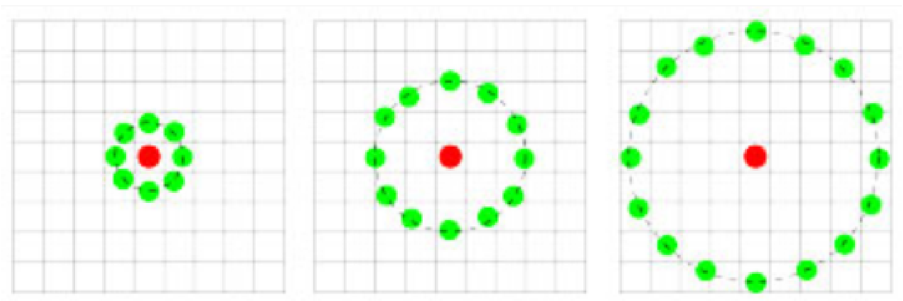


Figure 6: Three neighborhood examples with varying p and r used to construct Local Binary Patterns.

2.4 Support Vector Machine (SVM)

Image features such HOGs, LBPs are often used with SVM classifiers. Each feature description that is computed is fed to a SVM classifier to determine if the object was found or not. SVM is a algorithm that belongs to supervised learning group to classify data into distinct groups Support Vector Machine is a distinguished classifier defined by a separating hyperplane. With labeled training data, the algorithm outputs an optimal hyperplane in order that the distance to the nearest data point of both classes is maximized. This hyper-plane in two-dimensional space is a line dividing a plane to two parts where each class is on either side as in Fig.7.

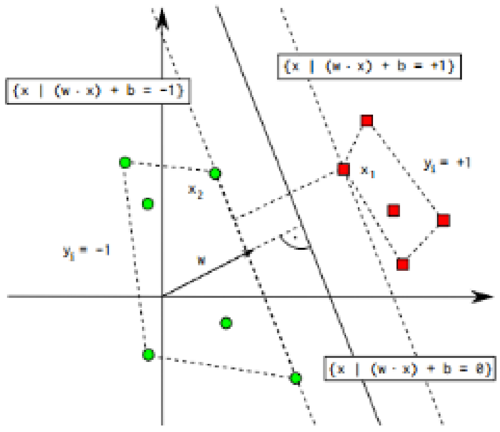


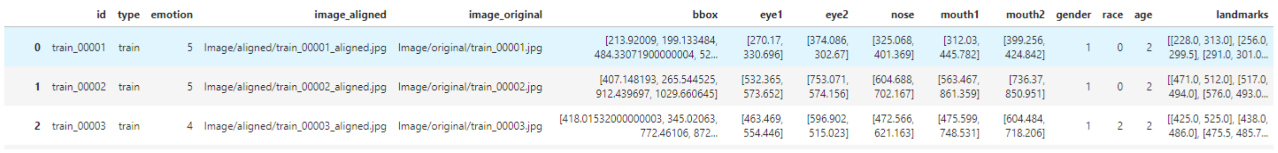
Figure 7: Supper Vector Machine (SVM) Classification

3 Project Contents and Algorithm

In this work, we use HOG and LBP features for facial emotion expressions. Finally, the descriptors are used by SVM (support-vector machine) to classify emotion recognition. Before we extract image features and classify them, we need pre-process data by creating indexing data which are saved to database table with id, direction, facial boundingbox, landmark positions in RAF-DB dataset.

3.1 Dataset preprocessing

First step, we download dataset RAF-DB and index image data, and information as bounding-box, 5 face basic points (left/right eye, tip of nose, and left/right mouth), 37 detail landmarks, etc. as in Fig.8.



	id	type	emotion	image_aligned	image_original	bbox	eye1	eye2	nose	mouth1	mouth2	gender	race	age	landmarks
0	train_00001	train	5	Image/aligned/train_00001_aligned.jpg	Image/original/train_00001.jpg	[213.92009, 199.133484, 484.33071900000004, 52...	[270.17, 330.696]	[374.086, 302.67]	[325.068, 401.369]	[312.03, 445.782]	[399.256, 424.842]	1	0	2	[[228.0, 313.0], [256.0, 299.5], [291.0, 301.0...
1	train_00002	train	5	Image/aligned/train_00002_aligned.jpg	Image/original/train_00002.jpg	[407.148193, 265.544525, 912.439697, 1029.660645]	[532.365, 573.652]	[753.071, 574.156]	[604.688, 702.167]	[563.467, 861.359]	[736.37, 850.951]	1	0	2	[[471.0, 512.0], [517.0, 494.0], [576.0, 493.0...
2	train_00003	train	4	Image/aligned/train_00003_aligned.jpg	Image/original/train_00003.jpg	[418.01532000000003, 345.02063, 772.46106, 872...	[463.469, 554.446]	[596.902, 515.023]	[472.566, 621.163]	[475.599, 748.531]	[604.484, 718.206]	1	2	2	[[425.0, 525.0], [438.0, 486.0], [475.5, 485.7...

Figure 8: RAF-DB database indexing

There are total 15,339 images with over 12,000 training images and 3,000 testing images. Distribution of emotion on the dataset is shown as Fig.9 below:

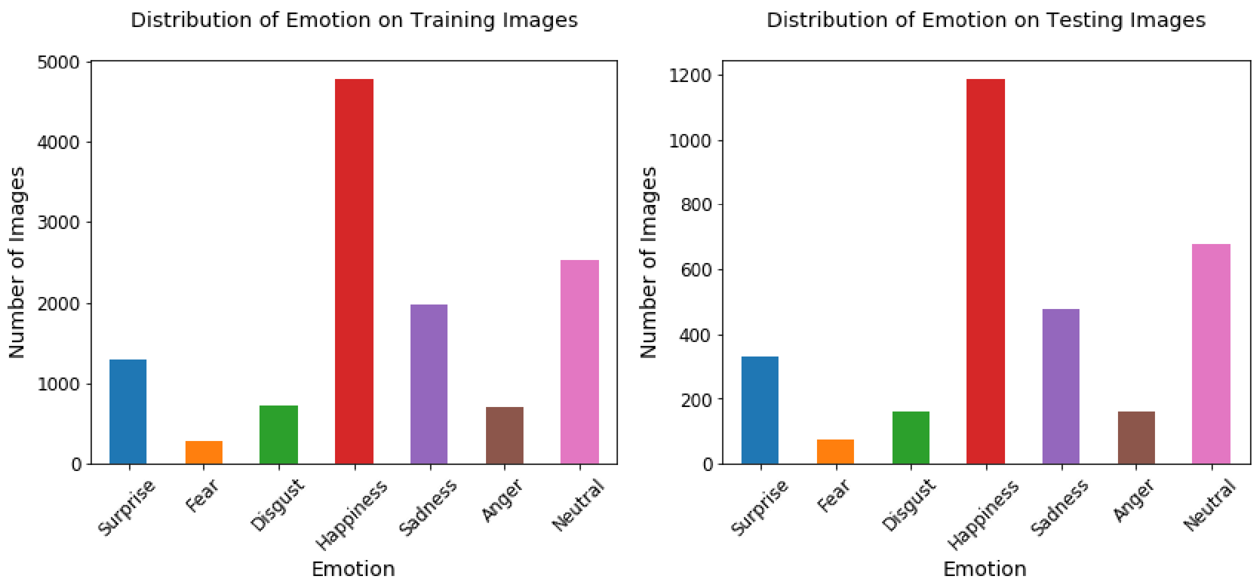


Figure 9: RAF-DB database distribution on Emotion

Every images will be annotated manually with bounding box, basic points, and landmarks as well as corresponding aligned images as in Fig.10 below:

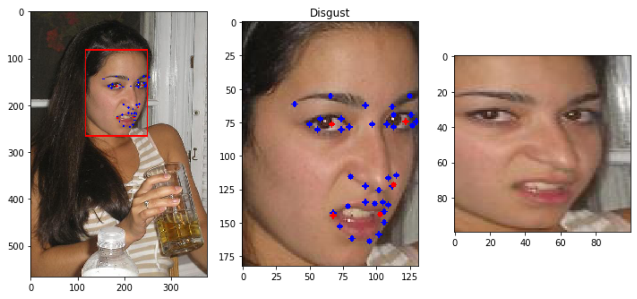


Figure 10: The annotation images

Some sample images in training and testing are shown in Fig.11 and Fig.12 below:

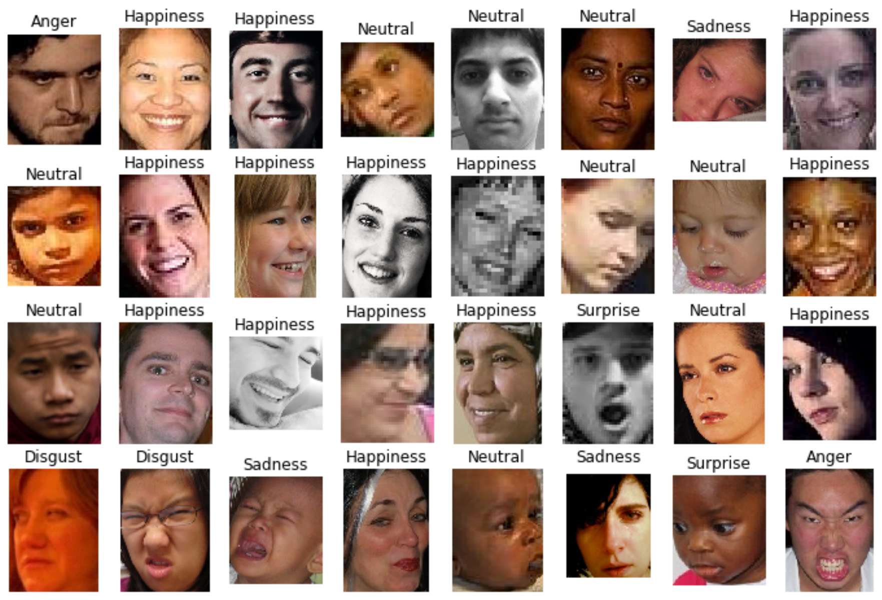


Figure 11: Training images

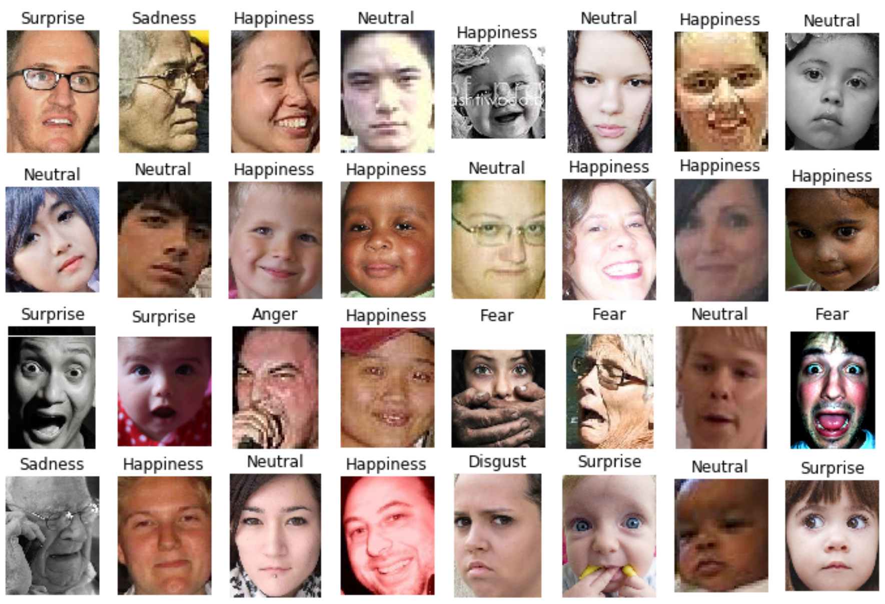


Figure 12: Testing images

3.2 Feature Extraction

We use sk-learn libraries to extract HOG features and LBP features. The HOG parameters in this project are pixels per cell $(8,8)$, cells per block $(3,3)$, and orientations 9 as Fig.13 below:

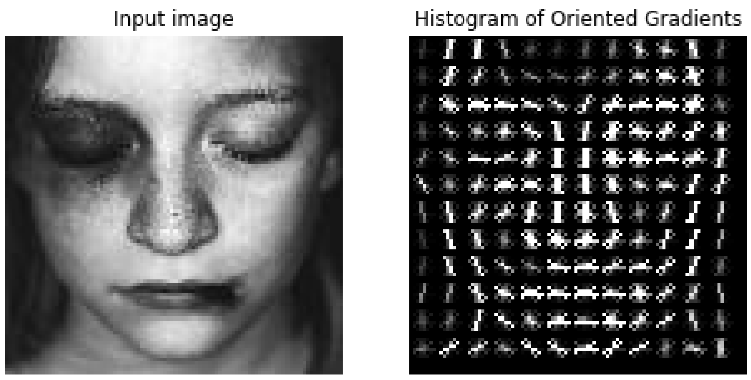


Figure 13: HOG Features

Fig.13 shows the HOG features present the face emotion well. Almost the emotion focuses on the eye, mouth, nose, and forehead. Besides, we extract the LBPs features on entire image. It shows in Fig.14 below:

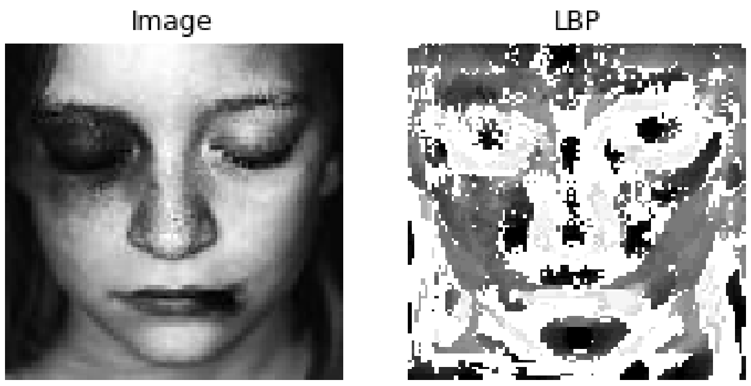


Figure 14: LBPs Entire Features

The LBPs features on entire image does not present well the emotion region. So, we extract LBPs features on patch images with block size 100 x 100 as Fig.15 below:

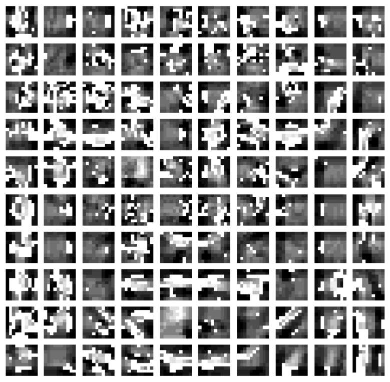


Figure 15: LBPs Patch Features

3.3 Classification

Final step in this proposed method as Fig.16 is classification. We use SVM to classify visual features. It constructs a hyperplane or set of hyperplanes in a high-dimensional space, which can be used for classification. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class.

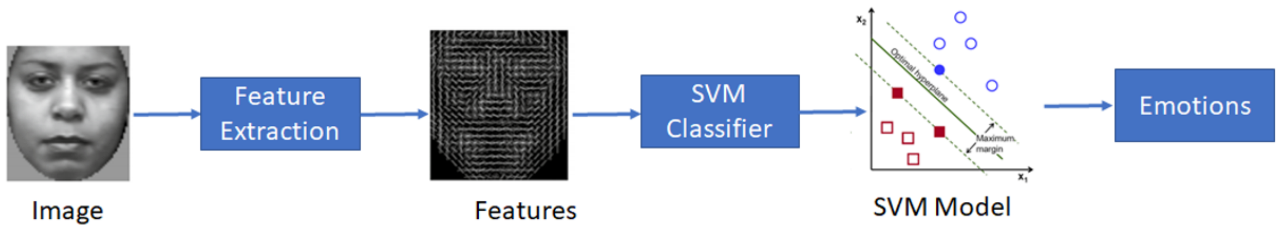


Figure 16: Overall proposed method

In real world application, finding perfect class for millions of training data set takes lot of time. We need to make turning parameters with regularization, kernel, gamma and margin. With kernel parameter, it defines whether we want linear/non-linear separation. For linear separation, we use kernel linear. Otherwise, polynomial and exponential kernels calculates separation line in higher dimension.

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’. In other words, with low gamma, points far away from plausible separation line are considered in calculation for the separation line. Where as high gamma means the points close to plausible line are considered in calculation as Fig.17.

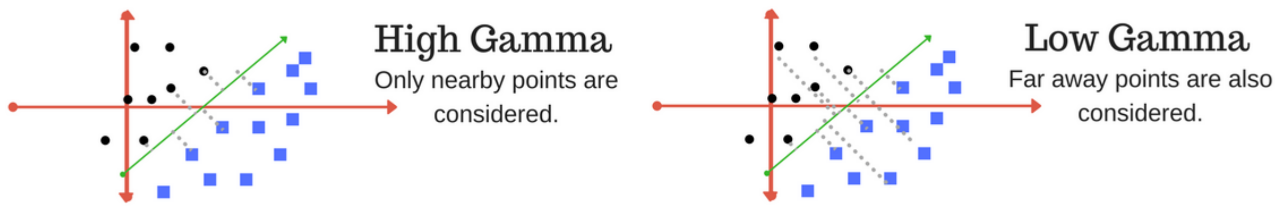

High Gamma
Only nearby points are considered.
Low Gamma
Far away points are also considered.

Figure 17: SVM gamma turning parameter C. Left: high gamma, right: low gamma

The regularization parameter (often termed as C parameter in python’s sklearn library) tells the SVM optimization how much you want to avoid mis-classifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

Fig.18 shows example of two different regularization parameter. Left one has some misclassification due to lower regularization value. Higher value leads to results like right one.

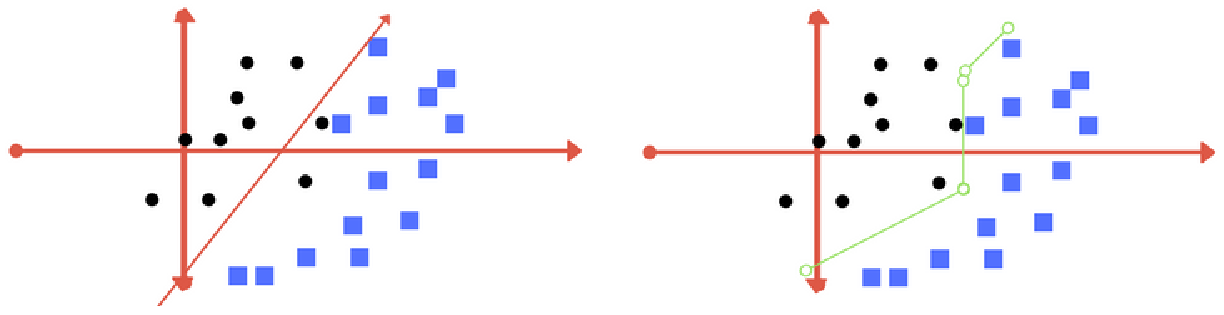


Figure 18: SVM regular turning parameter C. Left: low regularization value, right: high regularization value

4 Experiment and Result

4.1 HOG Features

In this experiment, we choose HOG parameters as following: pixels per cell $(8, 8)$, cells per block $(3, 3)$, and orientations 9. For training SVM, we use C regular parameter 100.0 and rbf kernel for non-linear separation. Fig.19 shows confusion matrix result:

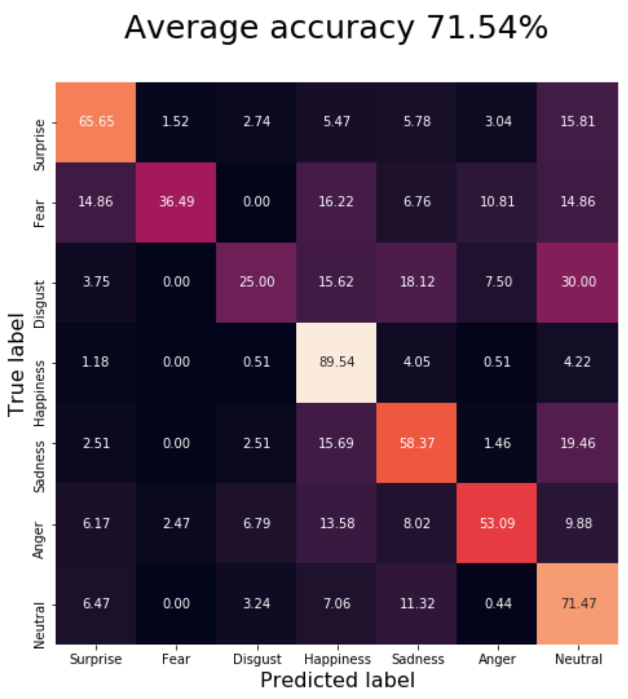
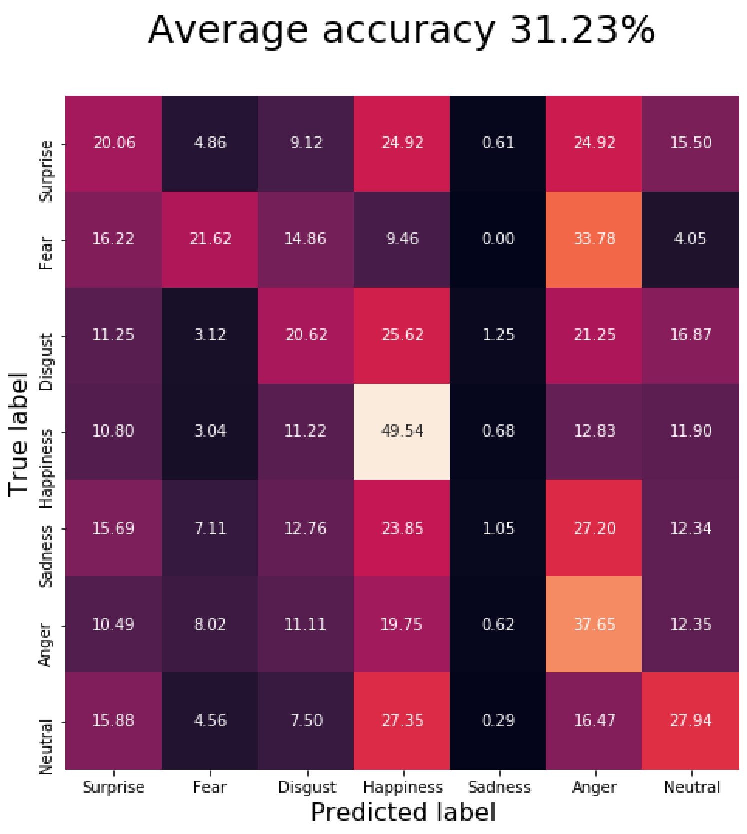


Figure 19: Confusion Matrix with HOG feature and SVM classification

The result shows good with accuracy 71.54%. Happy emotion has highest accuracy due to easy to detect and highest data. Digust, and Fear emotions have low accuracy because of low data as well as difficult emotions to recognize.

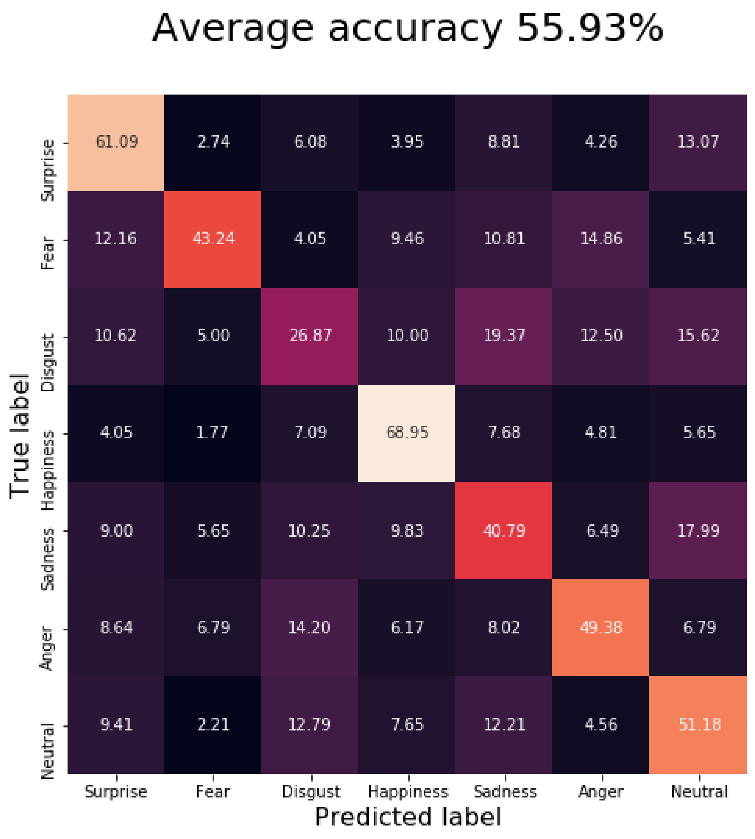
4.2 LBP Features

For LBPs, we achive accuracy 31.23% and 55.93% for entire and patch features, respectively as Fig.20 and 21. The reason for lowest accuracy on LBP features on entire images causes by not capturing well on emotion features on small detail regions at mouth, eye, forehead, etc. With using patch image features, we improve accuracy to 55.93%.



True label \ Predicted label	Surprise	Fear	Disgust	Happiness	Sadness	Anger	Neutral
Surprise	20.06	4.86	9.12	24.92	0.61	24.92	15.50
Fear	16.22	21.62	14.86	9.46	0.00	33.78	4.05
Disgust	11.25	3.12	20.62	25.62	1.25	21.25	16.87
Happiness	10.80	3.04	11.22	49.54	0.68	12.83	11.90
Sadness	15.69	7.11	12.76	23.85	1.05	27.20	12.34
Anger	10.49	8.02	11.11	19.75	0.62	37.65	12.35
Neutral	15.88	4.56	7.50	27.35	0.29	16.47	27.94

Figure 20: Confusion Matrix with LBPs entire and SVM classification



True label \ Predicted label	Surprise	Fear	Disgust	Happiness	Sadness	Anger	Neutral
Surprise	61.09	2.74	6.08	3.95	8.81	4.26	13.07
Fear	12.16	43.24	4.05	9.46	10.81	14.86	5.41
Disgust	10.62	5.00	26.87	10.00	19.37	12.50	15.62
Happiness	4.05	1.77	7.09	68.95	7.68	4.81	5.65
Sadness	9.00	5.65	10.25	9.83	40.79	6.49	17.99
Anger	8.64	6.79	14.20	6.17	8.02	49.38	6.79
Neutral	9.41	2.21	12.79	7.65	12.21	4.56	51.18

Figure 21: Confusion Matrix with LBPs patch feature and SVM classification

5 Conclusions

In this paper, we focus on facial emotion recognition using traditional feature and classification. Moreover, we use the dataset in-the-wild to recognize. We achive the best accuracy on HOG features using SVM classification.

References

[1] P. Ekman, "Facial expression and emotion.," *American psychologist*, vol. 48, no. 4, p. 384, 1993.

[2] S. Li and W. Deng, "Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 356–370, 2019.

[3] T. Surasak, I. Takahiro, C. H. Cheng, C. E. Wang, and P. Y. Sheng, "Histogram of oriented gradients for human detection in video," in *Proceedings of 2018 5th International Conference on Business and Industrial Research: Smart Technology for Next Generation of Information, Engineering, Business and Social Science, ICBIR 2018*, vol. 1, pp. 172–176, IEEE Computer Society, 2018.

[4] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 971–987, 2002.

[5] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.