ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 6, June 2016

Performance Analysis of Texture Based Feature Extraction Techniques for Facial Expression Recognition

Renu S. Patil, Vijay B. Baru

***Abstract*— Facial expression recognition (FER) has been a growing field of research that attracted the attention of many signal processing engineers. Various algorithms have been implemented to extract the facial features. These algorithms have a basis on two fundamental approaches, i.e. based on feature and based on texture. Feature-based algorithms have been proven to be better classifiers than feature descriptors. Texture based algorithms include Local Binary Pattern (LBP), its variant Local Ternary Pattern (LTP), higher order Local Derivative Pattern (LDP) and so on. Texture based algorithms outperform the feature-based algorithms in feature extraction hence, the focus is given on them. The aim is to implement the above-mentioned feature extraction algorithms and compare them on the basis of recognition efficiency. The performance evaluation is based on the parameters such as a number of feature vectors formed, training time and recognition time. The classifier used is Support Vector Machine.**

***Index Terms*—Local Binary Pattern, Local Ternary Pattern, Local Derivative Pattern, Support Vector Machine, Texture-based algorithms.**

I. INTRODUCTION

Facial expressions are an integrated part of communication for humans. Over the last few years, there has been a lot of research in this field to cope up with the challenges of extracting the features and recognizing the emotions. The six fundamental emotions that are studied in general are happiness, anger sadness, surprise, fear and disgust. The machine learning of facial expressions has been a progressing research area in computer vision. There is a notable progress in the last decade in the techniques of face detection and tracking, feature extraction methods and the mechanism implemented for classification of expression. Automatic FER systems constitute of three modules:

- Face Tracking and Face Detection
- Feature Extraction from the face and
- Expression Classification

Following is a framework of an FER system in its most generic form. The primary step in the analysis of facial expression is facial detection in an image or in a video sequence.

Fig 1. Framework of Facial Expression Recognition System

The process of locating a face in an image is called as face localization or face detection whereas obtaining location of the face and tracking it in sequential frames of a video is face tracking. Acquisition of data can be done both in 2-D or/and 3-D and also in the static mode or a dynamic mode. For faces that appear in complex scenes having backgrounds which are cluttered, face detection algorithms are used for location of the facial area present in the image, since almost all the methods require the accurate position of face for extracting features of interest from the face [1],[2]. Representation of facial expression is a process in which feature extraction is done, converting the facial information from a low-level representation such as 2-D pixel based or 3-D vertex-based, into a representation of the face which is higher-level, which constitutes of its landmarks, shape of the face, appearance, spatial configuration and/or motion. The features that are extracted play a vital role in the reduction of the dimensionality of original facial data as the input. A facial representation based on landmark uses characteristic points located near particular areas such as nose, eyebrows, mouth and edges of eyes since these areas are known to show prominent changes during facial articulation. Classification is the step in which the observed data is assigned to one of the facial expression categories that are predefined. This process design is specific and is dependent on the following

- Observation type (e.g. static or dynamic),
- Data representation style adopted and
- Classification algorithm itself.

II. TEXTURE BASED ALGORITHMS

This section consists of a brief introduction of the algorithms that are analyzed for their performance for the purpose of recognition of expression. The section gives a description of these algorithms followed by the mathematical formulation for the same.

Manuscript received June, 2016.
***Renu S. Patil**, Department of Electronics and Telecommunication, Sinhgad College of Engineering, Pune, India, 9403636311*
***Vijay B. Baru,** Department of Electronics and Telecommunication, Sinhgad College of Engineering, Pune, India.*

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 6, June 2016

A. *Local Binary Pattern*

It is an operator that assumes that texture locally has two complementary aspects, pattern and strength of it [3]. The version of operator designed for a 3 × 3-pixel image block is the original LBP operator. In this block, the central pixel value thresholds the neighboring pixels at 1 or 0. These binary values are then multiplied by powers of two and their summation is taken to get a label for the pixel of interest. As there are 8 pixels in the neighborhood, 2^8 = 256 distinct labels are obtained that depend on the relative gray-level values of the central pixel and the pixels of the surrounding. The derivation of a generic LBP operator is presented below. $I(x, y)$ is an image which is monochrome, is considered and the gray value of a pixel is denoted by g_c i.e. $g_c = I(x, y)$. Moreover, g_p denotes the gray level of a sample point in the neighborhood that is made up of evenly circular shape of P sample points and circumference of $2\pi R$ surrounding point (x, y):

$$g_p = I(x_p, y_p) \tag{1}$$

$$for\ p = 0, \ldots, P-1 \ \ and$$

Where,

$$x_p = x + R\cos 2\pi p/P\,,$$
$$y_p = y - R\sin 2\pi p/P.$$

Now, the approximation of this joint distribution is carried out considering that the central pixel boasts of statistical independence in case of differences. This enables the factorization of the joint distribution. The LBP code is formed by considering only the signs of the differences obtained after learning vector quantization is applied. The equation thus becomes

$$t(s(g_0 - g_c), s(g_1 - g_c), \ldots, s(g_{P-1} - g_c)) \tag{2}$$

Wherein, s(z) is a step function. The $LBP_{P,R}$ operator is defined as

$$LBP_{P,R}(x_c y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \tag{3}$$

In practice, (3) signifies that the sign of the differences that are obtained by comparing the center pixel and the pixels surrounding are seen as a P -bit binary number, that result in 2^P separate values for the LBP code. Uniform LBP patterns are used for better facial recognition as they are more stable and fewer samples are required for their reliable estimation of distribution. Also, use of uniform LBP patterns helps in dealing with the dimensionality problem, i.e. the 256 bin histograms are reduced to 59 bins only due to the uniform patterns that are considered.
The illustration of LBP code applied to an image matrix of 3*3 can be seen as follows.
Consider an image I(A) which is a 3*3 matrix given as

56	48	32
40	50	55
58	38	45

For this image, LBP code can be constructed by comparison of the values of neighboring pixels to the center pixel. Hence, the binary values that would be allocated to the neighboring pixels are given as follows

1	0	0
0		1
1	0	0

Here, the pixel values are converted into binary values by comparing to the pixel at the center. If the pixel value is greater than the pixel in the center, it is replaced with one, and if it is equal to or less than the center pixel, it is replaced with zero.

B. *Local Ternary Pattern*

Local Ternary Pattern is one of many variants of Local Binary Pattern. This is a three-level operator and was proposed to minimize the problems on image areas that are near constant [4]. LBPs have proven to give features that are highly discriminative in nature for texture classification and they show a certain amount of resistance to lighting effects so one can say that in a sense they show invariance to gray-level transformations that are monotonic. However, as the pattern operator thresholds at the center pixel value i_c, it tends to have a considerable noise sensitivity, particularly in regions that are non-uniform, and it smoothens the weak illumination gradients. Many facial regions show sufficient uniformity and it is beneficial to investigate whether improvement can be done in the robustness of the features in these regions. Basically, Local Ternary Pattern is a 3 valued code that is obtained with the implementation of the following function $step(i, i_c, t)$.

$$step(i, i_c, t) = \begin{cases} 1, & i \geq i_c + t \\ 0, & |i - i_c| < t \\ -1, & i \leq i_c - t \end{cases} \tag{4}$$

In which, i is the interest pixel, i_c is the pixel at the center and t is the value of the threshold that is decided on the amount of changes that are to be allowed in the pixel values without affecting the thresholding results. An example of a Local Ternary operator is given as follows. In the following example, the obtained LTP code is converted into two LBP codes in order to take care of the dimensionality reduction. These two LBP codes are subsequently called as upper LBP pattern and lower LBP pattern. Same as LTP there exists a Quaternary code wherein the values are split into five (-2,-1,0,1,-2). These can be split into four LBP patterns.
The illustration is as follows.

Ternary code:

1	1	0
0		0
0	-1	-1

1100(-1)(-1)00

Binary code:11000000

1	1	0
0		0
0	0	0

Binary code:00001100

0	0	0
0		0
0	1	1

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 6, June 2016

When the equation of LTP is obtained from LBP, slight modification is done in the formula as shown below

$$LTP_{P,R}(x_c y_c) = \sum_{p=0}^{P-1} s(g_p - g_c + |t|)2^P \quad (5)$$

C. Local Derivative Pattern

The Local Derivative Pattern operator was proposed for face recognition and the focused areas were feasibility of the high-order local patterns and their effectiveness for representation of face [5]. LDP operator that was proposed had the $(n-1)^{th}$-order derivative variations which are directional that depend on a coding function that is binary. In this approach, LBP, in concept, is regarded as the non-directional first-order operator, all-directional first-order derivative result is encoded by LBP and the higher-order derivative information which consists of detailed discriminative features is encoded by LDP. This information is such that the first-order pattern cannot be used for obtaining it from an image. Considering an image *I(A)*, the first order derivatives along the directions of 0°, 45°, 90° and 135° are denoted as $I'_\alpha(A)$ where α=0°, 45°,90° and 135°. Consider a 3*3 image I(A) where A_0 is the center pixel and $A_1,\ldots A_8$ as the neighboring pixels

A_1	A_2	A_3
A_8	A_0	A_4
A_7	A_6	A_5

The four derivatives of the first order at $A=A_0$ can be written as

$$I'_{0°}(A_0)=I(A_0)-I(A_4)$$
$$I'_{45°}(A_0)=I(A_0)-I(A_3)$$
$$I'_{90°}(A_0)=I(A_0)-I(A_2)$$
$$I'_{135°}(A_0)=I(A_0)-I(A_1)$$

The second order LDP, $LDP_\alpha^2(A_0)$ in a direction α at $A=A_0$ is defined as

$$LDP_\alpha^2(A_0) = \left\{ \begin{array}{l} f(I'_\alpha(A_0), I'_\alpha(A_1)), \\ f(I'_\alpha(A_0), I'_\alpha(A_2)), \\ \ldots, f(I'_\alpha(A_0), I'_\alpha(A_8)) \end{array} \right\} \quad (6)$$

In which $f(.,.)$ is a function that is used for coding that determines the type of transitions of the local pattern. The co-occurrence of two derivative directions at different pixels in the neighborhood is encoded by this function as

$$f(I'_\alpha(A_0), I'_\alpha(A_i)) = \begin{cases} 0, & if\ I'_\alpha(A_i).I'_\alpha(A_0) > 0 \\ 1, & if\ I'_\alpha(A_i).I'_\alpha(A_0) \le 0 \end{cases} \quad (7)$$
$$i = 1, 2, \ldots, 8.$$

And lastly, the second-order pattern, $LDP^2(A)$ is obtained by combining the four 8-bit directional LDPs in a concatenated form

$$LDP^2(A) = \{LDP_\alpha^2(A) \,|\, \alpha = 0°, 45°, 90°, 135° \quad (8)$$

It is observed that the LDP operator can be used for labeling of the pixels of an image by making a comparison of the two derivative directions at two pixels in the neighborhood and concatenation of the results is carried out as a 32-bit sequence. Contrary to the LBP that encodes the binary derivative gradient directions, the change of the derivative directions in the neighborhood is encoded by second-order LDP. This represents the local region's second-order pattern information.

III. Implementation Of Texture Based Algorithms

This section describes the implementation of the aforementioned algorithms and comments on their performance for recognition of facial expression. The database used for this comparison is created under controlled lighting conditions. The database consists of 98 images of 14 students expressing 6 basic emotions and a neutral expression. There is another database called as JAFFE (Japanese Female Facial Expression) database [6]. The database consists of a total of 213 images of 7 facial expressions (6 different + 1 neutral) posed by 10 Japanese female models. The section consists of each algorithm that has been implemented along with some observations regarding its performance for FER. The results obtained from the implementation are given on the created database.

A. Implementation of Local Binary Pattern

The algorithm for implementation of LBP program is given below.

1. Start
2. Read the image from the database.
3. Convert into grayscale form if required. Resize the image to 255*255.
4. Store the size of the image in two variables w and h.
5. Assign J=ltp(i,j). i varies from 2 to w-1 and j varies from 2 to h-1.
6. Select a block of the image and start the processing of the block from lbp(i-1,j-1)th pixel
7. Compare the value of the pixels in the surrounding to the pixel J in the center.
8. If the neighboring pixel value is greater than J, replace it with '1', if it is less than J, replace it with '0'.
9. Convert the obtained binary values into decimal values by multiplying with the appropriate powers of '2'.
10. Repeat steps 5 to 9.
11. Terminate when i=w-1 and j=h-1.

An example of an LBP image along with the original image is depicted in the figure below. The local binary pattern algorithm output is a histogram. These histograms are concatenated and are then converted into a feature vector by mapping and obtaining a 59 bin feature vector. It is obtained by using the uniform patterns that are present in the LBP codes. A uniform pattern is defined as a pattern in which there occur two transitions (a transition is counted as 0 to 1 or 1 to

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 6, June 2016

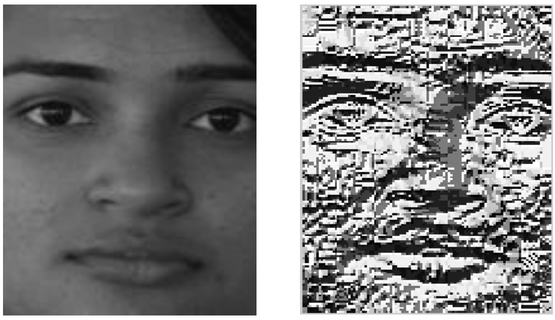

Fig 2. (a) Graylevel converted image and (b) LBP coded image

0). The uniform pattern count is assigned as the feature vector for the process of classification. As per the implementation that was carried out on the created database as well as JAFFE database, it is observed that Local Binary Pattern is good for expression recognition. However, it cannot be said that it is totally immune to noise. Its advantage is being operative in the local region, i.e. it shows invariance to gray scale changes which are monotonic. Nevertheless, the computation time that is required for the basic LBP implementation is more than is desirable for a real-time implementation. Hence, some of its other variants are experimented upon in research.

B. *Implementation of Local Ternary Pattern*

Algorithm for obtaining a local ternary pattern for an image classification is given as follows.

1. Start
2. Read the image from the database.
3. Convert into grayscale form if required. Resize the image to 255*255.
4. Store the size of the image in two variables w and h.
5. Assign J=ltp(i,j). i varies from 2 to w-1 and j varies from 2 to h-1.
6. Select a block of the image and start the processing of the block from lbp(i-1,j-1)th pixel
7. Predefine a threshold ‘t’ that gives a satisfactory range of values for which the ternary code can be ‘0’.
8. Compare the value of the neighboring pixels to the range of values J-t to J+t.
9. If the value of the neighborhood pixel is greater than J+t replace it with ‘1’, if it is less than J-t, replace it with ‘-1’ and if it lies between J+t to J-1, replace it ‘0’.
10. Convert the ternary code into upper and lower binary code by replacing the -1s by 0s and 1s by 0s.
11. Store the binary values obtained in the corresponding locations of another matrix Ipos and Ineg.
12. Convert the obtained binary values into decimal values by multiplying with the appropriate powers of ‘2’.
13. Repeat steps 5 to 10.
14. Terminate when i=w-1 and j=h-1.

An example of an image along with its upper LBP and lower LBP is shown below. These upper and lower LBP codes are obtained from reassigning the values for the ternary pattern. The ternary pattern image is separated into two binary pattern images of upper and lower versions. These images form the basis of the feature vector that needs to be constructed for the purpose of classification. Same as that in LBP, Local Ternary pattern also gives a histogram as the output. It is converted into a feature vector by using the uniform pattern mapping.

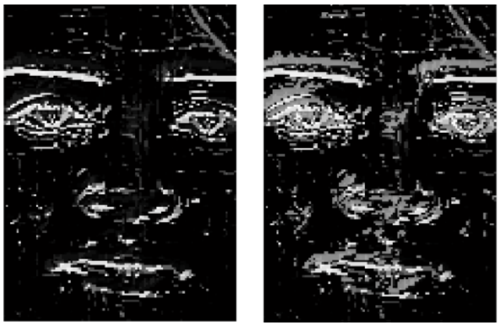

Fig 3. LTP coded image (a) positive and (b) negative

The advantage of using LTP is that in place of comparing the intensity of the center pixel as absolute, the neighboring pixels are compared to a range of values from $I(A_c)$*-t to* $I(A_c)$*+t* where $I(A_c)$ is the central pixel and *t* is the threshold value. This threshold value can be decided based upon the intensity variations of near constant image areas. Varying the threshold according to requirement gives an opportunity to suppress the influence of intensity variations that cannot be dealt with in LBP. The computation time required is, of course, double than that of LBP since there are two patterns to be computed and a double number of histograms are concatenated. In this case also, uniform patterns are looked for and information content is sought in them. Local Ternary Patterns are used not only for recognition of face and expression but for texture also.

C. *Implementation of Local Derivative Pattern*

The algorithm for implementation of a derivative pattern is given as follows.

1. Start
2. Read the image from the database.
3. Convert into grayscale form if required. Resize the image to 255*255.
4. Store the size of the image in two variables w and h.
5. Assign J=ldp(i,j). i varies from 2 to w-1 and j varies from 2 to h-1.
6. Obtain the binary codes for all the four directions specified in step 7.
7. Select a block of the image and start the processing of the block from lbp(i-1,j-1)th pixel.
8. Obtain the first order derivative at 45 degree, 90 degree, and 135 degree by taking the difference of neighboring pixel value from the center pixel.
9. Obtain the second order derivative and assign the binary values according to the function.

10. Store the binary values obtained in the corresponding locations of another matrix Ix0, Ix45, Ix90, and Ix135.
11. Convert the obtained binary values into decimal values by multiplying with the appropriate powers of '2'.
12. Repeat steps 5 to 10
13. Terminate when i=w-1 and j=h-1.

An example of an image and its LDP images along various angles are shown below.

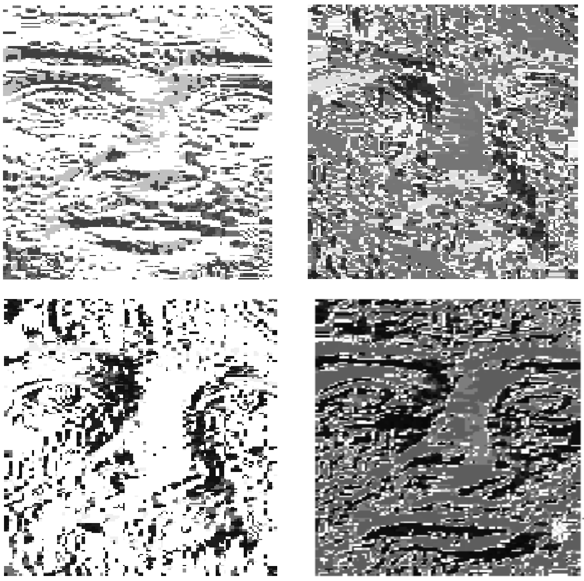

Fig 4. LDP coded image (a) at 0° direction (b) 45° direction (c) at 90° direction and (d) 135° direction

Local Derivative Pattern is the higher order pattern that can be used for expression recognition and texture recognition the best result for which is so far obtained by using second order derivatives. An LDP coded image is composed of the derivatives along the different directions (0°, 45°, 90° and 135°). LDP can be used to obtain more detailed information. This information contains discriminative features that are more detailed. These features cannot be obtained by the first-order pattern.

IV. Performance Of Algorithms

This section elaborates the performance of each of the above-mentioned algorithms. The performance is analyzed based on the number of feature vectors that are formed, the time required for the training of the database and the time required for the recognition of the expression. The recognized emotion is displayed by using a Graphical User Interface (GUI) created in MATLAB which is the platform on which this experiment is conducted. The version of MATLAB used is Matlab v.2013a. The timing analysis for all the implemented algorithms is given in a tabular form at the end of this section.

A. LBP

The LBP code was implemented and tested with JAFFE database. The feature vectors that were obtained by LBP implementation are given as an input to the Support Vector Machine (SVM). The total feature vectors is 59 out of which 56 are of uniform pattern and the non-uniform pattern is combines into remaining 4 feature vectors. SVM acts as a classifier for these feature vectors. The training time is 34.446 seconds. The recognition time obtained for LBP is 2.79 seconds. This is a very good recognition time. However, the low accuracy is a problem when this system would be implemented in real time.

B. LTP

The implementation of LTP code on JAFFE database gives a histogram of 64 bins. The feature vectors that are extracted are double in number i.e. 118. The time taken by LTP for training the database is 33.408 and the time taken for recognizing the expressions is 4.04 seconds. This time is almost double that of LBP. LTP is substantially more suitable for expression recognition. However, the time taken is almost double. Implementation of Local Ternary Pattern can be a part of the facial expression recognition system. However, real-time implementation could be complicated.

C. LDP

The performance of higher order derivatives for recognition of face is already demonstrated by Baochang Zhang. In this paper, the LDP is explored for usage in facial expression recognition. As there are four images to be obtained and their histograms to be concatenated, this vector becomes 128 bit. This is then converted to 32 bit with the help of mapping technique. This 32-bit feature vector is the input to the SVM classifier. The number of such feature vectors is 236. The time taken for training the database is 45.054 seconds and the recognition time is 7.31 seconds. This duration is almost four times that of LBP. The higher order derivative can surely by implemented in a recognition system. There could be a significant tradeoff between the recognition rate and the time taken for recognition. The performance of the above-mentioned algorithms is summarized in a tabular form. It displays the recognition rate of every expression for every algorithm.

TABLE I: COMPARISON OF LBP, LTP AND LDP

Parameters	**Algorithms**		
	LBP	***LTP***	***LDP***
Number of feature vectors	59	118	236
Training time (seconds)	34.446	33.408	45.045
Recognition time (seconds)	2.79	4.04	7.31

V. Conclusion

The training time required for Local Ternary Pattern is the lowest. However, the recognition time required is more than in the case of the basic LBP operator. If the length of the feature vector is reduced, it will result in a reduction of computation time. As the recognition time matters the most in real time applications, the basic LBP still holds an upper hand in expression recognition out of the analyzed techniques. There are other techniques who would be superior to those undertaken in this paper. Nevertheless, the fact would remain that the simpler the computation, faster is the execution speed. The recognition efficiency can be increased by using the adaptive boosting combined with Support Vector Machine, called as AdaSVM [7]. The use of these feature detection algorithms depends on the application where tradeoff can be made between the time required for recognition and the recognition rate.

ACKNOWLEDGMENT

The author would like to express deep gratitude towards Head of the Department, Dr. M.B. Mali, department of Electronics and Telecommunication SCOE, Pune for his encouragement for this paper. An earnest gratitude towards Principal, Dr. S.D. Lokhande, Sinhgad College of Engineering for giving this valuable opportunity. A sincere thanks all those who helped in making this a significant achievement.

REFERENCES

[1] Heisele, B.; Serre, T.; Pontil, M. & Poggio, T. Component-based face detection. In Proceedings on *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 657-662. 2001

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Comput. Vis. Pattern Recognit.*, vol. 1, pp. I–511–I–518, 2001.

[3] Ojala, T.; Pietikainen, M. & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.24 (7), pp. 971-987.

[4] X. Tan and B. Triggs, "Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions," pp. 168–182, 2007.

[5] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Pattern : Face Recognition With High-Order," vol. 19, no. 2, pp. 533–544, 2010.

[6] M. Lyons and S. Akamatsu, "Coding Facial Expressions with Gabor Wavelets", pp. 200–205, 1998.

[7] Marian Stewart Bartlett, Gwen Littlewort, Claudia Lainscsek, Ian Fasel and Javier Movellan." Machine Learning Methods for Fully Automatic Recognition of Facial Expressions and Facial Actions," In Proceedings of IEEE Conference on Systems, man & Cybernetics, The Hague, Netherlands, 2004.

Renu Patil received Bachelor's Degree in Engineering from Modern College of Engineering, Pune in the year 2013. Currently she is pursuing a Master's Degree in Electronics and Telecommunication with a specialization in Digital Systems from Sinhgad College of Engineering, Vadgao (Bk), Pune. Her areas of interest include image processing and computer vision. Under which she is most interested in working on real time face and facial feature recognition and Object recognition.

Prof. Vijay Baru is an Associate Professor in Sinhgad College of Engineering, Vadgao (Bk), Pune. He received Master's Degree in Engineering from College of Engineering, Pune in the year 1996. He has published more than 40 papers in national and international conferences. He is a renowned author of a book titled 'Electronic Product Design' by Wiley Publication. He has also written a book on Basic Electronics engineering by Dreamtech publication. He has 21 years of experience in teaching. His areas of interest include image processing and embedded systems.