Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 11, No. 1, November : 2023

EXPLORING ENHANCED EMOTION RECOGNITION: INTEGRATING DLTP FEATURE POINTS WITH 68 FEATURES USING VARIED MACHINE LEARNING ALGORITHMS

Mrs. N. Srilatha, PhD scholar, Dept. Of Computer Science, JNTUA Ananthapur, AP
Dr V. Lokeswara Reddy, Professor, Dept. Of CSE, KSRM College of Engineering, Kadapa, AP.

Abstract
Emotion recognition holds significant potential in human-computer interaction, affective computing, and various psychological studies. This research delves into advancing the accuracy of emotion recognition through the integration of Deep Local Texture Patterns (DLTP) feature points with a comprehensive set of 68 features, employing an array of diverse machine learning algorithms. The primary objective is to investigate the effectiveness of combining DLTP features with extensive feature representation, and subsequently, determine the most suitable machine learning algorithms for achieving heightened emotion recognition accuracy. The proposed methodology involves the extraction of DLTP feature points, complemented by 68 distinct features encompassing geometric, statistical, and texture-based attributes for high accuracy with less complex time. This combined feature set captures both local and holistic information present within the emotion-related facial expressions.

Keywords:
DLTP, Feature Extraction,68 feature extraction, Emotions

I. Introduction

Object recognition pertains to computer vision and image processing, encompassing the detection and classification of various entities like humans, buildings, and vehicles within digital images and video sequences. A notable facet is face recognition, which operates in verification and identification modes [1]. This paper's focus is on the identification mode, particularly in recognizing faces. Due to its multidimensional nature, face recognition requires robust computational analysis. The core challenge is accurately determining an individual's identity and making decisions based on this recognition outcome. While primarily crucial for security, it can also grant swift access to medical, criminal, or other records. This problem's solution bears significance, enabling preventive actions, improved service, secure access, and more.Face identification involves identifying a person in images or videos and validating their identity. It entails matching a query face against template images in a face database to ascertain the query's identity. This mode yields both positive and negative recognitions, with computational complexity escalating for larger template databases [2,3]. Our objective is identifying the person within the gallery corresponding to the query face. Upon submitting a query image, the normal map undergoes compression to compute feature indexes. These indexes narrow down the search to similar normal map clusters via a k-d-tree traversal [4].Over the years, academia and industry have developed diverse research and practical solutions to address face recognition challenges, particularly in pattern recognition and computer vision [5]. Facial recognition is intricate due to the susceptibility of facial morphology to factors like pose, lighting, and expression [3]. Faces share common components (eyes, nose, lips), necessitating efficient algorithms for similarity representation and distinct classification of diverse subjects. Local Binary Patterns (LBP) and k-Nearest Neighbor (K-NN) are notable solutions, initially used for texture classification but now tackling face recognition issues. LBP offers robustness and computational efficiency for real-time analysis [6]. K-NN excels in image classification due to its interpretability and low computation time [7,8]. Here, LBP and K-NN aim to extract and classify features from LBP histograms to enhance feature matching and identification rates.However, facial features might change during speech or partial occlusion. Rosenblum et al. [9] employ network techniques, dividing facial expression recognition (FE) complexity into three decomposition layers. In [10], occlusion-resistant FER is explored via localized

FE feature representation and classifier output fusion. Abboud and Davoine [11] propose a bilinear factorization expression classifier for facial recognition. Detecting and recognizing facial expressions play a pivotal role in nonverbal communication, with facial expressions carrying more message content than verbal language [12]. Recently, Support Vector Machines (SVM) have emerged within the framework of statistical learning theory [13],[14]. They have exhibited notable success across diverse applications, spanning from forecasting time series to facial recognition [15], and even processing biological data for medical diagnosis. This blend of theoretical underpinnings and practical achievements has kindled interest in exploring SVM's attributes and expanding its applications. This document provides a concise primer on SVM's theory and implementation, accompanied by a discussion of the five papers featured in the workshop. Initially conceptualized as a means to amalgamate multiple CART-style decision trees using bagging [18], random forests [16] have undergone development. Their inception was influenced by the random subspace method introduced in [19] and [20]'s work on feature selection. Core concepts underpinning random forests can also be traced back to early endeavors in assembling decision tree ensembles.

II. Methodology

Facial Expression Recognition (FER) stands as a pivotal pursuit in the realm of computer vision, garnering significant attention in recent times. The FER process typically encompasses several stages: face detection, facial alignment, feature extraction, and classification. Within this discourse, we delve into the core methodologies and techniques frequently employed for each of these phases. In the context of face detection and alignment, this article presents an exploration of prominent algorithms, notably Viola-Jones and Active Appearance Models. We delve into the merits and shortcomings inherent in each. Regarding feature extraction, we shed light on widely used approaches like Local Binary Patterns (LBP) and Gabor wavelets, elucidating their significance. Turning to the classification aspect, we furnish an overview of diverse techniques tailored for FER. These include Support Vector Machines (SVMs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs). We delve into the essence of each approach, highlighting their contributions and applicability. Moreover, we acknowledge the hurdles and restrictions that accompany each stage of FER. These challenges are outlined in tandem with potential avenues for overcoming them. The article concludes by casting a spotlight on the emergent domains for further FER research, charting the course for advancements in this field.

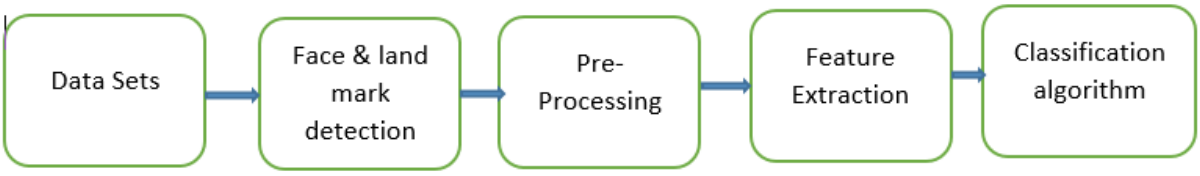


Fig1: Methodology for predicting expression

The insights presented within this article serve as a compass for steering the advancement of more precise and efficient Facial Expression Recognition (FER) systems, adaptable to an array of real-world contexts. The progression through the following steps facilitates this development:

1. Face Detection: Locate and pinpoint faces within images or video frames.
2. Pre-processing: Enhance input image or video quality, encompassing noise reduction, histogram equalization, and face alignment.
3. Feature Extraction: Extract pertinent features from pre-processed images or video frames, such as Local Binary Patterns (LBP), Histograms of Oriented Gradient (HOG), and deep features from Convolutional Neural Networks (CNNs).
4. Expression Classification: Utilize the extracted features to categorize facial expressions into predefined emotional states, e.g., happy, sad, angry, neutral.
5. Evaluation: Gauge the FER system's efficacy through metrics like accuracy, precision, recall, and F1 score.

The specifics of FER steps and techniques can vary contingent upon the distinct task, dataset, and model architecture in play.

2.1 Face Detection

Face detection assumes paramount importance within FER. It precisely identifies and isolates faces from backgrounds, paving the way for subsequent analysis. A plethora of methods, both traditional and deep learning-based, exists for face detection. Noteworthy techniques include the Viola-Jones algorithm, Histogram of Oriented Gradients (HOG) method, Single Shot Multibox Detector (SSD), and You Only Look Once (YOLO) algorithm. Subsequently, processed faces yield data for further analysis, encompassing Convolutional Neural Networks (CNNs) and Facial Landmark Detection (FLD) for feature extraction.

2.2 Pre-processing

Pre-processing enhances the integrity of input imagery in FER, facilitating information extraction and enhancing expression recognition precision. It entails:

1. Face detection.
2. Face alignment to a standardized orientation.
3. Image enhancement to remove noise and improve contrast.
4. Image normalization to ensure standardized intensity values.

These measures mitigate input variability and enhance the FER system's accuracy.

2.3 Feature Extraction

Feature extraction, a pivotal FER step, selects and distils relevant data from pre-processed images. Approaches encompass:

1. Geometric features, including facial landmarks and ratios.
2. Texture features like Local Binary Patterns (LBP) and Histograms of Oriented Gradient (HOG).
3. Deep features from CNNs trained on comprehensive datasets.

Method selection hinges on task specifics, with deep features excelling in complex scenarios and geometric/texture features offering computational efficiency.

2.4 Expression Classification

Expression classification, the ultimate FER step, assigns labels to images based on facial expressions. It incorporates techniques like statistical classifiers (SVM, Naive Bayes), neural networks (CNNs, RNNs), and ensemble methods. Contextual factors influence method choice, with neural networks thriving on complex data and statistical classifiers excelling in efficiency.

2.5 Evaluation

FER evaluation gauges system performance through metrics like accuracy, precision, recall, F1 score, confusion matrix, ROC curve, and AUC. Comprehensive evaluation compares against existing systems and embraces diverse datasets.

III. Existing Local Binary Patterns for Feature Extractions

Incorporating multiple LBP variants bolsters texture analysis, particularly for emotion recognition. These LBP variants include the original, Uniform, Circular, Rotation-Invariant, Extended, Completed, and Improved LBP. Each variant possesses distinctive characteristics, contributing to nuanced texture pattern recognition. Implementation involves applying LBP variants to specific facial regions, followed by classification using SVMs, Random Forests, or deep learning models. Performance comparison of these variants on benchmark datasets showcases their efficacy in emotion feature extraction. The choice of variant depends on dataset attributes and emotion recognition task nuances.

3.1 Original LBP

The initial LBP operator was presented by Ojala et al. [21]. This operator operates on the eight neighbouring pixels of a central pixel, utilizing the central pixel's intensity as a threshold. If a neighbouring pixel possesses a higher or equal gray value compared to the central pixel, it receives a value of one; otherwise, it's assigned zero. This procedure generates the LBP code for the central pixel, formed by concatenating the resulting eight binary values (as illustrated in figure 2).

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 11, No. 1, November : 2023

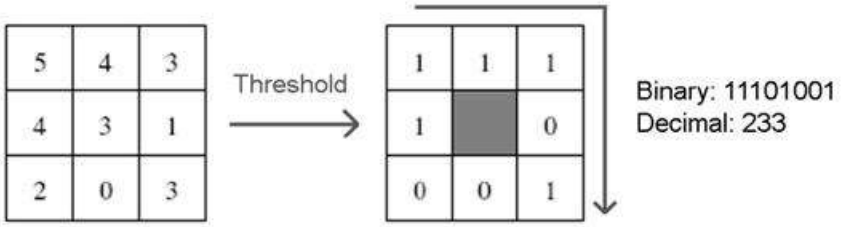


Figure 2: The Original LBP Operator

Subsequently, the LBP operator was expanded to incorporate neighbourhoods of varying dimensions. Here, a circular region with a radius of R is defined around the central pixel. P sampling points, evenly spaced along this circular edge, are assessed in comparison with the central pixel's value. To acquire the sampling point values across the neighbourhood for any radius and sampling point count, (bilinear) interpolation becomes essential. The notation (P, R) is employed to denote these neighbourhoods. In Figure 1.4, different (P, R) configurations are depicted to illustrate three distinct sets of neighboring points.

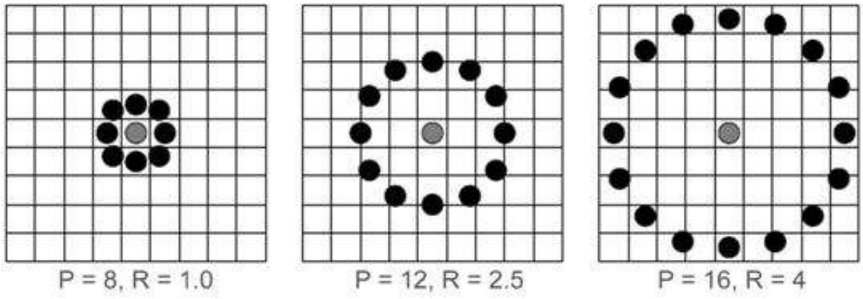


Figure 3: Circularly neighbor-sets for three different values of P and R

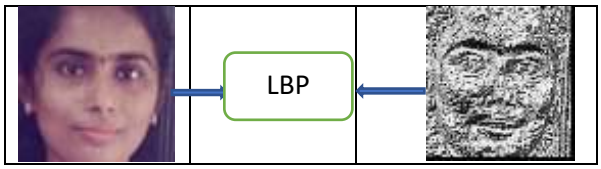


Fig4: Basic LBP image

If the coordinates of the center pixel are (xc, yc) then the coordinates of his P neighbours (xp, yp) on the edge of the circle with radius R can be calculated with the sinus and cosines:

xp = xc + Rcos(2$\pi\pi$p/P) (1)

yp = yc + Rsin(2$\pi\pi$p/P) (2)

If the gray value of the center pixel is gc and the Gray values of his neighbours are gp, with p = 0, ..., P –

1, than the texture T in the local neighbourhood of pixel (xc, yc) can be defined as:

$$T = t(g_c, g_0, \ldots, g_{P-1})$$

Upon attaining these point values, an alternative method of characterizing texture emerges. This involves subtracting the central pixel's value from the values of the points situated on the circular periphery. This approach effectively encapsulates local texture through a combined distribution of the central pixel's value and the resulting differences:

$$T = t(g_c, g_0 - g_c, \ldots, g_{P-1} - g_c) \quad (4)$$

Since t(gc) describes the overall luminance of an image, which is unrelated to the local image texture, it does not provide useful information for texture analysis. Therefore, much of the information about the textural characteristics in the original joint distribution (Eq. 3) is preserved in the joint difference distribution.

$$T \approx (g_0 - g_c, \ldots, g_{P-1} - g_c) \quad (5)$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 11, No. 1, November : 2023

While immune to grayscale shifts, the differences are susceptible to scaling. To ensure invariance against any monotonous transformation of the grayscale, only the sign of the differences is taken into account. Consequently, when a point on the circular periphery holds a Gray value higher than or equal to the centre pixel, it's assigned one; otherwise, it's assigned zero:

$$T \approx (s(g_0 - g_c), \ldots, s(g_{P-1} - g_c) \quad (6)$$

Where

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

In the last step to produce the LBP for pixel (xc, yc) a binomial weight 2p is assigned to each sign s(gp − gc). These binomial weights are summed:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{p-1} s(g_p - g_c)2^p \quad (7)$$

The Local Binary Pattern characterizes the local image texture around (xc, yc). The original LBP operator

in figure 1 is very similar to this operator with P = 8 and R = 1, thus LBP8,1. The main difference between these operators is that in LBP8,1 the pixels first need to be interpolated to get the values of the points on the circle.

3.2 Uniform LBP

A Local Binary Pattern earns the label of "uniform" if it encompasses a maximum of two bit-level shifts between 0 and 1, or vice versa. To clarify, this signifies that a uniform pattern either experiences no transitions or exactly two transitions. The scenario of a solitary transition is excluded since the binary sequence is regarded as circular. Noteworthy examples include the zero-transition patterns (e.g., 00000000 and 11111111) and uniform patterns with eight bits and two transitions, such as 00011100 and 11100001. With patterns involving two transitions, there are P (P − 1) potential combinations. For uniform patterns with P sampling points and radius R the notion $LBP^{u2}_{P,}$ is used.

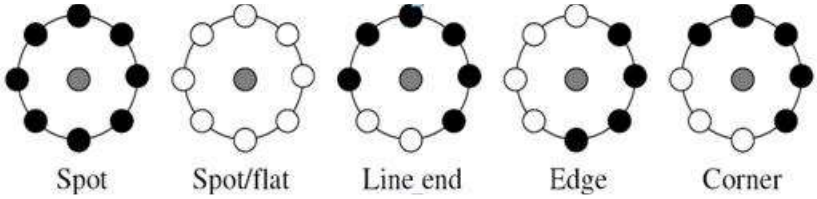


Fig5: Different texture primitives detected by the

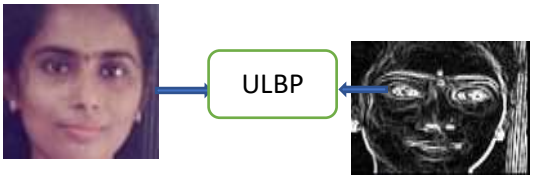


Fig6. Uniform LBP image

$LBP^{u2}_{P,}$ using only uniform Local Binary Patterns has two important benefits. The first one is that it saves memory. With non-uniform patterns there are 2P possible combinations. With $LBP^{u2}_{P,}$ there are P (P − 1) + 2 patterns possible. The number of possible patterns for a neighbourhood of 16 (interpolated) pixels is 65536 for standard LBP and 242 for LBPu2. The second benefit is that LBPu2 detects only the important local textures, like spots, line ends, edges and corners. See figure 6 for examples of these texture primitives.

3.3 Circular LBP

A circular neighbourhood is delineated through a collection of sampling points, uniformly distributed along a circle centred on the target pixel for labelling. Dictated by variables and respectively, the circular local neighbourhood's configuration is determined by the count of sampling points and the

circle's radius. Moreover, to accommodate sampling points not precisely coinciding with pixel canters, bilinear interpolation is employed. A depiction of the circular LBP operator is provided in Figure 7.

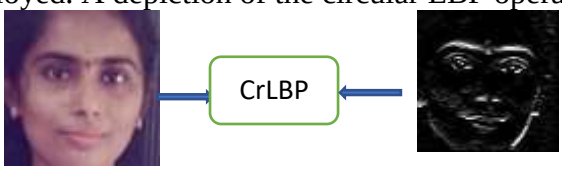


Fig7: Circular LBP image

Moreover, the theory of circular LBP operator is formally defined as follows: Given a pixel at location (x_c , y_c) and its circular neighbourhood (P R,) , sampling points locations (x_k ,y_k) are computed as

$$(x_k, y_k) = \left(x_c + Rcos\left(\frac{2\pi k}{P}\right), y_c - Rsin\left(\frac{2\pi k}{P}\right)\right) \quad (1)$$

And the intensity values at these sampling points denoted by $v_p = I(x_k, y_k)$ with k € { 0,1,…P-1}. The basic LBP code is then expressed in the decimal format as:

$$LBP_{P,R}(x_k, y_k) = \sum_{k=0}^{P-1} S(v_k + v_c)2^k \quad ,$$

Where v_k and v_c are respectively intensity values of the center pixel and the k^{th} neighborhood pixels in the circular neighborhood (P,R), and the thresholding function S(u) is defined as :

$$S(u) = \begin{cases} 1, u \geq 0 \\ 0, u < 0 \end{cases} \quad (3)$$

The above definitions express the properties of the LBP operator which are its resistance to illumination changes and its simplicity in computation. For P sampling points, 2^n LBP codes ranging between 0 and 2^{p-1} can be derived to form a LBP feature vector. This technique suffers from the curse of dimensionality. For example if one uses a circular (8,1) neighbourhood to describe an image divided into 64 blocks, then the resulting $LBP_{8,1}$-feature vector is of high dimension which is equal to 16,384. So high dimensionality of feature descriptors implies high discriminability but low classification effectiveness both at the training and testing stages. Therefore, a high discriminative feature descriptor with low feature dimension is required [22].

3.4 Rotation-Invariant LBP (RI-LBP)

The essence of the rotation-invariant LBP operator involves employing the standard LBP operator to generate a circular binary code. This code is then consistently rotated, yielding a sequence of LBP values. Throughout the rotation, the previous factor p2 in each code remains constant, and the smallest value after rotation is adopted as the ultimate LBP value. Concurrently, to address the challenge posed by an excess of binary modes, the LBP operator equivalent to the mode is employed to diminish the dimensionality of the rotation-invariant LBP operator's mode types. This process is formulated as (1):$$LBP_{P,R}^{n} = \min\left(ROR\left(LBP_{P,R}^{n} i\right) \middle| i = 01,2 \ldots P-1\right) \quad (1)$$

Where, ROR(x,i) is the rotation function, I is the number of bits of cycle shift(i<p), $LBP_{P,R}^{n}$ is the rotation invariant LBP operator, and $LBP_{P,R}^{n}$ is combined with the equivalent mode to obtain the equivalent mode of rotation invariant as shown in the below formula

$$LBP_{P,R}^{riu2} = \begin{Bmatrix} \sum_{p=0}^{P-1} s(g_p - g_c \; if \; U\left(L(P,R)\right) \leq 2 \\ p+1 \quad otherwise \end{Bmatrix} \quad (2)$$

Where, U(L(P,R) is the number of changes from 0 to 1 or from 1 to 0 fig () is the facial expression feature graph extracted by rotation invariant LBP operator [23] .

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 11, No. 1, November : 2023

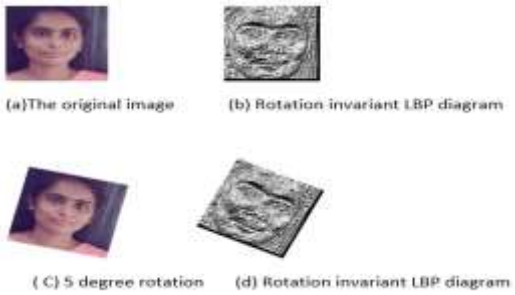


Fig8: ROR LBP image

3.6 Extended LBP (ELBP)

ELBP enhances the original LBP by considering additional information such as spatial relationships between pixels. It incorporates the radius and angular differences between neighbouring pixels, result [24].

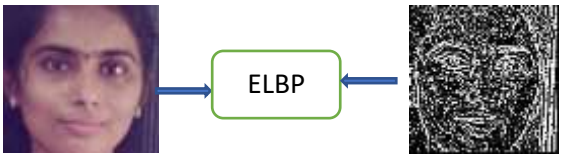


Fig 9: Elbp image

3.6.1 Radial Difference Local Binary Pattern (RDLBP)

LBP computation involves comparing the values of neighbouring pixels on a circular path with the central pixel value. This process solely encodes the connection between the central pixel and neighbouring pixels situated on the same ring (within a single scale), overlooking the second-order associations across varying rings (scales). For each image pixel, examination is extended to two rings: one with a radius of r and the other with a radius of r - δ, both centred at pixel xc. Additionally, p pixels are uniformly distributed along each ring. To produce the RDLBP codes, we first compute the radial differences $\{x_{r,p,n} - x_{r-\delta,p,n}\}$ n between pixels on the two rings and then threshold them against 0. The formal definition of the RDLBP code is as follows:

$$RDLA_{r,p,q}\ (x_c) = \sum_{n=0}^{p-1} S(P_{r,p,n} - P_{r-\delta\ ,p,n})2^k$$

3.6.2 Angular Difference Local Binary Pattern (ADLBP)

LBP's inadequacy in capturing second-order relationships among ring pixels is evident. Consequently, ADLBP addresses this limitation by incorporating angular comparisons (such as clockwise direction) between neighbouring pixels, excluding the central one. Mathematically, the computation of ADLBP can be expressed as follows:

$$ADLA_{r,p,q}\ (x_c) = \sum_{n=0}^{p-1} S(P_{r,p,n+1} - P_{r\ ,p,n})2^k$$

Compact and informative, RDLBP and ADLBP exhibit grayscale invariance and computational efficiency. Furthermore, they lend themselves to extensions that confer rotation invariance, uniformity, and even a 3D expansion of the LBP concept.

3.7 Completed LBP (CLBP)

The Completed Local Binary Pattern (CLBP) operator embodies a method for characterizing local regions. It encompasses the local contrast represented by the central Gray level and the Local Difference Sign Magnitude Transform (LDSMT). LDSMT is further decomposed into its sign and magnitude constituents. The Completed LBP encapsulates three distinct operators: CLBP_C, CLBP_S, and CLBP_M. CLBP_C encodes the central Gray level following global thresholding. CLBP_S and CLBP_M are responsible for coding the sign and magnitude components, respectively.

Subsequently, these three code maps are amalgamated to generate the CLBP feature map, which is then employed to create the CLBP histogram.[25]

The CLBP_S operator can be given by

$$CLBP_{S_{P,R}} = \sum_{P-0}^{P-1}(g_p - g_c)2^p \quad (1)$$

Where $(x) = \begin{cases}1, x \geq 0\\0, x < 0\end{cases}$, and the g_c is the gray value of the center pixel, g_p is the value of its neighbors, P is the number of involved neighbors.

The CLBP_M operator can be defined by

$$CLBP_{M_{P,R}} = \sum_{P=0}^{P-1} t(m_p, c)2^p \quad (2)$$

Where $t(x,c) = \begin{cases}1, x \geq 0\\0, x < 0\end{cases}$ and c is the threshold value, here it is set m_p which is the mean value for the whole image. The CLBP_C can be defined by

$$CLBP_{C_{P,R}} = t(g_c, C1) \quad (3)$$

These various adaptations of Local Binary Patterns provide distinct avenues for capturing texture details within images. The selection of a particular variant hinges on the unique attributes of the image data and the demands of the intended application. Often, experimentation and assessment using specific datasets are essential to ascertain the optimal Local Binary Pattern variant for a particular task.

iv. Proposed Methodology

The proposed Facial Expression Recognition (FER) pipeline's schematic is illustrated in Figure 10. The pipeline comprises six integral components: face detection & landmark localization, face alignment & registration, image enhancement, feature extraction, dimensionality reduction, and classification. When fed an input image, the face detection & landmark localization module identifies potential faces and their corresponding facial landmarks. In the subsequent phase, the face alignment & registration unit aligns and rescales facial images to a standardized resolution. Following this, image enhancement techniques are applied. Features are then extracted from these improved images utilizing the DLTP descriptor. Subsequently, the high-dimensional features are processed through 68 feature points, employed to extract features pertinent to Facial Expression Recognition (FER). These 68 feature points employ texture analysis to quantify spatial relationships amid pixel intensities in an image. Lastly, the reduced facial features undergo classification, with Machine Learning classifiers assigning expression labels. The subsequent section delves into comprehensive insights into each unit constituting the pipeline.

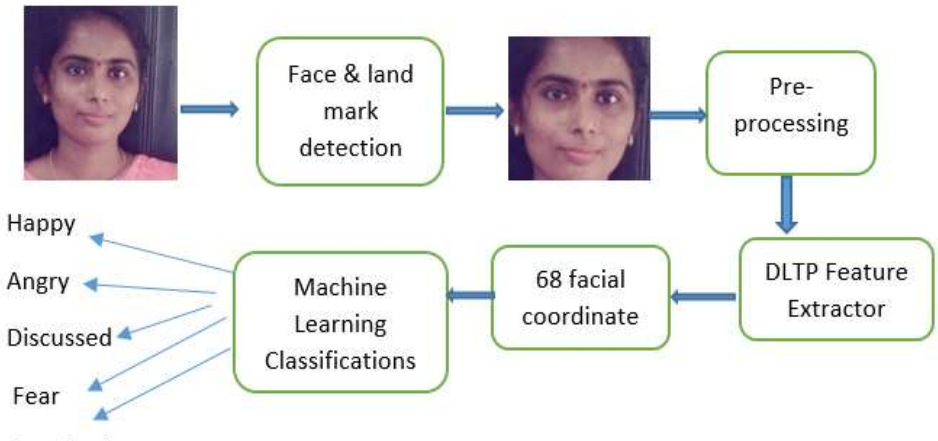


Fig10: Pipeline for the proposed facial smile emotions recognition system

Determining the pre-processing techniques to employ for Facial Expression Recognition (FER) hinges on a range of considerations, encompassing dataset attributes, the particular FER objective, and the computational resources at hand.

4.1 DLTP Feature Extraction

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 11, No. 1, November : 2023

Facial feature extraction in the designed FER pipeline involves the application of the Dynamic Local Ternary Pattern (DLTP) descriptor. Unlike the commonly used LTP descriptor, the DLTP descriptor employs an automated process to establish the threshold τ, drawing from Weber's law. Moreover, this descriptor adaptively adjusts the threshold according to pixel intensity values. Weber's law dictates that the alteration in a stimulus (such as lighting or sound) required for the change to be perceptible maintains a constant ratio with the initial signal. The variant of Weber's law integrated into DLTP is presented as (1).

$$\frac{\Delta I}{I} = \tau \qquad (1)$$

In equation (4), the term 1I represents the alteration in intensity I, while τ represents the consistent proportion. Within the context of DLTP, the term 1I becomes generalized as |In − Ic|, with I designated as Ic and In (where n = 1, 2... 8) representing neighbouring pixels. As a result, the formulation of Weber's law employed to autonomously ascertain the threshold can be mathematically articulated, as demonstrated in equation (5).

$$\frac{I_n - I_c}{I_c} = \tau \qquad (2)$$

Figure 8 illustrates the pattern encoding procedure utilizing the DLTP descriptor. An automatically determined threshold τ (using equation (2)) is applied around the center pixel value Ic of the 3 × 3 neighbouring pixels In (where n = 1, 2... 8), as depicted in Figure 9(b). Neighbouring pixels that fall within the range of Ic + τ and Ic - τ are quantized to 0, while those below Ic - τ are quantized to -1, and those above Ic + τ are quantized to 1, in line with equation (3). In this equation, SDLTP represents the quantized value of the neighbouring surroundings, as illustrated in Figure 3. Similarly, to LTP, the produced quantized value in DLTP is also segregated into negative patterns and positive patterns. The ensuing negative and positive binary patterns are then multiplied by predetermined weights and summed to yield the DLTP encoded lower and upper decimal values. Subsequently, the mean value of the lower and upper encoded values is calculated. The mathematical expressions employed to transform the upper and lower DLTP coded values into positive (upper) and negative (lower) decimal values are presented in equations (4) and (5), respectively.

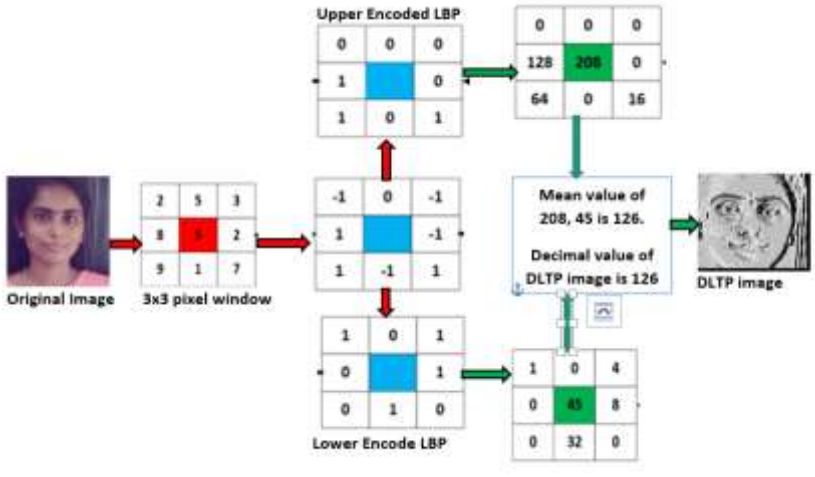


Fig 11: Systematic Representation of feature extraction scheme using DLTP descriptor

$$S_{DLTP}(I_c, I_n) = \begin{cases} -1, & if\ I_n < I_c - \tau \\ 0, & if\ I_c - \tau \le I_n \le I_c + \tau \\ +1 & if\ I_n > I_c + \tau \end{cases} \qquad (3)$$

$$P_{DLTP} = \sum_{i=0}^{7} (S_p(S_{DLTP}(i)) X 2^i \qquad (4)$$

Here

$$S_p(v) = \begin{cases} 1 & if\ v > u \\ 0 & otherwise \end{cases}$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 11, No. 1, November : 2023

$N_{DLTP} = \sum_{i=0}^{7} S_N(S_{DLTP}(i))X2^i$ (5)

here

$$S_N(v) = \begin{cases} 1 & if\ v < 0 \\ 0 & otherwise \end{cases}$$

Figure 8 delineates the process employed to capture textural details from a given facial image utilizing the DLTP descriptor. Starting with an input facial image, the procedure entails extracting the DLTP encoded positive (PDLTP) and negative (NDLTP) images by following the outlined sequence of steps. Subsequently, the feature extraction process segments these images into several m × n regions. Following this, local facial features are consolidated through histogram computations for each of these segments. Eventually, the DLTP-extracted positive and negative mean values are merged, resulting in the culmination of a high-dimensional facial feature.

$H_{DLTP}(\tau) = \sum_{r=1}^{m}\sum_{c=1}^{n}(P_{DLTP}(r,c),\tau)$ (6)

$H_{DLTP}(\tau) = \sum_{r=1}^{m}\sum_{c=1}^{n}(N_{DLTP}(r,c),\tau)$ (7)

Where

$$f(a,\tau) = \begin{cases} 1, & if\ a = \tau \\ 0 & otherwise \end{cases}$$

Equations (6) and (7) involve the parameters m and n, signifying the width and height of the encoded facial image region using DLTP and uDLTP, respectively. The value of τ spans from 0 to 58 for uDLTP and from 0 to 255 for DLTP. Extracted facial features via DLTP yield high-dimensional outcomes, with a considerable portion of these features proving redundant. Such elevated dimensionality impedes classifier performance and escalates computational requirements. Consequently, this study has employed Principal Component Analysis (PCA) for dimensionality reduction, effectively curtailing feature dimensions. Upon a closer examination of the scatter plot, the characteristics of the DLTP features become evident.

4.2 Face points recollection

By employing the code developed by Adrian Rosebrock, the detection of 68 facial landmarks along with their corresponding X and Y coordinates can be accomplished. A demonstration of these points is illustrated in below Figure 12.

Figure12: Localization in the human face of 68 facial coordinate points [26]

- 1-17 are points of the chin shape.
- 18-22 are points of the left eyebrow.
- 23-27 are points of the right eyebrow.
- 28-31 are points of the nose.
- 32-36 are points on the underside of the nose.
- 37-42 are points of the left eye.
- 43-48 are points of the right eye.
- 49-68 are points of the mouth.

The collected data has been organized into a CSV file, where each line corresponds to the X and Y coordinates of facial points within the dataset. Each row of this file corresponds to a distinct face within the dataset, while the columns represent X and Y coordinates for each facial landmark, structured as x0, y0, x1, y1, and so forth. Additionally, there exists another file indicating the specific emotion associated with each face. This file features the same number of rows as the previous one, but with only a single column since each face is linked to a single emotion.

V. Japanese Female Facial Expression [JAFFE] Dataset

For effective model training, it's crucial to possess datasets that exhibit clearly categorized emotions and a sufficient volume of data to optimize model performance. In this section, we introduce the datasets employed in the course of this study.

The Japanese Female Facial Expression (JAFFE) Database encompasses two hundred and thirteen images, each capturing seven distinct emotional expressions portrayed by ten Japanese female models [27]. This database was curated by the Psychology Department at Kyushu University. A glimpse of sample examples can be observed in Figure 13.

Figure 13: JAFFE example [27]

VI. Machine Learning Classifications

Machine learning is focused on exploring the recognition of patterns and facilitating computers to learn autonomously. It enables machines to acquire knowledge without the need for explicit programming. In this realm, algorithms play a pivotal role, deriving pertinent insights or conclusions from datasets, eliminating the necessity for human-generated instructions or code. The central objective of this field is to foster a collaborative interaction between humans and machines. This collaborative ethos is embodied by algorithms, which empower machines to perform tasks encompassing both general and specific domains. This learning process is executed through classifiers, algorithms that, upon receiving certain information about an object, can determine its category or class from a predefined set of possibilities.

To evaluate performance, four classifiers have been selected, representing prominent families of algorithms that are widely recognized

6.1 Support Vector Machine [SVM]

Support Vector Machines (SVMs) comprise a collection of supervised learning algorithms [28],[29] that are closely aligned with classification and regression challenges. In the context of a set of training examples, wherein classes are labelled, SVMs are harnessed to construct a model that can predict the class of a novel sample. The essence of an SVM lies in its capability to portray sample points in a spatial context, effectively partitioning classes into maximally spacious domains via a hyperplane. This hyperplane serves as the demarcation boundary. When new samples are introduced to the model, they are assigned to a specific class based on the partitioned regions they fall within. In simpler terms, this model represents sample points within a high-dimensional space, employing a hyperplane to demarcate classes. This approach is suitable for both classification and regression tasks. Effective separation between classes ensures accurate classification, as depicted in Figure 14.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 11, No. 1, November : 2023

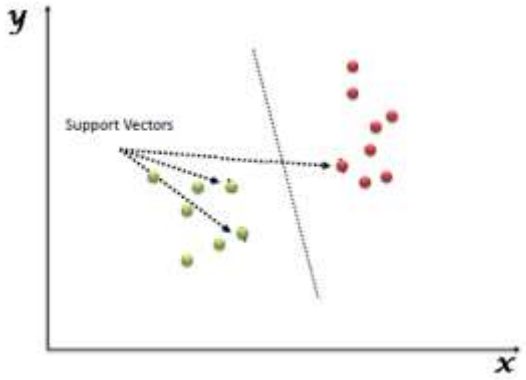


Fig14 : SVM feature selection1.

6.2 Decision Tree [DT]

Decision tree learning is a predictive modelling technique extensively employed in statistics, data mining, and machine learning [30],[31]. Within this framework, tree models tailored for scenarios where the target variable assumes a discrete set of values are referred to as classification trees. In such tree structures, class labels are situated in the leaves, while the branches encapsulate combinations of attributes that guide to those class labels. Conversely, when the target variable can assume continuous values, these tree models are termed regression trees. The primary goal of this classification technique is to construct a model capable of forecasting the outcome by mastering simple decision rules derived from data attributes.

6.3 Random Forest [RF]

The core concept behind random forests is to aggregate numerous models that might be noisy individually, yet exhibit a near-unbiased nature on average, thus mitigating variance. Trees are especially fitting candidates for bagging due to their capacity to encapsulate intricate interaction patterns within data, and if they reach sufficient depth, they maintain a comparatively low bias. As trees inherently exhibit noise, averaging them provides significant benefits [32],[33].

Each individual tree is crafted using the following procedure:

- Given N as the count of test cases and M as the number of variables in the classifier.
- Given m as the count of input variables utilized to determine a decision at a particular node; m should be significantly smaller than M.
- Select a training subset for this tree and employ the remaining test cases to estimate the error.
- For every node within the tree, randomly select m variables for decision-making. Compute the optimal partition of the training set based on the chosen m variables.

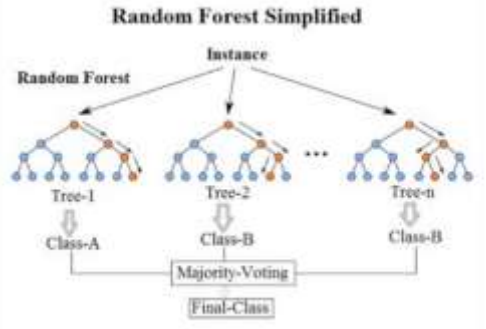


Figure 15: Scheme of RF performance2.

During prediction, a fresh case is traversed down the tree structure. It is eventually assigned the label of the terminal node it reaches. This sequence is repeated for all the trees in the ensemble, and the label that garners the highest occurrences is deemed the prediction. In essence, the estimator constructs multiple decision tree classifiers across various sub-samples of the dataset and leverages their averages to enhance predictive precision. A representation of this process's efficacy is depicted in Figure 11.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 11, No. 1, November : 2023

6.4 Experimental results

This chapter will showcase the outcomes of various experiments conducted. However, it's crucial to highlight that almost all of the images feature posed expressions, with only the FER database containing spontaneous expressions. Recognizing spontaneous expressions poses a greater challenge. As a result, the accuracy results presented here would likely fare worse in a real-world application of these learning architectures.

Metrics for evaluation

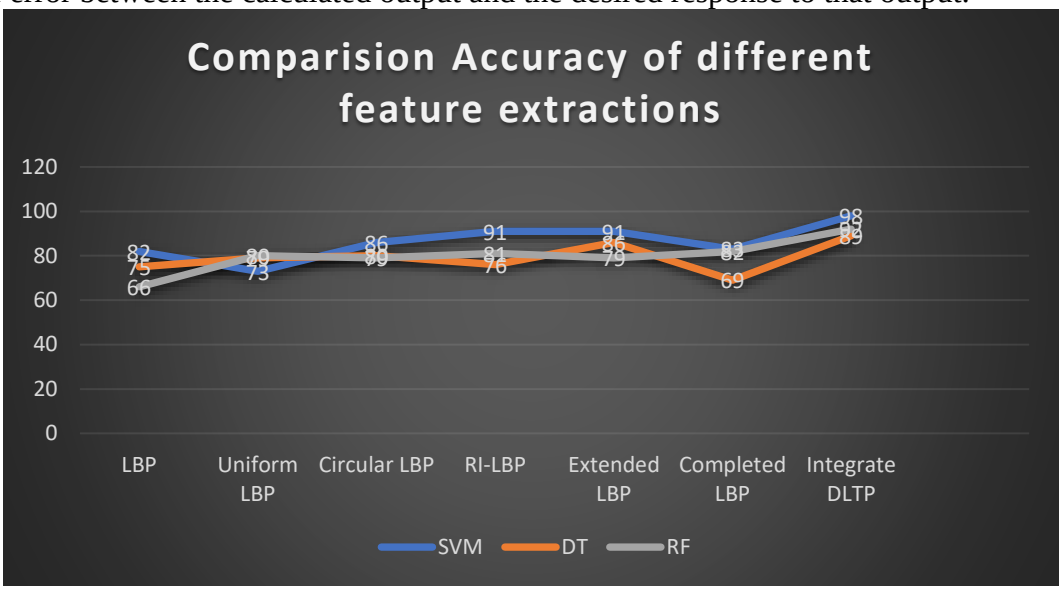
To assess the effectiveness of the learning algorithms, a diverse set of metrics is employed. Among these metrics, the confusion matrices and classification accuracy of our system stand out. These two evaluation methods have been selected due to their widespread usage by other researchers.

Accuracy: A parameter that quantifies the success of a model's predictions.

$$Accuracy = \frac{Correctly\ dected\ emotions}{Total\ emotions} * 100$$

Confusion matrix: A tool that provides insight into the distribution of predictions, indicating successful and incorrect predictions. The matrix values are presented as percentages.

Losses: A parameter that highlights the samples lost due to incorrect predictions. It quantifies the degree of error between the calculated output and the desired response to that output.



Graph 1 : Comparison of different feature extractions Accuracy using SVM,DT,RF

DLTP with 68 feature points feature extraction	ML Classifications	Accuracy (%)
LBP	SVM	82
	DT	75
	RF	66
Uniform LBP	SVM	73
	DT	79
	RF	80
Circular LBP	SVM	86
	DT	80
	RF	76
RI-LBP	SVM	91
	DT	76
	RF	81
Extended LBP	SVM	91

	DT	86
	RF	79
Completed LBP	SVM	83
	DT	69
	RF	82
Proposed LBP+68 Feature Points	SVM	98
	DT	89
	RF	92

Table 1: Comparisons for Different Feature Extraction methods

ML Classifications	Accuracy (%)	Time(in sec)
SVM	98	7
DT	89	15
RF	92	10

Table 2: Accuracy summary for machine learning on JAFFE dataset

Evidently, a substantial disparity exists in the effectiveness of the algorithms. Support Vector Machines and Multilayer Perceptron achieve higher accuracy at around 98% and 89%, respectively, while Decision Tree and Random Forest achieve approximately 92% accuracy. These results stem from the fact that SVM and RF are more intricate algorithms, capable of considering the complexities within the features. There are shared characteristics among all these algorithms. They excel in recognizing happiness emotions, but struggle with identifying sadness, often confusing it with neutral or anger emotions. This observation is supported by examining the confusion matrices depicting the performance of these three algorithms on the JAFFE database.

Emotions	Afraid	Angry	Disg.	Happy	Neutr.	Sad	Surpr.
Afraid	89	2	0	1	0	3	5
Angry	4	90	1	0	5	3	0
Disg.	0	2	92	0	6	0	0
Happy	0	0	2	88	4	0	5
Neutr.	2	0	4	0	86	2	4
Sad	1	2	0	3	0	90	4
Surpr.	2	0	3	0	2	0	93

Table3: Accuracy for SVM with JAFFE Dataset image.

Emotions	Afraid	Angry	Disg.	Happy	Neutr.	Sad	Surpr.
Afraid	90	2	0	2	2	0	4
Angry	0	89	4	2	4	0	1
Disg.	1	0	95	0	0	2	2
Happy	3	0	2	80	0	10	5
Neutr.	0	2	4	0	82	8	4
Sad	1	0	0	8	0	91	0
Surpr.	0	0	10	4	0	0	86

Table4: Accuracy for DT with JAFFE Dataset images

Emotions	Afraid	Angry	Disg.	Happy	Neutr.	Sad	Surpr.
Afraid	80	6	0	4	6	0	4
Angry	0	85	4	6	0	0	5
Disg.	10	0	75	5	0	10	0
Happy	3	0	7	80	0	5	5
Neutr.	1	0	4	0	81	8	4
Sad	1	0	0	9	0	90	0
Surpr.	0	0	9	4	0	1	86

Table5: Accuracy for RF with JAFFE Dataset images.

VII. Conclusion

In conclusion, this study explored Facial Expression Recognition (FER) using diverse machine learning algorithms. The research spanned preprocessing, feature extraction, dimensionality reduction, and classification, aiming to decode human emotions from facial images. Different Local Binary Pattern (LBP) variants, including DLTP, were applied to capture textural data effectively. The proposed methodology involves the extraction of DLTP feature points, complemented by 68 distinct features encompassing geometric, statistical, and texture-based attributes for high accuracy with less complex time. This combined feature set captures both local and holistic information present within the emotion-related facial expressions. These techniques demonstrated meaningful feature extraction for emotional portrayal. Evaluation revealed varying performance trends. SVM and Multilayer Perceptron excelled in complex feature understanding, while Decision Tree and Random Forest exhibited commendable but slightly lower accuracy. Recognizing emotions displayed varying proficiency, with happiness evident and sadness often confused with neutral or anger emotions. While accuracy progress was achieved, challenges persist, particularly with spontaneous expressions. The evolving synergy between machine learning and datasets propels FER advancement. As technology evolves, decoding emotions from facial images offers promising applications in affective computing, human-computer interaction, and more.

References:

[1] Ahonen, T.; Hadid, A.; Pietikainen, M. Face Recognition with Local Binary Patterns; Springer: Berlin/Heidelberg, Germany, 2004.
[2] Mau, S.; Dadgostar, F.; Lovell, B.C. Gaussian Probabilistic Confidence Score for Biometric Applications. In Proceedings of the 2012 International Conference Digital Image Computing Techniques and Applications (DICTA), Fremantle, WA, Australia, 3–5 December 2012 .
[3] Jones, M.J. Face Recognition: WhereWe Are and Where to Go from Here. IEE J. Trans. Elect. Inf. Syst. 2009, 129, 770–777.
[4] Abate, A.F.; Nappi, M.; Ricciardi, S.; Sabatino, G. One to Many 3D Face Recognition Enhanced Through k-d-Tree Based Spatial Access; Candan, K.S., Celentano, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 5–16.
[5] Phillips, P.J.; Moon, H.; Rizvi, S.A.; Rauss, P.J. The FERET Evaluation Methodology for Face-Recognition Algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 2000, 22, 1090–1104.
[6] Rahim, A.; Hossain, N.; Wahid, T.; Azam, S. Face Recognition using Local Binary Patterns (LBP). Glob. J. Comput. Sci. Technol. 2013, 13, 469–481.
[7] Manvjeet Kaur, D. K-Nearest Neighbor Classification Approach for Face and Fingerprint at Feature Level Fusion. Int. J. omput. Appl. 2012, 60, 13–17.

[8] Ebrahimpour, H.; Kouzani, A. Face Recognition Using Bagging KNN. In Proceedings of the International Conference on Signal Processing and Communication Systems (ICSPCS'2007), Gold Coast, Australia, 17–19 December 2007.
[9] M. Rosenblum, Y. Yacoob, and L. S. Davis, "Human Expression Recognition from Motion using a Radial Basis Function Network Architecture," IEEE Transactions on Neural Networks, Vol.7, No.5, pp.1121-1138, 1996.
[10] F. Bourel, C. Chibelushi, and A. Low, "Recognition of FacialExpressions in the Presence of Occlusion", Proceedings ofthe12th British Machine Vision Conference, Vol. 1, pp. 213– 222,2001.
[11] B. Abboud and F. Davoine, "Appearance Factorization for Facial Expression Recognition and Synthesis," Proceedings of International
Conference on Pattern Recognition, pp.163–166, 2004.
[12] Manja Pantic and Leon J.M. Rothkrantz,"Automatic Facial Expressions : The state of the Art", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol 22, No 12 December 2000.
[13] Vapnik V., "Statistical Learning Theory", Wiley, New York, 1998.
[14] Vapnik V. and Chervonenkis A., "On the uniform convergence of relative frequencies of events to their probabilities", in "Th. Prob. and its Applications", 17(2): 264--280, 1971.
[15] Tefas A., Kotropoulos C., and Pitas I., "Enhancing the performance of elastic graph matching for face authentications by using Support Vector Machines", ACAI99.
[16] Breiman, L. Consistency for a simple model of random forests. Technical report, University of California at Berkeley, 2004.
[17] Breiman, L., Friedman, J., Stone, C., and Olshen, R. Classification and Regression Trees. CRC Press LLC, 1984.
[18] Breiman, L. Bagging predictors. Machine Learning, 24(2):123– 140, 1996.
[19] Dietterich, T. G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning, pp. 139–157, 2000.
[20]. Amit, Y. and Geman, D. Shape quantization and recognition with randomized trees. Neural Computation, 9:1545–1558, 1997.
[21]. T. Ojala, M. Pietikainen and D. Harwood, "A comparative study of texture measures with classification based on feature distributions" Pattern Recognition vol. 29, 1996.
[22]. Ignace TCHANGOU TOUDJEU, Jules-Raymond TAPAMO," Circular Derivative Local Binary Pattern Feature Description for Facial Expression Recognition" in Advances in Electrical and Computer Engineering Volume 19, Number 1, 2019
[23] Yu Wang1*, Yongsheng Zhao2 and Yi Chen, Wang "Texture classification using rotation invariant models on integrated local binary pattern and Zernike moments" EURASIP Journal on Advances in Signal Processing 2014, 2014:182 .
[24]. Li Liu 1 , Paul Fieguth 2 and Gangyao Kuang " Generalized Local Binary Patterns for Texture Classification" in Image and Vision Computing · February 2012 .
[25]Completed Modeling of Local Binary Pattern Operator for Texture Classification Article in IEEE Transactions on Image Processing · March 2010.
[26].Adrian Rosebrock. Face alignment with opencv and python, 2017. https://www.pyimagesearch.com/2017/05/22/face-alignment-with-opencv-and-python/.
[27].Michael J Lyons, Shigeru Akamatsu, Miyuki Kamachi, Jiro Gyoba, and Julien Budynek. The Japanese female facial expression (JAFFE) database. In Proceedings of third international conference on automatic face and gesture recognition, pages 14–16, 1998.
[28].Evgeniou, Theodoros & Pontil, Massimiliano. Support Vector Machines: Theory and Applications. In 2049. 249-257. 10.1007/3-540-44673-7_12 (2001).
[29].Dakhaz Mustafa Abdullah , Adnan Mohsin Abdulazeez ,"Machine Learning Applications based on SVM Classification: A Review" in Qubhan Academic journal, 81-90.

[30].Ronny Kohavi, Ronny Kohavi. Data mining tasks and methods: Classification: decision-tree discovery Handbook of data mining and knowledge discovery Pages 267-276. Oxford University Press, Inc. New York, NY, USA c 2002.
[31].Chi-Chun Lee a,⇑ , Emily Mower a , Carlos Busso b , Sungbok Lee a , Shrikanth Narayanana. "Emotion recognition using a hierarchical binary decision tree approach", in Speech Communication · January 2011.
[32].M Denil, D Matheson, N De Freitas. Narrowing the Gap: Random ForestsIn Theory and In Practice in Proceedings of The 31st International Conference on Machine Learning, pp. 665–673.(2014).
[33].Kamlesh Tiwari & Mayank Patel "Facial Expression Recognition Using Random Forest Classifier" in International Conference on Artificial Intelligence: Advances and Applications 2019 .