Scientific African 8 (2020) e00465

Contents lists available at ScienceDirect

Scientific African

journal homepage: www.elsevier.com/locate/sciaf

Facial micro-expression recognition: A machine learning approach

Iyanu Pelumi Adegun[a,b,c,*], Hima Bindu Vadapalli[c]

[a] Department of Computer Science, Federal University of Technology Akure, Nigeria
[b] Department of Computer Science, Rufus Giwa Polytechnic, Owo, Nigeria
[c] School of Computer Science, University of the Witwatersrand, Braamfontein 2000, Johannesburg, South Africa

ARTICLE INFO

Article history:
Received 29 February 2020
Revised 16 June 2020
Accepted 26 June 2020

Keywords:
Micro-expressions
Feature extraction
Support vector machine
Emotions
Extreme learning machine

ABSTRACT

Micro-expression recognition is a growing research area owing to its application in revealing subtle intention of humans, especially while under high stake conditions. With the rapid increase in security issues all over the world, the use of micro-expressions to understand one's state of mind has received major interest. Micro-expressions are characterized by short duration and low intensity, hence, efforts to train humans in recognizing them have resulted in very low performances. Automatic recognition of micro-expressions using machine learning techniques thus promises a more effective result and saves time and resources. In this study, we explore the use of Extreme Learning Machine (ELM) for micro-expression recognition because of its fast learning ability and higher performance when compared with other models. Support Vector Machine (SVM) is used as a baseline model and its recognition performance and its training time compared with ELM training time. Feature extraction is performed on apex micro-expression frames using Local Binary Pattern (LBP) and on micro-expression videos divided into image sequences using a spatiotemporal feature extraction technique called Local Binary Pattern on Three Orthogonal Planes (LBP-TOP). Evaluation of the two models is performed on spontaneous facial micro-expression samples acquired from Chinese Academy of Sciences (CASME II). Results obtained from the experiments show that ELM produces a higher recognition performance than SVM in terms of accuracy, precision, recall and F-score when temporal features are used. Comparison between SVM and ELM training time also shows that ELM learns faster than SVM. An average training time of 0.3405 seconds is achieved for SVM while an average training time of 0.0409 seconds is achieved for ELM for the five selected micro-expression classes. This study shows that automatic recognition of micro-expressions is produces a better result when temporal features and a machine learning algorithm with fast learning speed are used.

[*] Corresponding author.
E-mail addresses: iyanupelumi22@gmail.com (I.P. Adegun), hima.vadapalli@wits.ac.za (H.B. Vadapalli).

https://doi.org/10.1016/j.sciaf.2020.e00465

Introduction

Expressions reveal what takes place in the human mind at a time. These are often displayed through speech, body gestures or facial expressions. Of all the existing modes of expression, facial expression appears to be the most expressive means through which humans show their emotions [4]. The vital role of facial expressions in day to day life of humans has made several researchers to develop automated systems for their recognition and interpretation. For instance, Vural et al. [21] proposed a system that identifies the level of drowsiness in drivers. Another example of facial expression recognition system is that of Whitehill et al. [25] which was used to get response/feedback from students while being taught.

Facial expressions can be either categorized as “macro-expressions” or “micro-expressions". Macro-expressions are the normal expressions that are seen in our day to day interaction with people and last between 1/2 seconds and 4 seconds. However, in high stake situations, people often conceal or suppress their true emotions because of the fear of being caught [20]. These concealed emotions take place within the duration of 1/5 and 1/25 seconds and are known as “micro-expressions”. Micro-expressions were initially detected by Haggard and Isaacs [10] and later by Ekman and Friesen [6]. Aside from the short duration of micro-expressions, they also possess low intensity. These two unique features make the recognition of micro-expressions by humans more difficult even when they are trained to perform such tasks [6,27].

Humans can be specially trained to recognize micro-expressions, but it often yields very low performances [7]. The limited ability of humans to recognize micro-expressions effectively makes it necessary to develop automated systems that can automatically recognize them. Automatic micro-expression recognition involves the use of computer-based methods for recognition of micro-expressions. Both macro-expressions and micro-expressions can be expressed in seven different forms which include sadness, happiness, fear, anger, surprise, disgust and contempt.

Machine learning algorithms have been used in previous studies for various classification problems, however only a few studies have been carried out in this area. In the existing studies, supervised machine learning algorithms were used which includes decision trees, neural networks, k nearest neighbors (KNN), Support Vector Machine (SVM), ELM and random forest classifiers. SVM [5] has been widely used for both macro- and micro-expression recognition and for other classification tasks while ELM has not been used by many for the same purpose. Support Vector Machine appears to be the mostly used machine learning model for facial expression recognition because of its good generalization performance irrespective of bias in training sample [2]. However, SVMs have some major drawbacks which include their complexity and slow learning speed [3].

In this study, SVM is used as the baseline recognition model, but because of its slow learning speed, ELM which has a faster learning speed is also explored for micro-expression recognition. Features are extracted from apex micro-expression frames using Local Binary Patterns (LBP) and extracted from the entire micro-expression videos using (LBP-TOP). The performance of the two models (SVM and ELM) are compared on both static and temporal features and their overall training time was compared. The models were evaluated using CASME II micro-expression database [28].

Related works

Micro-expression can either be recognized from static facial images or temporal facial images (video sequences). Many of the past studies conducted their micro-expression recognition on temporal data while a few used static data for recognition. Wu et al. [26]; Liong et al. [12] and Liong et al. [13] evaluated the performance of their models using apex micro-expression images and static feature extraction techniques with low recognition results. Some other studies (Yan et al. [28]; Guo et al. [8]; Guo et al. [9] and Adegun & Vadapalli [1]) have used temporal data (image sequences/videos) in their work and were able to achieve good result.

Wu et al. [26] proposed a means of recognizing micro-expressions using a frame by frame approach. Gabor filter [17] was used for feature extraction from facial image frames while Gentle-SVM (a combination of Gentle boost and SVM algorithm) was used for classification. Their system recorded an accuracy of 85.42% for recognition. However, there is the need to increase the training set in order to achieve better results. Yan et al. [27] performed a baseline evaluation of CASME II database using LBPTOP for feature extraction and SVM for classification. SVM classification results revealed that the highest average accuracy obtained was 63.41% at LBP-TOP radii values of 1, 1 and 4. Wang et al. [23] used discriminant tensor subspace analysis (DTSA) to extract features and Extreme Learning Machine (ELM) for classification of micro-expressions. In their work, ELM models were built via ten-fold cross validation to ensure that all test sets were independent. Guo et al. [9] also combined centralized binary pattern on three orthogonal planes (CBP-TOP) with ELM as classifier to obtain a better classification result. Jia et al. [11] proposed a macro to micro-expression transformation model. In their work, they used LBP and LBP-TOP for macro-expression and micro-expression for recognition respectively. They evaluated their work on CASME II database and had a better result than earlier studies. Wang et al. [24] proposed a micro-expression recognition approach that was based Eulerian Motion Magnification (EVM) using CASME II database for evaluation. This approach helps in revealing subtle changes and hidden information in micro-expressions. It was also found to be effective on various LBP-TOP spatial block partitions and neighborhood sizes during micro-expression recognition. Zhang et al. [29] developed a discriminative feature descriptor that are less sensitive to variants in pose and illumination for a better micro-expression recognition. They used face alignment algorithm to locate facial points in each video frame. The faces were then divided into several regions based on the facial points. They used LBP-TOP for feature extraction and random forest for classification on CASME II database.

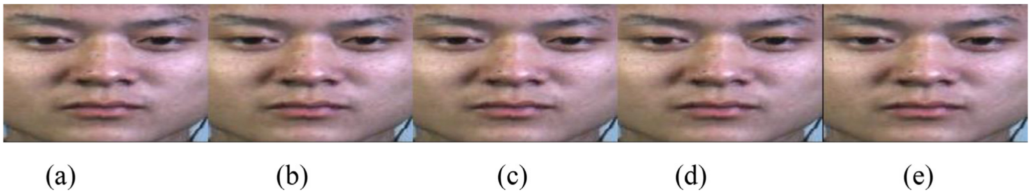


Fig. 1. CASME II Happiness frame sequence showing onset, offset and apex frames: (a) onset frame; (b) random frame between onset and apex frame; (c) apex frame; (d) random frame between apex and offset frame and (e) offset frame [28].

In many of these studies reviewed, images were divided into blocks before applying LBP-TOP feature extraction on each block which resulted in an accuracy of 63.41%. In this study, we used the micro-expression in a holistic manner which means that images were not divided into blocks before extraction of features to reduce computational complexity as expressed by Guo et al. [9].

Materials and methods

The process of micro-expression recognition can be divided into three major phases. These include data collection/preparation, feature extraction and classification.

Dataset description and preparation

Micro-expression datasets can be categorized into acted or spontaneous samples. Acted micro-expression samples are those elicited by asking subjects to act out micro-expression after a careful explanation of what such expressions entails. For spontaneous samples, subjects' emotions are stimulated by real-time emotional occurrence. Micro-expression samples elicited spontaneously gives a true picture of what it really is when compared with the acted samples [27]. Existing micro-expression database samples that were acted include USFHD database [28] and Polikovsky's database (Polikovsky et al., 2014) while spontaneous ones include Spontaneous Micro-expression database (SMIC) [15], Chinese Academy of Sciences Micro-expression database (CASME) and its updated version, CASME II. Of these three spontaneous databases, we considered CASME II in this study since it is the most recent, up-to-date and publicly available for research.

Chinese Academy of Sciences Micro-expression (CASME II) database [28] is an improved version of CASME database. The dataset consists of 247 facial micro-expression samples retrieved from 26 participants. These samples were retrieved from 18 participants who were made to watch highly emotional video clips. While watching the videos, a screen was placed before each of the participants and a high-resolution camera was used to record their emotions. After watching the clips, participants were told to rate the intensity of the video clips using a 7-point Lickert scale with 0 as the lowest and 6 as the highest. Video recordings from each participant were divided into frames and pre-processed by removing facial and body movements that were not regular. Final micro-expression samples were selected based on recordings that had a total duration of less than 500 milliseconds or an onset duration of less than 250 milliseconds. These samples were labelled using Facial Action Coding System (FACS) but their labeling criteria are not the same with that of ordinary facial expressions. The micro-expression samples consist of seven (5) classes which includes disgust, happiness, surprise, repression and others.

Micro-expression samples from CASME II were already pre-processed before being made available to the public. Some of the pre-processing carried out by the database owners include face detection and registration, division of recorded videos into frames, labeling of samples with relevant action units and coding with onset, apex and offset frames. Onset frame is the first frame where changes from the neutral expression occurs. Apex frame is the frame where the highest intensity of the expression is reached while offset frame is the last frame before facial expression changes to neutral CASME II database has a total of 247 micro-expression samples with a sampling rate of 200fps. One of the samples from CASME II database is shown in Fig. 1.

A total of 230 samples were used for the experiment involving the use of temporal data (image sequence) while 220 samples were used for the experiments involving the use of static data (apex micro-expression frames). The variation in the number of samples used for the two set of experiments is as a result of some samples whose coding did not include their correct apex, onset and offset labels. The samples without correct labels were left out, hence, we had 220 apex frame samples. Details on the number of samples for each micro-expression class are presented in Table 1. Pre-processing involves conversion of frames from RGB into grey-scale images.

Feature extraction

The process of extracting relevant features from data is critical to micro-expression recognition. To recognize micro-expressions accurately and effectively, features (both static and temporal) were extracted from both static and temporal micro-expression samples. These features were extracted using a holistic approach whereby spatio-temporal features are extracted from whole facial images instead of blocks of facial regions. This approach was used so as to reduce information redundancy and avoid complexity during feature extraction as expressed in Guo et al., [8]

Table 1
Number of samples for each micro-expression class in CASME II and number of selected apex frames and image sequences.

Class	Original CASME II data	No. of selected image sequences	No. of selected apex frames
Disgust	60	59	57
Happiness	33	30	25
Repression	27	24	27
Surprise	25	25	23
Others	102	92	88
Total Samples	247	230	220

Feature extraction from apex micro-expression frames (static data) using LBP

Local Binary Patterns (LBP) was proposed by Ojala et al. [18] and their intention was to use it as a means of describing 2D textures of static images. The main idea of LBP is to compare the value of the center pixel C of an image with the value of its neighboring pixel P. If the center pixel value is greater than the neighboring pixel value, then, 0 is assigned, otherwise, 1 is assigned. This results into an 8-digit binary number comprising of 0s and 1s which is converted to decimal number and serves as the LBP value of the center pixel.

Local Binary Pattern is described mathematically as follows:

Given a pixel p with intensity value v_p, radius r and N neighboring pixels, a binary label is assigned to each of the neighboring pixels. If the intensity value of the given pixel is greater than that of the center pixel, then 1 is assigned, otherwise, 0 is assigned to the pixel. LBP value is then given as:

$$LBP_{x_p,\ y_p} = \sum_{i=0}^{N-1} f(g_i - \ g_p)\ 2^i \tag{1}$$

where x_p, y_p represents the co-ordinates of the center pixel, g_p denotes the intensity value of the center pixel and g_i is the intensity value of the i^{th} neighboring pixel. 2^i is the weight that corresponds to the neighboring pixel locations and f(x) is a sign function defined as $f(x) = \{^{1,\ x\geq0}_{0,\ x<0}.\}$

The feature vector can then be derived by calculating the histogram of all the LBPs as described in (2).

$$H_i = \sum_{x=0}\sum_{y=0} I\ \{LBP\ (x, y) = i\},\ \ (i = 0,\ ..\ .., n-1) \tag{2}$$

where n represents the total number of labels produced and $I(X) = \{^{1,\ \text{if X is true}}_{0,\ \text{if X is false}}.\}$How Feature Extraction was performed with LBP

Feature extraction was performed using LBP technique on apex micro-expression frames. As presented in Table 1 above, a total of 220 samples that includes 57 disgust samples, 25 happiness samples, 27 repression samples, 25 surprise samples and 88 others samples were used for this experiment. Selection of these samples was performed based on the labeling provided by the database owners. For each grey-scale image, LBP values were obtained by comparing the center pixel values with all the neighboring pixel values resulting in an 8-digit binary number converted into decimal. Thereafter, the histograms of these LBP values were calculated with intensity levels on y coordinates and pixel values (from 1 to 256) on x co-ordinates which resulted in a feature vector size of 1 X 256 for each sample.

Feature extraction from image sequence (temporal data) using LBP-TOP

Local Binary Patterns on Three Orthogonal Planes was selected as a technique for feature extraction in this study because of its ability to extract temporal features from micro-expression samples. LBP-TOP [30] is one of the spatio-temporal descriptors for dynamic textures (textures in motion) which was created in order to overcome the drawbacks of ordinary LBP. The major drawback of LBP is that it can only extract features from still images. LBP-TOP is one of the variants of ordinary LBPs and was proposed by Zhao & Pietikainen [30]. It was proposed as a result of the need to analyze textures that are time-dependent (i.e. videos).

For a given video with time length T, LBP value is calculated in three planes; XY, XT and Y T, where XY provides spatial information while XT and YT supplies both spatial and temporal information about space-time transitions [30]. The LBP value for each of the planes is calculated using Eq. (3) and is concatenated into a single histogram which serves as the final feature vector.

The corresponding LBP-TOP feature is given as $LBP\ --TOP_{\ PXY,\ PXT,\ PYT,\ RXY,\ RXT,\ RYT}$ where P represents the number of neighboring points and R represents the radius, while the histogram is described in Eq. (3).

$$H(i) = \ \sum \left\{f_j\ (x, y, t) = i\right\},\ \ i = 0,\ \ .\ .\ ., n_j - 1; j = 0, 1, 2. \tag{3}$$

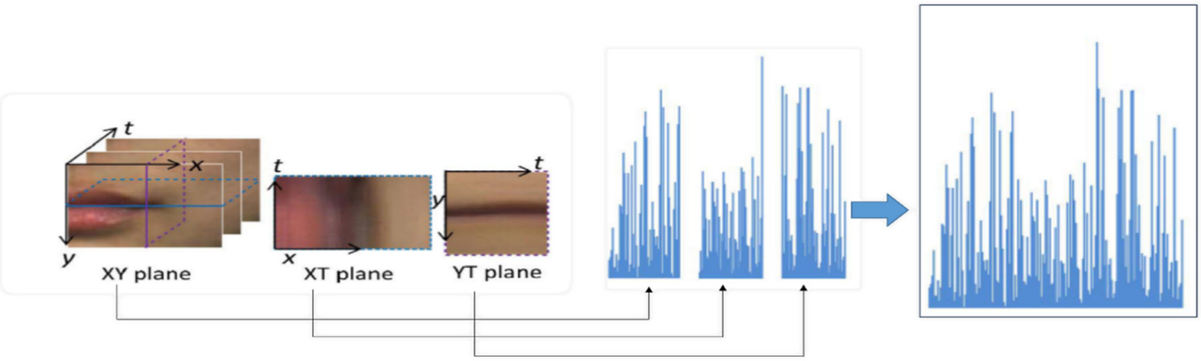


Fig. 2. How LBP-TOP features are extracted for a CASME II micro-expression sample, showing XY, XT and YT planes with their corresponding histogram (Source: Adopted from [28]).

Table 2
Positive and negative samples with their given labels in parenthesis.

Positive samples	Negative samples			
Disgust (1)	Happiness (0)	Repression (0)	Surprise (0)	Others (0)
Happiness (1)	Disgust (0)	Repression (0)	Surprise (0)	Others (0)
Repression (1)	Happiness (0)	Disgust (0)	Surprise (0)	Others (0)
Surprise (1)	Happiness (0)	Disgust (0)	Disgust (0)	Others (0)
Others (1)	Happiness (0)	Disgust (0)	Repression (0)	Surprise (0)

where n_j represents the number of interest patterns in the j^{th} plane and $f_j(x, y, t)$ represents the LBP code at pixel positions (x, y, t) along the j^{th} plane. The final histogram is normalized using Eq. (4).

$$N_{i,j} = \frac{H_{i,j}}{\sum_{n_j-1}^{k=0} H_{k,j}} \tag{4}$$

To extract LBP-TOP features, grey-scale image sequences were first read (i.e., micro-expression videos readily converted into frames of varying lengths). Thereafter, LBP-TOP features were extracted. In this experiment, radii values at x, y planes were varied between 1 and 4 while for t plane radii values were varied between 2 and 4. Number of neighboring points for Y, XT and YT planes were set to 8 which gave a total of 2^8 patterns (i.e. 256 patterns) for each of the samples.

For each image sequence, LBP values were calculated in Y, XT and YT planes along with their respective histograms as seen in Fig. 2. The histogram derived for each of the three planes were concatenated which produced a 3 × 256 feature vector for each image sequence. To reduce the length of the feature vector, uniform binning was applied to extract only uniform patterns from the 256 histogram bins. Application of uniform patterns reduced the length of the feature vector from 256 to 59. This produced a 3 × 59 feature vector for each image sequence which was converted into a single row vector of size 1 × 177. Therefore, for all 230 samples used for these experiments, we had 230 × 177 features.

Classification (micro-expression recognition)

Five micro-expression classes were used for all experiments (disgust, happiness, repression, surprise and others) via a “one vs all” multi-classification approach. Training was performed using all samples belonging to each class as positive samples labeled as 1 while samples from the remaining four (4) classes were used as negative samples labelled as 0. The description of the labels is given in Table 2.

Training with Support Vector Machine Model

Selection of SVM as baseline model was motivated by its success in past micro-expression studies like [8,19,28]. SVM uses a hyperplane to separate the group of data into their appropriate classes, considering that we have two a dataset with two separate classes that are linearly inseparable. There could be more than a single hyperplane separating the classes and the one with the largest margin is chosen as the best/most correctly classified. Since SVM is a binary classifier, each micro-expression class was trained separately, and the average of their performance was calculated.

In this study, five-fold cross validation was used to divide all the samples into five subsets. SVM models were built using the two data formats that we had (apex frames and image sequences). A total of 220 samples were used for apex frame experiment while 230 samples were used for experiments performed using image sequences. Details on training that was performed and how parameters were optimized are presented in the next sections.

Table 3
Training accuracy of SVM using apex on LBP features.

Micro-expression class	Training accuracy (%)
Disgust	94.99
Happiness	97.62
Repression	97.62
Surprise	98.12
Others	93.16
Average	96.30

Table 4
Training accuracy records at varying LBP-TOP radii values using SVM model. (Accuracy in %).

$\mathbf{R_x}$, $\mathbf{R_y}$, $\mathbf{R_t}$	Surprise	Disgust	Happiness	Others	Repression	Average
1, 1, 2	95.22	86.96	87.83	91.30	97.39	91.74
1, 1, 3	89.15	91.95	93.80	97.61	97.50	94.00
2, 2, 2	88.70	93.04	95.65	93.04	88.70	91.82
2, 2, 3	88.26	89.57	96.52	93.92	95.65	92.78
2, 2, 4	88.70	89.57	97.83	93.48	95.65	93.05
3, 3, 2	86.09	88.70	97.39	85.65	89.13	89.39
3, 3, 3	86.09	88.69	97.39	85.22	90.44	89.57
3, 3, 4	86.52	89.13	96.96	88.70	87.82	89.83
4, 4, 2	85.22	87.83	96.96	86.52	87.83	88.87
4, 4, 3	83.04	83.05	85.22	92.61	94.78	87.74
4, 4, 4	83.91	85.22	85.65	94.78	94.78	88.86

SVM training with LBP features

Training SVM on apex micro-expression frames was performed by loading (1 X 177) feature vector for 220 apex samples acquired after extraction of LBP features. Thereafter, five-fold cross validation was performed to divide the samples into five independent subsets. Linear SVM kernel was used for training. An average training accuracy of 96.30% was achieved with LBP features obtained from apex frames as presented in Table 3.

SVM training with LBP-TOP features

Similar procedure used for apex frame experiments was followed to train SVM on micro-expression image sequences. Feature vector of size 1 × 177 for 230 image sequences acquired from LBP-TOP feature extraction were loaded into each classifier. Optimization of the models was performed by recording training accuracy at varying LBP-TOP radii values in x, y and t planes. These values were varied between 1 and 4 for x and y planes while variation for t plane was between 2 and 4. Training result shows that the highest average training accuracy (94.00%) was achieved at $R_x = 1$ $R_y = 1$ and $R_t = 3$ as presented in Table 4. Training results as presented reveals that there is no more increase in average training accuracy as from radii values $R_x = 3$ $R_y = 3$ and $R_t = 2$ to $R_x = 4$ $R_y = 4$ and $R_t = 4$.

Training with extreme learning machine model

Extreme Learning Machine is a learning algorithm for the single hidden layer feedforward neural networks (SLFN) proposed by Huang et al., [14]. This learning algorithm was proposed to overcome the drawbacks of traditional feed-forward neural networks. According to Huang et al., [14], one of the major drawbacks of traditional feed-forward neural networks is their slow learning speed. Some of the advantages of ELM over other traditional learning algorithms of SLFN are highlighted below:

- ELM does not require parameter tuning
- ELM has an extremely fast learning speed as compared with other learning algorithms such as back propagation (BP) algorithm
- ELM is very useful in training SLFNs with many non-differentiable activation functions. [22,23]

According to Wang et al. [23], the most superior and impressive of these features is the fast training speed compared to other traditional learning algorithms. The mathematical model for ELM is described below:

For N distinct samples x_i, t_i where $X_i = (X_{i1}, X_{i1}, \ldots X_{in})^T \varepsilon R^n$ and $t_i = (t_{il}, t, \ldots, t_{in})^T \varepsilon R^m$, SLFN can be modeled using Eq. (5).

$$\sum_{i=1}^{\tilde{N}} \beta_i g\left(w_i.x_j + b_i\right) = P_j,\ j = 1,\ .\ .\ .,\ N \tag{5}$$

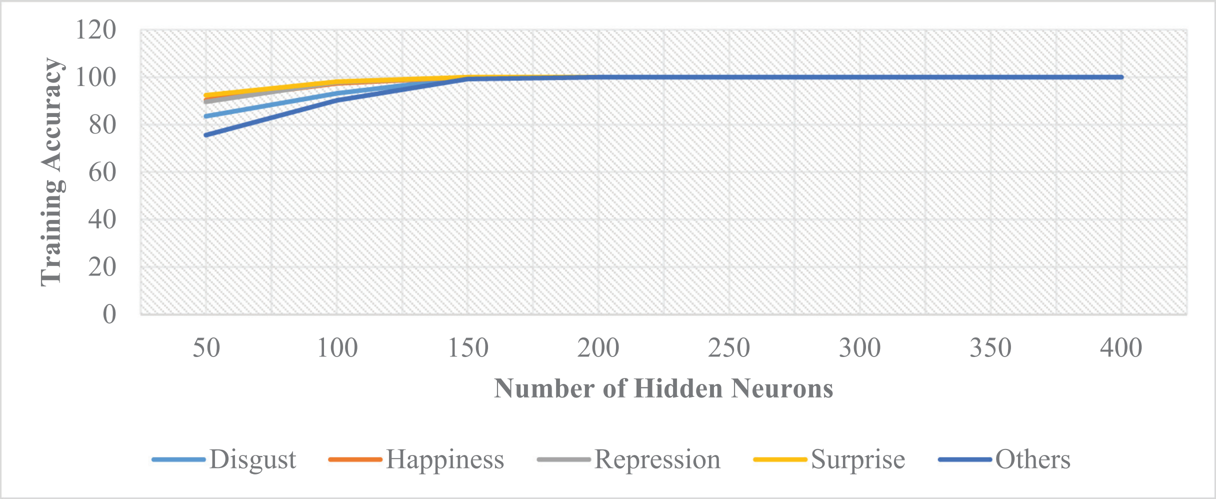


Fig. 3. Optimization of number of hidden neurons using LBP features.

where $g(x)$ represents the activation function, $\tilde{N}$ represents the number of hidden neurons, the weight vector that connects the i^{th} hidden node and the input nodes are represented by $w_i = (w_{i1}, w_{i2}, \ldots, w_{im})$ and b_i represents the threshold of the hidden node. $\beta_i = (\beta_{i1},\ \beta_{i2,\ .\ .\ .,}\ \beta_{im})^T$is the weight vector connecting the i^{th} node and the output nodes while $w_i.x_j$ is defined as the inner product of w_i and x_j. Eq. (5) can be re-written linearly as:

$$H\beta = T \tag{6}$$

where $H\ (w_1,\ .\ .\ .,\ \mathrm{w}_{\tilde{N},}\ \ \mathrm{b}_1,\ .\ .\ .,\ \mathrm{b}_{\tilde{N}},\ \ \mathrm{x}_1.\ .\ .,\ \mathrm{x}_\mathrm{N}) = \begin{bmatrix} \mathrm{g}(\mathrm{w}_1.\mathrm{x}_1 +\ \mathrm{b}_1) & \cdots & \mathrm{g}(\mathrm{w}_{\tilde{N}}.\mathrm{x}_\mathrm{i} +\ \mathrm{b}_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ \mathrm{g}(\mathrm{w}_1.\mathrm{x}_1 +\ \mathrm{b}_1) & \cdots & \mathrm{g}(\mathrm{w}_{\tilde{N}}.\mathrm{x}_\mathrm{N} +\ \mathrm{b}_{\tilde{N}}) \end{bmatrix}_{\mathrm{N\ X\ \tilde{N}}}$

$\beta = \begin{bmatrix} {\beta_1}^\mathrm{T} \\ \vdots \\ {\beta_{\tilde{N}}}^\mathrm{T} \end{bmatrix}_{\mathrm{\tilde{N}\ X\ M}}$ and $\mathrm{T} = \begin{bmatrix} {\mathrm{t}_1}^\mathrm{T} \\ \vdots \\ {\mathrm{t}_{\tilde{N}}}^\mathrm{T} \end{bmatrix}_{\mathrm{\tilde{N}\ X\ M}}$ where H represents the hidden layer output matrix. According to Huang's Theorem (Huang et al., 2014), it is assumed that since input weights $\mathrm{w_i}$ and hidden layer biases H are randomly generated, the output weight β can be determined by finding the minimum norm least square (LS) solution to the linear system $\mathrm{H}\beta = \mathrm{T}$. The LS solution is described in Eq. (7)

$$\hat{\beta} =\ \mathrm{H}^{-1}\mathrm{T} \tag{7}$$

where H^{-1} is the Moore-Penrose generalized inverse of H.

In this work, ELM models were built using both LBP and LBP-TOP features and using five-fold cross validation to partition the data samples into five subsets. ELM model was selected because of its learning speed and more effective training ability [14]. Training this model was performed using four subsets out of the five subsets while the remaining one subset was reserved for testing purpose. This process was repeated five times and the average training accuracy was calculated.

ELM Training with LBP Features

Training accuracy was recorded for varying number of hidden neurons (between 50 and 400) at an interval of 50 using LBP features. It was discovered that training accuracy increased with increasing number of hidden neurons. There was a constant 100% training accuracy from 200 hidden neurons for the five classifiers as shown in Fig. 3. This informed the decision to select 300 as number of hidden neurons during validation of the model.

ELM training with LBP-TOP features

Optimization of parameters was performed by recording training accuracy at varying LBP-TOP radii values in x, y and t planes. These values were varied between 1 and 4 for x and y planes while variation for t plane was between 2 and 4 as presented in Table 5. Training results showed that the highest average training accuracy (97.57%) using ELM was achieved at $\mathrm{R_{x=}}1$, $\mathrm{R_{y=}}1$ and $\mathrm{R_{t=}}3$. To select optimal number of hidden neurons, average training accuracy was recorded between 50 and 500 at an interval of 50 for each class of micro-expression. The highest average training accuracy (97.54%) was achieved at 200 hidden neurons.

Table 5
Training accuracy records at varying LBP-TOP radii values using ELM model. (Accuracy measured in %).

R_x, R_y, R_t	Surprise	Disgust	Happiness	Others	Repression	Average
1, 1, 2	92.07	91.74	99.67	99.02	99.24	96.35
1, 1, 3	94.13	94.35	99.67	99.68	100.00	97.57
1, 1, 4	98.37	98.37	92.17	92..66	100.00	96.24
2, 2, 2	94.78	100.00	96.85	91.74	100.00	96.67
2, 2, 3	94.13	99.57	97.72	91.41	100.00	96.57
2, 2, 4	93.80	100.00	97.18	91.52	100.00	96.50
3, 3, 2	93.59	99.68	93.58	90.65	93.04	94.11
3, 3, 3	94.35	99.57	93.48	90.44	93.04	94.17
3, 3, 4	94.13	99.57	93.91	90.65	93.15	94.28
4, 4, 2	93.69	99.57	93.59	91.09	92.72	94.13
4, 4, 3	92.07	92.39	96.96	90.33	99.68	94.28
4, 4, 4	92.50	92.72	97.17	90.65	100.00	94.61

Table 6
Details on Final ELM model built for each classifier.

Details	Apex frames	Image sequence
Number of Samples	220	230
Feature Vector Size	1 × 256	1 × 177
ELM Model Type	Classification	Classification
Number of Input Neurons	256	177
Number of Output Neurons	2	2
Label	1 (Positive Samples) 0 (Negative Samples)	1 (Positive Samples) 0 (Negative Samples)

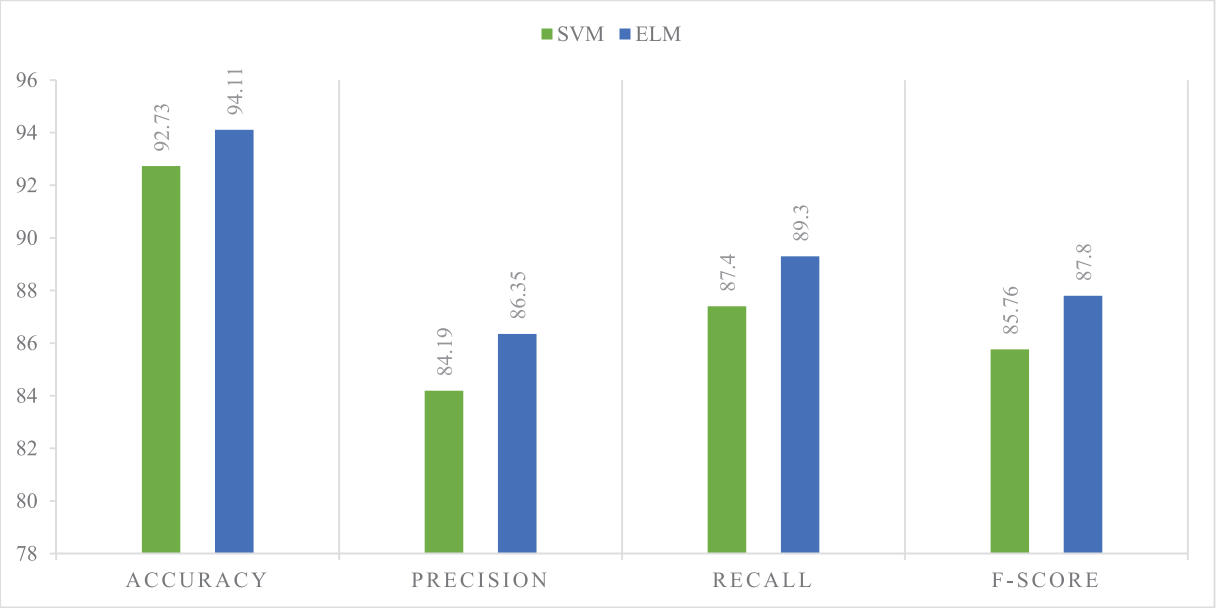


Fig. 4. Comparative results for SVM and ELM using LBP features.

Final ELM model architecture

This section presents a summary of the final architecture on which the actual classification (testing) was performed. Details on resulting ELM model using both LBP (static) and LBP-TOP (temporal) features are presented in Table 6. These details include type of model (regression/classification), number of input and output weights, number of input and output neurons and labels. Optimal LBP-TOP radii values in x, y and t planes were 1, 1 and 3 respectively based on highest average training accuracy of 94.00% using SVM model as presented in Table 3. Optimal LBP-TOP training accuracy was also achieved at Rx = 1, Ry = 1 and Rt = 3.

For ELM-based model training using LBP features from apex frames, optimal average training accuracy of 100% was achieved with 300 hidden neurons (See Fig. 4). This selection was made based on training accuracy recorded for apex

Table 7
Confusion matrix.

	Predicted (1)	Predicted (0)
Actual (1)	TP	FP
Actual (0)	FN	TN

Table 8
Performance of SVM model using LBP features.

Class	Disgust	Happiness	Repression	Surprise	Others	Average
Accuracy (%)	90.87	96.52	95.45	90.36	90.43	92.73
Precision (%)	82.05	84.28	81.43	84.28	88.89	84.19
Recall (%)	83.33	90.00	86.66	90.00	87.02	87.40
F1 (%)	82.69	87.05	83.96	87.05	87.95	85.76

frames at varying number of hidden neurons between 50 and 400 at an interval of 50. For ELM-based model using image sequences and LBP-TOP features, 200 hidden neurons were optimal with an average training accuracy of 97.54%. (See Table 6).

Experimental result, analysis and discussion

In this section, results obtained from all experiments are presented, analyzed and discussed. These include test performance for SVM and ELM models using LBP (static) features from apex micro-expression frames. It also includes test performance for SVM and ELM models using LBP-TOP (temporal) features from micro-expression samples. Comparative analysis was carried out to show which of the feature extraction and classification models performed better, based on the nature of data (static/temporal) used.

Accuracy, precision, recall and F1 measures were used to evaluate the models. Accuracy performs the function of measuring the correctness of the classifier. Precision helps to measure the relevance of the classifier while recall helps to measure the completeness of the classifier. Confusion matrix performs the function of describing the performance of the classifier based on the test sample data. The formula for calculating accuracy, precision, recall and F1 is illustrated in Eqs. (8), 9, (10) and (11). Confusion matrix is described in Table 7. In the confusion matrix, TP represents True Positive, TN represents True Negative, FP represents False Positive, and FN represents False Negative.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{8}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{F1} = \frac{2*\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

Classification with LBP features from apex frames

Classification was performed for each of the five micro-expression classes, which produced five classifiers. For this experiment, a total of 220 samples (57 disgust, 25 happiness, 27 repression, 25 surprise and 88 others) were divided into training and test sets using five-fold cross validation. The performance of the classifiers was recorded five times (five-fold) and their mean calculated. Number of training samples used for each classifier was within the range of 174 and 176 while number of test samples was within the range of 44 and 46 samples.

After LBP features were extracted and training performed, classification of the micro-expression samples was performed on test subsets for each of the five classifiers. The mean accuracy, precision, recall and F-score performances were recorded for each class using SVM and the result is presented in Table 8. Average testing accuracy of 92.73%, precision of 84.19%, recall of 87.40% and F-score of 85.76% was achieved for SVM model.

Table 9 describes the result recorded from test subsets (samples) for each of the five classifiers using ELM. Average testing accuracy of 94.11%, precision of 86.35%, recall of 89.30% and F-score of 87.80% was achieved for ELM model. The result presented in Table 8 and Table 9 shows that there is a higher recognition performance for ELM model compared with SVM when static (LBP) features were used. This is illustrated with a chart in Fig. 4.

Table 9
Performance of ELM model using LBP features.

Class	Disgust	Happiness	Repression	Surprise	Others	Average
Accuracy (%)	91.55	94.54	95.00	98.18	91.30	94.11
Precision (%)	89.96	82.86	79.05	88.57	90.11	86.35
Recall (%)	92.03	82.86	84.28	100.00	88.24	89.30
F1 (%)	90.98	82.86	81.58	93.94	89.17	87.80

Table 10
Performance of SVM model using LBP-TOP features.

Class	Disgust	Happiness	Repression	Surprise	Others	Average
Accuracy (%)	91.30	95.65	100	98.70	95.65	96.26
Precision (%)	71.43	81.43	100	90.00	83.33	85.24
Recall (%)	71.43	86.66	100	100	83.33	88.28
F1 (%)	71.43	83.96	100	94.74	83.33	86.73

Table 11
Performance of ELM model using LBP-TOP features.

Class	Disgust	Happiness	Repression	Surprise	Others	Average
Accuracy (%)	95.65	96.09	100	97.83	98.70	97.65
Precision (%)	91.67	83.81	100	86.66	96.84	91.80
Recall (%)	91.67	86.66	100	96.67	100	95.00
F1 (%)	91.67	85.21	100	91.39	98.39	93.37

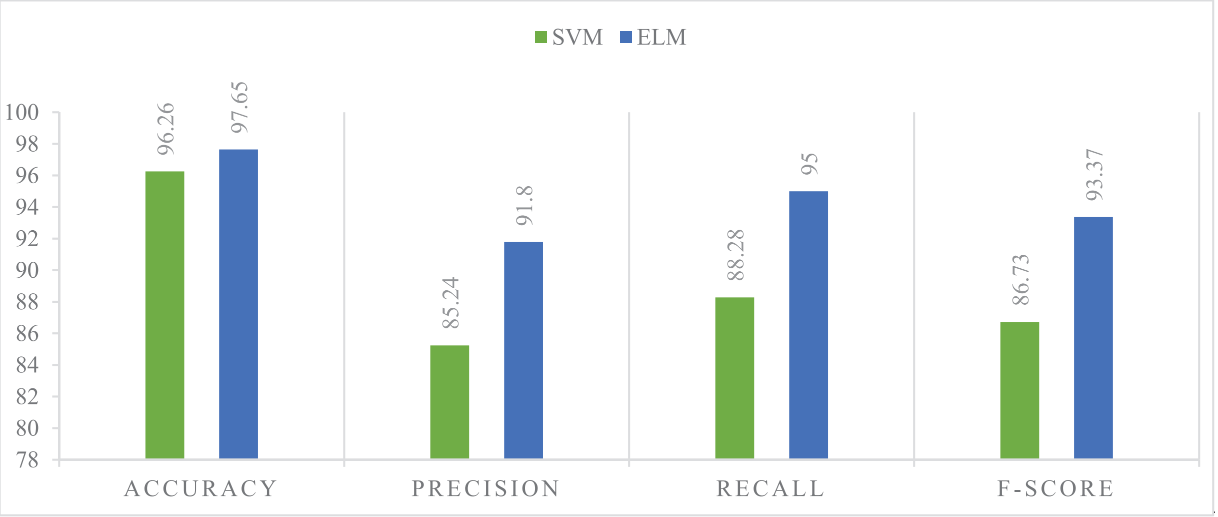
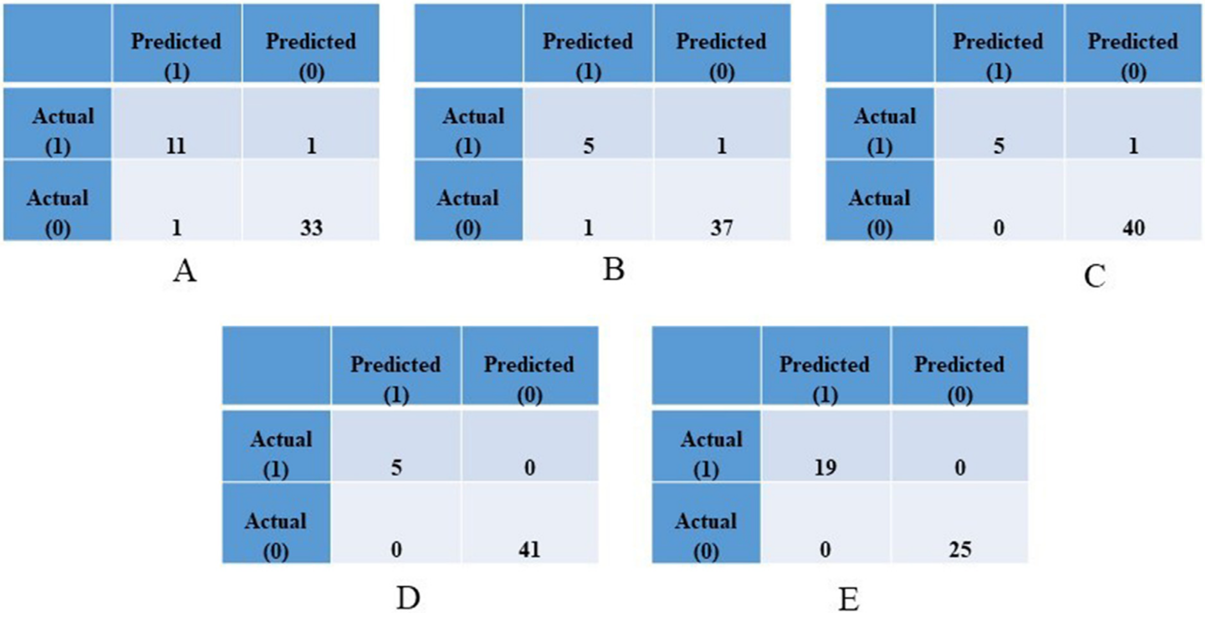


Fig. 5. Comparative results for SVM and ELM using LBP-TOP features.

Classification with LBP-TOP features from videos (image sequences)

For this experiment, a total of 230 image sequence samples (59 disgust, 30 happiness, 24 repression, 25 surprise and 92 others) were divided into training and test sets via a five-fold cross validation. Since one versus all classification was used, we had a total of five classifiers. Number of training samples used for each classifier was within the range of 184 and 186 while number of test samples was within the range of 44 and 46 samples. After LBP-TOP features were extracted and training performed, classification was performed on the test subsets using feature vectors from LBP-TOP and trained SVM model for each of the five classifiers. The results presented in Table 10 and Table 11 shows that there is a higher recognition performance for ELM model compared with SVM when temporal features were used. This is illustrated with a chart in Fig. 5 while the confusion matrices for the five classes is described in Fig. 6.



A

	Predicted (1)	Predicted (0)
Actual (1)	11	1
Actual (0)	1	33

B

	Predicted (1)	Predicted (0)
Actual (1)	5	1
Actual (0)	1	37

C

	Predicted (1)	Predicted (0)
Actual (1)	5	1
Actual (0)	0	40

D

	Predicted (1)	Predicted (0)
Actual (1)	5	0
Actual (0)	0	41

E

	Predicted (1)	Predicted (0)
Actual (1)	19	0
Actual (0)	0	25

Fig. 6. Confusion matrices for the five classes, labeled as A, B, C, D and E representing disgust, happiness, repression, surprise and others respectively.

Table 12
Training time (in seconds) of SVM and ELM using LBP-TOP features.

Class	SVM	ELM
Disgust	0.2808	0.0468
Happiness	0.3806	0.0499
Repression	0.2434	0.0593
Surprise	0.2995	0.0530
Others	0.3182	0.0406
Average Training Time	0.3405	0.0499

Comparative results from classification using static and temporal data

In this section, the performance results from the use of static (apex micro-expression frames) features is compared with that of temporal features (image sequences) is presented. Result shows that there is a higher and better performance with the use of temporal features in terms of all metrics used (accuracy, precision, recall and F-score) used. This applies to both SVM and ELM model. Details are presented in Fig. 7a and 7b.

Comparative result for SVM and ELM training time

Training time for each class of micro-expression was recorded and the average calculated. Results were recorded using both SVM and ELM. Average training time using SVM was 0.3405 seconds while ELM had an average training time of 0.0499 seconds. Hence, we can deduce that ELM learns faster than SVM. The difference between the training time of SVM and ELM for the five classes of micro-expressions using LBP-TOP features is presented in Table 12.

Comparison of previous related studies with this study

Results from this study were compared with results from reviewed literature. These results show that the combination of methods adopted for this study produces a higher recognition performance when compared with previous studies. Table 13a and Table 13b shows the comparative results using apex/static frames and image sequences respectively. A similar study was carried out by Yan et al. [28] while evaluating the performance of their micro-expression algorithm using LBP-TOP for feature extraction and SVM for classification. In their work, images were divided into blocks before applying LBP-TOP feature extraction on each block which resulted in an accuracy of 63.41%. In this study images were used in holistic manner which means that images were not divided into blocks before extraction of features to reduce computational complexity as expressed by Guo et al. [9].

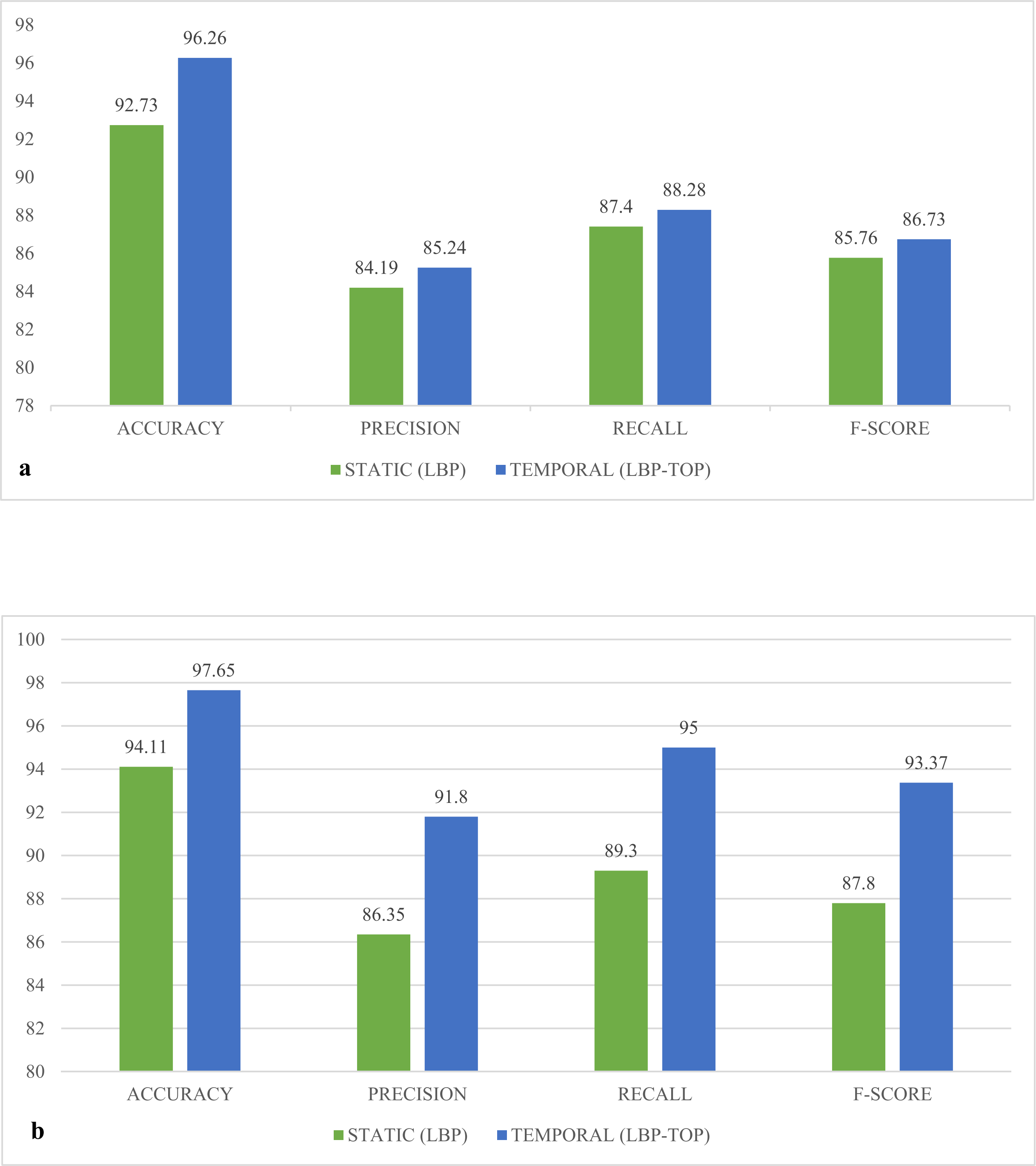


Fig. 7. (a) Comparative Results for Static (LBP) vs Temporal (LBP-TOP) Data using SVM. (b) Comparative Results for Static (LBP) vs Temporal (LBP-TOP) Data using ELM.

Table 13a
Comparative Analysis of our Study Vs Previous works (Static).

Author (s)	Features	Algorithm	Dataset	Performance
Liong et al. [12]	Constraint Local Models (CLM), LBP and Optical Strain	Not specified	CASME II	Over 20% improvement when compared with baseline methods
Liong et al. [13]	Bi-weighted Oriented Optical Flow (Bi-WOOF), LBP and LBP-TOP	Not specified	CASME II	61% F-measure 58.85% Accuracy (for Bi-WOOF + CASME II)
This Study	LBP	ELM	CASME II	94.11% Accuracy, 86.35% Precision, 89.30% Recall and 87.80% F-score.
This Study	LBP	SVM	CASME II	92.73% Accuracy, 84.19% Precision, 87.40% Recall and 85.76% F-score.

Table 13b
Comparative Analysis of our Study Vs Previous works (Temporal).

Author (s)	Features	Algorithm	Dataset	Performance
Yan et al. [28]	LBP-TOP	SVM	CASME II	63.41% Accuracy
Wang et al. [23]	Robust Principal Component Analysis (RPCA) and Local Spatio-temporal Descriptors	Not specified	CASME II	65.45% Accuracy
Wang et al. [24]	LBP-TOP	EVM	CASME II	75.30% Accuracy
Jia et al. [11]	LBP-TOP	KNN	CASME II	65.50% Accuracy
Zhang et al. [29]	LBP-TOP	Random Forest	CASME II	62.5 % Accuracy
This Study	LBP-TOP(without dividing faces into blocks)	SVM	CASME II	96.26% Accuracy, 85.24% Precision, 88.28% Recall and 86.73% F-score.
This Study	LBP-TOP(without dividing faces into blocks)	ELM	CASME II	97.65% Accuracy, 91.80% Precision, 95.00% Recall and 93.37% F-score.

Furthermore, our results from LBP-TOP with SVM model produces an average testing accuracy of 96.26% while results from LBP-TOP features with ELM model produces an average testing accuracy of 97.65%. This shows a higher recognition performance when compared with the study in Yan et al. [28].

From our results, we can also deduce that recognition of micro-expressions using LBP-TOP features (holistic) for SVM and ELM on CASME II database outperformed other methods. ELM provides a faster learning model when compared with SVM. The learning speed of ELM is believed to contribute to the overall performance of the micro-expression recognition process.

Conclusion

This study reveals that it is possible to recognize micro-expressions automatically and achieve promising results using temporal feature extraction technique (LBP-TOP) and a machine learning algorithm with an efficient and very fast learning speed (ELM). Two data formats were used for the experiments. The first format includes apex frames extracted from CASME II micro-expression samples. The second data format includes temporal image sequences from CASME II micro-expression samples. The first experiment was conducted by extracting LBP features from apex frame samples using SVM and ELM for classification. The second experiment was conducted by extracting LBP-TOP features from micro-expression image sequence samples using SVM and ELM for classification. Results show that supervised machine learning algorithms (SVM and ELM) can successfully classify micro-expressions into their appropriate classes. It also shows that recognition of micro-expression using temporal features is more effective than recognition of micro-expression using static features. However, this depends on the classification model used and the number of samples. Comparison between SVM and ELM training time also shows that ELM learns faster than SVM. An average training time of 0.3405 seconds is achieved for SVM while an average training time of 0.0409 seconds is achieved for ELM for the five selected micro-expression

Future work should include employing a better means of reducing LBP-TOP feature vector size and optimizing the features (feature selection). In this study, we reduced the feature vector size from (1 × 256) to (1 × 177) by applying uniform patterns. We could not ascertain if this means of reduction to 177 features is efficient in improving classification. The use of optimization algorithms such as genetic algorithm or ant colony could be adopted in the future. This will help to optimize the features used for micro-expression recognition and hence improve performance as identified in Loderer & J. Pavlovicova [16]. The use of other kernels for SVMs aside from linear kernel and other activation functions for ELMs is also an area of interest that can be further studied.

On a more general note, recognition of micro-expressions can assist in identifying criminals with bad intentions and are trying to suppress their emotions. This can act as a useful tool in achieving the 16th Sustainable Development Goal (Peace, Justice and Strong Institutions).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The support of the DST-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS) towards the research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the CoE. The Institute of Psychology, Chinese Academy of Sciences is appreciated for releasing CASME II micro-expression database for this research.

References

[1] I.P. Adegun, H.B. Vadapalli, Automatic recognition of micro-expressions using local binary patterns on three orthogonal planes and extreme learning machine, in: 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), IEEE, 2016, pp. 1–5.

[2] L. Auria, R.A. Moro, Support vector machines (SVM) as a technique for solvency analysis. German Institute for Economic Research, 2008.

[3] C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov. 2 (2) (1998) 121–167.

[4] I. Cohen, N. Sebe, A. Garg, L.S. Chen, T.S. Huang, Facial expression recognition from video sequences: temporal and static modeling, Comput. Vision Image Underst. 91 (1) (2003) 160–187.

[5] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

[6] P. Ekman, W.V. Friesen, Non-verbal leakage and clues to deception, Psychiatry 32 (1) (1969) 88–106.

[7] M. Frank, M. Herbasz, K. Sinuk, A. Keller, C. Nolan, I see how you feel: training laypeople and professionals to recognize fleeting emotions, The Annual Meeting of the International Communication Association, 2009.

[8] Y. Guo, Y. Tian, X. Gao, X. Zhang, Micro-expression recognition based on local binary patterns from three orthogonal planes and nearest neighbor method, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2014, pp. 3473–3479.

[9] Y. Guo, C. Xue, Y. Wang, M. Yu, Micro-expression recognition based on CBP-TOP feature with ELM, Opt. Int. J. Light Electron Opt. 126 (23) (2015) 4446–4451.

[10] E.A. Haggard, K.S. Isaacs, Micro-momentary facial expressions as indicators of ego-mechanisms in psychotherapy, in: Methods of Research in Psychotherapy, Springer, 1966, pp. 154–165.

[11] X. Jia, X. Ben, H. Yuan, K. Kpalma, W. Meng, Macro-to-micro transformation model for micro- expression recognition, J. Comput. Sci. 25 (2017) 289–297, doi:10.1016/j.jocs.2017.03.016.

[12] S.-T. Liong, J. See, K. Wong, A.C. Le Ngo, Y.-H. Oh, R. Phan, Automatic apex frame spotting in micro-expression database, in: 3rd IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, 2015, pp. 665–669.

[13] S.T. Liong, J. See, K. Wong, R.C.W. Phan, Less is more: Micro-expression recognition from video using apex frame, Signal Process. Image Commun. 62 (2017) 82–92.

[14] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: A new learning scheme of feedforward neural networks, in: Proceedings of International Joint Conference on Neural Networks, IEEE, 2004, pp. 985–990.

[15] X. Li, T. Pfister, X. Huang, G. Zhao, M. Pietikainen, A spontaneous micro-expression database: inducement, collection and baseline, in: 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), IEEE, 2013, pp. 1–6.

[16] M. Loderer, J. Pavlovicova, Optimization of LBP parameters, in: 56th International ELMAR Symposium, IEEE, 2014, pp. 1–4.

[17] M. Lyons, S. Akamatsu, M.& Kamachi, J. Gyoba, Coding facial expressions with Gabor Wavelets, in: Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 200–205.

[18] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,", IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.

[19] T. Pfister, X. Li, G. Zhao, M. Pietikainen, Recognising spontaneous facial micro-expressions, in: IEEE International Conference on Computer Vision (ICCV), 2011, pp. 1449–1456.

[20] L. Su, M.D. Levine, High-stakes deception detection based on facial expressions, in: 22nd IEEE International Conference on Pattern Recognition (ICPR), 2014, pp. 2519–2524.

[21] E. Vural, M. Bartlett, G. Littlewort, M. Cetin, A. Ercil, J. Movellan, Discrimination of moderate and acute drowsiness based on spontaneous facial expressions, in: 20th International Conference on Pattern Recognition (ICPR), IEEE, 2010, pp. 3874–3877.

[22] Y. Wang, F. Cao, Y. Yuan, A study on effectiveness of extreme learning machine, Neurocomputing 74 (16) (2011) 2483–2490.

[23] S.-J Wang, H.-L Chen, W.J Yan, Y.H. Chen, X Fu, Face recognition and micro- expression recognition based on discriminant tensor subspace analysis plus extreme learning machine, Neural Process. Lett. 39 (1) (2014) 25–43.

[24] Y. Wang, J. See, Y.-H. Oh, R.C. Phan, Y. Rahulamathavan, H.C. Ling, Effective recognition of facial micro-expressions with video motion magnification, in: Multimedia Tools and Applications, 76, Springer, 2017, pp. 21665–21690, doi:10.1007/s11042-016-4079-6.

[25] J. Whitehill, M. Bartlett, J. Movellan, Automatic facial expression recognition for intelligent tutoring systems, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'08), 2008, pp. 1–6.

[26] Q. Wu, X. Shen, X. Fu, The machine knows what you are hiding: An automatic micro-expression recognition system, in: Affective Computing and Intelligent Interaction, Springer, 2011, pp. 152–162.

[27] W.-J. Yan, Q. Wu, Y.-J. Liu, S.-J. Wang, X. Fu, CASME database: A dataset of spontaneous micro-expressions collected from neutralized faces, in: 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 2013, pp. 1–7.

[28] W.-J. Yan, X. Li, S.-J. Wang, G Zhao, Y.-J. Liu, Y.-H. Chen, X. Fu, CASME II: an improved spontaneous micro-expression database and the baseline evaluation, PLoS One 9 (1) (2014) e86041, 2014.

[29] S. Zhang, B. Feng, Z. Chen, X. Huang, Micro-expression recognition by aggregating local spatio- temporal patterns, in: International Conference on Multimedia Modeling (Reykjavik: Springer), 2017, pp. 638–648.

[30] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 915–928.