



Elite Firefly and Long Short Term Memory Based Model for Texture Classification

Naga Siva Jyothi Kompalli^{1*} Dasika Sree Rama Chandra Murthy²

¹*Sri Satya Sai University of Technology and Medical Sciences, Sehore, India*

²*Geethanjali College of Engineering and Technology, Hyderabad, India*

* Corresponding author's Email: Jyothi.kompalli@gmail.com

Abstract: The texture is an important feature in the visual content of the images and plays a significant role in the computer vision system. Many existing methods were involved in applying the feature selection and classifiers for texture classification. Existing methods have the limitation of poor convergence in the feature selection and overfitting problems in the classification. In this research, the Elite FireFly – Long Short Term Memory (EFF-LSTM) model is proposed to improve the efficiency of the texture classification. The EFF method has the advantage of replacing the low fitness firefly with a high fitness firefly in the search process and the LSTM model has the advantage of a store the important features in long term for the classification. This process provides good convergence in the feature selection and important features are applied to overcome the overfitting problem. Two datasets such as UIUC and KTHTIPS-2b were used to test the performance of the proposed EFF-LSTM model. The Min-Max normalization method is applied to enhance the quality of the images. The Local Directional Ternary Pattern (LDTP), Dual-Tree Complex Wavelet Transform (DTCWT), and Grey-Level Co-occurrence Matrix (GLCM) were the extracted features from the pre-processed images. The EFF method is applied to select the relevant features with good convergence. The LSTM method is applied to classify the images based on the EFF selected features. The proposed EFF-LSTM model has an accuracy of 96.5 % and the existing Convolutional Neural Network (CNN) has 94.3 % accuracy.

Keywords: Convolutional neural network, Elite fireFly, Local directional ternary pattern, Long short term memory, Texture classification.

1. Introduction

Texture feature is the important source of information in the images and plays a significant role in computer vision applications. In computer vision, texture feature extraction is an important step in various applications such as face recognition, image matching, image segmentation, image retrieval, and texture classification [1]. Texture features provide rich information about the environment in the image and Texture features in computer vision provide various visual information about the characteristics and appearance of the image. Various methods were applied to extract texture features from the images in computer vision applications due to their significant [2]. An image texture region of characterizing is analyzed based on the spatial distribution of image

intensity or color in the non-uniform or varying pattern. If the scene has an inherent pattern, then the captured scene variations of non-uniformity are analogous. The texture features play a significant role in pattern recognition and computer vision classification [3]. The unique feature denotes the various textures that define each class. The perspective changes, illumination, rotation are the same class in images that are fundamental processes in classification and texture analysis [4].

Deep features recently show the effective performance in the image representation and most prominent performance in various computer vision. Deep features' main drawback in the visual texture classification requires a large amount of labeled data for training, and time-consuming labeling [5, 6]. Various techniques were present to extract the texture information of images. The Local Binary Pattern

(LBP), Wavelet Transform, Gabor Filter, Markov Random Field, and Co-occurrence matrix were the various features for the classification. These features were applied in various applications such as palm print recognition, medical image analysis, ear recognition, and face recognition [7, 8]. The binary pattern is constructed from image pixels between neighborhood pixels and the center in the traditional local binary pattern method [9, 10]. In this research, the EFF-LSTM model is proposed to improve the efficiency of texture classification. Two datasets such as UIUC and KTHTIPS-2b were used to test the performance of the developed method. The EFF method replace the low fitness firefly with high fitness firefly to improve the learning and LSTM method stores important features for long term. The EFF method provides good convergence in the feature selection and provide relevant features for classification. The good convergence and store important features for long terms helps to overcome the overfitting problem in classification.

The paper is organized as follows: a review of recent texture classification methods was provided in Section 2 and the explanation for feature extraction, EFF, and LSTM was provided in Section 3. The EFF-LSTM model simulation setup is in Section 4 and the result of the EFF-LSTM model in texture classification is given in Section 5. The conclusion of this research is given in Section 6.

2. Literature review

Texture information from the images plays a significant role in the visual cue describing the characteristics of many types of images. Texture classification is the fundamental process in computer vision and is used in many applications such as medical analysis, image retrieval, face recognition, etc.

Florindo [11] applied non-linear filter and scattering networks to provide descriptors with a recursive scheme and apply them to the Convolutional Neural Network (CNN). The descriptor of Schroedinger distance transform was applied to extract the texture features from the input images. Three datasets such as UIUC, UMD, and KTHTIPS-2b were used to evaluate the performance of descriptors with CNN in texture classification. The Brazilian plant species identification was carried out to test the efficiency of the descriptor with the CNN model. The descriptors with the CNN method have a higher performance compared to the existing method. The CNN model has the limitation of overfitting problem that affects the performance of the model.

Bu [12] proposed a descriptor based on convolutional layer activation of end-to-end learning framework for texture classification. The locality-aware layer was applied to learn the locality constraint that simultaneously learns the encoding representation and dictionary. The locality-aware was performed based on backpropagation on the layer for the training process in the network. The KTH-T2b, FMD, and DTD datasets were used to evaluate the performance of convolution with a locality-aware method for texture classification. The result shows that convolution with locality-aware method has higher performance compared to existing methods in texture classification. The CNN model has the limitation imbalance data problem and low performance in high variance of data.

Ershad [13] proposed Improved Local Ternary Pattern (ILTP) for bark texture classification for tree identification. The ternary pattern was applied as two binary patterns and classified into uniform or non-uniform classes in the proposed ILTP model. The degree of uniformity was measured to label the extracted patterns and the probability occurrence was measured to label the extracted features. The four theories in the number of hidden layers of multilayer perceptron were applied for the classification of bark. The ILTP method has the advantage of rotation invariant and Noise-resistance in the classification. Two benchmark datasets were used to evaluate the performance of the ILTP method in texture classification. The multilayer perceptron model hyperparameter was needed to be optimized for better performance in texture classification.

Ghazouani and Barhoumi [14] applied Local Edge Signature (LES) to represent the texture feature in the classification. Edge pixel arrangement of statistical information and a specific local region orientation of LES are insensitive to scale and rotation changes. Genetic Texture Signature (GTS) was a genetic programming method applied to learn a global texture descriptor. The global texture features were learned based on the individual tree representation and intra-class homogeneity was applied to analyze the fitness value of the model. Six benchmark datasets were used to evaluate the performance of the developed method. The genetic programming method has the limitation of being easily trap into local optima and have lower efficiency.

Garg and Dhiman [15] applied discrete wavelet transformation, correct and approximate coefficient in RGB channel to obtain multi-level decomposition. The dominant rotated local binary pattern was applied with the correct and approximate coefficient

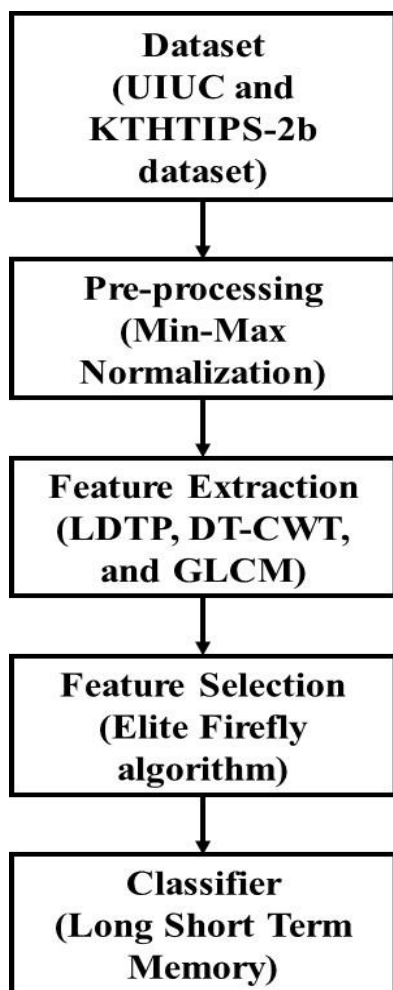


Figure. 1 The EFF-LSTM model in texture classification

for rotation invariant and to compute the texture feature effectively. The local binary pattern extracted structural information and magnitude information was provided for the classification. The GLCM descriptor was applied to analyze the statistical information for textural image classification. The CORAL dataset was used to evaluate the performance of the developed method in content retrieval. Various classifiers were applied to test the performance of the developed feature extraction in the classification. The result shows that the developed method has higher performance compared to the existing method.

3. Proposed method

In this study, the EFF-LSTM model is proposed to improve the efficiency of texture classification. The EFF method applied the elite method to replace the less fitness firefly with a high-fitness firefly to improve the exploitation of the search process. Two datasets such as UIUC and KTHTIPS-2b were used to test the performance of the developed model. The LDTP, DT-CWT, and GLCM features were exacted

from the pre-processed images and applied for the feature selection. The EFF method is applied to select the features with good convergence and apply them to the classifier. The LSTM method receives the features selected from the EFF and performance classification. The EFF-LSTM model in the texture feature block diagram is shown in Fig. 1.

3.1 Pre-processing

The differences in the original data affect the performance of the classification. Normalization technique is applied to enhance the intensity of the images to provide clear information of the image and improve the performance of classification. The intensity normalization is shown in Eq. (1).

$$I = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where image minimum intensity is denoted as x_{min} , image maximum intensity is denoted as x_{max} , normalized image is denoted as x_N and the input image is denoted as I .

3.2 Feature extraction

The LDTP, DT-CWT, and GLCM features were extracted in this model and apply to EFF method for feature selection.

3.2.1. Local directional ternary pattern

Each neighborhood principal axis was analyzed to extracts the texture information on the LDTP operator. The LDTP operator performs on the principal information and other operators attempt to analyze the available information that involves in creating errors sometimes. The LDTP operator provides effective performance in scene recognition and facial expression recognition.

Each pixel edge response values F_i were calculated using Kirsch masks to obtain the LDTP operator, as given in Eq. (2).

$$F_i = |I \times M_i|, (i = 0,1, \dots,7), \tag{2}$$

where input image is denoted as I , the i^{th} Kirsch mask is M_i , and the convolution operation is denoted as “ \times ”.

Each pixel response value F_i are sorted to compute the two principal directions. Eq. (3) shows each pixel's first principal directional number P_{dir}^1 .

$$P_{dir}^1 = argmax\{F_i | 0 \leq i \leq 7\} \tag{3}$$

Similarly, each pixel second principal directional number P_{dir}^2 are selected and this is the second maximum response of the order.

The eight neighborhood pixels gray value is denoted as m_0, \dots, m_7 and center pixel gray value is denoted as m_c . Neighbourhood gray value pixel difference d_n is measured in every two principal directions, as given in Eq. (4).

$$d_n = \begin{cases} m_{P_{dir}^n} - m_{P_{dir}^n+4} & P_{dir}^n \in \{0,1,2,3\} \\ m_{P_{dir}^n} - m_{P_{dir}^n-4} & P_{dir}^n \in \{4,5,6,7\} \end{cases} \quad (4)$$

Eq. (5) provides encode of each difference.

$$D_f(d_n) = \begin{cases} 0, & -\varepsilon \leq d_n \leq \varepsilon \\ 1, & d_n < -\varepsilon \\ 2, & d_n > \varepsilon, \end{cases} \quad (5)$$

where the gray value of the difference d_n is measured based on Eq. (3), the specific threshold is denoted as ε , and the difference d_n encoded value is denoted as $D_f(d_n)$.

The LDTP is measured for each pixel based on concatenating directional encoded difference of first and second $D_f(d_1)$ and $D_f(d_2)$. The 8 possible values of the first principal directional number and 3 possible values of encoded difference to measure the LDTP histogram dimension is 72 ($8 \times 3 \times 3$). A detailed description of the LDTP operator is given in the research [16].

3.2.2. Dual-tree complex wavelet transform (DTCWT)

In image and signal processing, DWT is widely applied and has some limitations of lack of shift-invariance, transform domain in aliasing effects, singularities in coefficient oscillations [17]. The DTCWT is enhanced from the DWT method with the additional property of directionally selective and shift-invariance.

At each level decomposition, low frequency of two complex sub-bands and high frequency of six complex sub-bands are generated by the DTCWT from the segmented image I_s . The output-oriented angles of $\pm 15^\circ$, $\pm 45^\circ$, and $\pm 75^\circ$ were denoted in six directional filters. The low-frequency coefficients are represented as the Eq. (6).

$$x(\lambda, L, u, v) = Re(x(\lambda, L, u, v)) + jIm(x(\lambda, L, u, v)); \quad (6)$$

where $L \in \{L_1, L_2\}$

The coefficient of high frequency is denoted as Eqs. (7) to (10).

$$x(\Lambda, \theta, u, v) = Re(x(\Lambda, \theta, u, v)) + jIm(x(\Lambda, \theta, u, v)) \quad (7)$$

For

$$\theta \in \{-75, -45, -15, +15, +45, +75\} \quad (8)$$

$$0 \leq u \leq \frac{N}{2^\Lambda} - 1 \quad (9)$$

$$0 \leq v \leq \frac{M}{2^\Lambda} - 1 \quad (10)$$

where $Re(.)$ and $Im(.)$ denotes the real and imaginary parts, respectively. The low-frequency sub-bands of first and second tree decomposition is represented as L_1 and L_2 . The θ denotes the sub-band direction and Λ denotes the decomposition level. The input image size is denoted as N and M . The variables of u and v denote the location of each sub-band.

3.2.3. Grey-level co-occurrence matrix

The GLCM is one of texture feature extraction [18] and measures the intensity of local variation in a repeating pattern. This provides intensities or colors of spatial arrangement information, characterized by intensity levels spatial distribution in a neighborhood. The GLCM is a statistical method to measure region intensity arrangement and various works show the benefits of GLCM.

The GLCM is the frequency occurrence related to the pixel intensity of different combinations in an image. The relation between two pixels of reference and neighbor pixel were measured and the relation is given in co-occurrence matrix form. The GLCM equation is described in Eq. (11).

$$P_{d_{i,j}} = n_{i,j} \quad (11)$$

The pixel values (i, j) number of occurrence is denoted as $n_{i,j}$ at length $d_{i,j}$ in the image. The occurrence matrix P_d is in the size of $n \times n$ and grey levels in the images are denoted as n . The quantization is used to specify the grey levels in the images. The elements of P_d are normalized using Eq. (12).

$$N_{i,j} = \frac{P_{i,j}}{\sum_{i=1}^n \sum_{j=1}^n P_{i,j}} \quad (12)$$

After normalization, matrix $P_{d_{i,j}}$ elements are presented in the range of 0 to 1 and probability function is applied for manipulation.

The co-occurrence matrix is used to extract features and texture properties of GLCM is captured. The feature vector for extraction is selected based on contrast. An image's local variations $C(k, n)$ are measured using Eq. (13).

$$C(k, n) = \sum_{i=1}^n \sum_{j=1}^n (i - j)^2 N[i, j] \quad (13)$$

3.3 Elite firefly algorithm

Firefly algorithm is developed based on the fireflies group behavior and simulated for the optimization process. Attractiveness and light intensity are the two important elements in the firefly algorithm [19]. The optimization process is described as follows:

Firefly light intensity is measure in Eq. (14).

$$I = I_0 \times e^{-\gamma r_{ij}} \quad (14)$$

where original light intensity is I_0 , the distance between firefly i and j is r_{ij} , and the light absorption coefficient is γ .

A firefly's attractiveness is given in Eq. (15).

$$\beta = \beta_0 \times e^{-\gamma r_{ij}^2} \quad (15)$$

where, β_0 is the attractiveness at $r_{ij} = 0$.

Eq. (16) provides the attraction a of firefly i movement towards firefly j .

$$a = x_i + \beta(x_j - x_i) + \alpha \times (rand - 1/2) \quad (16)$$

where firefly i and j location are denoted as x_i and x_j , uniform distribution of random numbers is denoted in the range of $[0, 1]$, and the step factor is denoted as α .

Around the current best solution, the local search attempts to find a better solution. The elite method replaces the new solution New in the worst one Wst if a new solution has higher fitness than worst learning of firefly, as given in Eq. (17).

$$w_i = \begin{cases} New, & \text{if } f(New) < f(Wst) \\ Worst, & \text{else} \end{cases} \quad (17)$$

where current population worst one is w_i and the corresponding solution objective function is denoted as $f(\cdot)$. Each decision variable in firefly linearly decreases with fitness F increases to increases the effectiveness of the local search process, as given in Eq. (18).

$$\mu = 1 - F/Max_F \quad (18)$$

Chaotic sequence provides the random-like process in dynamic systems of non-linear properties, which is bounded, non-converging, and non-periodic. The logistic map is the simplest chaotic map is applied in this study. The chaotic dynamics are applied in the model, the control parameter is set to 4 and the initial value is denoted as $X_{n+1} = 4 \times X_n \times (1 - X_n)$. The chaotic map is discussed as follows.

The n^{th} chaotic iteration value is denoted as X_n and iteration number is denoted as n .

Eq. (19) calculates mutation variable μ .

$$\mu = 2 \times X_n - 1 \quad (19)$$

The New formulation is given in Eq. (20).

$$New(1, k) = \begin{cases} teacher(1, k) + 2 \times X_n - 1, & \text{if } rand < \mu \\ teacher(1, k) & \text{else} \end{cases} \quad (20)$$

The best learning of firefly is replacing the worst learning of firefly.

3.4 Long short term memory

The LSTM method not only requires the recent data and also the historic data for the prediction process. The LSTM model has the advantage of long-term dependence problems based on the hidden layer self-feedback mechanism [20]. LSTM consists of three special gates and a memory cell to store the information. The LSTM cell structure is shown in Fig. 2.

The LSTM cell receives the input data of x_t and the output of the previous moment LSTM cell is denoted as h_{t-1} , and memory cell value is denoted as c_t , and the LSTM cell output is denoted as h_t . The LSTM process is explained as follows:

The candidate memory cell value is calculated as \tilde{c}_t , the bias is denoted as b_c and the weight matrix is denoted as W_c , as shown in Eq. (21).

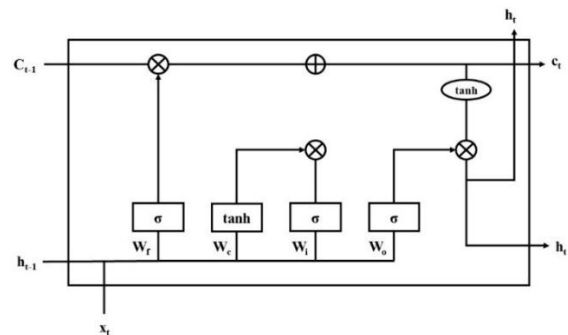


Figure. 2 The LSTM cell structure

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (21)$$

The input gate i_t value is calculated, the current input data update is controlled by the input gate to the memory cell state value, the weight matrix is denoted as W_i , the sigmoid function is denoted as σ , and the bias is denoted as b_i , as shown in Eq. (22).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (22)$$

The forget gate f_t value is calculated, the historical data update is controlled by the forget gate and store memory cell state value, the bias is denoted as b_f , the weight matrix is denoted by W_f , as shown in Eq. (23).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (23)$$

The value of current moment memory cell c_t is calculated, and the last LSTM unit state value is c_{t-1} , as shown in Eq. (24).

$$c_t = f_t \times c_{t-1} + i_t * \tilde{c}_t \quad (24)$$

where dot product is denoted as “ \times ”. The input and forget gate controls the memory cell update, which is dependent on the state value of the last cell and the candidate cell.

The output gate o_t value is calculated, the memory cell state output value is controlled by the output gate, the bias is denoted as b_o , the weight matrix is denoted as W_o , as given in Eq. (25).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (25)$$

The LSTM unit output h_t is calculated in Eq. (26).

$$h_t = o_t * \tanh(c_t) \quad (26)$$

LSTM method is applied to store, reset and update the information in long term easily based on the memory cell and three control gates. The output dimension is adjusted by setting the weight matrix dimensions based on LSTM internal parameters sharing mechanism. The LSTM architecture maintains a continuous error flow and due to this, the gradient will neither disappear nor explode in the internal state of the memory cell.

4. Simulation setup

In this study, the Elite FireFly with LSTM is proposed to improve the efficiency of textural classification. This section provides detailed

information about the datasets, metrics, system requirements, and parameter settings.

Dataset: The UIUC [21] dataset consists of 1000 images with 25 categorical of images and images have variations in scale, illumination, perspective, and albedo. The KTHTIPS-2b [22] dataset consists of 4752 images and has 11 classes. The images represent the materials rather than the objects.

Metrics: Accuracy, Sensitivity, and Specificity were the three metrics used to evaluate the performance of the proposed method. The formulas for accuracy, sensitivity, and specificity are given in Eq. (27, 28, & 29).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (27)$$

$$Sensitivity = \frac{TP}{TP+FN} \times 100 \quad (28)$$

$$Specificity = \frac{TN}{TN+FP} \times 100 \quad (29)$$

where TP denotes True Positive, TN denotes True Negative, FP denotes False Positive, and FN denotes False Negative.

System Requirement: The system consists of Intel i7 processor, 16 GB of RAM, and a 6 GB graphics card. The MATLAB 2018b software was used to implement and test the performance of the developed method.

Parameter Settings: The learning rate of the LSTM is set as 0.4, the number of an epoch is set as 50, and the dropout rate is set as 0.001. The number of populations is set as 50 and the number of iterations is set as 50 in the Firefly algorithm.

5. Results and discussion

In this study, the EFF-LSTM model is proposed to improve the efficiency of textual classification. Two datasets such as UIUC and KTHTIPS-2b were used to test the efficiency of the proposed model in textual classification. The metrics such as accuracy, sensitivity, and specificity were measured to analyze the efficiency of the model. The proposed feature selection methods were compared with other feature selection techniques. The proposed LSTM model is compared with other classifiers to analyze the performance.

The proposed EFF-LSTM model is evaluated in the UIUC dataset and compared with other feature selection and classifiers, as shown in Table 1.

The LDTP, DTCWT, and GLCM features were extracted and applied for feature selection. Table 1 shows that the EFF-LSTM model has a higher

Table 1. The proposed EFF-LSTM model performance analysis in UIUC dataset

Feature Selection	Classifiers	Accuracy (%)	Sensitivity (%)	Specificity (%)
GA	RF	81	82.3	84.5
	SVM	81.5	83.1	82.4
	ANN	83	84.5	86.2
	LSTM	85.2	86.2	86.7
PSO	RF	83.6	85.4	85.7
	SVM	85.4	87.3	87.6
	ANN	87.1	89.2	89.7
	LSTM	90.4	92.4	92.5
Firefly	RF	88.2	90.4	91.7
	SVM	88.4	92.1	91.6
	ANN	89.7	92.3	92.4
	LSTM	92.3	95.4	95.6
Elite Firefly	RF	90.2	93.2	93.4
	SVM	92.4	95.1	95.3
	ANN	94.2	94.6	94.7
	LSTM	96.5	97.2	97.4

efficiency compared to the other feature selection and classifier models. The EFF-LSTM model has the advantage of replacing the firefly of loss fitness value with a high fitness firefly to improve the probability of finding a better solution. This process improves the convergence of the method and finds effective performance. The existing methods such as GA, PSO, and Firefly have the limitation of lower convergence and easily trap into local optima. The LSTM model has the advantage of select the relevant features for the long term and backpropagation is applied to improve the efficiency of the classifier. The RF, SVM and ANN model has the limitation of overfitting, data imbalance problem, and feature redundant in the classification, respectively. The EFF-LSTM model has an accuracy of 96.5 % and the EFF-ANN model has 94.2 % accuracy.

The proposed EFF-LSTM model accuracy in UIUC dataset is compared with existing feature selection and classification, as shown in Fig. 3. The proposed

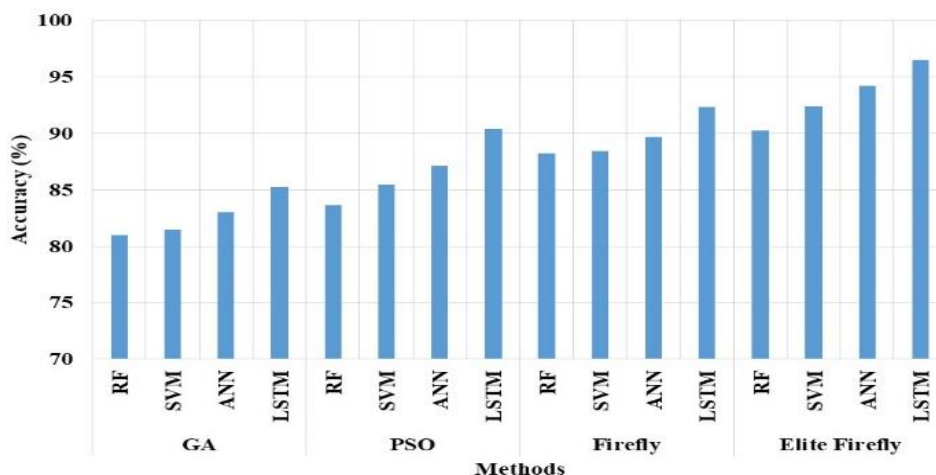


Figure. 3 The proposed EFF-LSTM model accuracy in the UIUC dataset

EFF-LSTM model has higher accuracy compared to other methods due to the advantage of replacing the best fit firefly instead of the low fit firefly in the search process. This improves the performance of the search process and also improves the convergence of the search. The LSTM model has the advantage of holding the important features for the long term and apply for classification. The LSTM model has a higher performance in various feature selection methods in classification due to these advantages. The RF, SVM and ANN methods have the limitations of overfitting, imbalance data, and redundant feature in the classification respectively. The EFF method has a higher efficiency compared to the Firefly algorithm in feature selection. The EFF-LSTM model

has an accuracy of 96.5 % and the EFF-ANN model has 94.2 % accuracy.

The EFF-LSTM model is analyzed in the KTHTIPS-2b dataset and compared with the existing feature selection and classifier methods, as shown in Table 2. The table shows that the EFF-LSTM model has a higher efficiency than other feature selection and classifiers. The proposed EFF-LSTM method has lower performance in the KTHTIPS-2b dataset compared to UIUC dataset due to the KTHTIPS-2b dataset has more images and textural features are similar. The EFF method has the advantage of replacing the low-fit firefly with the high-fit firefly to improve the convergence and the LSTM method has the advantage of a store the important features for the

Table 2. The EFF-LSTM model performance analysis in KTHTIPS-2b dataset

Feature Selection	Classifiers	Accuracy (%)	Sensitivity (%)	Specificity (%)
GA	RF	61.7	62.3	63.5
	SVM	62.9	65.1	65.7
	ANN	63.1	65.6	65.8
	LSTM	65.7	67.2	67.9
PSO	RF	62.3	65.3	65.7
	SVM	68.2	70.3	71.2
	ANN	71.2	73.5	74.3
	LSTM	72.8	76.6	76.8
Firefly	RF	71.7	73.5	75.1
	SVM	72.6	75.6	75.6
	ANN	74.2	77.1	77.4
	LSTM	76.3	79.3	79.6
Elite Firefly	RF	74.1	77.4	77.6
	SVM	76.2	77.6	77.9
	ANN	80.3	84.3	86.2
	LSTM	82.5	85.6	86.5

long term for effective classification. The GA, PSO, and Firefly algorithms have the limitations of poor convergence and easily trap into local optima. The RF method has lower efficiency in more number of trees and overfitting problem in the less number of trees. The SVM method has the limitation of data imbalance problem and ANN has the limitation of redundant features in the classification. The EFF-LSTM method has 82.5 % accuracy and the Firefly-LSTM method has 76.3 % accuracy.

The proposed EFF-LSTM model is evaluated in the KTHTIPS-2b dataset compared to the existing methods, as shown in Fig. 4. The proposed EFF-LSTM model has higher performance compared to other feature selection and classifiers. The proposed EFF-LSTM model has the advantage of selecting the relevant features in the feature selection due to the replacement of the best fit firefly and LSTM stores the important features for the long-term to improve the efficiency of the classifier. The EFF-LSTM method has an accuracy of 82.5 % and the EFF-ANN method has 80.3 % accuracy.

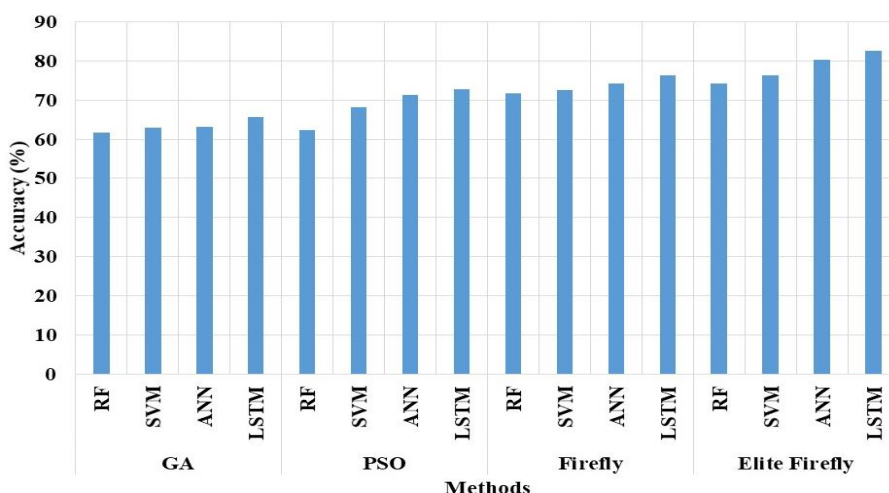


Figure. 4 The proposed EFF-LSTM model performance analysis in the KTHTIPS-2b dataset

5.1 Comparative analysis

The proposed EFF-LSTM method is compared with existing methods in textual classification to analyze the effectiveness of the model. The proposed and existing methods were compared in the UIUC and KTHTIPS-2b datasets.

The proposed EFF-LSTM model accuracy is compared with existing methods in UIUC and KTHTIPS-2b datasets, as shown in Table 3. Table 3 shows that the EFF-LSTM model has higher accuracy than existing methods in textual classification. The proposed EFF-LSTM model has the advantage of good convergence due to the

replacement of low fit firefly in the search process and the LSTM method store the important information in the long term for effective classification. The DSTNet [11], and CNN [12] have the limitation of overfitting and data imbalance problem. The ILTP[13], LES-GA [14] and texture feature [15] method have the limitation of redundant feature selection. The proposed EFF-LSTM model has the accuracy of 96.5 % and the existing CNN [12] model has 94.3 % accuracy.

The EFF-LSTM method accuracy is compared with existing methods in two datasets such as UIUC and KTHTIPS-2b datasets, as shown in Fig. 5. The proposed EFF-LSTM model has higher

Table 3. The proposed EFF-LSTM accuracy compared with existing methods

Methods	UIUC	KTHTIPS-2b
DSTNet [11]	93.6	61
CNN [12]	94.3	63.2
ILTP [13]	93.4	60.62
LES - GA [14]	92.1	58.4
Texture features [15]	93.2	60.4
EFF-LSTM	96.5	82.5

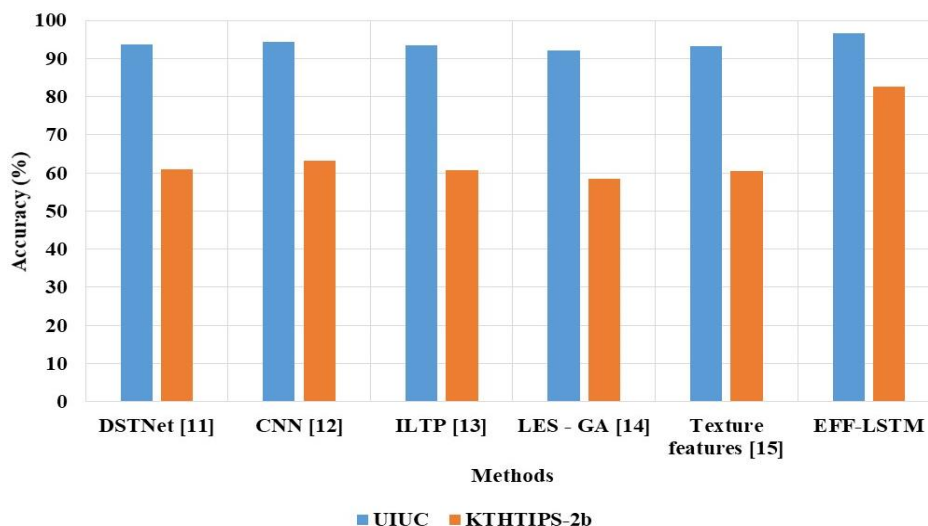


Figure. 5 The EFF-LSTM model accuracy comparison analysis

performance in both datasets due to the advantage of replacing the less fit firefly in the search process and store the important features in long term for classification. The existing methods have the limitation of low convergences in feature selection and overfitting problems in the classification.

6. Conclusion

Texture classification is required in many applications such as the medical field, face recognition, material identification, etc. The existing methods in texture classification have the limitation of poor convergence in the feature selection and overfitting problem in the classifier. In this study, the EFF-LSTM model is proposed to improve the efficiency of texture classification. The EFF-LSTM model has the advantage of replacing low fitness fireflies with high fitness fireflies to improve the convergence and LSTM stores the important features in long terms for classification to avoid overfitting problems. Two datasets such as UIUC and KTHTIPS-2b datasets were used to test the performance of the EFF-LSTM. The result shows that EFF-LSTM has higher accuracy compared to the existing method. The EFF-LSTM model has higher performance compared to the existing feature selection and classifier method in texture classification. The EFF method has higher efficiency

compared to the firefly algorithm in the feature selection for texture classification. The EFF-LSTM model has the accuracy of 96.5 % and the existing CNN has 94.3 % accuracy in the UIUC dataset. The future work of this method involves in apply deep feature extraction to improve the effectiveness of the model in texture classification.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper background work, conceptualization, methodology, dataset collection, implementation, result analysis and comparison, preparing and editing draft, visualization have been done by first author.

The supervision, review of work and project administration, has been done by second author.

References

[1] T. Song, J. Feng, S. Wang, and Y. Xie, "Spatially weighted order binary pattern for color texture classification", *Expert Systems with Applications*, Vol. 147, pp. 113167, 2020.

[2] Y. Almakady, S. Mahmoodi, J. Conway, and M. Bennett, "Rotation invariant features based on three dimensional Gaussian Markov random

- fields for volumetric texture classification”, *Computer Vision and Image Understanding*, Vol. 194, p. 102931, 2020.
- [3] S. K. Roy, D. K. Ghosh, S. R. Dubey, S. Bhattacharyya, and B. B. Chaudhuri, “Unconstrained texture classification using efficient jet texture learning”, *Applied Soft Computing*, Vol. 86, p. 105910, 2020.
- [4] N. Alpaslan and K. Hanbay, “Multi-resolution intrinsic texture geometry-based local binary pattern for texture classification”, *IEEE Access*, Vol. 8, pp. 54415-54430, 2020.
- [5] Y. Ruichek, D. Chetverikov, and R. Touahni, “O3S-MTP: Oriented star sampling structure based multi-scale ternary pattern for texture classification”, *Signal Processing: Image Communication*, Vol. 84, p. 115830, 2020.
- [6] A. Akoushideh, B. M. N. Maybodi, and A. Shahbahrani, “Features' value range approach to enhance the throughput of texture classification”, *IET Image Processing*, 2020.
- [7] S. Nithya and S. Ramakrishnan, “Wavelet domain directional binary pattern using majority principle for texture classification”, *Physica A: Statistical Mechanics and Its Applications*, Vol. 545, pp. 123575, 2020.
- [8] S. K. Roy, B. Chanda, B. B. Chaudhuri, D. K. Ghosh, and S. R. Dubey, “Local jet pattern: a robust descriptor for texture classification”, *Multimedia Tools and Applications*, Vol. 79, No. 7, pp. 4783-4809, 2020.
- [9] A. Ganesan and S. M. Santhanam, “Local Neighbourhood Edge Responsive Image Descriptor for Texture Classification Using Gaussian Mutated JAYA Optimization Algorithm”, *Arabian Journal for Science and Engineering*, pp. 1-20, 2021.
- [10] S. Veerashetty and N. B. Patil, “Novel LBP based texture descriptor for rotation, illumination and scale invariance for image texture analysis and classification using multi-kernel SVM”, *Multimedia Tools and Applications*, Vol. 79, No. 15, pp. 9935-9955, 2020.
- [11] J. B. Florindo, “DSTNet: Successive applications of the discrete Schroedinger transform for texture recognition”, *Information Sciences*, Vol. 507, pp. 356-364, 2020.
- [12] X. Bu, Y. Wu, Z. Gao, and Y. Jia, “Deep convolutional network with locality and sparsity constraints for texture classification”, *Pattern Recognition*, Vol. 91, pp. 34-46, 2019.
- [13] S. F. Ershad, “Bark texture classification using improved local ternary patterns and multilayer neural network”, *Expert Systems with Applications*, Vol. 158, p. 113509, 2020.
- [14] H. Ghazouani and W. Barhoumi, “Genetic programming-based learning of texture classification descriptors from local edge signature”, *Expert Systems with Applications*, Vol. 161, p. 113667, 2020.
- [15] M. Garg and G. Dhiman, “A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants”, *Neural Computing and Applications*, Vol. 33, pp. 1311-1328, 2021.
- [16] H. Zeng, R. Zhang, M. Huang, and X. Wang, “Compact local directional texture pattern for local image description”, *Advances in Multimedia*, 2015.
- [17] J. Liu, J. Li, J. Ma, N. Sadiq, U. A. Bhatti, and Y. Ai, “A robust multi-watermarking algorithm for medical images based on DTCWT-DCT and Henon map”, *Applied Sciences*, Vol. 9, No. 4, p. 700, 2019.
- [18] Z. Zainal, R. Ramli, and M. M. Mustafa, “Grey-level cooccurrence matrix performance evaluation for heading angle estimation of moveable vision system in static environment”, *Journal of Sensors*, 2013.
- [19] M. Tian, Y. Bo, Z. Chen, P. Wu, and C. Yue, “A new improved firefly clustering algorithm for SMC-PHD filter”, *Applied Soft Computing*, Vol. 85, p. 105840, 2019.
- [20] A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network”, *Physica D: Nonlinear Phenomena*, Vol. 404, pp. 132306, 2020.
- [21] E. Hayman, B. Caputo, M. Fritz, and J. O. Eklundh, “On the significance of real-world conditions for material classification”, *European Conference on Computer Vision*, pp. 253-266, 2004.
- [22] S. Lazebnik, C. Schmid, and J. Ponce, “A sparse texture representation using local affine regions”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 8, pp. 1265-1278, 2005.