



A Review of Local Feature Algorithms and Deep Learning Approaches in Facial Expression Recognition with Tensorflow and Keras

Kennedy Chengeta^(✉)

Computer Science Department, University of KwaZulu Natal, Durban, South Africa
216073421@ukzn.ac.za

<http://www.kaributechs.com/>

Abstract. In facial expression identification classification and lower processing times are key in choosing the algorithms to use in the facial detection, preprocessing, feature extraction or classification step. Facial expression recognition is based on deep learning, feature and holistic algorithms. Feature based algorithms like local binary patterns, local directional patterns (LDP) extract features from various facial components like nose, mouth or ears into a histogram. Deep learning involves using convolutional neural networks for image analysis with several hidden layers as opposed to artificial neural or shallow networks. The most popular models are AlexNet, VGG-Face and GoogleNet. The study evaluates computational accuracy and efficiency of deep learning algorithms and compares them to local feature based algorithms. The FER2013, Yale Faces, AT&T Database of Faces, JAFFE and CK+ datasets were used for analysis. Popular frameworks deep learning frameworks called Keras and Tensorflow backends are used to classify data and give better accuracy than a variant of local binary patterns. The processing time is shorter for feature based algorithms than the deep learning algorithms. To improve time on the deep learning approaches the study used pre-trained models to achieve greater accuracy with low execution times as well. A combination of preprocessed multi block binary patterns, PCA, multilayer perceptron, support vector machines and extra trees classifier gave competitive results to the superior established convolutional network for small datasets within a percentage range. Preprocessing used canny edge detection and histogram equalization.

Keywords: Keras · Tensorflow · Multi-block local binary patterns

1 Introduction

Use of automated facial expression has relied on improved classification methods like deep learning in order to reduce human bias and dependency. Wide scale

use has been witnessed in hospitals, social media, manufacturing with lean systems, oil industry as well as security, search and rescue operations [4, 6]. One's facial expressions is depicted in 7 different facial expressions namely surprise, fear, joy, contempt, sadness, neutral and anger. The recognition process involves facial detection, facial alignment, feature extraction, feature selection and classification [4, 6, 15, 17, 19]. Facial detection involves use of software and hardware systems to identify facial images from a human image from a video or static picture. There has been wide interest in facial recognition using deep learning as opposed to using feature based approaches [2, 8, 17]. The emergency of convolutional neural networks through use of GPUs has had a multiplier effect on their processing timelines and accuracy [1, 2, 20]. Transfer learning techniques have also been used on the facial expression datasets to reduce the training time by reusing trained data [7].

The study used a deep neural network to compare its accuracy with a feature based algorithm. The MB-Local Binary Pattern variant or multi-block local binary patterns approach classified the features using an ensemble of neural networks (multi-layer perceptron), extra-trees classifier and support vector machines [17]. The former has an input module, recognition module and output module. The study used popular frameworks Keras and Tensorflow as the backend.

2 Related Work

Facial expression recognition research has exceeded expectations in different fields like medical, travel, education, security and manufacturing [8, 14, 18]. The key expressions include sadness, anger, fear, neutral and disgust. The key stages include image detection using Viola Jones and Haar Cascade, preprocessing of images using histogram equalization and edge detection algorithms [9, 14]. The popular edge detectors to have been used with great popularity include canny edge detector, kirsch and LoG detectors. Deep learning algorithms or local feature based algorithms are then used to retrieve features and classify emotions from images [4, 12] and videos [8]. Local descriptors are good on images with varying illumination changes due to their use of grey scale images [1, 10, 13, 17]. Some of the popular feature based algorithms include local binary patterns and their variants like central-symmetric local binary patterns, multi-block local binary patterns, local directional patterns and rotational local invariant local binary patterns. Deep learning algorithms accuracy and performances have risen due to better processing power including Nvidia GPUs and other better processing devices. Deep learning algorithms use loss functions through the use of softmax function. Activation functions are also more varied from sigmoid function to rectifier RELu and Tanh function [2, 4].

2.1 Local Feature Extraction Methods

Feature based algorithms analyze facial components as separate components like mouth, nose or forehead and the features are aggregated using a histogram. Local Binary and Directional Patterns are popular algorithms to have been used

in facial expression recognition [12, 14]. Different LBP variants were successfully used in facial expression recognition that include, TLBP or Ternary Local Binary Patterns, Over-Complete Local Binary Patterns (OCLBP) and ELBP or elliptical local binary patterns and rotational local binary patterns [3, 12, 17, 20]. These eliminated the challenges of basic LBP algorithm like illumination or rotation [13, 14]. **Multi-Block Local Binary Patterns (MB-LBP)** uses rectangular regions in encoded format to derive their local binary operator to enable local structure image diversity [10–12]. The block regions are used in place of one pixel. The algorithm divides the input as horizontal/spawn processes (Fig. 1).

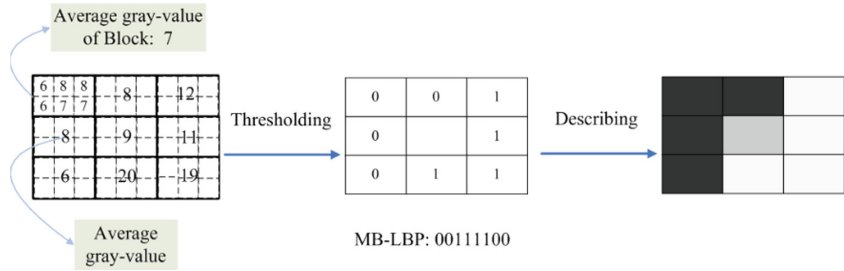


Fig. 1. Multi-Block Local Binary Patterns (MB-LBP) [14, 17]

The algorithm also encodes the image’s micro and macrostructures. The average of sub-region blocks is used to remove locality disadvantages [11, 12]. The algorithm is based on the Haar-like rectangular features and represented in the form below [10, 14, 17]:

$$MB - LBP(X, Y)(x_c, y_c) = \sum_{X=1}^{X=8} 2^X s(g_x - g_c), s(x) = \begin{cases} 1, & (x \geq 0) \\ 0, & (x < 0) \end{cases} \quad (1)$$

2.2 Artificial Neural Networks

Artificial neural networks have been used to recognize facial expression with relative success. The ANN is a black box model with labelled inputs and output with set of predicted vectors grouped as probability distribution labels. It consists of artificial neurons based on biology [15, 16]. The data is fed through dense networks that include hidden layers [15, 16]. The initial layer is given as a parameter of the initial dense layer and hidden layer can be one or more layers. Single layers are termed shallow networks and deep networks for multiple hidden layered networks [15, 16]. The algorithm has the output layer where data is returned from the network. Several activation functions successfully used in deep networks include RELu, Tanh and sigmoid functions. Inputs are assumed standard and standardization to a mean of zero and variance of 1 is always recommended [16]. The output values from the neural network are in either in binary, continuous or categorical/multiple values form. Popular algorithms include the feed forward and backpropagation algorithms [15, 16, 18, 20].

Feed Forward and Backpropagation. Feed forward or multi-layer perceptron neural networks are shown as linear (regression) and nonlinear (classification) models and have generalized activations functions. The layers undergo affine transforms and normally they contain a single hidden layer represented with a continuous function as shown in the following equation [2, 15, 19, 20].

$$y(x, w) = f\left(\sum_{j=1}^m w_j \theta_j(x)\right) \quad \mathbf{z}^{(l)} = \mathbf{W}^{(l)} \cdot \mathbf{a}^{(l-1)}, \quad \mathbf{a}^{(l)} = \sigma(\mathbf{z}^{(l)}). \quad (2)$$

Backpropagation networks handle nonlinear problems through use of partial derivatives on errors for the given activation functions [15, 18, 20]. The current layer's error proportion is calculated and subsequent splits propagate errors to previous layers and nodes through assigning weights. Optimization of the errors is then done to minimize the errors [2, 15, 19]. The error fraction per weight is shown in the equations below:

$$E_{mse} = \frac{1}{M} \sum_{\mathcal{D}} \frac{1}{2} \|y - \hat{y}\|_2, \quad MSE = \left(\frac{1}{n}\right) \sum y_{true} - y_{pred}^2 \quad (3)$$

$$\log(\hat{y}_j) = \log\left(\frac{e^{z_j}}{\sum_{i=1}^n e^{z_i}}\right) = \log(e^{z_j}) - \log\left(\sum_{i=1}^n e^{z_i}\right) = z_j - \log\left(\sum_{i=1}^n e^{z_i}\right) \quad (4)$$

3 Deep Learning

Deep learning convolutional neural networks produced exceptional results in image classification to create a boom in classification. This was aided by rise of GPU machines as well as improved processing power. The rise of cognitive cloud solutions like AWS Cognitive solution, Microsoft Azure Cloud and other cloud solutions [1] allowed for cheaper infrastructure models based on per use basis to execute deep learning models. Key advantages include huge storage advantages for readily available web data. With CNN one or more convolutional layers are used together with a pooling and fully connected layers [5]. Key deep learning models include AlexNet, VGG-Face and GoogleNet [1, 2] and a fusion with feature based algorithms has improved accuracy [3]. Deep learning enables feature extraction and classification together based on multilayer networks (hidden, input and output) as well use of Softmax to allow for classification using a probability function model [1, 5] and this is depicted in the following equation:

$$S(a, b) = (V * Z)(a, b) = \sum_x \sum_y Z(a - x, b - y) V(x, y) \quad (5)$$

Convolutional neural networks are based on mathematical convolutions for several layers of filters and kernels(k) and for an image in two dimensional format the neural network at time t given values u and s, the value s(t) is shown as below: [2, 4]

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t - a) \quad (6)$$

$$i \otimes k = \sum_{y=0}^c \left(\sum_{x=0}^r i(x - a, y - b)k(x, y) \right) \tag{7}$$

The convolutional neural networks are made up of one or more pooling layers and output layers [2,4]. Key pooling options include max pooling, L2 norm pooling as well as average pooling. The output layers receive features from a multitude of hidden layers to generate output classes and error predictions are based on given loss functions [5,6] in a single forward and backward pass cycle [2,3].

Convolutional Layers. The convolution filters denoted $f(k)$ are determined through sharing weights of nearby neurons allowing for smaller weights to be trained. Through max pooling, the input is reduced by applying the maximum function on the input.

Activation Function. Activation functions map a node’s inputs to its outputs through transformations in hidden layers. They derive the weighted sum and also enhance it though bias in a nonlinear functional model. Key hidden layer activation functions include Rectified Linear Unit (ReLU), Sigmoid and Tanh [5] and are key in deciding whether a neuron can be activated or not [6,8]. The sigmoid function or logistic denoted with ‘S’ shaped graph has smooth edges in some parts [1,22]. The tanh function allows for back propagation through a hyperbolic function. The Rectifier function (ReLU) allows for back propagation of errors and activates neurons on different layers [2]. The activation function allows non-linearity into the key functions as shown in the following equations.

$$o(x) = \sigma(w_0 + \sum_h \sigma(w_o^h + \sum_i w_i^h x_i)) \quad Y = activation(\sum(W * I) + b) \tag{8}$$

$$sigm_{f(x)} = (1 + e^{-x})^{-1} \quad tanh(x) = 2 \, sigm(2x) - 1 \quad reluf(x) = max(0, x) \tag{9}$$

Softmax Function. Softmax is used in the output layer based on the sigmoid function or multiclass regression to classify images in computer vision. The layer’s nodes match the output layer nodes where the classifier returns class probabilities that add up to 1 in a probability distribution and in normalized form [1,2,21,23]. In cases of binary classification, a logistic regression model is applied as $yk = \sigma(ak)$ and multiclass models(Maximum Entropy Classifier) softmax based on an extended multi-class logistic regression model is applied [4,21]. The latter uses decimal probability models whose sum is 1 and follows $[1, -2, 0] \rightarrow [e^1, e^{-2}, e^0] = [2.71, 0.14, 1] \rightarrow [0.7, 0.04, 0.26]$.

$$\sigma_x = \Pr(Y_i = k) = \frac{e^{\beta_k \cdot \mathbf{X}_i}}{\sum_{0 \leq c \leq K} e^{\beta_c \cdot \mathbf{X}_i}} \quad for \ i = 1 \dots k \quad and \ z = (z_1 \dots z_k) = R^k \tag{10}$$

3.1 TensorFlow, Theano and Keras

The Google Open Source library, Tensorflow is Python and C++ language driven [21,22]. The framework is used to recognize facial expressions using GPU or CPUs. The mathematical operations are defined as nodes and edges are used to define node input and output and association. The data is transferred using tensors, a multi layered array. Execution is done asynchronously and in parallel [21,23] (Fig. 2).

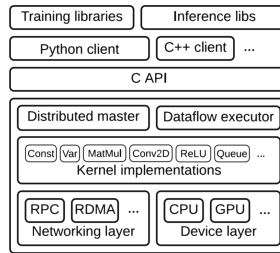


Fig. 2. TensorFlow architecture diagram [21,23]

The input node values and their connected proportions/weights are used as the aggregate inputs. where y takes values $y = f(\sum_{i=1}^D w_i * x_i) \dots y = f(w_1 * x_1 + w_2 * x_2 + \dots w_D * x_D)$.

- *# Define predictions based on slope and intercept variables*
- *pred = tf.add(tf.multiply(m,x), b)*
- *loss_x = tf.reduce_mean(tf.pow(pred - y, 2)) # Create loss function*
- *optim = tf.train.AdamOptimizer(learning_rate = 0.01).minimize(loss_x) # optimizer*
- *with tf.Session() as sess: sess.run(init) # Initialize TensorFlow session*

An optimizer like Stochastic Gradient Descent (SGD), ADAM and RMSprop is required and the loss algorithm MSE used to minimize errors in the model.

Keras on TensorFlow. Whilst tensorflow uses graph execution based on sessions and maintains state using variables, Theano based programs train and run simple neural networks based on fully connected layers (with, convolutional, max pooling and softmax) [3,21–23,23]. They activate neurons using activation functions (sigmoid, tanh, and rectified linear units). Keras is a modular high level framework that runs open neural networks on top of Tensorflow or Theano backends [21,23]. The latter is a high level framework and the former are low level complex apis. The Keras model has multiple layers with a defined network graph and the sequence is shown as below [21,23]:

- *Import 'Sequential' from 'keras.models' from keras.models import Sequential*
- *Import 'Dense' from 'keras.layers' from keras.layers import Dense*

- Initialize a constructor sequential model
- Add an input layer `model.add(Dense(12, activation is relu, input_shape=(11,)))`
- Add a hidden layer `model.add(Dense(8, activation is relu))`
- Add an output layer `model.add(Dense(1, activation is sigmoid))`, `Dropout(0.5)(x)`
- Compile model `model.compile(optimizer is rmsprop' and loss is mse)`

Transfer Learning. Speed of execution is based on resources and data availability challenges for neural networks which rely on huge datasets. To reuse previous models, transfer learning has been used in facial expression recognition based on pre-trained models. Transfer learning enables pre-trained feature extractors in deep learning using convolutional layers and only the output layers are replaced/altered based on the data available [7].

4 Implementation Framework

The study used a Keras on Tensorflow approach on facial expression images and compares the results to a proposed local binary pattern variant, MB-LBP which is preprocessed with Gabor Filters, histogram equalization, edge detection and also feature reduction with PCA algorithm. The proposed approach used a classification approach based on support vector machines, stochastic gradient descent (SGD), multilayer perceptron and an extra trees classifier with a 3:1:2:2 classifier ratio. The 2 approaches were compared in terms of accuracy and processing time. The Keras deep learning approach uses GPUs. The SGD classifier is based on the `sklearn.linear_model.SGDClassifier python implementation`. The stochastic gradient descent also acts as an optimizing algorithm hence the choice and the support vector machine enables loss derivation.

Tools. The study used python to implement both the deep learning approaches and feature based approaches. Flask a mini web python framework was used to design the frontend for the solution. The implementation used python running on a docker engine based on linux. The docker was configured and run on an Amazon AWS cloud environment to take advantage of GPU use. Other popular frameworks considered included other popular frameworks like Apache MXNet, PyTorch and Microsoft Cognitive Toolkit but Amazon cloud environment was chosen due to the easiness of use.

Approach and Databases. The study compared the MB-LBP with PCA and Gabor Filters as well as the deep learning CNN algorithm in terms of accuracy of classification of the FER-2013 dataset, AT&T Database of Faces, Yale Faces, Cohn-Kanade (CK) and JAFFE databases. Facial detection was done using the Viola Jones Open CV detection algorithm. Feature extraction for the local feature approach was done using the algorithms MB-LBP algorithm. Classification

was through machine learning classifiers namely multilayer perceptron, support vector machines (SVM), SGD classifier and extra trees classifier all in weighted proportions [2]. The first step involved using histogram equalization and canny edge detector to preprocess the images. Gabor filters were then used for filtering before the feature based algorithm was used to extract expression features for classification (Fig. 3).

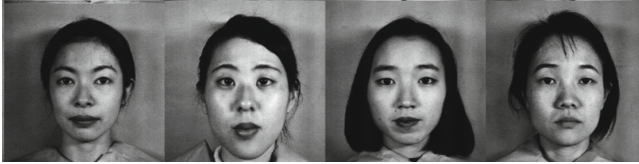


Fig. 3. Example expression images: JAFFE dataset. [24,25]

The study used FER-2013 dataset, AT&T Database of Faces, Yale Faces, Cohn-Kanade (CK) AU-coded expression dataset and JAFFE (Japanese Female Expression) datasets. The latter is made up of 10 Japanese subjects with 213 expression images giving 7 different expressions namely anger, sadness, neutral, fear, disgust and happiness. The Yale Faces has 165 grayscale images in GIF format and the CK+ dataset is based on a mixture of images of American, Latin and Asian descent. The AT & T Database of Faces is composed of 10 unique images of each of 40 distinct subjects. The deep learning computation involved using decay factor, a learning rate of 0.00001 and 3000 epochs which indicates when all queued images have been successfully processed. The deep learning datasets were normalized using *preprocessing.MinMaxScaler()* preprocessor. The pooling layers reduce spatiality of data and dense layers used the features from the convolution layers. The drop out layers remove some neurons to cater for overfitting and normalization is done using batch normalization by subtracting the mean and dividing with the batch mean [1].

Deep Learning Facial Expression Analysis Results. The deep learning algorithm executed was executed against the FER-2013 dataset, AT&T Database of Faces, Yale Faces, Cohn-Kanade (CK) AU-coded expression dataset and JAFFE (Japanese Female Expression) datasets and had accuracy values as shown in the following tables where 7 different expressions were the output class layers. The batch size was 64 and number of epochs varied but best results was found when 3000 epochs were used.

The Keras model was based on a 5 dense layers value 512, softmax activation and an input of 4624 compiled based on Stochastic gradient descent (SGD) with values *SGD(lr = 0.0001, decay = 1e-6, momentum = 0.9)*. The optimiser was extended with the adaptive learning rate based on the Adam(Adaptive Moment Estimation) optimiser with value of *keras.optimizers.Adam(lr = 0.00001)*.

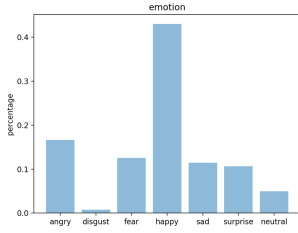


Fig. 4. Selected CK+ image (a)-happy mood

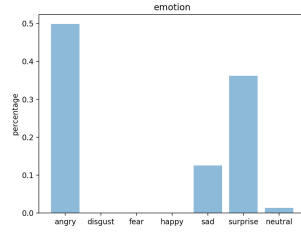


Fig. 5. Selected JAFFE image (b)-angry mood

The other options include Adaptive Gradient Algorithm (AdaGrad) and the Root Mean Square Propagation (RMSprop). The different optimisers *optimizers.Adadelta()*, *optimizers.Adagrad()*, *optimizers.Adam()*, *optimizers.Adamax()*, *optimizers.SGD()*, *optimizers.RMSprop()* were executed and the best results came by using the SGD and the Adam optimiser.

Deep Learning Vs Feature Based MB-LBP Algorithm		
Feature	MB-LBP	Keras backed TensorFlow
Design	Uses local facial regions, mouth, nose,eyes	Based on whole image
Speed	Faster, runs in sub minute times	Slower but speeds improving due to use of GPUs and Cloud, from days to hours
Light	Gray scale images reduce light impact	Affected by light
Error Detection	Use SVM to reduce loss impact	Use MSE to detect losses and rectify
Preprocessing	Edge Detection,HE, Gabor Filters, PCA	Images normalised to mean of 0 and a norm of 10
Dataset Size	Good for small and large datasets	Very good on large data sets and good on small datasets

The FER2013 dataset has 3 convolutional Conv2D(128) layers, dropout at 0.2, two FC dense(1024) layers all based on RELU activation function and softmax for the classification based on an input of (48, 48, 1). The Figs. 4, 5, 6 and 7 show the emotion distribution and confusion matrices for the 7 different expressions namely angry, fear, disgust, neutral, surprise, fear and happy.

The deep learning accuracy losses are shown in the following training loss against number of epochs executed. The deep learning algorithm gave an accuracy of 87.88%, 88.38% on the FER2013, Yale Faces, CK+ and JAFFE datasets. The loss function is shown in Figs. 6 and 8 as shown where the graph shows the number of epochs against the loss averaging at 1.5. The following diagram shows the loss function for both the training and testing phase for the deep learning test based on 100 images in this case and the training phase averaged just above zero from 1.5 and 0.7. For one facial image the distribution predicted showed high frequency on the happy class image (Fig. 9).

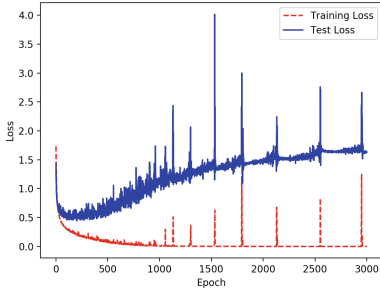


Fig. 6. Epochs = 3000, lr = 0.0001 loss function

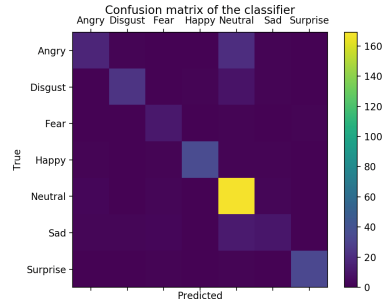


Fig. 7. 3000 epochs confusion matrix

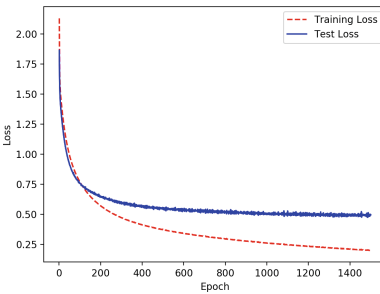


Fig. 8. Epochs = 1600, lr = 0.00001 loss function

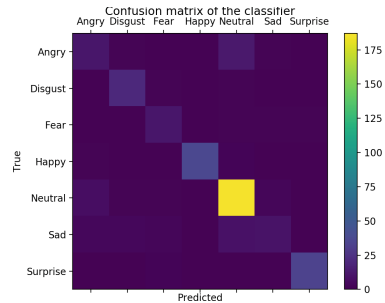


Fig. 9. 1600 epochs confusion matrix

The MB-LBP algorithm showed high accuracy with preprocessing with percentages of 87%, 85% and 86% respectively for CK+ and JAFFE, Yale and AT&T Faces on small datasets. Deep learning approaches showed better accuracy but with more processing power and execution time was longer. The accuracy of the deep learning algorithm was better than the feature based algorithm MB-LBP on the CK+ and JAFFE databases. For smaller datasets, the MB-LBP algorithm accuracy was almost the same as the convolutional neural network approach since deep learning algorithms work better on bigger datasets. This convolutional neural network approach was executed over 50, 100, 500, 1000, 2000 and 3000 epochs on the FER2013, Yale Faces, AT&T Faces, CK+ and JAFFE datasets.

4.1 Conclusion

Deep learning approaches showed much better accuracy than feature based local algorithms like local binary patterns and their variants. For smaller datasets as in this study, the accuracy was almost closely matched between the 2 approaches but on big datasets, the deep learning approach showed much improved accuracy compared to the local based approach. The Keras implementation

Accuracy and Performance Results				
Algorithm	Dataset	Preprocessors	Layers/Classifier	Accuracy(%)
Keras with Tensorflow	CK+JAFFE	Histogram Equalization		88.38%
	Yale Faces	Histogram Equalization		87.78%
	AT&T Faces	Histogram Equalization		89.33%
	FER2013	Histogram Equalization		75.09%
TensorFlow	FER2013	Histogram Equalization		76.43%
MB-LBP	CK+JAFFE	Histogram Equalization Canny Detector/GF	SVM,ExtraTrees, MLP,SGDClassifier	87.56%
	Yale Faces	Histogram Equalization Canny Detector/GF	SVM,ExtraTrees, MLP,SGDClassifier	85.33%
	AT&T Faces	Histogram Equalization Canny Detector/GF	SVM,ExtraTrees, MLP,SGDClassifier	86.64%
	FER2013 Kaggle	Histogram Equalization Canny Detector/GF	SVM,ExtraTrees, MLP,SGDClassifier	73.77%

is simpler and high level than the tensorflow alone approach but classification results are similar. Changing the learning rates to smaller values improves the results. The proposed approach in this study of MB-LBP, Gabor Filters, canny edge detector and Histogram Equalization classified by a weighted classifier of support vector machines, extra trees classifier and multi-layer perceptron showed a marked improvement of 4–5% points compared to a basic local binary pattern (LBP) on the FER2013 dataset. The feature based approach executes with much faster processing times than the Keras/Tensorflow approach. The deep learning approach showed improved processing times when executed on GPUs on the Amazon AWS cloud environment and also when transfer learning was used in the case of the FER2013 dataset. The study concludes that feature based approaches like local binary patterns' accuracy was marginally less compared deep learning approaches but processing times favour the feature based approaches.

References

1. Pramerdorfer, C., Kampel, M.: Facial expression recognition using convolutional neural networks: state of the art (2016). arXiv preprint [arXiv:1612.02903](https://arxiv.org/abs/1612.02903)
2. Tang, Y.: Deep learning using linear support vector machines. arXiv preprint [arXiv:1306.0239](https://arxiv.org/abs/1306.0239) (2013)
3. Levi, G., Hassner, T.: Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In: ACM International Conference on Multimodal Interaction, pp. 503–510. ACM, November 2015
4. Mollahosseini, A., Chan, D., Mahoor, M.H.: Going deeper in facial expression recognition using deep neural networks. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–10. IEEE, March 2016
5. Zhao, X., Shi, X., Zhang, S.: Facial expression recognition via deep learning. IETE Tech. Rev. **32**(5), 347–355 (2015)

6. Hemalatha, G., Sumathi, C.P.: A study of techniques for facial detection and expression classification. *Int. J. Comput. Sci. Eng. Surv.* **5**(2), 27 (2014)
7. Xu, M., Cheng, W., Zhao, Q., Ma, L., Xu, F.: Facial expression recognition based on transfer learning from deep convolutional networks. In: 11th International Conference on Natural Computation (ICNC), pp. 702–708. IEEE, August 2015
8. Kahou, S.E., et al.: Emonets: multimodal deep learning approaches for emotion recognition in video. *J. Multimodal User Interfaces* **10**(2), 99–111 (2016)
9. Tripathi, S., Acharya, S., Sharma, R.D., Mittal, S., Bhattacharya, S.: Using deep and convolutional neural networks for accurate emotion classification on DEAP dataset. In: Twenty-Ninth IAAI Conference, February 2017
10. Cai, Z., Gu, Z., Yu, Z.L., Liu, H., Zhang, K.: A real-time visual object tracking system based on Kalman filter and MB-LBP feature matching. *Multimedia Tools Appl.* **75**(4), 2393–2409 (2016)
11. Girish, G.N., CL, S.N., Das, P.K.: Face recognition using MB-LBP and PCA: a comparative study. In 2014 International Conference on Computer Communication and Informatics, pp. 1–6. IEEE, January 2014
12. Dhavalikar, A.S., Kulkarni, R.K.: Face detection and facial expression recognition system. In: 2014 International Conference on Electronics and Communication Systems (ICECS), pp. 1–7. IEEE, February 2014
13. Zhou, S., Yin, J.: Face detection using multi-block local gradient patterns and support vector machine. *J. Comput. Inf. Syst.* **10**(4), 1767–1776 (2014)
14. Zhang, L., Chu, R., Xiang, S., Liao, S., Li, S.Z.: Face detection based on multi-block LBP representation. In: Lee, S.-W., Li, S.Z. (eds.) *ICB 2007*. LNCS, vol. 4642, pp. 11–18. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74549-5_2
15. Iftikhar, S., Younas, R., Nasir, N., Zafar, K.: Detection and classification of facial expressions using artificial neural network. *J. Inf. Technol. Electr. Eng.* **3**, 18–22 (2014)
16. Chaudhari, M.V., Student, M.E., Bhusaval, S.S.G.B.C.O.E.T., Patil, Y.S., Patil, D.D.: Facial expression recognition using ANN & Gabor filter **1**(6) (2017)
17. Pietikinen, M., Hadid, A., Zhao, G., Ahonen, T.: *Computer Vision Using Local Binary Patterns*, vol. 40. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-0-85729-748-8>
18. Poornima, P., Radhapriya, S.: Survey of automatic facial recognition based on classification schemes **4**(10) (2017). ISSN: 2454-6933
19. Ruiz, L.Z., Alomia, R.P.V., Dantis, A.D.Q., San Diego, M.J.S., Tindugan, C.F., Serrano, K.K.D.: Human emotion detection through facial expressions for commercial analysis. In: Conference on Humanoid, Nanotechnology, (HNICEM) (2017)
20. Han, J., Kamber, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco (2001)
21. Gulli, A., Pal, S.: *Deep Learning with Keras*. Packt Publishing Ltd., Birmingham (2017)
22. Breuer, R., Kimmel, R.: A deep learning perspective on the origin of facial expressions (2017)
23. Xia, X.L., Xu, C., Nan, B.: Facial expression recognition based on tensorflow platform. In: ITM Web of Conferences, vol. 12, p. 01005. EDP Sciences (2017)
24. The Japanese Female Facial Expression (JAFFE) Database. <http://www.kasrl.org/jaffe.html>
25. Kanade, T., Tian, Y., Cohn, J.F.: Comprehensive database for facial expression analysis. In: IEEE International Conference Automatic Face Gesture Recognition (2000)