

Deep Convolutional Neural Network Based Extreme Learning Machine Image Classification

G. D. Praveenkumar^{1*}, Dr. R.Nagaraj²

¹Research Scholar, Computer Science, Kaamadhenu Arts and Science College, sathyamangalam, Tamil Nadu, India

²Associate Professor, Computer Science, Kaamadhenu Arts and Science College, sathyamangalam, Tamil Nadu, India

ABSTRACT

Article Info

Volume 8, Issue 5

Page Number : 30-38

Publication Issue :

September-October-2021

Article History

Accepted : 01 Sep 2021

Published: 05 Sep 2021

In this paper, we introduce a new deep convolutional neural network based extreme learning machine model for the classification task in order to improve the network's performance. The proposed model has two stages: first, the input images are fed into a convolutional neural network layer to extract deep-learned attributes, and then the input is classified using an ELM classifier. The proposed model achieves good recognition accuracy while reducing computational time on both the MNIST and CIFAR-10 benchmark datasets.

Keywords : Neural Network, Convolutional Neural Network, Feature Extraction, ELM Classifier, Image Classification.

I. INTRODUCTION

Neural networking is a computer device that recognizes trends. Their architectures are inspired by human structure and consist of three layers: input layer, hidden layer, and output layer. Each network layer is made up of interconnected nodes that perform computations [1]. Image classification is the method of categorizing images based on their characteristics. A function in an image may be the pixel intensity. The changes in pixel value among other things. An image is made up of smallest indivisible segments known as pixels, and each pixel has a strength known as the pixel intensity. The digital image is made up of RGB channels. To implement image processing, the RGB color channels have several pixels. Since the conventional approach

fails to manage digital images, the convolutional neural network (CNN) was introduced as a class of deep learning neural network.

CNN is requesting a significant breakthrough in image recognition. The artificial neural network, which is composed of multiple neurons, is incapable of extracting features from an image [14]. Images are generated by combining convolutional layers and deep learning. The CNN functions by extracting features from images. The features have not been trained. They are found as the network trains on a series of images. As a result, deep learning models are extremely reliable for computer vision tasks [5]. CNN is an artificial neural network technology that processes images using a convolution filter. CNN can perform image processing more effectively by

reducing the amount of weights used in the synthetic product [6, 7]. The CNN is a subset of the neural network embedding a convolutional structure of small respective fields. The CNN based classifiers recently have been considering as one of the most powerful classifiers [22]. ELM was proposed by Guang-Bin and Qin-Yu, which was aim to train single hidden layer feedforward networks (SLFNs). The performance of the SLFNs should be appropriate for the system model for data such as threshold value, weight, activation function so that higher learning can be performed[2,3]. In gradient-based learning approaches, all of the parameters are iteratively changed for the appropriate value [13]. ELM widely applied in a variety of learning problems, such as classification, regression, clustering, feature mapping [19].

In this paper, we concentrate on a classifier based on a deep convolutional neural network with extreme learning machine (DCNN-ELM). The first step is to extract features using the CNN layer, and then the feature values are fed into the ELM classifier to improve classification accuracy. ELM has a high training precision as well as a fast training rate in a short running time [15]. The proposed DCNN-ELM model is divided into two sections, feature extraction and classification, as illustrated in Figure1: At first, an input data is acquired into the CNN for feature extraction, and the image is then divided into output classes. The CNN utilizes the ReLU function as a nonlinearity activation function to pass the composite value to the output stage or the hidden stage, which transforms the cumulative value of the input information into a value within a specific range.

The ReLU feature also reduces overfitting issues and speeds up the overall convergence speed of the CNN architecture [14]. The softmax classifier is widely used in shallow neural networks, but the ELM classifier is used in the DCCN-ELM model to predict the class label. In this paper, the proposed model focuses on

multiclass classification and employs the ELM classifier to address the shortcomings of traditional approaches.

- The DCNN-ELM technique is presented to improve image classification accuracy.
- The DCNN-ELM technique is specific in that it uses extreme learning machine classifier hyper parameters to train the CIFAR -10 and MNIST datasets.
- The proposed deep learning method is used to reduce error and computing time.
- The ELM classifier function is used in the output layer to classify input images into corresponding classes in order to enhance image classification accuracy.
- Using a standard benchmark dataset, we demonstrate that the proposed method achieved impressive outcomes.

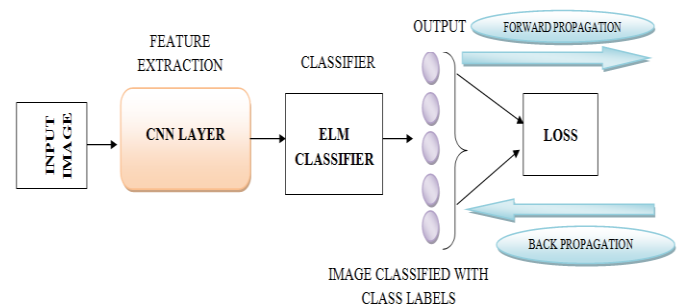


Figure 1 : Block diagram of proposed method

II. RELATED WORK

Y. Park *et al.*, [21] created CNN is taught layer by layer and blends the approach with an alternate technique of random convolutional filters and semi-supervised filters. By utilizing a deep network, the design automatically takes advantage of the efficiency of random and unsupervised feature extraction. However, because to speed and scalability difficulties, there are a huge number of hidden nodes. A.Khella *et al.*, [1] approach combine an ELM based learning algorithm to train the discriminative features

extraction and ELM classification. However the training of infrared images not improves the accuracy of classification performance. S.Ali *et al.*, [20] developed the CNN models incorporating a sigmoid activation function with a large number of derivatives have low efficiency in terms of feature extraction. The sigmoid function is upgraded with the ELM classifier along with ReLU function are used as the feature extractor. However, ReLU activation need for an efficient effective model in accuracy and computational time. The framework is used a minimum number of hidden layer.

R.Murgan *et al.*, [18] the proposed CNN method has trained and test with the publicly available COVID-19, pneumonia and normal dataset. The pre-trained method are used to classify the best image is occur covid or not. The CNN is combined with ELM based classifier is called EDiCoNet algorithm to applied medical fields. Wireless capsule endoscopy approach using CNN as a data driven feature extractor and the cascade ELM as a strong classifier in the deep CNN classification system. The ELM methods are used in supervised manner with hybrid system by cascading CNN feature extractor and ELM classifier.

Jung-Soo Han *et al.*, [14] created a model that integrated CNN with a ReLU-based ELM classifier. The CNN model, which employs a pre-trained architectural model to learn hundreds of images, is simple to use for feature extraction and image classification. The ReLU function is used to determine the best output weight for an ELM classifier. H.Chang Shin *et al.*, [9] created a deep convolutional network to solve a CAD problem. The approach employs transfer learning and a pre-trained ImageNet for the identification and categorization of lung illnesses. CNN is being evaluated for two separate CAD applications of lung disease. All of the researches have been suggested for image classifications based on benchmark datasets employing CNN, which are

exposed to a substantial disadvantage despite being an excellent image handling system.

III. DEEP CONVOLUTIONAL NEURAL NETWORK BASED EXTERME LEARNING MACHINE IMAGE CLASSIFICATION

The CNN models have proven to be successful in a variety of image processing applications. CNN are more efficient visual frameworks that produce feature hierarchies. Table I describes the detailed steps of each layer in the proposed DCNN-ELM, which consists of two convolutional layers, two pooling layers, two fully connected layers, and an ELM classifier.

Table I. Description of the CNN-ELM architecture

LAYERS	PARAMETERS
CONVOLUTIONAL LAYER_1	Filters:5x5x32 Stride:1 Activation function: ReLU
POOLING LAYER_1	Method: max Size:3x3 Stride:1
CONVOLUTIONAL LAYER_2	Filters:5x5x64 Stride:1 Activation function: ReLU
POOLING LAYER_2	Method: max Size:3x3 Stride:1
FULLY CONNECTED -1	Filters:1x1x1000 x 512 Stride:1
FULLY CONNECTED -2	Filters:1x1x1000 x 512 Stride:1
ELM CLASSIFIER	Classifier

The CNN focus the input feature map with a series of convolutional layer kernels, then appends the derivation to the output and applies a nonlinear activation function to the previous result. In the proposed approach, the ReLU function is used to display the unit value of the feature map. The training feature of CNN methods derived mathematically expressed using below,

$$x_j^l = f \sum_{i \in M_j} (x_i^{l-1} k_{ij}^l + b_j^l) \quad (1)$$

From the above mathematical equation (1) where l represent the number of convolutional layer f is the neural activation where the i^{th} feature map of the l layer can be obtained calculation, k represents the convolutional kernel, M_j represent a collection of input image b_j^l represent the bias matrix corresponding of the j^{th} characteristics graph of the layer.

The pooling layer is commonly associated with the convolutional layer to reduce the dimension of feature parameters and achieve translation invariant effects. The sub sampling layer perform on max pooling operation on the feature map of the previous convolutional layer to obtain in a new feature map is calculated by equation 1 and sub sampling of feature derived mathematically expressed using below,

$$x_j^l = f(\beta_j^l \text{down}(x_j^{l-1}) + b_j^l) \quad (2)$$

From the above mathematical equation (2) where x_j^{l-1} represent the characteristics graph obtained after the pooling function, β_j^l is multiplicative factor, b_j^l is the bias matrix and down represent a pooling function. The most common pooling operation is maximum pooling. The feature map values are transformed into fully connected layers after the convolutional layer and pooling layer. It is a neural network in which every neuron in the previous layer connects with every neuron in the next layer from

equation 1 and 2 features derived mathematically expressed using below,

$$x_{ij} = f(\sum_{l=0}^{n(l-1)} x^{l-1} \cdot w_{ji}^l + b_j^l) \quad (3)$$

From the above mathematical equation (3) where $n(l-1)$ is the number of neuron in the preceding layer $(l-1)$, w_{ji}^l is the weight for connection from neuron i in layer $(l-1)$ to neuron j in layer (l) and b_j^l is the bias of neuron j in layer and f represent the activation function of layer (l) .

The values are then fed into the ELM classifier. ELM was originally designed for single-hidden layer feed forward neural networks [10, 11]. ELM was then applied to generalized single-hidden layer feed forward networks, in which the hidden layer could be neuron-like or not. Since the input weights and hidden node biases are obtained at random, the ELM prediction model can be trained hundreds of times faster than traditional neural nets, and its output layer weights can be computed analytically using a least-squares approach [12].

The most prominent thing of ELM is that the hidden layer parameters are chosen at random. It is assumed that training of SLFNs with K hidden neurons and activation function $g(x)$ to learn N distinct samples (x_i, t_i) , where (x_i, t_i) , where $X_i = [X_{i1}, X_{i2}, \dots, X_{in}]^T \in R^m$ and $T_i = [T_{i1}, T_{i2}, \dots, T_{in}]^T \in R^m$.

In DCNN-ELM, the input weights and hidden biases are at random, generated in its place of tuning. In its responsibility so, the linear system has to be converted to a non-linear system derived mathematically expressed using below,

$$H\beta = T\hat{\beta} = H^+T \quad (4)$$

The minimum norm solution is distinct which has the smallest norm between all the solutions. In the case of learning an arbitrary function with zero training error, Baum had presented several constructions of SLFNs with sufficient hidden neurons. Even though, whilst simulation, the number of hidden neurons required achieving a proper generalization performance on novel patterns is less when compared with other existing solutions. The resulting training error might not approach to zero but can be minimized by solving the following problem derived mathematically expressed using below,

$$\min_{W_i, b_i, \beta} \|H(w_1, \dots, w_N, b_1, \dots, b_N) \beta - T\|^2 \quad (5)$$

Where

$$T = [t_1^T \dots \dots t_N^T] N \times M \quad (6)$$

From the above mathematical equation (5,6) DCNN-ELM randomly assigns and fixes the input weights w_i and biases b_i based on some continuous probability distribution function in the case of learning a structured function, only leaving output weights β_i to be adjusted according to derived mathematically expressed using below,

$$\min_{\beta} \|H\beta - T\|^2 \quad (7)$$

From the above mathematical equation (7), the above problem is well established and known as a linear system optimization problem. Its unique least-squares solution with minimum norm is derived mathematically expressed using below,

$$\hat{\beta} = H^+T \quad (8)$$

The generalization performance of a SLFNs tends to be better with smaller magnitude of output weights. From this sense, the solution produced by DCNN-ELM in the above mentioned equation achieves the minimum square training error along with the best

generalization performance on novel patterns. The ELM is summarized as follows:

- (i) Given a training set $X = \{(X_1(k, t_1 k) | X_1 k \in R^{\uparrow} n, t_1 k \in R^{\uparrow} m, k = 1, \dots, N)\}$ an activation function $g(x)$ and the number of hidden neurons \hat{N} ,
- (ii) Assign input weights w_i and biases b_i according to some continuous probability density function.
- (iii) Calculate the hidden layer output matrix H .
- (iv) Calculate the output weights

$$\beta_i: \hat{\beta} = H^+T \quad (9)$$

In the experiments with DCNN-ELM, the activation function is ReLU function derived mathematically expressed using below,

$$ReLU(F) = Max(0, x) \quad (10)$$

From the above mathematical equation (10), the probability density function (F) is a uniform distribution function in the range from -1 to 1 and < 0 its break down the learning. ELM is as SLFNs with N hidden neurons are learn and mentioned as N that has distinct samples with zero error. Even if the number of hidden neurons (N) is lesser than the number of distinct samples (N), ELM manages by assigning random parameters to the hidden nodes and calculate the output weights using pseudo inverse of H resulting in occurrence of only a small error $\epsilon > 0$. The hidden node parameters of ELM a_i and b_i (input weights and biases or centers and impact factors) are tuned by the fuzzy logic mechanism during training and may simply be assigned with random values. The pseudo code of the DCNN-ELM procedure is summarized as in Algorithm 1.

Algorithm 1: DCNN-ELM CLASSIFIER			
Input:	MNIST(D1),	CIFAR-10(D2)	X ∈
	(D1...n, D2...n)		
Output:	Classification of input image		

```

1: ending epochs=200
2: initialize the neural network model with parameter
3: while (training epochs < ending epochs) do
4: for D1,D2 X in dataset do
5: preprocessing image // data augmentation
6: Extracting feature from CNN by using equation(1)
7: Extracting feature from pooling layer by using
equation(2)
8: Extracting feature from fully connected layer by
using equation (3).
9: ELM image classification by using equation(4-10)
10:if(testing data== classification result) then
    Classification image into corresponding class
    Calculate the accuracy , computation time
11:Else // incorrectly classified labels
    Updating gradient descent with backpropagation
algorithm
12:end if
13:end for
14: end while.

```

When the ELM training was completed, it was fine-tuned by starting with a subset of the dataset. The parameter model was optimized using a minibatch stochastic gradient descent (SGD) algorithm [8]. The batch size is set to 30, the learning rate is set to 0.001, and the epoch size is set to 200. The SGD algorithm is used to maximize the ELM' weight. The hidden layer's neuron number is set at 1000. If the least error or the maximum number of iterations is reached, the DCNN-ELM model training phase is terminated. The train data with hidden neurons input and output weight, biases and labels. Testing data and output weight are used in the classification of test data. If the test model accuracy is less than desirable, then fine-tune the hyperparameter of the backpropagation model applied to CNN with ReLU unit to solve complex learning tasks. Backpropagation from output to input will effectively calculate the basic concept of backpropagation in the gradient [5, 17]. The backpropagation algorithm is the most commonly

used form of learning algorithm. To assess the derivate of all training results, the ELM classifier hidden layer retrains gradient descent.

IV. EXPERIMENTAL RESULT

The experiments have been executed on MATLAB (2019 a) with Windows 8(64 bit) and the hardware requirements are Intel®-Core™ i7-2670 QM GHz Processor, 8GB Ram, Open Gl3.3 support graphics card with 1GB GPU memory running on Microsoft platform. Experiments carried out in benchmark datasets MNIST [16] & CIFAR-10 [4].MNIST dataset consist of 70,000 images with 28 * 28 pixels. It contains 60,000 training images and 10,000 testing images. The CIFAR-10 data set contain 60,000 images with 32*32 pixels. In CIFAR-10 dataset, there are 50,000 training images and 10,000 test images. The two dataset are divided into 70% of training, 30% testing and proceed the learning rate is 0.001. The performance of the model is based on the metrics like accuracy and computational time.

The overall accuracy is calculated by using below equation 11.The number of corrected classes divided by the total number of classes. The overall classification performance is function derived mathematically expressed using below

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

From the above mathematical equation (11), where TP denotes True Positive, TN denotes True Negative, FP denotes False Positive, and FN denotes False Negative. TP denotes a result in which the model accurately predicts the negative class. An FP is a result in which the model predicts the positive class incorrectly. FN is a function of the model incorrectly predicting the negative class.

During training, the neuron is executed in a hidden layer every iteration and excludes zero, random nodes are accessed in the subsequent activation function. Following the completion of the training operation, we will proceed to the testing operation, which will involve the use of all neurons. If incorrectly classified images occur in the datasets, we will use the backpropagation algorithm to retrain the model fine-

tuned hyperparameter to achieve the best class labels. The test results show that the ReLU function improves classification accuracy and allows the network to converge much faster than the sigmoid function. The accuracy of an image classification is determined by the accuracy of the training and testing total image classification when compared to the state-of-the-arts method.

Table II. Performance of accuracy on MINIST AND CIFAR-10 datasets

DATA SETS	ALGORITHMS	NUMBER OF HIDDEN NEURONS	TRAINING ACCURACY	TESTING ACCURACY
MINIST	ELM [20]	1000	0.9400	0.9300
	ELMCON_NET_1 [21]	1000	0.9735	0.9930
	PROPOSED	1000	0.9650	0.9750
CIFAR-10	ELM[20]	1000	0.7477	0.6452
	ELMCON_NET_1[21]	1000	0.8250	0.7650
	PROPOSED	1000	0.8992	0.7842

Table III

Performance of computational time on MINIST AND CIFAR-10 datasets

DATA SETS	ALGORITHMS	COMPUTATIONAL TIME
MINIST	ELM[20]	35.39
	ELMCON_NET_1[21]	34.70
	PROPOSED	32.16
CIFAR-10	ELM[20]	45.53
	ELMCON_NET_1[21]	50.57
	PROPOSED	42.17

Table II. shows the image classification precision for all classes of MNSIT and CIFAR-10 datasets, while Table III. shows the computational time for the various deep neural network techniques. The comparison is based on the model accuracy rate and computational time. The proposed approach outperformed the traditional models in terms of

training accuracy (0.9650 (MNSIT), 0.89920 (CIFAR-10) and testing accuracy (0.9750 (MNSIT), 0.7842 (CIFAR-10)). The proposed model classification accuracy was higher than the conventional methods. In Figure 2: shows the computational time of compared different neural network models, the proposed model reduces the computational time.

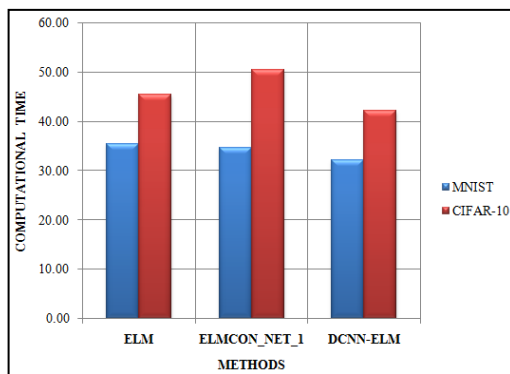


Figure 2 : Performance of computational time

V. CONCLUSION

In this paper, we concluded that deep neural networks can be used to classify images. The CNN is used to extract features first, and then the DCNN-ELM model trains the neural network using the gradient descent technique before the learning target accuracy is reached. The ELM classifier with ReLU activation function replaces the Softmax classifier in classifying the input images. When compared to previous works, the proposed approach obtained significantly improved results. In a deep learning-based approach, the proposed method achieves effective accuracy while reducing computing time. In future work, the proposed DCNN-ELM used the ReLU activation function during training if the number of hidden nodes is not met, causing the network to encounter a problem known as the vanishing gradient problem. The gradient weight updates the learning rate on a regular basis, and then we concentrate on pre-trained deep neural networks with various datasets to minimize memory and processing time.

VI. REFERENCES

- [1]. Atmane Khellal, Hongbin Ma, Qing, Convolutional Neural Network Based on Extreme Learning Machine for Maritime Ship Recognition in Infrared Images, *Sensors*, 18, 1490, 2018.
- [2]. B.Kaviyarasu, A.V.Senthil Kumar, Web Interaction Mining using Improved Extreme Learning Machine Classifier, *International Journal of Research in Science Engineering and Technology*, vol.3, no.12, pp. 45-51, 2016.
- [3]. C.he, H.Kang, T.Yao, X.Li, An effective classifier based on convolutional neural network and regularized ELM, *MBE*, vol.16, no.6, pp.8309-8321.
- [4]. CIFAR-10 and CIFAR-100 datasets: <https://www.cs.toronto.edu/~kriz/cifar.html>.
- [5]. Enrique S. Marquez, Jonathon S. Hare, and Mahesan Niranjan, Deep Cascade Learning, *IEEE Transactions on Neural Networks and Learning Systems*, 29, pp.5475 – 5485, 2018.
- [6]. G.D.Praveenkumar, M.Dharmalingam, Pruned Cascade Neural Network Image Classification, *International Journal Recent Technology and Engineering*, vol 8, pp.6454-6457, 2019.
- [7]. G.D.Praveenkumar, M.Dharmalingam, Recurrent Cascade Neural Network for Image Classification, *International Journal of Scientific and Technology Research*, vol.8, pp. 1009-1012, 2019.
- [8]. Hakan Cevikalp, Burak Benligitay, Omer Nezhik Gerek, Semi-Supervised robust deep neural networks for multi-label image classification, *Pattern Recognition*, vol.100, 107164, 2020.
- [9]. Hoo-Chang Shin et al., Deep CNN for Computer-Aided Detection: CNN Architectures, Dataset characteristics and Transfer Learning, *Arxiv*, pp.1-14, 2016.
- [10]. Huang GB, Zhu QY, Siew CK, Extreme learning machine: theory and applications. *Neurocomputing, Neural networks selected papers from the 7th Brazilian symposium on neural networks (SBRN '04)*, vol.70, no.13, pp.489–501, 2006.
- [11]. Huang GB, Zhu QY, Siew CK, Extreme learning machine: a new learning scheme of feedforward neural networks. In: *Proceedings of the*

- International Joint Conference On Neural Networks, pp.985– 990, 2006.
- [12]. J.Du et al., Supervised Extreme learning machine auto-encoder for discriminative feature learning, IEEE ,vol 8, pp.11700-11709, 2020.
- [13]. Jia et al ., A hybrid convolutional neural network with extreme learning machine for WCE image classification, In: Proceeding of IEEE , Robotics and Biomimetics,2015.
- [14]. J-S.Han, G-Bae cho, K-C.Kwak, A design of convolutional neural network using ReLU based ELM classifier and its applications, In: Proceeding of ICMLC, Springer, 2017.
- [15]. L.Li, K.zhoa, S.Li, R.Sun, S.Cai, Extreme learning machine for supervised classification with self paced learning, Neural Processing Letters, Springer, 2020.
- [16]. M.Malik, N.Roy, ELM-based image classification model using hand written digit database, application of artificial intelligence techniques in engineering, Advances in Intelligent System and Computing, 697.
- [17]. R.Amalraj, M.Dharmalingam, A Work Point System Coupled with Back-Propagation for Solving Double Dummy Bridge Problem, Neurocomputing. 168,pp.160-178, 2015.
- [18]. R.Murugan,Tripti Goel, E-DiCoNet: Extreme learning machine based classifier for diagnosis of COVID-19 using deep convolutional neural network, Journal of ambient intelligence and humanized computing ,2021.
- [19]. S.Madhusudhanan, S.Jaganathan, Jayashree, Incremental learning for classification of unstructured data using Extreme learning machine, Algorithms,vol.11,158, 2018.
- [20]. Saqib Ali et al., An effective and improved CNN-ELM for handwritten digit recognition and classification, Symmetry, vol.12, 1742, 2020.
- [21]. Y.Park,H.S.Yang , Convolutional neural network based on extreme learning machine for image classification, Neurocomputing, 339,pp.66-76, 2019.
- [22]. Y.Yoo,S.Y.Oh, Fast training of CNN classifier through extreme learning machine, IEEE, pp. 1702-1708, 2016.

Cite this article as :

G. D. Praveenkumar, Dr. R. Nagaraj, "Deep Convolutional Neural Network Based Extreme Learning Machine Image Classification", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 8 Issue 5, pp. 30-38, September-October 2021. Available at doi : <https://doi.org/10.32628/IJSRSET1218475> Journal URL : <https://ijsrset.com/IJSRSET1218475>