International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 08 | Aug - 2021 ISSN: 2582-3930

Detection of COVID-19 using Pattern Network

Arvind Kumar[1], Sartaj Singh Sodhi[2]

[1]PhD Scholar, USICT, Guru Gobind Singh Indraprastha University, Delhi

[2]Associate Professor, USICT, Guru Gobind Singh Indraprastha University, Delhi

***Abstract*:** Coronavirus disease of 2019 (COVID-19) is a disease caused by the deadly coronavirus. COVID-19 has probably originated in December 2019 from Wuhan, China. The spreading rate of a corona virus is very high. Timely detection of coronavirus is crucial in timely start of treatment. We detect corona virus disease with the help of neural network. Narin, Kaya and Pamuk [1], detected corona virus with the help of a deep convolutional neural network (DCNN). But DCNN takes more time, so we use pattern network to achieve an accuracy close to 96%.

Keywords: *Pattern-Network (Pat-Net), Deep-Network (DeepNet), Convolutional Neural Network (CNN), deep learning, corona virus, chest X-ray*

1. INTRODUCTION

COVID-19 is a disease which probably has started from Wuhan, China in December, 2019. More than 1,07,46,183 confirmed cases of the COVID-19, include 1,54,274 death cases upto 31st January, 2021 found in India alone [2]. The Coronavirus is caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [2]. Coronavirus is a very dangerous disease. In this paper, we use pattern network with a median filter to detect a coronavirus. The pattern network is a part of the Artificial Neural Network (ANN). Artificial Neural Network is inspired by the biological human brain. There are mainly three layers of ANN. They are inputs layer, outputs layer and one or more hidden layers. The learning process is an important process for the training of a network. Generally, there are two types of learning, firstly, supervised learning and secondly, unsupervised learning. In supervised learning, desired outputs are given for the problem at hand like recognition and classification purpose. But in unsupervised learning process, the desired outputs are not given. In the field of neural network, the first successful given by McCulloch & Pitts in 1943 [3,4,6,7]. A single-layer neural network given by Rosenblatt in 1958 has limited function and used for classification of linear patterns only. Then, Widrow and Hoff's in 1960 gave ADALINE and LMS algorithm. This algorithm is based on a single linear neuron with adjustable weights, limits the computing power of the algorithm. To remove the limitations of the perceptron and the LMS algorithm another network called Multilayer Perceptron (MLP), was developed by Minsky and Papert in 1996. With the help of the back propagation technique, Sekhon and Agarwal achieved 86% accuracy in 28462 iterations [5]. With the help of pattern net, Yousaf et al., [8], trained 19422, English alphabets sample and 7720 digits sample that is written through 150 different writers in various styles of handwriting and achieves 95.69% precision without using conventional feature extraction techniques. Bihday et al., recognized the handwritten image using Multilayer Neural Network [9]. The Multilayer Neural Network is also used for denoising the image [10]. Achkar et al., classified a Brain tumor using Back Propagation Algorithm in Multilayer Perceptron (MLP)[11]. Pacheco and Lopez [12], used the k-nearest neighbors (K-NN) algorithm with the MLP neural network for tomato color classification. Dino and Abdulrazzaq [13], got recognition rate at 93.53% with Support Vector Machine (SVM), 82.97% with MLP

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 08 | Aug - 2021 ISSN: 2582-3930

classifier and 79.97% with KNN classifier. Velioglu and Vural made a stacked denoised autoencoder with the help of fMRI dataset, and proved that this network is better than the PCA, ICA [14]. In [15], the authors classify the 3-D Functional Brain fMRI data with the help of the Convolutional Neural Network. Vu, Kim and Lee classify fMRI data with the help of 3D-CNN and proved that performance of 3D-CNN is superior compared to fully connected feedforward deep neural network (fcDNN) in the automated feature extraction [16]. Kamonsantiroj, Charoenvorakiat and Pipanmaekaporn extract features of fMRI data with the help of autoencoder [17]. Huang et al., [18], developed auto-encoder (DCAE) for extracting features from Task-based functional magnetic resonance imaging (tfMRI) data set. Deep learning Methods are 'blank boxes' in the field of medicine, because accountability is important here and it is often not enough to have a good prediction system [19]. Deep learning requires more data for processing, so performance grows only logarithmically with the amount of data [20]. L. Gondara also denoised medical image with the help of stacked denoising autoencoder [21].

2. Methodology

This network has input layers, output layers and one or more hidden layers. Each layer has its own weight matrix (**w**), its own bias vector (**b**), a net input vector (**n**) and an output vector (**a**). There are mainly three layers of neural network as shown in figure 1. In figure 1, first layer is called input layer, second layer is hidden layer(s) and third layer is output layer. Multi-layer neural networks are more powerful than a single-layer network [22]. Training of pattern network can be done by back-propagation algorithm. This network has also several layers. Each layer has its own weight matrix, its own bias vector, a net input vector and an output vector.

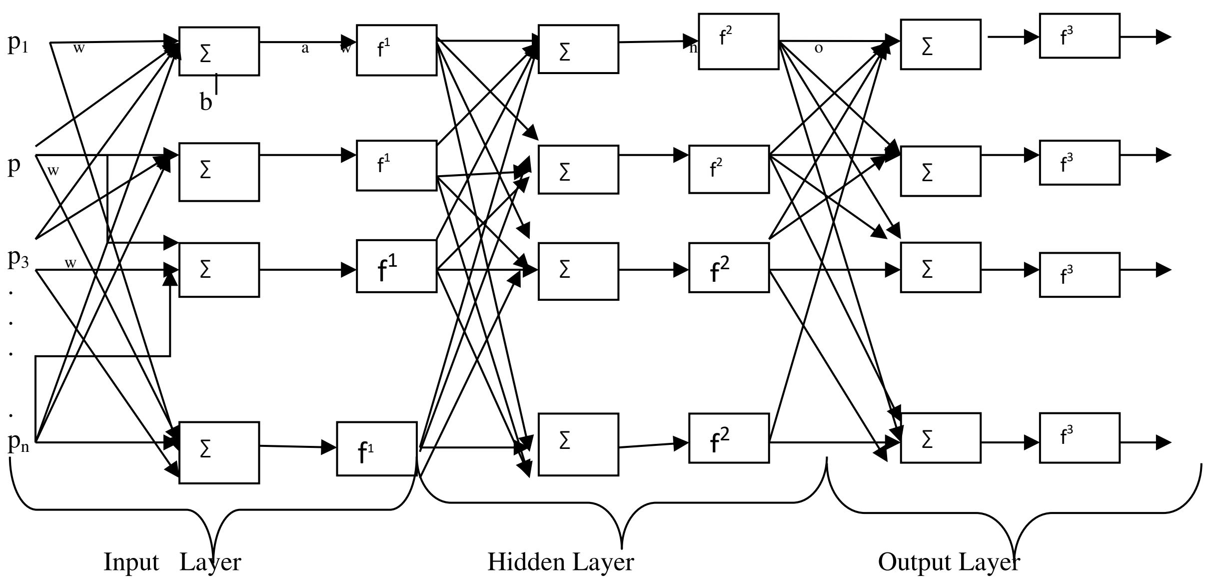


Fig.1 Pattern Network, having p_1, p_2, ...p_n as inputs. w represents weights and b is bias. ∑ is summation of product of inputs and weights and after summation we add bias into this result. f^1, f^2, ...f^n are the activation functions.

In this network, there are two phases of training [2]:

(i) Forward phase: In this phase layer-by-layer, the synaptic weights of the network are calculated and the input signal is propagated through the network until we reach the outputs.

(ii) Backward phase: In this phase, first of all we calculate,

Error = (given target) - (induced output)

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 08 | Aug - 2021 ISSN: 2582-3930

This error is again propagated through the network [layer by layer] in the backward direction. Again, we calculate the synaptic weights and output. For find of output we use equation (1).

$$v_j(n)=\sum_{i=0}^{m} wy \quad (1)$$

where, $w = w_{ji}(n)$ = weight or the first layer
and $y = y_i(n)$ = input of the first layer
m=total number of inputs (including bias).

For finding error, we use equation (2):

$$e_j = d_j(n) - y_j(n) \quad (2)$$

This error is again propagated with the help of chain rule given in equation (3)

$$\frac{d(\varepsilon j(n))}{d(wji(n))} = \frac{\partial \varepsilon j(n)}{\partial ej(n)} \times \frac{\partial ej(n)}{\partial yj(n)} \times \frac{\partial yj(n)}{\partial vj(n)} \times \frac{\partial vj(n)}{\partial wji(n)} \quad (3)$$

weight and bias are modified by the equations (4) and (5) as under:

$$w_{new} = w_{old} + \Delta w_{ji}(n) \quad (4)$$
$$b_{new} = b_{old} + \eta \text{ (error)} \quad (5)$$

where, $\Delta\ w_{ji}(n) = -\ learning\ rate * error * input$ (6)

3. EXPERIMENTS

3.1. Data acquisition: We take 92 chest X-ray images. Out of them 46 images are COVID-19 Chest X-ray images taken from the open-source GitHub repository (See Fig. 2a). Other 46 are normal Chest X-ray images taken from Kaggle website (Pneumonia) (See Fig. 2b).

3.2. Algorithm: (See flowchart 1.A and 1.B)

Step 1: We take all the images from database. (n-samples of one image, similarly n-sample of second image………upto m-images. i.e., total no of images = m x n)

Step 2: Following steps are done for pre-processing of images:

Step (a): Resize them into 20x20 small images so that processing can be faster, by using formula (imresize).

Step (b): Then we convert RGB to Gray by using formula (rgb2gray).

Step (c): For achieving good contrast and removing some illumination effect from the images, we do histogram equalization by using formula (histeq).

Step (d): Then, we do double operation on it.

Step 3: Then we save them into a new database named pre_processed_image database. This database consists m x n images and size of each image is 20x20.

Step 4: In the first layer of the network, we take size of inputs as 400. In the second layer of the network (i.e., hidden layer), we take hidden neurons as 30. Now, we initialize the weights and the bias. Then, we use tan sigmoid activation function in hidden layer and softmax activation function in the output layer. Outputs are either normal or coronavirus. Softmax function given in equation (7) is continuous and differentiable [5].

$$\text{Softmax}(z)_I = \exp(z_i)/\sum_j \exp(z_j) \quad (7)$$

Following steps create neural network:

Step (a): For the input layers, we take all the images from "pre_processed_image" database.

Step (b): We convert it into column_wise_ matrix.

Step (c): We take some initial weight and bias of the hidden layer.

Step (d): We take activation function of the hidden layer.

Step (e): For output layer, we take some initial weight & bias of this layer.

Step 5: Now, for performing a network, we take the following value in MATLAB:

```
net2.trainParam.min_grade = 1e-10;
net2.trainParam.goal = 1e-10;
```

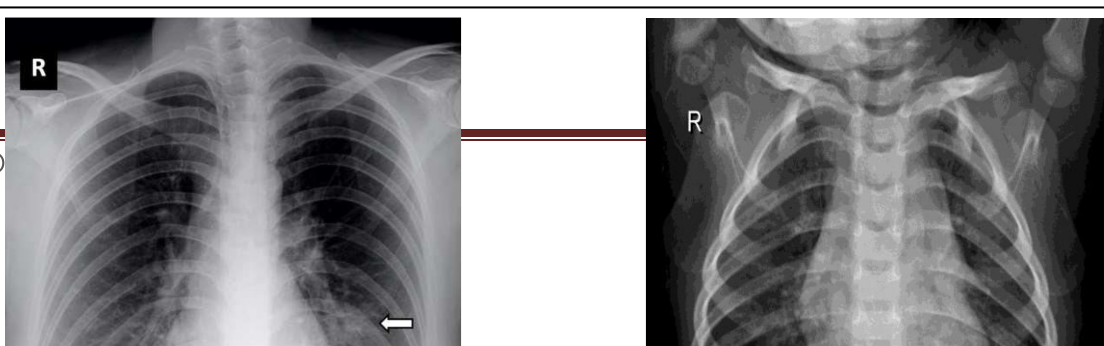
Step 6: Now, take some iteration (epochs) for proper assigning of weights & bias in MATLAB.

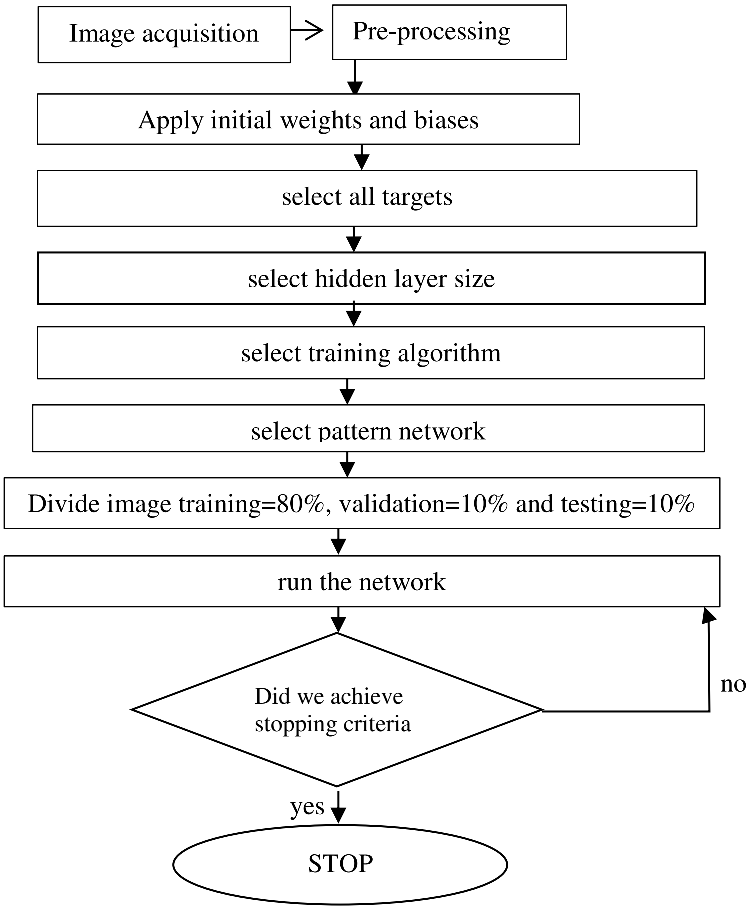
```
net2.trainParam. epochs = 1500;
```

Step 7: Then, we train this network in MATLAB using Levenberg-Marquardt algorithm. (After finishing training, we get figure 3.a)

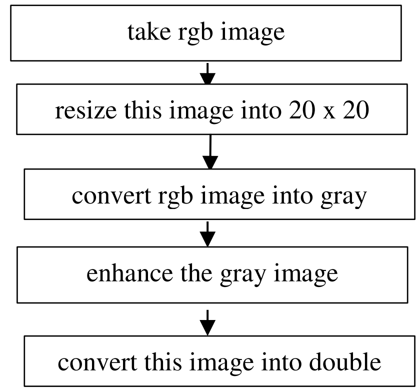
```
[net2, tr] = train (net2, inputs, targets);
```

After completing the above process, we get exact weight and bias of the network for both hidden and output layers.



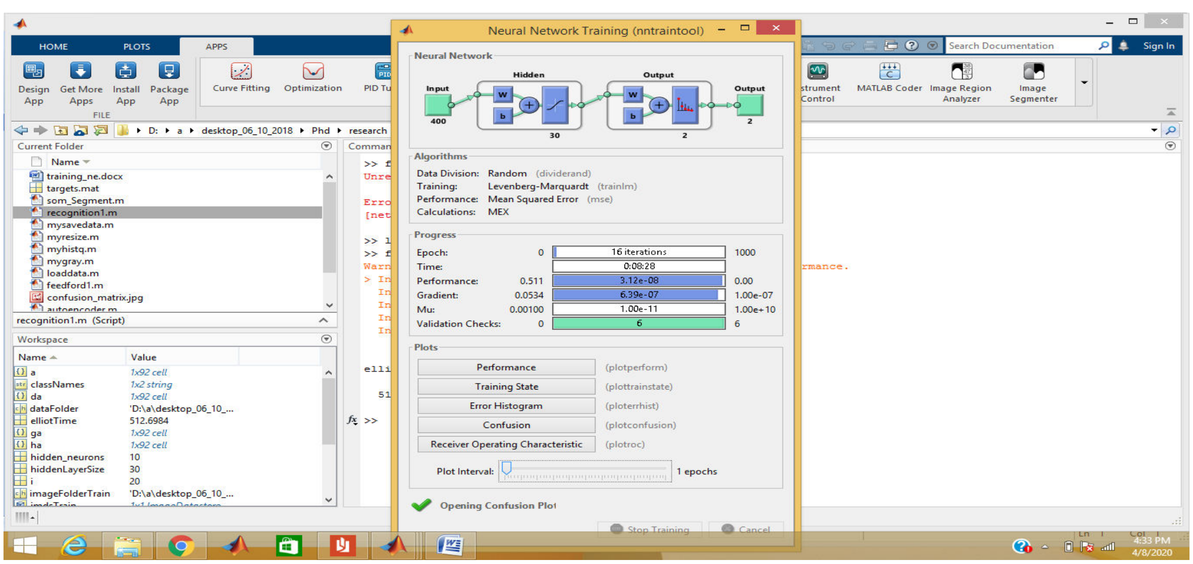


(A)



(B)

Flowchart 1 (A) Methodology of this experiment and (B) Image pre-processing



International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 08 | Aug - 2021 ISSN: 2582-3930

Fig.3. (a) Pattern Network made by MATLAB software

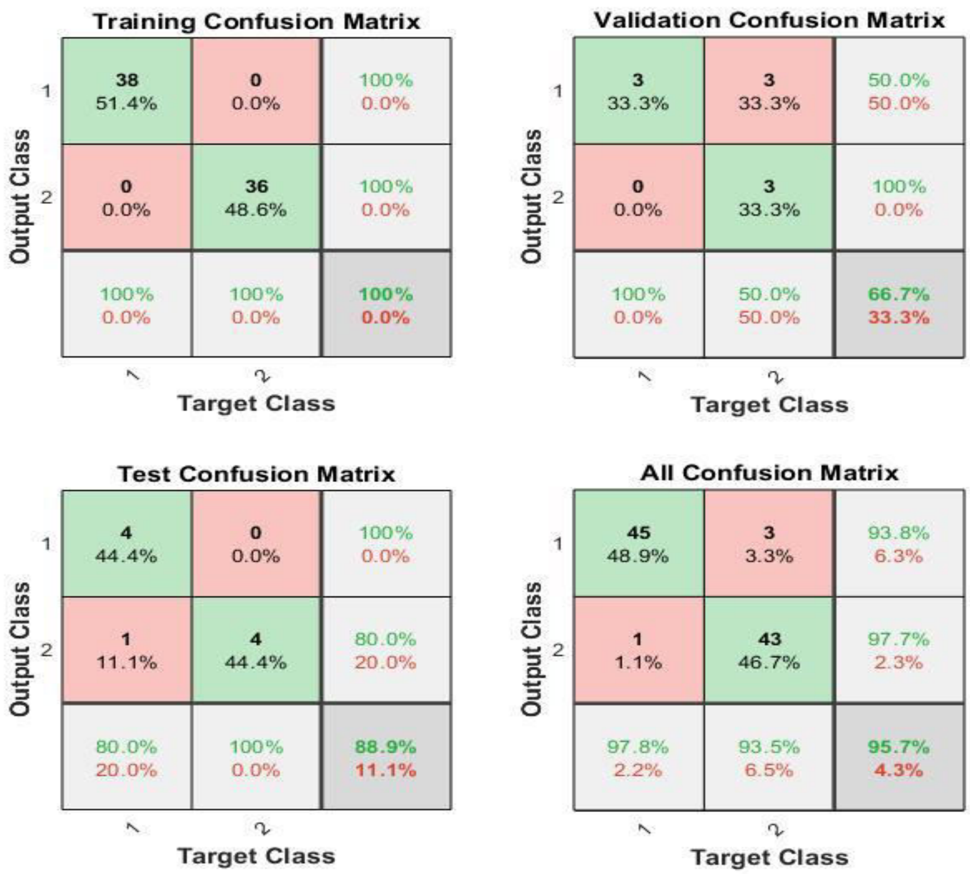


Fig.3. (b) Four confusion matrices.

4. RESULTS

With the help of 92 images and pattern network we made four confusion matrices We found the following results in all the confusion matrices:
In the figure 3.b, in the training confusion matrix, we found 38-images are True positive (TP) images and 36 images are True Negative (TN) images. This implies that we found 100% accuracy during training. In the validation confusion matrix, we found 3 images are True positive (TP) images, 3 images are True Negative (TN) images and 3 images are False Positive (FP) images. This means that we found 66.7% accuracy during validation. In the testing confusion matrix, we found 4 images are True positive (TP) images, 4 images are True Negative (TN) images and 1 image is False Negative (FN) image. This calculates to 88.9% accuracy during testing. In the all confusion matrix, we found 45 images are True positive (TP) images, 43 images are True Negative (TN) images, 3 images are False Positive (FP) and 1image is False Negative (FN) image. This leads to an accuracy of 95.7%.

5. CONCLUSION AND FUTURE SCOPE

Coronavirus is a very dangerous disease. So, it is necessary to find out which type of Chest X-ray image is a normal image and which type of Chest X-ray is a coronavirus infected Chest X-ray image. For this reason, we take two types of images: the first is a coronavirus infected Chest X-ray images and the second is a normal Chest X-ray image. After than we make a pattern network. With the help of this pattern network, we analyze all the images and we got 100% accuracy during training, 66.7% accuracy during validation, 88.9% accuracy during testing and we got 95.7% overall accuracy.

Future work can be done on a Pattern Network for a finding corona-19 virus with even higher accuracy. This work can also be extended to some modified version of the Pattern Network by altering training algorithm, activation functions and changing neurons in the hidden layer.

References

[1] A. Narin, C. Kaya and Z. Pamuk, "Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks", *arXiv preprint arXiv:2003.10849* (2020).

[2] https://www.indiatoday.in/india/story/coronavirus-india-reports-13-052-new-cases-127-deaths-in-last-24-hours-10-points-1764400-2021-01-31

[3] Martin T. Hagan, " Neural Network Design", 2nd Edition, Cengage Learning India, 2008.

[4] Simon Haykin, "Neural Networks, and Learning Machines", 3rd Edition, Pearson Prentice Hall, 2009.

[5] Mrs. A. Sekhon, Dr. P. Agarwal, “Face Recognition using Back Propagation Neural Network Technique”, IEEE, pp. 226-230, 2015.

[6] B. Yegnanarayana, "Artificial Neural Networks", Prentice Hall of India, 2006.

[7] I. GoodFellow, Y. Bengio and A. Courvlle, " Deep Learning", MIT Press, 2016.

[8] A. Yousaf, M. J. Khan, M. J. Khan, N. Javed, H. Ibrahim, K. Khurshid and K. Khurshid, “Size Invariant Handwritten Character Recognition using Single Layer Feedforward Backpropagation Neural Networks”, IEEE, 2019.

[9] V. Bihday, V. Brygilevych, Y. Hychka, Z. Liubun, N. Pelypets and V. Rabyk, "Recognition of Handwritten Images Using Multilayer Neural Networks," ***2019 XIth International Scientific and Practical Conference on Electronics and Information Technologies (ELIT)***, Lviv, Ukraine, 2019, pp. 59-62.

[10] A. Rubel, O. Rubel, V. Abramova, G. Proskura and V. Lukin, "Improved Noisy Image Quality Assessment Using Multilayer Neural Networks," ***2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)***, Lviv, Ukraine, 2019, pp. 1046-1051.

[11] R. Achkar, M. Owayjan, B. Harajli, D. Khazaal, M. Dbouk and G. Magnifico, "Brain Tumor Classification Using Back Propagation Algorithm in MLP," ***2019 Fourth International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)***, Beirut, Lebanon, 2019, pp. 1-5.

[12] W. D. N. Pacheco and F. R. J. López, "Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms K-NN, MLP, and K-Means Clustering," ***2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)***, Bucaramanga, Colombia, 2019, pp. 1-5.

[13] H. I. Dino and M. B. Abdulrazzaq, "Facial Expression Classification Based on SVM, KNN and MLP Classifiers," ***2019 International Conference on Advanced Science and Engineering (ICOASE)***, Zakho - Duhok, Iraq, 2019, pp. 70-75.

[14] B. Velioglu and F. T. Y. Vural, “Transfer Learning for Brain Decoding using Deep Architectures”, IEEE, pp. 65-70, 2017.

[15] D. Ren. Y. Zhao, H. Chen, Q. Dong, J. Lv and T. Liu, “3-D Functional Brain Network Classification using Convolutional Neural Networks”, IEEE pp. 1217-1221, 2017.

[16] H. Vu, H. C. Kim, J.H. Lee, “3D convolutional neural network for feature extraction and classification of fMRI volumes”, IEEE, 2018.

[17] S. Kamonsantiroj, P. Charoenvorakiat and L. Pipanmaekaporn, "Learning Representation for fMRI Data Analysis Using Autoencoder," ***2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)***, Kumamoto, 2016, pp. 560-565.

[18] H. Huang ***et al.***, "Modeling Task fMRI Data Via Deep Convolutional Autoencoder," in ***IEEE Transactions on Medical Imaging***, vol. 37, no. 7, pp. 1551-1561, July 2018.

[19] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. Setio, F. Ciompi, M. Ghafoorian, J. A.W.M. Laak, B. Ginneken and C. I. Sanchez, “A Survey on deep learning in Medical Image Analysis”, Elsevier, vol 42 pp. 60-88, 2017.

[20] A. Maier, C. Syben, T. Lasser and C. Riess, “A gentle introduction to deep learning in medical image processing”, Elsevier, vol 29, pp. 86-101, 2019.

[21] L. Gondara, "Medical Image Denoising Using Convolutional Denoising Autoencoders," ***2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)***, Barcelona, 2016, pp. 241-246..

[22] O. I. Abiodun ***et al***., "Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition," in ***IEEE Access***, vol. 7, pp. 158820-158846, 2019.