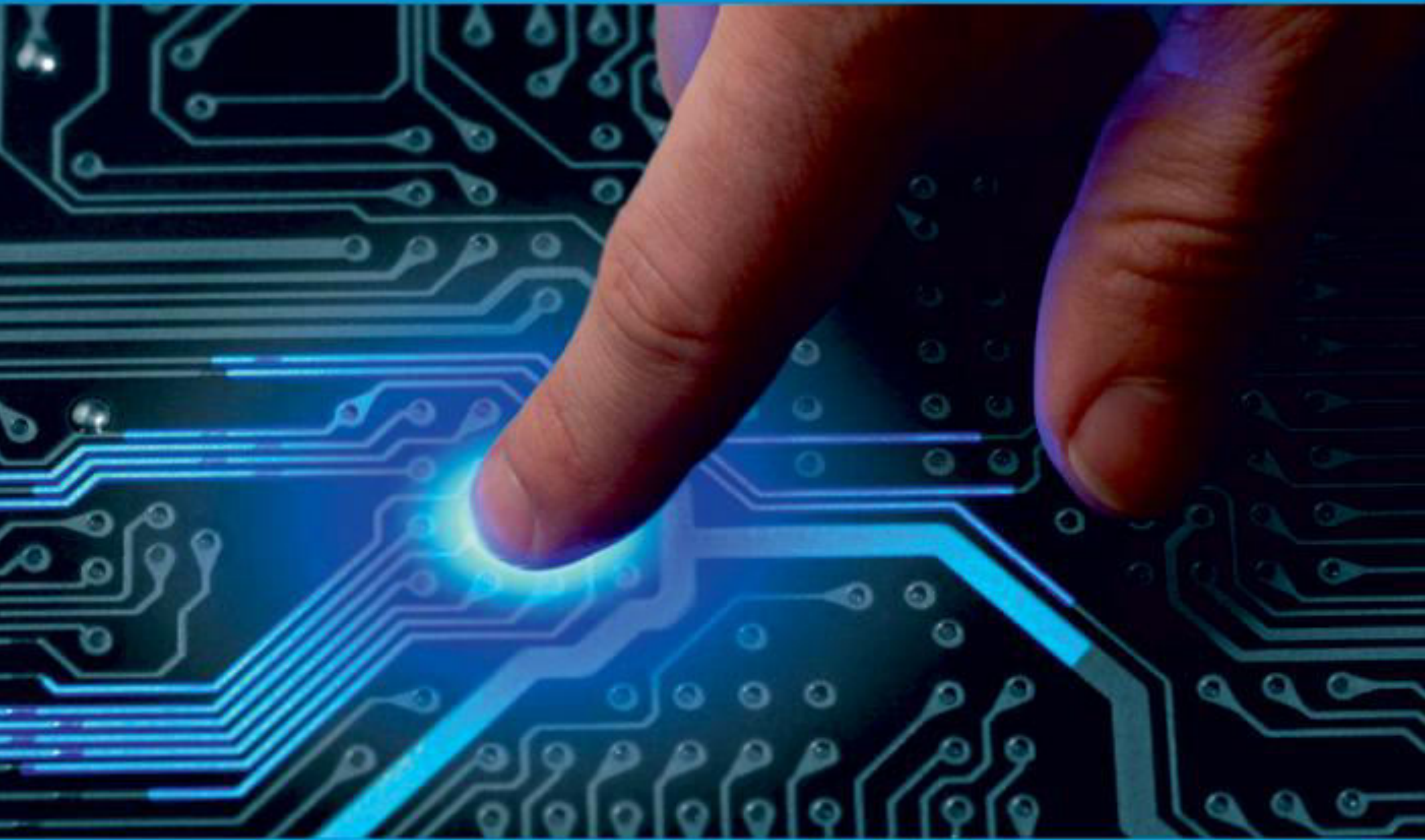
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 8, August 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.542

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 8, August 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0908028 |

Image Classsification using DCT and DWT in CNN

Anita Shirol, Dr Santosh Saraf

Digital Communication & Networking (M-Tech), KLS, Gogte Institute of Technology, Belagavi, Karnataka, India

Dept. of Electronics & Communication Engineering, KLS, Gogte Institute of Technology , Belagavi, Karnataka, India

ABSTRACT: Object detection with minimum computation time and lesser number of features is gaining importance from the last few years. As Conventional Neural Network (CNN) model is reliable yet data greedy, which needs thousands of images for training, lots of memory is needed to store the training data set hence not suitable for embedded devices and systems with lesser memory. To address this scenario instead of directly passing the images, it is converted into the DCT and DWT co-efficient to the CNN network for training purpose. The time taken for compilation is lesser compared to the time taken by the CNN with images. Since single image is divided into various components of DWT (LL, HL, LL, LH) and a set of DCT coefficients. The system is trained for this co-efficient a separate model is generated from every co-efficient mentioned above. All these models are used for testing purpose out of six results which classes gets the larger predication class number that is considered as the final predication class. The proposed framework is implemented in pycharm with the system configuration of 4GB RAM, 3rd Generation Intel Processor with training time of five minutes with the accuracy of 94%.

KEYWORDS: DCT, DWT, CNN, Model, Accuracy, minimum computation.

I. INTRODUCTION

The conventional CNN model takes image as input and gives the prediction of the image as the output. Here we deviate from conventional ways and try to decrease burden on processing images. We obtain the DCT and DWT of the same components. These components are fed to the system to process and create models for classification of images. The block diagram shown in **Figure 1**, shows the model generation using the inputs and testing classification using the generated model.**Modeling:** The obtained values are fed to the Convolutional neural network (CNN). The CNN used here has following components:

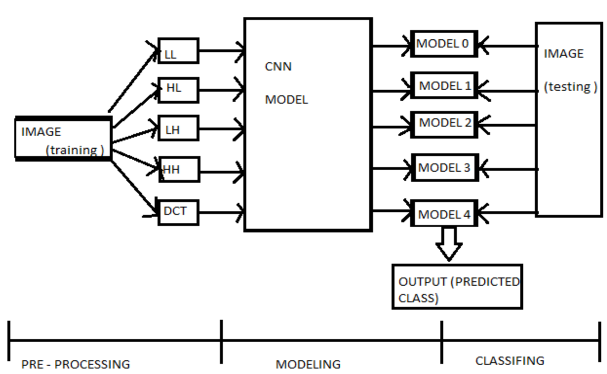


Figure 1: Block Diagram of the CNN model

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 8, August 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0908028 |

***Preprocessing*:** Here each image is taken and split into Augmented, Vertical, Horizontal and Diagonal components using Discreet Wavelet Transforms(DWT) and Discreet Cosine Transform (DCT) is applied to the same image. The Coefficients are obtained. The components of DWT and coefficients are stored in a list. ***Modeling*:**The obtained values are fed to the Convolutional neural network (CNN). The CNN used here has following components:

1.Convolutional Layer: This is the first as well as core layer, that extracts various features from the input layers. This layer gives convoluted results of input data to give the output result. The inputs are dimensions of image, batch size, color type (RGB or gray scale, activation functions).

a. **Filters:** Filters are features detectors applied on original image, where features are capture like edges to create feature maps of the original image.

b. **Activation functions:** ReLU is used as activation function, the API returns max (x, 0) i.e., element wise max of 0 or input tesnsor. Here the output will be of same shape and data type as that of input. There are many activation functions available like softmax, sigmoid, linear, softplus, tanh in this implementation relu is used as an activation function.

c. **Kernel Size:** Dimensions of the convolutional filters i.e., size of the matrix (n x n).

Input layer: Takes shape of the input image and number of channels (3 for RGB and 1 for grey scale.

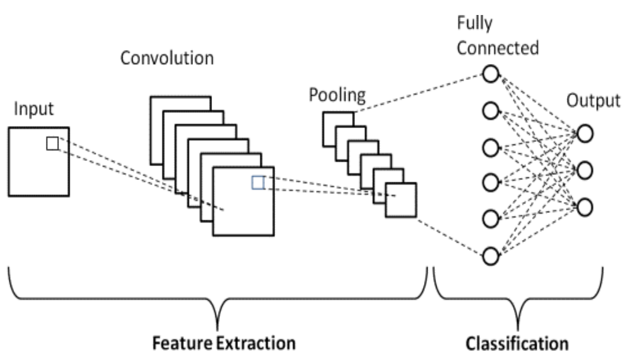


Figure 2: Fully connected model of a CNN

Image credit: Internet source

2. ***Pooling layer*:** This layer decreases the size of the convoluted output so that the computation cost of the system reduces. Thisis performed by reducing the connection between the layers and operates independently on each feature locally. There are many types of pooling used some of them are Max pooling, Average pooling, sum pooling.
3. **Flatten layer:** This layer converts amultidimensional array into single dimension arrayi.e., data is converted to single Coolum vectors which can be fed to fully connected layers.
4. **Fully Connected layer:** Consists of weights and bias which have connection between input and outputs. Here all neurons are connected to every other neuron, hence the name fully connected layer. This layer has many dense layers
5. **Output layer:** This layer has 2 basic functions: Compiling and Fitting of model. Compiling needs the following parameters: Optimizer: Algorithm used to calculate weights for the CNN model, Evaluation Matrix: Matrix used to evaluate the performance of the matrix, Loss: This is used to evaluate the loss i.e., difference between predicted value and the actual value.
6. Fitting of model has the following parameters:*Batch size*: It is the number of images to be taken at a time while training. If its not specified the system by default uses the corresponding values.

Epochs: It is the number of iterations of the training of images to be carried out. More the number of epochs better is the training. A catchy point arises here. As the number of epochs increase beyond certain iterations model gets over

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 8, August 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0908028 |

trained. This is taken care by Drop out layers, where few parts of image are randomly dropped so that model is still able to recognize and classify the test images while testing.The model is generated and can be saved.

This model can be used to test any related images to classify and testing. The model generated can be used any number of times on any machine like mobile, PC, Laptops. These models once created can be used n number of times, hence, this model generating is important. In the real time applications, which can be seen in Facebook and many other applications, Lakhs of images are used to train these models for the better accuracy and to work smoothly and in expected manner. There is trained model available so that the time and efforts needed to train again is reduced

The data set in this research is CIFAR-10. This dataset has 60,000 color pictures, each of which is 32x32 pixels, and includes 6,000 images in each class. Each dataset consists of 50,000 training pictures and 10,000 testing images. Every class is completely separate from every other class.

II. METHODOLOGY

Images from Cifar 10 data set are taken and are converted into gray scale and resized to 60*60. Each Image is then converted into DWT components (LL, HL, LH, HH). Also, gray scaled version of original images are converted to DCT coefficients, coefficients obtained are stored in list. These components are given to CNN where in training is started for images. The parameters are given like number of epochs, input filter size, batch size,

Here CNN model considered is a sequential model, the first layer is Convolutional layer where in feathers are extracted. As the image proceeds deeper into the layers, image is down sized with each feature handling independently. This model has 2 Conv2D layers. Activation function used is ReLU. A fully connected layer is dense layer where in each neuron is connected to every other neuron, 3 dense layers are used in this model. Since the output is a classification of images into 10 classes. We have the last output layer with 10 as the parameter to the output layer. The number of neurons in the intermediate layers must be such that, layers nearing the input are more than decorrelating performance of the output sides. Because the clustering has to happen such that image classify into number of classes specified.Next step is compiling the model, data set is split onto training and testing depending on the ratio provided. In this case we are using 70 % for training and 30% for testing. With the inputs as list, the model is generated as per parameters provided like epochs, batch size if provided else batch size is determined by the system. Model generated above is saved in the specified location. Once the model is generated the testing of the model happens and the accuracy of each model is given as the output to the user.

Testing: The model is ready and is up for testing and checking its accuracy and testing if the predicted class of the test image is correct or not. The image to be tested is taken and converted into DWT and DCT coefficients. These values are fed to the model, then prediction is made by model. The decision is made by voting by all the models. Majority wins. The decision is made based on voting by all models. In worst case when system fails to get majority number of votes, Decision is done in favor of that model whose prediction value is greater than 55%.

Summarizing the methodology as follows:

Step I: *Images are converted to special domain.*

Step II: *Image is compressed using DCT and DWT.*

Step III: *Image is convoluted using TestCNN.*

Step IV: *The models are generated and stored.*

Step V: *Testing of images for its classification accuracy and results for its fluency to classify an image is checked.*

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

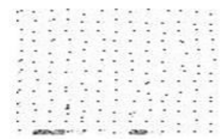
|| Volume 9, Issue 8, August 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0908028 |

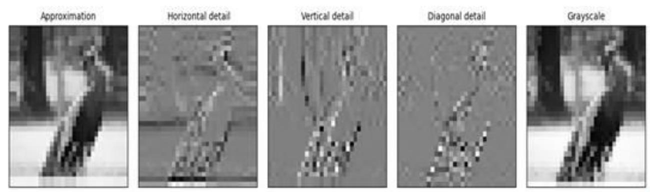
III. EXPERIMENTS AND RESULTS

1. Software used: python 3.7 with tensor flow.Data set used: Cifar10.Operating system: Windows 10.

Original Image



DCT of the original image



Various DWT components of Original Image shown above

Figure 3: Image in various forms of orignal image

The figure 3 shows the various versions of same image when they are converted into DCT and DWT Components.

The Training and testing accuracy obtained for the system for each module is as shown in the graph below:

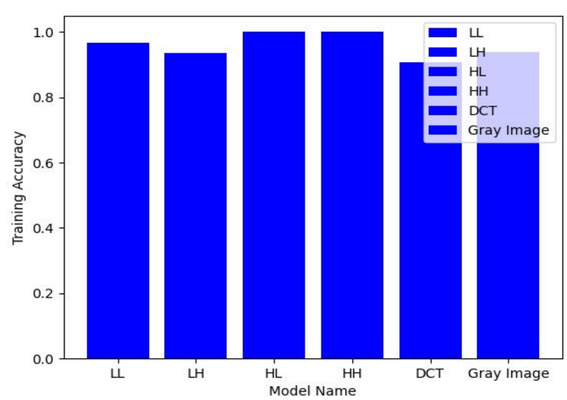


Figure 4: Training accuracy of the model

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542
|| Volume 9, Issue 8, August 2021 ||
| DOI: 10.15680/IJIRCCE.2021.0908028 |

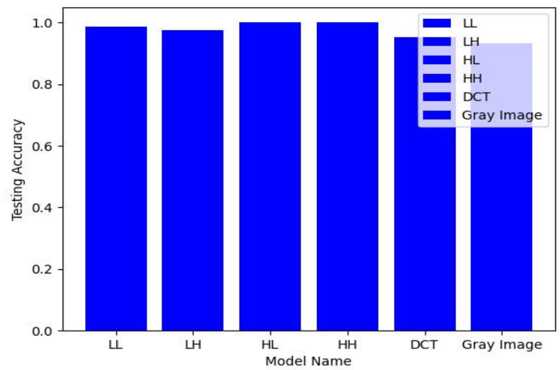


Figure 5: Testing accuracy of the model

The graph of accuracy (training and validations) with respect to number of epochs are as shown below:

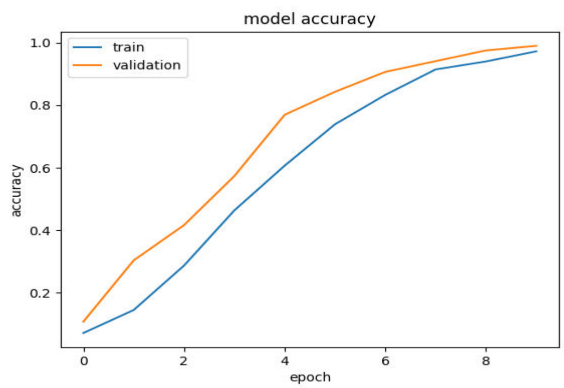


Figure 6: Model Accuracy with respect to number of epochs

The advantages of the model obtained are:

1. Minimum computation.
2. Light weight.
3. Usable in embedded Applications.

IV. FURTHUR SCOPES

The study can be further extended to other data sets available. This can be implemented into real time systems with lower memory consumption and has better results

V. CONCLUSION

This thesis has presented a methodology to detect the object using the Convolution neural Network (CNN) and have employed Tensorflow 2.3 with the python 3.7 version. For training the CNN the CIFAR-10 database has been employed. Instead of providing the images directly to the CNN, a DCT and DWT is obtained for all the images

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 8, August 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0908028 |

in database. These co-efficient are fed to the network for training, the ADAM is the optimizer employed before fitting the data for training. The model is successfully trained and provided an accuracy of more than 90% and this shows that the system can be employed for the real time and it works reasonably.

REFERENCES

[1] Watson, Andrew B. "Image compression using the discrete cosine transform." *Mathematica journal* 4.1 (1994): 81.

[2] Kant, Sonaal, et al. "Leukonet: Dct-based cnn architecture for the classification of normal versus leukemic blasts in b-all cancer." *arXiv preprint arXiv:1810.07961* (2018).

[3] Wang, Yunhe, et al. "CNNpack: Packing Convolutional Neural Networks in the Frequency Domain." *NIPS*. Vol. 1. 2016.

[4] Er, Meng Joo, Weilong Chen, and Shiqian Wu. "High-speed face recognition based on discrete cosine transform and RBF neural networks." IEEE Transactions on neural networks 16.3 (2005): 679-691.

[5] Farinella, Giovanni Maria, et al. "Representing scenes for real-time context classification on mobile devices." Pattern Recognition 48.4 (2015): 1086-1100.

[6] Ulicny, Matej, and Rozenn Dahyot. "On using cnn with dct based image data." Proceedings of the 19th Irish Machine Vision and Image Processing conference IMVIP. Vol. 2. 2017.

[7] Gueguen, Lionel, et al. "Faster neural networks straight from jpeg." Advances in Neural Information Processing Systems 31 (2018): 3933-3944.

[8] Oyallon, Edouard, et al. "Compressing the input for cnns with the first-order scattering transform." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[9] Williams, Travis, and Robert Li. "Advanced image classification using wavelets and convolutional neural networks." 2016 15th IEEE international conference on machine learning and applications (ICMLA). IEEE, 2016.

[10] Said, Salwa, et al. "Deep wavelet network for image classification." 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2016.

[11] Kutlu, Hüseyin, and Engin Avcı. "A novel method for classifying liver and brain tumors using convolutional neural networks, discrete wavelet transform and long short-term memory networks." Sensors 19.9 (2019): 1992.

[12] Prabukumar, Manoharan, et al. "Three-dimensional discrete cosine transform-based feature extraction for hyperspectral image classification." Journal of Applied Remote Sensing 12.4 (2018): 046010.

[13] Abbasi, Aamir Naveed, and Mingyi He. "CNN with ICA-PCA-DCT Joint Preprocessing for Hyperspectral Image Classification." 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2019.

[14] Elgallad, Elaraby A., Wael Ouarda, and Adel M. Alimi. "Dense Hand-CNN: A Novel CNN Architecture based on Later Fusion of Neural and Wavelet Features for Identity Recognition." (2019).

[15] Chamain, Lahiru D., and Zhi Ding. "Faster and accurate classification for jpeg2000 compressed images in networked applications." arXiv preprint arXiv:1909.05638 (2019).

[16] Rao, Yuan, and Jiangqun Ni. "A deep learning approach to detection of splicing and copy-move forgeries in images." 2016 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, 2016.

[17] Bendjillali, Ridha Ilyas, et al. "Improved facial expression recognition based on DWT feature for deep CNN." Electronics 8.3 (2019): 324.

[18] Jarrett, Kevin, et al. "What is the best multi-stage architecture for object recognition?." 2009 IEEE 12th international conference on computer vision. IEEE, 2009.

[19] Zeiler, Matthew D., and Rob Fergus. "Stochastic pooling for regularization of deep convolutional neural networks." arXiv preprint arXiv:1301.3557 (2013).

[20] Boureau, Y-Lan, et al. "Learning mid-level features for recognition." 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2010.

[21] Boureau, Y-Lan, Jean Ponce, and Yann LeCun. "A theoretical analysis of feature pooling in visual recognition." Proceedings of the 27th international conference on machine learning (ICML-10). 2010.

[22] Bell, Anthony J., and Terrence J. Sejnowski. "The "independent components" of natural scenes are edge filters." Vision research 37.23 (1997): 3327-3338.

[23] Le, Quoc, et al. "ICA with reconstruction cost for efficient overcomplete feature learning." Advances in neural information processing systems 24 (2011): 1017-1025.

[24] Korekado, Keisuke, et al. "A convolutional neural network VLSI for image recognition using merged/mixed analog-digital architecture." International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer, Berlin, Heidelberg, 2003.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 8, August 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0908028 |

[25] Korekado, Keisuke, et al. "A convolutional neural network VLSI for image recognition using merged/mixed analog-digital architecture." International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer, Berlin, Heidelberg, 2003.

[26] Hubel, David H., and Torsten N. Wiesel. "Receptive fields of single neurones in the cat's striate cortex." The Journal of physiology 148.3 (1959): 574-591.

[27] Leng, Lu, et al. "Dynamic weighted discrimination power analysis in DCT domain for face and palmprint recognition." 2010 international conference on information and communication technology convergence (ICTC). IEEE, 2010.

[28] Leng, L., et al. "Cancelable palmcode generated from randomized gabor filters for palmprint protection." 2010 25th International Conference of Image and Vision Computing New Zealand. IEEE, 2010.

[29] Lu, Jiwen, et al. "Palmprint recognition via locality preserving projections and extreme learning machine neural network." 2008 9th International Conference on Signal Processing. IEEE, 2008.

[30] Masood, Hassan, et al. "Wavelet based palmprint authentication system." 2008 International Symposium on Biometrics and Security Technologies. IEEE, 2008.

[31] Murukesh, C., and G. Arul Elango. "Multi-algorithmic palmprint authentication system based on score level fusion." International Journal on Smart Sensing and Intelligent Systems 1.18 (2018): 1-11.

[32] Sapijaszko, Genevieve M., and Wasfy B. Mikhael. "Facial Recognition System Using Mixed Transform and Multilayer Sigmoid Neural Network Classifier." Circuits, Systems, and Signal Processing 39 (2020): 6142-6161.

[33] Sapijaszko, Genevieve M., and Wasfy B. Mikhael. "Facial Recognition System Using Mixed Transform and Multilayer Sigmoid Neural Network Classifier." Circuits, Systems, and Signal Processing 39 (2020): 6142-6161.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 7.542

doi crossref

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com