iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45856

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

Review on Iris Recognition Research Directions- A Brief Study

Dr. Lokesh M R[1], B K Amruth Gowda[2], Prajwal B L[3], Vaibhav M P[4], Haseeb Ahmed Danish[5]
[1]*Proforesor,* [2, 3, 4, 5]*UG Student, Department of CSE, Maharaja Institute of Technology Mysore*

Abstract : Biometric methods, which identify people based on physical or behavioral characteristics, are of interest because people cannot forget or lose their physical characteristics in the way that they can lose passwords or identity cards. Among these biometric methods, iris is currently considered as one of the most reliable biometrics because of its unique texture's random variation. It is found that this method for Iris Recognition design offers good class discriminacy. The iris recognition technique consists of iris localization, normalization, encoding and comparison. The Neural Classifier will be a feed forward network with three hidden layers and be used after normalization and feature extraction phase. Simulation results will be very promising in person identification.

Keywords: Biometric methods, passwords, iris, normalization, neural classifier, encoding

I. INTRODUCTION

Identity verification and identification is becoming increasingly popular. Initially fingerprint, voice and face have been the main biometrics used to distinguish individuals. Advances in the field have expanded the options to include biometrics such as iris and retina. Among the large set of options, it has been shown that the iris is the most accurate biometric.

The iris is the elastic, pigmented, connective tissue that controls the pupil. Daugman proposed an iris recognition system representing an iris as a mathematical function.

Mayank Vatsa proposed a support-vector-machine-based learning algorithm selects locally enhanced regions from each globally enhanced image and combines these good-quality regions to create a single high-quality iris image. proposes algorithms for iris segmentation, quality enhancement, match score fusion, and indexing to improve both the accuracy and the speed of iris recognition.

M.Gopikrishnan used hamming distance coupled with Neural Network based iris recognition techniques are discussed. Perfect recognition on a set of 150 eye images has been achieved through this approach ; Further, Tests on another set of 801 images resulted in false accept and false reject rates of 0.0005% and 0.187% respective

II. RELATED WORK

Creating Iris segmentation based on a possible fuzzy method is used to identify the local vectors already available algorithms use two circular templates to identify the eye but are not in a standard circle show that leads to iris legacy and difficult to find proper identification edge detection method, k-algorithm is used to match the exact image for the given input data by making input pepsin as centric and the related pupil forming the usage position the eye images the method provides the effective adaptation technique to detect pupil and it can increase the accuracy for iris resonation.

A physiological characteristic is relatively stable physical properties such as fingerprint iris, facial, hand shadow image the type of measurement is fundamentally invariable and unchangeable without substantial compulsion. In this application a secure manner the person or the object itself is a password user verification system that use a single biometric display are disrupted by noisy data limited freedom degrees and error rates several biometrics attempt to overcome these drawbacks by providing multiple identity features of the same identity so that performance can be increased.

The human iris can be tested frequently throughout time its permanent and invariant only when the individual has not consumed alcohol comprises of unique iris pattern even the left and right iris of an individual varies iris recognition based personal authentication system is known to be reliable over biometric methods identifying two people with same pattern probability is more over zero.

Traditional iris segmentation methods provide good results. When iris image are recorded under ideal imaging conditions check the segmentation accuracy of an iris recognition system considerably influence the Active count the method is used for conidial iris image algorithm for count the model describe the bounders of shape in a images.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

III. MODEL OVERVIEW

In this analysis, we tend to purpose an economical and efficient Iris detection and recognition technique by rising existing Iris recognition algorithms. This IRIS recognition algorithmic program should be effective, accurate and efficient.

We tend to are providing a completely unique and sturdy feature extractor for Iris options together with Zernike moments and Neural Network primarily based IRIS detection for the aim of authentication wherever the system can detect and acknowledge the person's exploitation the IRIS sample pictures.

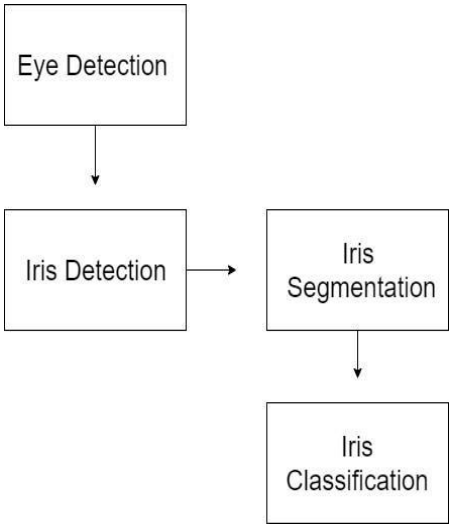


Fig 1 shows the architecture of the iris recognition system

The new algorithmic program can be flexible to numerous IRIS detection primarily based on the authentication systems respectively. This algorithmic program would be designed with the help of combinations of sturdy feature descriptor, Zernike moments and neural network model for versatile and strong meta-heuristic Iris recognition model.

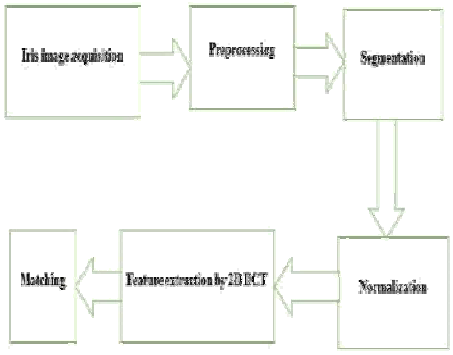


Fig -2: Purposed methodology steps

IV. IRIS RECOGNITION MODEL

Convolutional Neural Networks have a different architecture than regular Neural Networks. Regular Neural Networks transform an input by putting it through a series of hidden layers. Every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Finally, there is a last fully-connected layer — the output layer— that represent the predictions.

Convolutional Neural Networks are a bit different. First of all, the layers are organised in 3 dimensions: width, height and depth. Further, the neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it. Lastly, the final output will be reduced to a single vector of probability scores, organized along the depth dimension.

1) *Module 1:* Region Proposal. Generate and extract category independent region proposals, e.g. candidate bounding boxes.
2) *Module 2:* Feature Extractor. Extract feature from each candidate region, e.g. using a deep convolutional neural network.
3) *Module 3:* Classifier. Classify features as one of the known class, e.g. CNN classifier model. Architecture:

©IJRASET: All Rights are Reserved

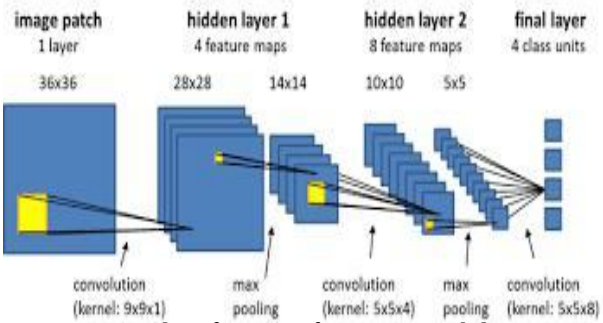
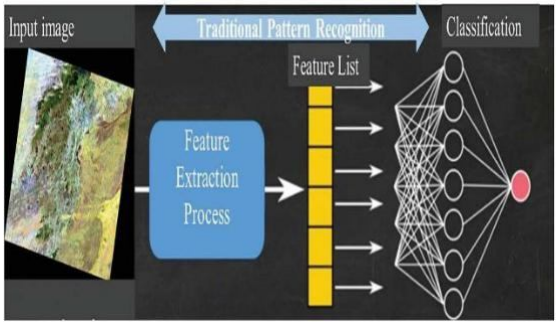


Fig-3: Classification of image in each layer

Convolutional Neural Networks have the following layers

1) *Convolution Layer:* Convolutional neural networks apply a filter to an input to create a feature map that summarizes the presence of detected features in the input.



2) *ReLU Layer:* In this layer, we remove every negative value from the filtered images and replaces them with zeros.It is happening to avoid the values from adding up to zero.
Rectified Linear unit(ReLU) transform functions only activates a node if the input is above a certain quantity. While the data is below zero, the output is zero, but when the information rises above a threshold. It has a linear relationship with the dependent variable.
3) *Pooling Layer:* In the layer, we shrink the image stack into a smaller size. Pooling is done after passing by the activation layer. We do by implementing the following 4 steps:

a) Pick a window size (often 2 or 3)
b) Pick a stride (usually 2)
c) Walk your

Window across your filtered images

4) *Fully Connected Layer:*

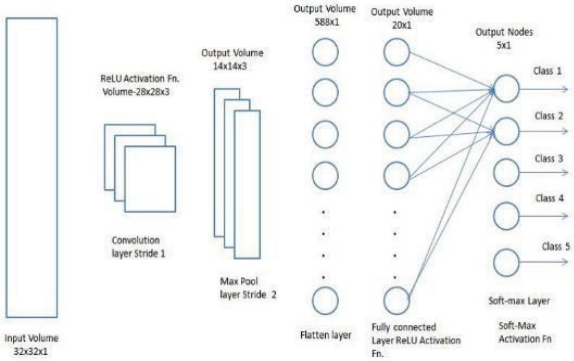


Fig-4 : This image shows how the internal parts are connected

The last layer in the network is fully connected, meaning that neurons of preceding layers are connected to every neuron in subsequent layers.

This mimics high-level reasoning where all possible pathways from the input to output are considered.

Then, take the shrunk image and put into the single list, so we have got after passing through two layers of convolution relu and pooling and then converting it into a single file or a vector.

Some researchers divide the deep learning algorithms into four categories: Convolutional Neural Networks, Restricted Boltzmann Machines, Auto encoders and Sparse Coding (Guo et al., 2016). In this work of images classification, we opted for the Convolutional Neural Networks (CNNs) (Fig. 4) because they are the widely adapted for image classification with good performances by making convolutional networks fast to train (Barbedo, 2013; Krizhevsky et al., 2012).

Convolutional Neural Networks (CNN) approaches are commonly used with architectures having multiple layers that are trained. There are two parameters (weights and bias) in each layer. The first layers find low-level features for instance edges, lines and corners and the other layers find mid-level and high-level features for instance structures, objects, and shapes (LeCun et al., 2015).Then the prediction output is used to compute the loss cost to the ground truth labels. Second, based on the loss cost, the backward stage computes the gradient of each parameter with chain rules. All the parameters are updated based on the gradients and are prepared for the next forward computation. After a number of iterations of the forward and backward stages, the network learning can be stopped.

Clarification of what was meant by the term “convolution” would be useful, since it is used to describe mathematical principles and ideas about the feature transformation approach and process. In mathematics, specifically in algebraic topology, convolution is a mathematical transformation on two functions (u and v); it generates a third function, which is normally viewed as a transformed version of one of the initial functions. In the case of two real or complex functions, u and v the convolution is another function, which is usually denoted u ∗ v and which is defined by (Hirschman and Widder, 2017):

$$(u \; v \; x)\left(\int_{-\infty}^{+\infty} u(x _ \;) = t \; v \; t \; dt\right)( ) \qquad (1)$$

The convolution satisfies some algebraic properties (e.g. commutativity, associativity, distributivity and multiplicative identity) in order to ensure that the characteristics of their corresponding geometric pictures are preserved. Commutativity

$$u \; v = v \; u \qquad (2)$$

Associativity

$$u \; (v \; h) = (u \; v) \; h \qquad (3)$$

Associativity with scalar multiplication

$$a \; u \; v() = (au) \; v \qquad (4)$$

Distributivity

$$u(v + =h) \; (u \; v) + (u \; h)$$

In artificial intelligence, convolutional neural networks apply multiple cascaded convolution kernels with deep learning applications (Fig. 5). Formally, the filtering operation performed by a feature map is a discrete convolution. Discrete convolution can be seen as multiplication by a matrix. Though, the matrix has several entries with constraints to be equal to other entries.

For functions defined on the set of integers, the convolution of two finite sequences is determined by extending the sequences to finitely applicable functions on the set of integers (e.g. a finite summation may be used for a finite support). The discrete convolution can be defined by following formula (Hirschman and Widder, 2012).

```
for each image row in input image:
  for each pixel in images row:

    set accumulator to zero

    for each kernel row in the kernel:
      for each element in the kernel row:

        if element position  corresponding to pixel position then
          multiply element value  corresponding to pixel value
          add result to accumulators
        endif

      Endfor
    Endfor
  Endfor
Endfor
set output image pixel to accumulator
```

The basic CNN architecture includes Convolution layer, Pooling layer, ReLU layer and Fully connection layer (as described in Table 1). Convolution layer

The real power of deep learning, especially for image recognition, comes from convolutional layers. It is the first and the most important layer. In this layer, a CNN uses different filters to convolve the whole image as well as the intermediate feature maps, generating various feature maps (Fig. 1). Feature map consists of a mapping from input layers to hidden layers. (5)

Depth of the output volume controls the number of neurons in the layer that connect to the same region of the input volume. All of these neurons will learn to activate for different features in the input. For instance, if the first Convolutional Layer takes the raw image as input, then different neurons along the depth dimension may activate in the presence of various oriented edges, or blobs of color. The depth is 5 in the Fig. 6.

There are three main advantages of the convolution operation

(Zeiler and Fergus, 2014)

a) the weight sharing mechanism in the same feature map reduces the number of parameters

b) local connectivity learns correlations among neighboring pixels

c) invariance to the location of the object.

One interesting approach to handling the convolutional layers is the Network In Network (NIN) method (Lin et al., 2013), where the main idea is to substitute the conventional layer with a small multilayer perceptron consisting of multiple fully connected layers with nonlinear activation functions, thereby replacing the linear filters with nonlinear neural networks. This method achieves good results in image classification.

5) ReLU Layer: it is the Rectified Linear Units Layer. This is a layer of neurons that applies the non-saturating non-linearity function or loss function:

$$f(x) = \max(0, x) \quad (7)$$

It yields the nonlinear properties of the decision function and the overall network without affecting the receptive fields of the convolution layer. We have saturated nonlinear functions that are much slower (Krizhevsky et al., 2012). Also, we have the tanh function:

$$f(x) = \tanh(x) \quad (8)$$ or the logistic sigmoid function:

$$1 \; Fx() = (1+e) \quad (9)$$

ReLU results in the neural network that is training several times rapidly, without making a significant difference to generalization accuracy.

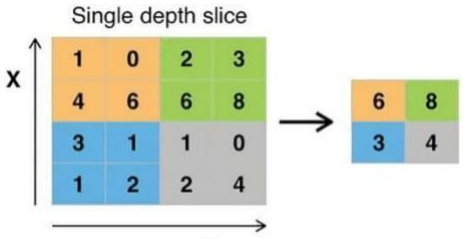


Fig-5: Max pooling process

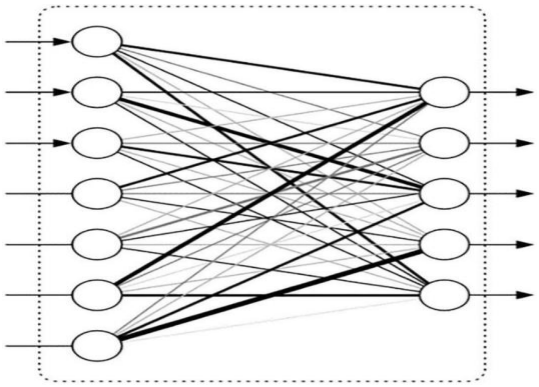


Fig-6 : This image shows how image will get segregated

6) *Pooling layer:* Its task consists to simplify or reduce the spatial dimensions of the information derived from the feature maps. We have three types of pooling: the first one is the average pooling, the second is the L2-norm pooling and finally the most popular use is the max pooling (Scherer et al., 2010) because of its speed and improved convergence. This basically takes a filter (normally of size 2 × 2) (fig. 10) and a stride of the same length. It then applies it to the input volume and outputs the maximum number in every sub region that has the filter which convolves around.
7) *Fully connection layer:* The input to this layer is a vector of numbers. Each of the inputs is connected to every one of the outputs hence the term "fully connected". It is the last layer, in general after the last pooling layer in CNN process (Fig. 11). Fully connected layers perform like a traditional neural network and contain about 90% of the parameters in CNN. This layer basically takes as input the output of the last pooling layer and outputs an N dimensional vector where N is the number of classes that the program has to choose from. It allows us to feed forward the neural network into a vector with a predefined length. For image classification, we could feed forward the vector into certain number categories (Krizhevsky et al., 2012). The output is also a vector of numbers.
8) *Loss layer:* Loss layer uses functions that take in the model's output and the target, and it computes a value that measures the model's performance. It can have two main functions:

a) Forward (input, target): calculates loss value based input and target value.
b) Backward (input, target): calculates the gradient of the loss function associated with the criterion and return the result.

V. RESULTS AND DISCUSSION

Testing is the way toward running a framework with the expectation of discovering blunders. Testing upgrades the uprightness of the framework by distinguishing the deviations in plans and blunders in the framework. Testing targets distinguishing blunders – prom zones. This aides in the avoidance of mistakes in the framework. Testing additionally adds esteems to the item by affirming the client's necessity.

The primary intention is to distinguish blunders and mistake get-prom zones in a framework. Testing must be intensive and all around arranged. A somewhat tried framework is as terrible as an untested framework. Furthermore, the cost of an untested and under-tried framework is high. The execution is the last and significant stage. It includes client preparation, framework testing so as to guarantee the effective running of the proposed framework. The client tests the framework and changes are made by their requirements. The testing includes the testing of the created framework utilizing different sorts of information. While testing, blunders are noted and rightness is the mode.

A. Top-Down Testing

In high levels of a system are tested before testing the detailed components. The application is represented as a single abstract component with sub-components represented by stubs. Stubs have the same interface as the component but very limited functionality. After the top-level component has been tested, its sub-components are implemented and tested in the same way. This process continues recursively until the bottom - level components are implemented. The whole system may then be completely tested. Top-down testing should be used with top-down program development so that a system component is tested as soon as it is coded. Coding and testing are a single activity with no separate component or module testing phase. If top-down testing is used, unnoticed design errors may be detected at an early stage in the testing process. As these errors are usually structural errors, early detection means that extensive re-design re-implementation may be avoided. Top-down testing has the further advantage that we could have a prototype system available at a very early stage, which itself is a psychological boost. Validation can begin early in the testing process as a demonstrable system can be made available to the users.

B. Bottom-Up Testing

Bottom-Up Testing is the opposite of Top-Down. It involves testing the modules at the lower levels in the hierarchy, and then working up the hierarchy of modules until the final module is tested. This type of testing is appropriate for object-oriented systems in that individual objects may be tested using their own test drivers. They are then integrated and the object collection is tested.

C. Validations

Toward the consummation of the reconciliation testing, the product is totally amassed as bundle interfacing blunders have been revealed and adjusted and a last arrangement of programming tests starts in approval testing.

Approval testing can be characterized from multiple points of view, however a straightforward definition is that the approval succeeds when the product work in a way that is normal by the client. After approval test has been directed as follows:

1) The capacity or execution qualities adjust to detail and are acknowledged.
2) A deviation from the particular is revealed and a lack list is made.
3) Proposed framework viable has been tried by utilizing an approval test and discovered to be working acceptably.

[10]:

	File	Classid	Class
0	person1/4.bmp	0	person1
1	person1/1.bmp	0	person1
2	person1/2.bmp	0	person1
3	person1/6.bmp	0	person1
4	person1/8.bmp	0	person1

Table-1 : this table shows the small dataset of how it looks

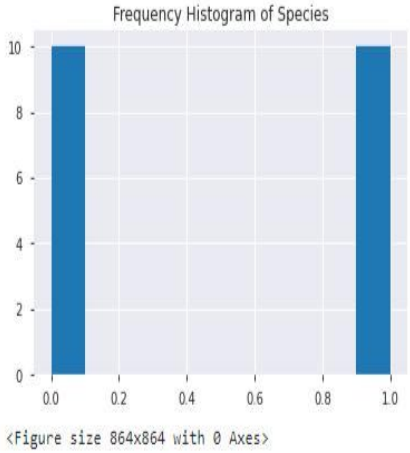


Fig-7 : This image shows frequency histogram of species

```
94773248/94765736 [==============================] - 1s 0us/step
94781440/94765736 [==============================] - 1s 0us/step
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 64, 64, 3)]       0
_________________________________________________________________
conv2d (Conv2D)              (None, 64, 64, 3)         84
_________________________________________________________________
resnet50 (Functional)        (None, None, None, 2048)  23587712
_________________________________________________________________
global_average_pooling2d (Gl (None, 2048)              0
_________________________________________________________________
batch_normalization (BatchNo (None, 2048)              8192
_________________________________________________________________
dropout (Dropout)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 256)               524544
_________________________________________________________________
batch_normalization_1 (Batch (None, 256)               1024
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
root (Dense)                 (None, 2)                 514
=================================================================
Total params: 24,122,070
Trainable params: 24,064,342
Non-trainable params: 57,728
```

[31]: Text(0.5, 29.5, 'Predicted')

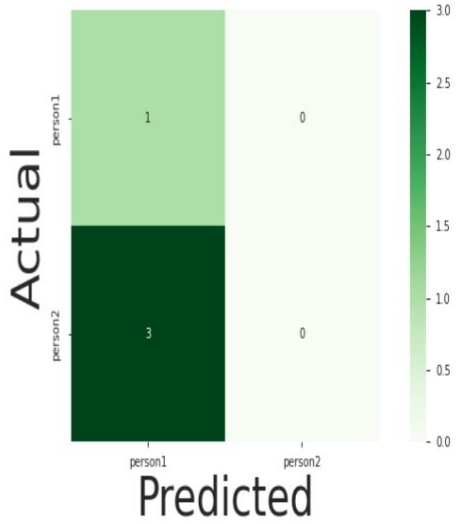


Table-2 : This table shows how the image is been Fig-8 : This image shows actual prediction of iris processed

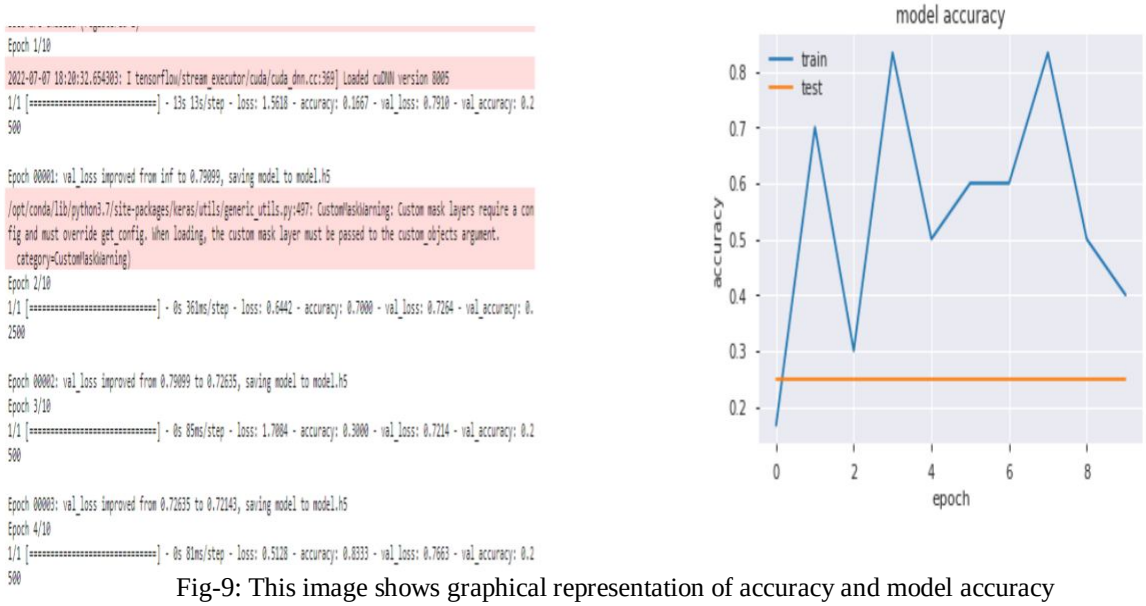


Fig-9: This image shows graphical representation of accuracy and model accuracy

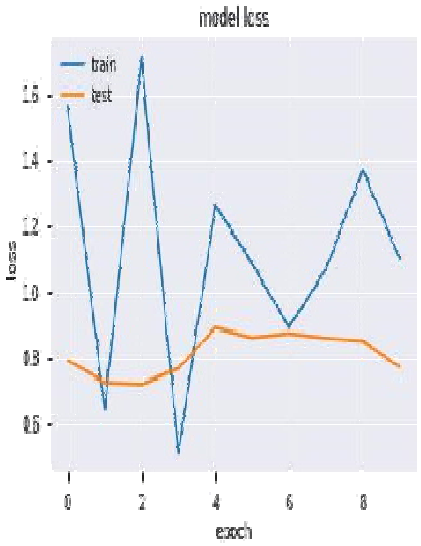


Fig-10: This image shows graphical representation of loss and model loss

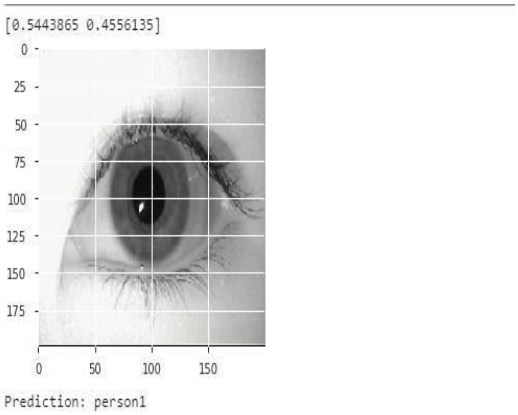


Fig-11: This image shows the actual prediction of the iris

VI. CONCLUTION

In this work we propose a dilated deep learning framework for iris recognition, by fine-tuning a pretrained convolutional model on ImageNet. We focused on iris recognition task, and chose a dataset with a large number of subjects, but limited number of images per subject, and proposed a transfer learning approach to perform identity recognition using a deep residual convolutional network. This framework is applicable for other biometrics recognition problems, and is specially useful for the cases where there are only a few labeled images available for each class.

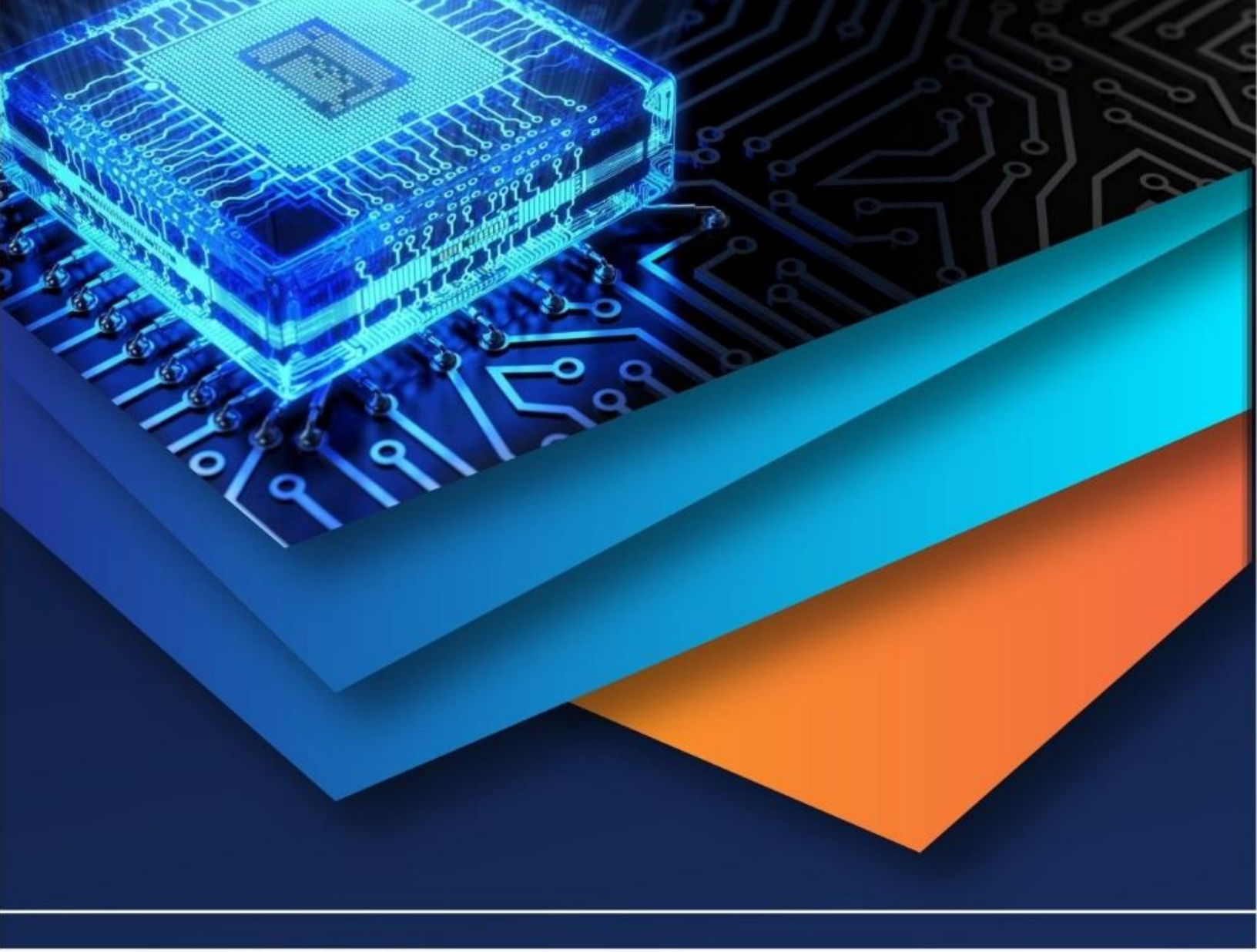
We apply the proposed framework on a well-known iris dataset, CASIA, IIT-Delhi, and achieved promising results, which outperforms previous approaches on this datasets. We train these models with very few original images per class. We also present a visualization technique for detecting the most important regions while doing iris recognition.

Conflict of Interest: The author declares that he has no conflict of interest.

REFERENCES

[1] J. Daugman, “Iris recognition border-crossing system in the UAE,” International Airport Review, vol. 8, no. 2,2004.
[2] K. W. Bowyer, K. Hollingsworth, and P. J. Flynn, “Image understanding for iris biometrics: A survey,” Computer Vision and Image Understanding, vol. 110, no. 2, pp. 281– 307,2008.
[3] L. Masek, “Recognition of human iris patterns for biometric identification,” The University of Western Australia, 2003.
[4] J. Daugman, “How iris recognition works,” in The essential guide to image processing. Elsevier, 2009, pp. 715–739.
[5] D. M. Monro, S. Rakshit, and D. Zhang, “Dct- based iris recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 4, pp. 586–595, 2007.
[6] K. Miyazawa, K. Ito, T. Aoki, K. Kobayashi, and H. Nakajima, “An effective approach for irisrecognition using phase-based image matching.” IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 10, pp. 1741–1756, 2008.
[7] Y. Zhou and A. Kumar, “Personal identification from iris images using localized Radon transform,” in
[8] O. C. Reyes, R. Vera-Rodriguez, P. Scully, and K. B. Ozanyan, “Analysis of spatio-temporal representations for robust footstep recognition with deep residual neural networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.
[9] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J.OrtegaGarcia, “Exploring recurrent neural networks for on-line handwritten signature biometrics,” IEEE Access, vol. 6, no. 5128-5138, pp. 1–7,2018. Graphics, Patterns and Images (SIBGRAPI, Brazil, pp.289-296, 2018.
[10] Proença, H. and Alexandre, L., Iris recognition: Analysis of the error rates regarding the accuracy of the segmentation stage. Image and Vision Computing, Vol.28, Issue.1, pp.202- 206, 2010.
[11] Sanchez-Gonzalez Y, Chacon-Cabrera Y, Garea-Llano E. A Comparison of Fused Segmentation Algorithms for Iris Verification. In: Salinesi C, Norrie MC, Pastor Ó, eds. Advanced Information Systems Engineering, Berlin, Heidelberg: Springer Berlin Heidelberg, Vol 7908, pp.112- 119, 2014
[12] Nigam A, Gupta P. Iris Recognition Using Consistent Corner Optical Flow. In: Lee KM, Matsushita Y, Rehg JM, Hu Z, eds. Computer Vision– ACCV 2012. Berlin, Heidelberg: Springer Berlin Heidelberg, Vol 7724, pp.358-369, 2013.
[13] Bellaaj M, Elleuch JF, Sellami D, Kallel IK. An Improved Iris Recognition System Based on Possibilistic Modeling. In: Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia - MoMM, ACM Press, Brussels, Belgium, pp.26-32, 2015
[14] Minaee S, Abdolrashidi A, Wang Y. An Experimental Study of Deep Convolutional Features For Iris Recognition.in Conferene of IEEE Signal Processing in Medicine and Biology Symposium, USA, 2017
[15] Zanlorensi LA, Luz E, Laroca R, Britto Jr. AS, Oliveira LS, Menotti D. The Impact of Preprocessing on Deep Representations for Iris Recognition on Unconstrained Environments. , Conference on Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, 2010, pp. 2840–2843.
[16] Bhateja, A., Sharma, S., Chaudhury, S. and Agrawal, N., Iris recognition based on sparse representation and k-nearest subspace with genetic algorithm. Pattern Recognition Letters, Vol. 73, pp.13-18, 2016.
[17] Proenca, H. and Alexandre, L., Toward Noncooperative Iris Recognition: A Classification Approach Using Multiple Signatures. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, Issue. 4, pp.607-612, 2007.
[18] Sarhan AM. Iris Recognition Using Discrete Cosine Transform and Artificial Neural Networks. J of Computer Science, Vol.5, Issue.5, pp.369-373, 2009.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)