IJDI-ERET

INTERNATIONAL JOURNAL OF DARSHAN INSTITUTE ON
ENGINEERING RESEARCH AND EMERGING TECHNOLOGIES
Vol. 11, No. 2, 2022
www.ijdieret.in

(Research Article)

Classification Model for Multi-Classes Iris Image Using Deep Learning Neural Networks

F.O Aranuwa[1*], Fawehinmi, O.B[2]

[1,2]Department of Computer Science, Adekunle Ajasin Uiversity, Akungba – Akoko, Ondo State, NIGERIA*

Abstract

Iris image has been adjudicated one of the most reliable biometric traits for authentication as a result of its unique patterns, stringency to spoof attack and reliability. However, studies have revealed that inefficient classification sorts and techniques leading to classification errors and inaccurate matching has characterized its processes in many authentication and identification applications. Additionally, most of the existing works in the domain are based on single image classification, and major drawbacks of these methodologies remains insufficient learning input and classification errors. The current work uses Deep learning neural networks (DLNN), a subset of artificial intelligence that are effective in learning complex features from data, such as images. Majorly, the work is focused at classifying both left and right human iris images. Data for the work was acquired from the CASIA-Iris-lamp dataset (http://biometrics.idealtest.org). The dataset contains 16,163 iris datasets, and 20 iterations were passed on the data during modeling to determine the accuracy of the model. Performance metrics such as sensitivity, specificity and accuracy were used to evaluate the performance of the model. Experimental results show that the model performed well with classification accuracy of 99.57% which is relatively an improvement over the existing model that was used as a benchmark with 93.35%. The model correctly classified and predicted all images belonging to the right category as right irises, while it wrongly predicted 14 images belonging to the left category. Connotionally, the right iris recorded higher accuracy compared to that of the left iris.

Keywords: Iris, Classification, Deep Learning, CNN, Accuracy

1. Introduction

The iris is the colored ring component of the eye located between the pupil and white sclera within the human eye [1]. The trait is well known for high precision, stringency to spoof attack and reliability [2]; [3]. It possesses a good resistance to replication and its unique patterns do not change over time [4]; [5]. Every iris is unique to an individual, thus making it an ideal and reliable form of biometric verification [6]; [7]. Figure 1 shows the anatomy of the human eye.

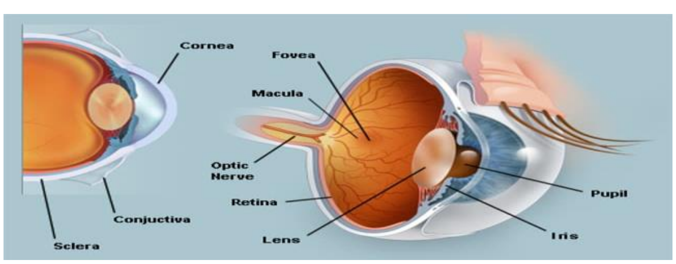


Figure 1. Anatomy of the eye [8]

In biometrics, the iris belongs to the physiological traits of human being, and its function is to control the amount of light entering through the pupil. Major applications of iris recognition systems include criminal investigation, eye tracking, immigration and border controls, access and security applications, and so on [9]. It is an extremely reliable and accurate identification method. However, it has some challenges. Studies have revealed that inefficient classification methodologies and procedures leading to classification errors and inaccurate matching has characterized its process in many authentication and identification systems. Studies also have revealed that various methods proposed to address these challenges are majorly based on single iris image classification, and the drawbacks of these approaches remains insufficient learning input and inadequate classifications sorts [5]; [10]; [11]; [12]; [13]. To the best of our knowledge, few works that considered multiple classes (left and right) iris image classification based on traditional neural networks suffers also some trade-off between accuracy and computational complexities [14];[15];[16].

The works in [5] was carried out on genetically identical irises. The research was motivated by the need to examine the texture similarity that is not detected by iris biometrics. The research showed that, by using Daugman's algorithm, the left and right

*Corresponding Author: e-mail: felix.aranuwa@aaua.edu.ng
Tel-+2347031341911,*
ISSN 2320-7590 (Print) 2583-3863 (Online)

irises of the same person are as different as irises of unrelated individuals, and the eyes of twins are as different as irises of unrelated individuals. The work reported an accuracy of 75%. In [11], a research on the diagnosis of iris nevus using a neural networks and deep belief network (DBN) was presented. The work was motivated by the need to study the Iris nevus eye defect caused by pigmented growth (tumor) found in the front of the eye or around the pupil. The objective was to describe the automated diagnosis of iris nevus using neural network-based systems for the classification of eye images as “nevus affected" and “unaffected". The research recorded accuracy of 93.35%, but suffers some computational complexity as a result of the combination of two complex algorithms.

A research on periocular recognition under less constrained environment using semantics-assisted convolutional neural network (SCNN) was presented in [13]. The research was motivated by the need to solve the challenge of achieving accurate biometric identification under real environments to meet growing demands for higher security. The objective was to develop a framework to efficiently and accurately match periocular images that are automatically acquired under less-constrained environments. An overall accuracy of 77.06% was reported using the cuda-covnet-SCNN approach.

The work in [14] presented a study on gender classification of iris images using deep learning. The aim of the research was to predict the gender of an individual from iris image by adopting an unsupervised training stage using a Restricted Boltzmann Machine (RBM) and a supervised training stage using neural networks. An overall accuracy of 84.66% was achieved combining the two classifiers considered. An accuracy of 90% on average was achieved. The limitation of the work, according to the authors, was the limited amount of learning data used. Furthermore, the authors used single iris classification without considering the correlation of the left and right.

In [15], the covariate influence on gender and race prediction from near-infrared (NIR) ocular images was analyzed. The research was motivated by the need to examine the problem of predicting race and gender from NIR ocular images. The objective was to use simple texture descriptors, such as binarized statistical image features and local binary patterns, to extract gender and race attributes from a near-infrared ocular image that is used in a typical iris recognition system. The research contributed to knowledge by providing a method for the prediction of race and gender of an individual by examining near-infrared (NIR) ocular images with an accuracy of 90%. However, single iris classification was used.

In summary, the limitations of the above includes inadequate classification sorts leading to inaccurate matching, trade-off between accuracy and computational complexities. Therefore, the current work uses deep learning neural networks (DLNN), a subset of artificial intelligence that are effective in learning multiple and complex features from data. It also possessed the capability to handle the complexities in the existing works. The work is majorly focused at classifying both left and right human iris images for accurate prediction.

2. Deep Learning Neural Networks

Deep learning is a subfield of machine learning, a new approach on learning representations from data that puts emphasis on learning successive layers of increasingly meaningful representations. It is a modern update to Artificial Neural Networks that are concerned with building of large and complex neural networks with remarkable results in the domain [17];[18]. Deep learning techniques are effective in learning complex features from data, such as images, texts, audio, and videos. Its learning process can be supervised, semi-supervised or unsupervised depending on the data structure. The technique allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These layered representations are learned via models called “neural networks”, structured in literal layers stacked one after the other. Figure 2 depicts the simple neural and deep learning neural networks respectively.

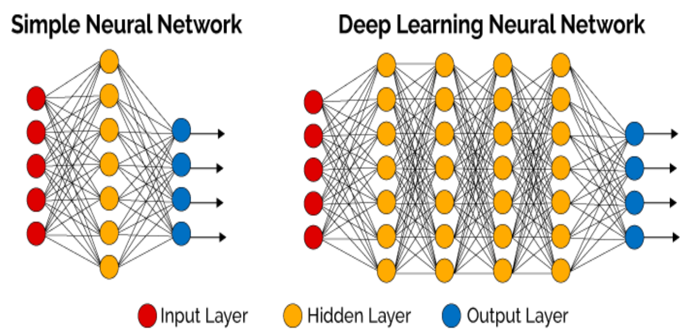


Figure 2. Showing the simple neural and deep learning neural networks.Source: [18]

According to [19], deep learning algorithms have been used to solve classical artificial intelligence problems with the main goal of learning high level abstractions from data [16]. Popular deep learning techniques include: Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs), Stacked Auto-Encoders, Deep Boltzmann Machine (DBM), and Deep Belief Networks (DBN). Meanwhile, much attention is been given to Convolutional Neural Networks (CNN) because of its effectiveness in computer vision task such as image analysis and classifications. It has become one of the most appealing approaches in the recent time, hence it is considered very suitable for this work. CNN, also known as ConvNet, is one of the best neural networks for classification, segmentation, natural language processing (NLP), and video processing [20]. Its application has become most demanding due to its ability to learn features from images automatically, involving massive amount of training data and high computational resources like graphic processing units (GPUs). According to [21], the reason behind better performance of CNN is due to its ability to work on raw data without having prior knowledge of the data. The architecture of CNN is very similar to the ordinary Neural Networks, made up of neurons that have learnable weights and biases. According to [22], a convolutional neural network is a feed-forward neural network (FNN) that is generally used to analyze visual images by processing data with grid-like topology. When using CNN algorithm for image classification. The image is usually fed directly to the convolutional neural networks, then the algorithm thereafter extracts the best features of the image. CNNs are regularized versions of multilayer perceptron where each neuron in one layer is connected to all neurons in the next layer. Meanwhile, this full connection of these networks makes them prone to over-fitting. However, pooling process is a down-sampling operation that reduces the dimensionality of the feature map [23]. The architecture is divided into four important layers, namely: the convolution layers, ReLU layer, pooling layers, and fully connected layers [20;24]. Figure 3 shows CNN architecture.

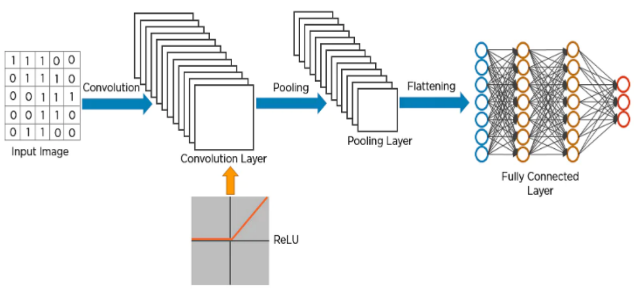


Figure 3. Architecture of CNN (fully connected layers) source: [25]

3. Methodology

To achieve the objectives of the work, the architecture of the classification model was first designed as depicted in figure 4. The model is composed of five major components: the image processing, the convolution layer feature maps, the rectifier linear unit, pooling layer features maps and the fully connected layers (classification layers). The image preprocessing in the model architecture involves feature extraction from the image pixels. In the convolution layer, the convolution operations are performed by combining the image with a feature detector (learnable filter) in order to extract data from neighboring pixels. The rectified linear unit (ReLU) is used to increase the non-linearity of the output image in the convolution layer after the required features have been filtered. Pooling is applied to the output from the ReLU to extract the most meaningful information. In order to avoid over fitting of the learned features, a down-sampling method (max pooling) was used to reduce the size of the pooled image, and the number of values as input are forwarded to the classification layer containing fully connected layer of neurons for image classification.

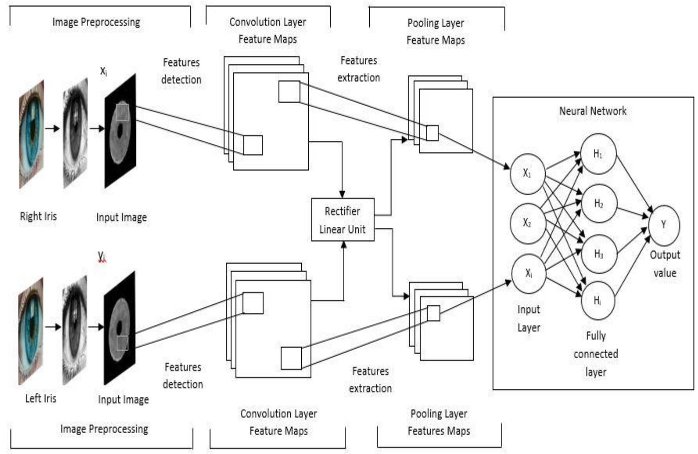


Figure 4. The architecture of the CNN classification model

Mathematically, the processes is presented in equation 1:

$$\begin{aligned} &Preprocessed\ image \rightarrow (Conv \rightarrow ReLU \rightarrow Pool) * M \\ &\qquad \rightarrow (FC \rightarrow ReLU) * K \rightarrow FC \\ &\qquad \rightarrow Classification \ldots\ldots\ldots\ldots\ldots\ldots\ldots. \qquad (1) \end{aligned}$$

Where:

a. Conv: is the Convolutional Layer
b. ReLU: is the Rectified Linear Unit
c. FC: is the Fully Connected Layer
d. M and K: are numbers representing the number of times each operation is performed.

3.1 Image Acquisition and Processing: The data for the work was acquired from the CASIA-Iris-Lamp dataset (http://biometrics.idealtest.org). The CASIA-Iris-Lamp dataset is a robust iris feature representation. The iris images are 8-bit gray-level JPEG files, collected under near infrared illumination. A total of 16,163 iris data was considered in the research work. The training set contained 12,931 iris images, while the test set contained 3,232 data. 20 iterations were passed on the model to determine the accuracy of the model.

The image processing involves segmentation, normalization and salient feature extraction, of the acquired images. As shown in Figure 5.

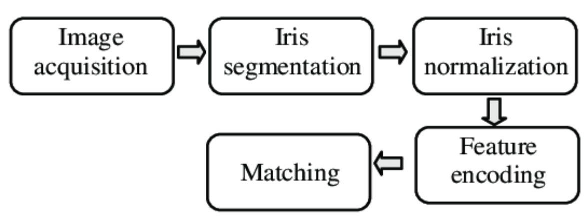


Figure 5. Main image processing

The image acquired was enhanced through segmentation of the required region of the trait, followed by normalization process to form a fixed pattern in polar coordinates. At the segmentation phase, the separation of the iris from the whole eye area took place. At this stage, the position of the upper and lower eyelids was determined. Iris segmentation acts directly on the image, seeking the maximum normalized standard area along the path. The centre of the coordinates and the radius of the circle were sought for, and edges of the iris determined. The purpose of the segmentation stage is to enhance the image in order to produce high clarity iris image that serves as input to the classification model. The enhancement reduces computational complexity of the model with a goal to avoid decline of matching performance. For the segmentation process, circular Hough Transform was applied to detect the boundaries of the iris and the pupil. This also involves performing Canny Edge detection to generate an edge map [26]. The parametric equation is:

$$(x-a)^2 + (y-b)^2 = r^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where x and y are the detection points, *a* and *b* are the coordinates of the center of the circle and *r* is the radius.

The Canny Edge detection contains five stages: Smoothing, Finding gradients, Non-maximum suppression, double thresholding, and edge tracking through hysteresis [27]. The normalization process transformed the localized iris to a defined format in order to allow comparisons with other iris codes (in binary code). The number of values in binary codes as input are forwarded to the classification layers.

3.2 Matching Process: This process establishes correspondences between the set of the images acquired and the template created in the database. The approach involves detecting a set of interest points each associated with image descriptors from the images. The performance of the matching methods is based on the properties of the underlying interest points and the choice of associated iris image descriptors [28]. In this work, the radius of the iris is considered. Table 1: shows the sampled interest point scores.

Table 1. Sampled interest point scores

Identity	Right iris		Left iris		Tx	Ty
	x_i (mm)	xi (mm)	yi (mm)	y (mm)	(mm)	(mm)
1	10.1	10.3	10. 3	10.4	10.2	10.35
2	10.2	10.5	11.5	11.6	10.35	11.55
3	11.0	11.4	11.5	12.0	11.2	11.75
4	12.2	12.3	12.5	12.8	12.25	12.65
5	11.5	12.0	12.1	12.6	11.75	12.35

xi and yi are the detection points for both right and left iris, *Tx and Ty are threshold values for the right and left iris measurements.*

3.3 Performance Evaluation: Sensitivity, specificity and accuracy were used to evaluate the performance of the classification model coupled with the confusion matrix report.

Values of TN, TP, FN and FP were used to calculate the performance metrics considered.

$$\textbf{\textit{Sensitivity}} = \frac{TP}{TP+FN} = \frac{1602}{1602+14} = 0.9913 \qquad (99.13\%)\ldots\ldots (3)$$

$$\textbf{\textit{Specificity}} = \frac{TN}{TN+FP} = \frac{1616}{1616+0} = 1.0000 \qquad (100\%)\ldots\ldots (4)$$

$$\textbf{\textit{Accuracy}} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{1602+1616}{1602+1616+0+14} = 0.9957(99.57\%)\ldots.(5)$$

Where,
TP represents the True Positives: These are cases in which the model predicted YES (the images are left irises), and they are actually left irises.

TN represents the True Negatives: The model predicted NO for left irises, and they are actually NOT left irises.
FP represents False Positives: The model predicted YES for left irises, but they are actually NOT left irises.
FN represents False Negatives: The model predicted NO for left irises, but they are actually left irises. Table 2 showed the confusion matrix report for the classification of the iris images considered.

Table 2. Confusion matrix report for classification of iris images

Actual class	Predicted class	
	Left iris	Right iris
Left iris	TN = 1,616	FP = 0
Right iris	FN = 14	TP = 1,602

4. Experimental Setup and Data Presentation

The derived pooled feature map in equation (1) was flattened into a long vector to serve as input layer for the neural network with fully connected neuron layers for the image classification. During the experiment, the dataset was fed into the Anaconda platform installed on Processor Intel(R) Core(TM) i7-8550U CPU at 1.80 GHz, 1992 MHz, 4 Core(s), 8 Logical Processor(s) system with 1TB HDD and 8GB RAM for the purpose of the work. The software is an open-source application for management of multiple Python versions on one computer. Anaconda provides a large collection of highly optimized, commonly used data science libraries to carry out experiments. It has a powerful IDE with advanced editing, interactive testing, debugging and introspection features called Spyder. Figure 6 presents the Anaconda Navigator (Individual Edition (Distribution) Interface).

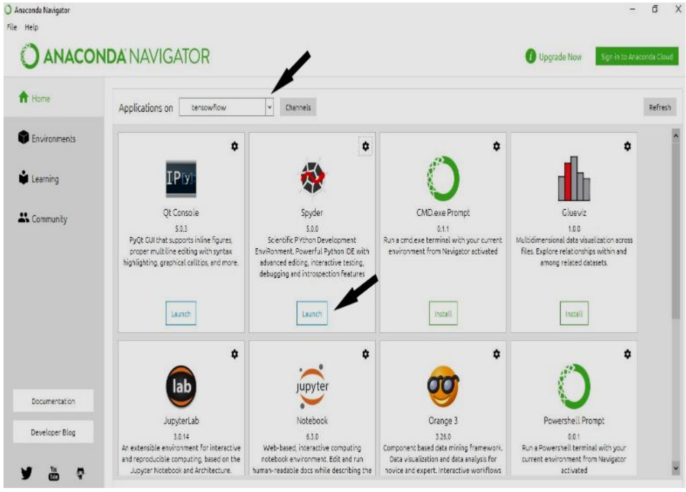


Figure 6. Anaconda navigator (individual edition (distribution) interface)

A specific internal environment (tensorflow) was created to manage the IDE and python libraries (Theano, TensorFlow, Keras). Figure 7 presents the IDE and Python Libraries Environment

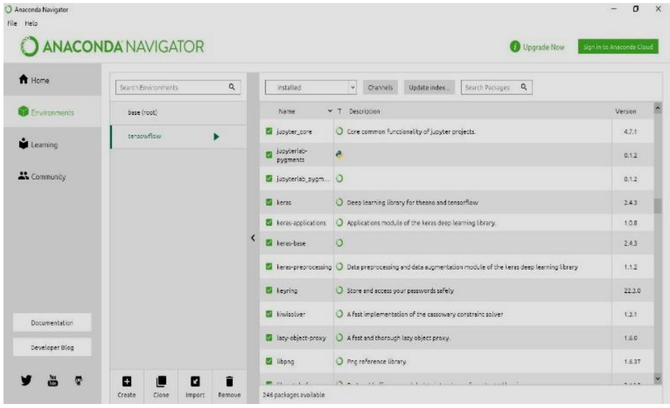


Figure 7. IDE and python libraries environment

Figure 8 presents the implementation in Spyder IDE and describes the following three major aspects: the dataset was fed into the platform by setting it as working directory. The dataset was splitted into two folders (*training set*, *test set*) within the working directory using 80/20 ratio (that is 80% of the total dataset for training the model and the remaining 20%) for validating the model. Since this work focuses on binary classification of the left and right iris image, two sub-folders (*left*, *right*) were created within the *training set* and *test set* folders to hold the two classes of labelled iris images (left irises, right irises). Secondly, the CNN model was built using Python programming language which supports scientific/numeric computing and effective integration of systems, and Thirdly, a console in figure 8 shows the classification report as the model is trained and validated with the dataset.

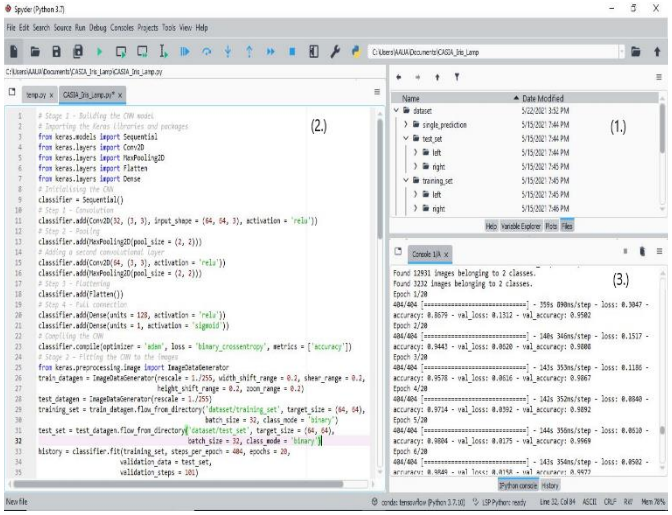
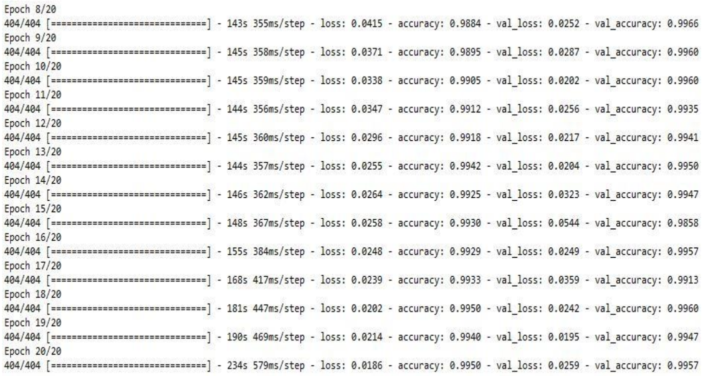


Figure 8. Implementation in spyder IDE

5. Results and Findings

This work adopts the scaling law ratio for training set composed of 80% of the total dataset, while the remaining 20% is allocated for the test set. 20 iterations were passed on the model to determine the accuracy of the developed model. The snapshot of the iteration process is shown in Figure 9.



```
Epoch 8/20
404/404 [==============================] - 143s 355ms/step - loss: 0.0415 - accuracy: 0.9884 - val_loss: 0.0252 - val_accuracy: 0.9966
Epoch 9/20
404/404 [==============================] - 145s 358ms/step - loss: 0.0371 - accuracy: 0.9895 - val_loss: 0.0287 - val_accuracy: 0.9960
Epoch 10/20
404/404 [==============================] - 145s 359ms/step - loss: 0.0338 - accuracy: 0.9905 - val_loss: 0.0202 - val_accuracy: 0.9960
Epoch 11/20
404/404 [==============================] - 144s 356ms/step - loss: 0.0347 - accuracy: 0.9912 - val_loss: 0.0256 - val_accuracy: 0.9935
Epoch 12/20
404/404 [==============================] - 145s 360ms/step - loss: 0.0296 - accuracy: 0.9918 - val_loss: 0.0217 - val_accuracy: 0.9941
Epoch 13/20
404/404 [==============================] - 144s 357ms/step - loss: 0.0255 - accuracy: 0.9942 - val_loss: 0.0204 - val_accuracy: 0.9950
Epoch 14/20
404/404 [==============================] - 146s 362ms/step - loss: 0.0264 - accuracy: 0.9925 - val_loss: 0.0323 - val_accuracy: 0.9947
Epoch 15/20
404/404 [==============================] - 148s 367ms/step - loss: 0.0258 - accuracy: 0.9930 - val_loss: 0.0544 - val_accuracy: 0.9858
Epoch 16/20
404/404 [==============================] - 155s 384ms/step - loss: 0.0248 - accuracy: 0.9929 - val_loss: 0.0249 - val_accuracy: 0.9957
Epoch 17/20
404/404 [==============================] - 168s 417ms/step - loss: 0.0239 - accuracy: 0.9933 - val_loss: 0.0359 - val_accuracy: 0.9913
Epoch 18/20
404/404 [==============================] - 181s 447ms/step - loss: 0.0202 - accuracy: 0.9950 - val_loss: 0.0242 - val_accuracy: 0.9960
Epoch 19/20
404/404 [==============================] - 190s 469ms/step - loss: 0.0214 - accuracy: 0.9940 - val_loss: 0.0195 - val_accuracy: 0.9947
Epoch 20/20
404/404 [==============================] - 234s 579ms/step - loss: 0.0186 - accuracy: 0.9950 - val_loss: 0.0259 - val_accuracy: 0.9957
```

Figure 9. Iteration process

At the 20th iteration level, the classification report gave a training accuracy of 99.50% with validation accuracy of 99.57% while a training loss of 1.86% was observed with validation loss of 2.59%. A difference of 0.07% was observed between the training and validation accuracy which shows that the model was able to overcome over-fitting by effectively closing the gap between the training and validation accuracy. The model is relatively an improvement over the existing model that was used as a benchmark with 93.35%. Table 3 showed the classification report for the training and validation process. Figure 10 presents a graph of the Training and Validation Loss while figure 11 presents the Training and Validation Accuracy using matplotlib in python.

Table 3. Classification report for training and validation

Epoch	Training loss (%)	Validation loss (%)	Training accuracy (%)	Validation accuracy (%)
1	30.47	13.12	86.79	95.02
2	15.17	6.20	94.43	98.08
3	11.86	6.16	95.78	98.67
4	8.40	3.92	97.14	98.92
5	6.10	1.75	98.04	99.69
6	5.02	1.58	98.49	99.72
7	4.53	1.89	98.56	99.78
8	4.15	2.52	98.84	99.66
9	3.71	2.87	98.95	99.60
10	3.38	2.02	99.05	99.60
11	3.47	2.56	99.12	99.35
12	2.96	2.17	99.18	99.41
13	2.55	2.04	99.42	99.50
14	2.64	3.23	99.25	99.47
15	2.58	5.44	99.30	98.58
16	2.48	2.49	99.29	99.57
17	2.39	3.59	99.33	99.13
18	2.02	2.42	99.50	99.60
19	2.14	1.95	99.40	99.47
20	1.86	2.59	99.50	99.57

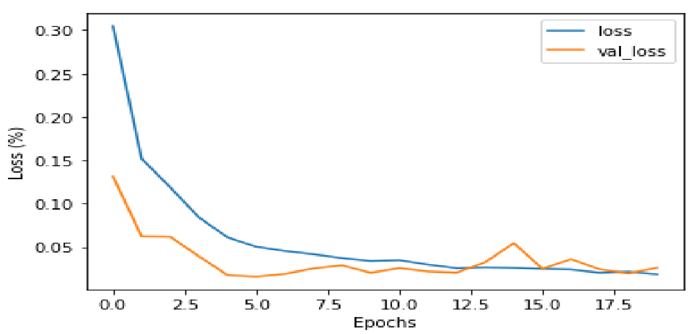


Figure 10. Training and validation loss

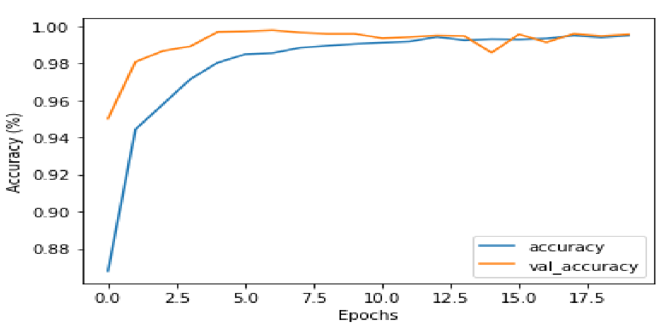


Figure 11. Training and validation accuracy

Figure 12 showed the snapshot of confusion matrix report.

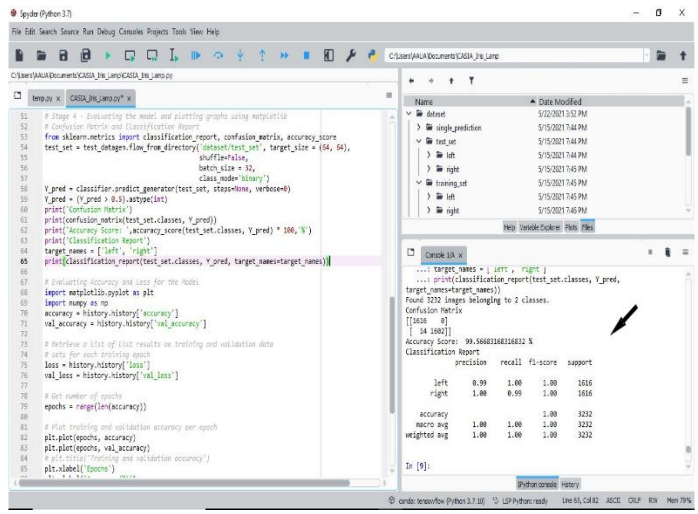


Figure 12. Snapshot of confusion matrix report

Interpreting the confusion matrix report, the model reported 0 false positives (FP) and 14 false negatives (FN) on the test set. This suffice to say the model correctly predicted all images belonging to right category as right irises while it wrongly predicted 14 images belonging to the left category. By implication, there was higher accuracy on the right iris compared to the left iris.

Table 4. Comparison of current work with existing works

Related work	Description	Dataset	Accuracy
Hollingsworth *et al.*, (2011)	using Daugman's algorithm (standard biometric algorithm)	ND-IRIS-0405	75%
Oyedotun and Khashma (2016)	using a CNN and DBN	The Eye Cancer Foundation, Eye Cancer Network	93.35%
Tapia and Aravena (2017)	using RBM and CNN	Private	84.66%
Zhao and Kumar (2017)	using semantics-assisted convolutional neural network (SCNN)	CASIA-Iris-Distance	77.06%

Bobeldyk and Ross (2019)	using simple texture descriptors, such as binarized statistical image features (BSIF) and local binary patterns (LBP)	GFI, BioCOP2009	90%
Zhao *et al.*, (2019)	using capsule network architecture	CASIA-Iris-Lamp	93.87%
Current research	using CNN	CASIA-Iris-Lamp	99.57%

6. Conclusions

This work presented an enhanced classification model based on CNN, a branch of DLNN for the classification of both left and right iris images to proffer solution to the inadequacies of the classification sorts and the challenges of the single mode classification techniques. Data for the work acquired was segmented and normalized and were later fed into the Anaconda model building platform installed for the work. The classification model developed was tested and had an accuracy of 99.57% which is relatively a great improvement over existing iris feature detectors and classification techniques. The confusion matrix reported 0 false positives (FP) and 14 false negatives (FN) on the test set. The shows that the model correctly predicted all images belonging to right category as right irises while it wrongly predicted 14 images belonging to the left category. By implication, there was higher accuracy on the right iris compared to the left iris. Based on the performance of the model presented, the research work has established that CNN based model as an improvement compared to other models in the domain. The current model in no small measure will increase the efficiency of biometric security applications and assist eye medical practitioners. Researcher in the future work will consider the case of multiple classes and larger database.

References

1. George, A., Image Based Eye Gaze Tracking and its Applications. Retrieved 21/06/2021 at 10.13140/RG.2.2.33992.88329, 2017.
2. Kong, W. K., Zhang D. and Kamel, M. S. An Analysis of Iris Code. IEEE Trans. on Image Processing, Vol. 19, No 1, pp. 522-532. 2010.
3. Shah, N. and Shrinath, P., Iris Recognition System – A Review. International Journal of Computer and Information Technology, Vol. 3, No.1. pp. 13-20 2014.
4. Elgamal, M. and Al-Biqami N., An Efficient Feature Extraction Method for Iris Recognition Based on Wavelet Transformation. International Journal of Computer and Information Technology, Vol. 2, No. 3, pp. 1-7, 2013.
5. Hollingsworth, K., Bowyer, K. W., Lagree, S., Fenker, S. P. and Flynn, P. J. Genetically Identical Irises have Texture Similarity that is not Detected by Iris Biometrics. Computer Vision and Image Understanding, Vol. 115 No. 1, pp 493–1502, 2011.
6. Dehkordi, A. B., and Abu-Bakar, S. A., A Review of Iris Recognition System. Journal Teknologi, Vol. 77 No. 1, pp. 234-245, 2015.
7. NEC, What is Iris Recognition and how does it work? Retrieved on 24/05/2022 from https://www.nec.co.nz/market-leadership /publications-media/what-is-iris-recognition-and-how-does-it-work/ 2019.
8. WebMD, A Picture of the Eye. WebMD, LLC. Retrieved on 23/03/2022 at https://www.webmd.com/eye-health/picture-of-the-eyes. 2015.
9 Olatinwo, O., Shoewu, O. and Omitola, O.O, Iris Recognition Technology: Implementation, Application, and Security Consideration. The Pacific Journal of Science and Technology. Vol.14, No. 2, pp.228-233. http://www.akamaiuniversity.us/PJST.htm. 2013.
10. Thiyaneswaran, B. and Padma, S. Iris Recognition using Left and Right Iris Feature of the Human Eye for Bio-Metric Security System. International Journal of Computer Applications (0975 - 8887), Vol. 50. No 12, pp. 211-218. 2012.
11. Oyedotun, O., and Khashman, A. , Iris Nevus Diagnosis: Convolutional Neural Network and Deep Belief Network. Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 25, pp. 1106-1115, 2017.
12 Arsalan, M., Hong, H. G., Naqvi, R. A., Lee, M. B., Kim, M. C., Kim, D. S., Kim, C. S., and Park, K. R. (2017). Deep Learning-Based Iris Segmentation for Iris Recognition in Visible Light Environment. Symmetry Vol. 9, No. 263, pp. 1-25, doi: 10.3390/sym9110263, 2017
13. Zhao, Z., and Kumar, A. Accurate Periocular Recognition Under Less Constrained Environment Using Semantics-Assisted Convolutional Neural Network. In IEEE Transactions on Information Forensics and Security, Vol 12, No 5, pp. 1017-1030, 2017.
14. Tapia, J., & Aravena, C., Gender Classification from NIR Iris Images Using Deep Learning. In Advances in Computer Vision and Pattern Recognition PartF1, pp. 219-239, Springer London, https://doi.org/10.1007/978-3-319-61657-5_9, 2017
15. Bobeldyk, D., and Ross, A., Analyzing Covariate Influence on Gender and Race Prediction from Near-infrared Ocular Images. IEEE Access 2019, Vol 7, No 1, 7905–7919, 2019.
16. Khalifa, N., Taha, M., Hassanien, A. E., & Mohamed, H. (2019). Deep Iris: Deep Learning for Gender Classification Through Iris Patterns. Journal of the

Society for Medical Informatics of Bosnia and Herzegovina, Vol. 27, No. 2, pp 96–102, 2019.

17. Nguyen, K., Fookes, C., Ross, A., and Sridharan, S., Iris Recognition with off-the-shelf CNN Features: A Deep Learning Perspective. IEEE Access 2018, Vol. 6, pp. 18848–18855, 2018.
18. Favio, V., Deep Learning Made Easy with Deep Cognition. BBVA Data & Analytics. Retrieved on 30/3/2022. at: https://www.kdnuggets.com/2017/12/deep-learning-made-easy-deep-cognition.html , 2022.
19. Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and McCool, C. DeepFruits: A Fruit Detection System Using Deep Neural Networks. Sensors 2016, Vol. 16 No. 8, pp. 12-22; https://doi.org/10.3390/s16081222 https://doi.org/10.5455/aim.2019.27.96-102 2016.
20. Muhammad, A. S., Norhalina, S., Fazli, W., Muhammad, A. Ali, S. and Mukhtaj, K., Comparative analysis of recent architecture of convolutional neural network Vol. 22, No. 1, pp. 1-9, 2022.
21. Senan, N. Aamir, M. Ibrahim, R. Taujuddin, N. and Muda, W. W (2020). An efficient convolutional neural network for paddy leaf disease and pest classification," International Journal of Advanced Computer Science and Applications. Vol. 11, No. 1, pp. 27-36, 2020.
22. Bilswal, A., Convolutional Neural Network Tutorial. Simplilearn. Retrieved online on 25/05/2022. at: https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network. 2022 Vol. 11, No. 1, pp. 27-36, 2020.
23. Mandal, M Introduction to Convolutional Neural Networks (CNN). An article published as a part of the Data Science Blogathon. Retrived online at https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/ on 25/05/2022.
24. O'Shea, K., and R. Nash, An Introduction to Convolutional Neural Networks. CoRR abs/1511.08458: http://arxiv.org/abs/1511.08458. 2015.
25. Sultana, F. Sufian, A.and Dutta, P. (2018). Advancements in image classification using convolutional neural network in Proceedings of the 2018 4th International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pp. 122–129, IEEE, Kolkata, India, November 2018.
26. Sangwan, S., and Rani, R., A Review on: Iris Recognition. International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 6, No. 4, pp.3871-3873, 2015.
27. Jayachandra, C., and Reddy, H. V., Iris Recognition based on Pupil using Canny edge detection and KMeans Algorithm. International Journal of Engineering and Computer Science ISSN: 2319-7242. Vol 2, No 1, 221-225, 2013.
28. Tyagi, D., Introduction to Feature Detection and Matching. In Data Breach. Available online - https://medium.com/data-breach/introduction-to-feature-detection-and-matching-65e27179885d, Accessed 22nd June, 2021, 2019.

Biographical notes

Dr. Felix Aranuwa is a senior faculty member in the Department of Computer Science, Adekunle Ajasin University, AkungbaAkoko, Nigeria· He has held administrative positions as Head of Department of Computer Science. He possessed B.Sc (Hons), M.Sc and Ph.D in Computer. Science in year 1997, 2010 and 2015 respectively. His research interest covers; Biometrics, System Modelling, Data Science, Machine Learning and Human Computer Interaction.

Fawehinmi Olatunji Bolu possessed a PGDCSC and M.Sc in Computer Science. His research interest covers: Soft Computing, Software Engineering, Data Science, and Deep Learning. He currently works as a Backend Developer at Seamless HR.