International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Jan-2022)

https://dx.doi.org/10.12785/ijcds/110143

A Real-time Machine Learning-Based Person Recognition System With Ear Biometrics

Shahadat Hossain[1], Sanjida Sultana Mitu[1], Sadia Afrin[1] and Shamim Akhter[1]

[1]*Department Of Computer Science and Engineering, International University of Business Agriculture and Technology (IUBAT) Dhaka, Bangladesh*

Received 16 Apr. 2021, Revised 3 Jul. 2021, Accepted 4 Jan. 2022, Published 20 Jan. 2022

Abstract: Biometric authentication is a common way of granting access to a system or device. The ear, like fingerprints, retina, iris, face, voice, and so on, is a biometric modality. Compared to other biometric organs, the anatomy of a human's ear remains stable from birth to old life. As a visible organ with an easily acquired image, it may also be a source of a biometric signature that may be used to identify individuals. This research demonstrates two approaches to recognizing a person from 2D ear images: non-deep ML models and deep learning-based ML models. The first, or classic, model investigates computer vision preprocessing techniques such as converting an RGB image to monochrome, then rescaling and locating the entropy. The key weighted characteristics from the ear images were extracted using Independent Component Analysis (ICA) and Principal Component Analysis (PCA). A Gaussian Process Classifier (GPC) is then utilized for classification and several kernels such as RBF, Rational Quadratic, and Matern. In the second technique, a deep learning-based ML model called You Only Look Once (YOLO) is utilized to categorize the ear images and identify the source individual without preprocessing. We gathered a standard ear dataset (EarVN1.0 Dataset) from 164 people, totaling 27,592 training images. For testing reasons, 820 images were chosen randomly, five images from each of 164 people. The models were built on the Google Colaboratory server using the Python language framework and GPU-based implementation on the Jupyter Notebook.

Keywords: Ear Biometrics, Person Identification, YOLO Machine Learning, Independent Component Analysis (ICA), Principal Component Analysis (PCA), Gaussian Process Classifier; Radial Basis Function (RBF), Rational Quadratic, and Matern

1. INTRODUCTION

Many applications need secure authentication, such as identifying a person to enter complex systems, monitoring and recognizing undesired persons, and safeguarding assets against various risks. A Biometric system is a reasonable means of identifying a person since it examines biological characteristics such as retinal vein patterns, voices patterns, DNA patterns, and fingerprint patterns, and it is exceedingly challenging to fabricate. As a result, many studies have verified the use of biometric features to access several essential resources. Automatic recognition uses encrypted passwords, pins, or IC embedded cards for authentication. However, these types of information can be lost, stolen, or forgotten. The biometric characteristics were contrary to this. Biometric modalities are unique, permanent, measurable, and acceptable for universal users, and challenging to masquerade. Fingerprint recognition systems are standard in cellphones, laptops, and other authenticated devices.

Similarly, ear biometric modalities can also play a significant role in user identification and authentication. Because individuals are more comfortable shooting ear images than facial shots for verification, ear recognition is sometimes more accepted or preferred than face recognition. Recent research [1], [2], [3], [4], [5], [6] has demonstrated that an individual's biological and geometrical ear features differ from others' and can be utilized to distinguish them. Furthermore, the ear's structure has remained constant over time. Therefore, this study aims to detect people from static images.

To distinguish ear characteristics, template matching, morphological methods, and other approaches can be employed. Image processing collects required information such as geometry, contour, boundary, curvatures, and graphs, requiring many manual adjustments [7]. The emergence of various Convolutional Neural Networks (CNNs) [8] has recently transformed image segmentation, categorization, and scene understanding techniques into a simple automated process, reducing the need for human intervention. As a result, we describe two distinct ways for identifying a person: non-deep traditional learning and deep learning. Developing a person recognition system with a non-deep version uses computer vision, advanced feature (ICA and PCA) extraction, and classification methods (Gaussian Process Classifier-GPC). The target of the entire model is to find

E-mail address: shahadathossain8721@gmail.com, sanjidamitu456@gmail.com, sadiaafrinetu753@gmail.com, shamimakhter@gmail.com

out the best combination among feature extraction models with different types of kernels for GPC implementations to gain maximum accuracy for the application.

The deep learning version uses machine-learning approaches. Namely, CNN and You Only Live Once (YOLO). In object identification and classification applications, CNN and its progeny NNs (VGGNet, R-CNN, and so on) play an essential role. However, they have a bad reputation due to their complicated construction and sluggish training pace. In 2016, YOLO [9] was launched to recognize real-time objects from images or videos. YOLO estimates the object's inner bounding boxes and calculates the class probabilities. In terms of speedy object detection, YOLO outperforms CNN and R-CNN. Our goal is to employ the YOLO paradigm to authenticate authorized users in a short amount of time. As a result, we used YOLO as a deep ML model to identify the individual from an ear image in real-time. We will start by evaluating the application's performance using the YOLO model and then compare the performance of the CNN and YOLO models. We trained the ML models to identify the individual in real-time using ear biometric data from the EarVN1.0 dataset [10]. There are 28,412 photos in the EarVN1.0 collection from 164 distinct sources. The images varied in scale, illumination, occlusion, resolution, and lighting circumstances. This section presents a high-level summary of the research on person identification using ear biometrics. Section II briefly presents related work on person identification. Section III describes the data sets and feature setup. Section IV describes the proposed work's technique as well as the entire YOLO architecture. Section V explains and analyzes the findings. Finally, section VI wraps up the article with closing remarks and future work.

2. Related Work

In 1890, ear biometric traits were employed to identify persons [11]. A manual ear identification system was developed in 1949 [12]. [13] presented a graph model built from the edges and contours of ears and utilized a graph-based matching approach to authenticate. Authors in [14] showed a completely automated ear identification system with several extracted properties such as ear points and ear shapes Morphological operators, and Fourier characteristics were used to segment curved areas of interest [15]. The outer and inner ear shapes as feature vectors in [16]. Both used a Neural Network (NN) to categorize and identify individuals. In [17], a Genetic Technique (GA) was used as a search algorithm to find the identical input and output ear images. [18] presented a hybrid system combining Independent Component Analysis (ICA) with an RBF network. Furthermore, the authors compared ICA and Principal Component Analysis (PCA).

Machine learning has grown in popularity over the last several decades since it has played an essential role in increasing the performance of object recognition and classification applications. In [19], deep convolutional neural network (CNN) models were employed for ear identification. [20] employed two separate convolutional neural networks (CNNs) to distinguish the front and side perspectives of ears. According to the test findings, the front view ear images system produced 84% accuracy. Meanwhile, the side-view image-based system achieved an accuracy of 80%. [21] combined K-nearest neighbor (KNN) and CNN classifiers with SURF features and tested their ear identification accuracy on AWS datasets. In [22], deep learning with geometric morphometrics was applied for automated ear identification. The CNN's performance was tracked and compared to those of pre-manual landmarks. [23] employed a novel variation of CNN multiple-scale of the faster region with CNN (R-CNN) to detect ears from input photos. The detecting process used two distinct phases to validate the ear position. First, three separate scale regions were discovered, and the ear placement inside the picture was semi-confirmed; second, filtering was utilized to avoid false-positive ear locations. [24] employed a convolution encoder and decoder network to achieve binary classification between the ear and non-ear classes. [25] demonstrated an ensemble of three convolutional neural networks (CNNs). The end outcome was the weighted average of their outputs and used to identify the ear regions. The ensemble of networks outperformed a single CNN. Another research [26] employed an ensemble of three (3) CNN networks. They did not differ in the member networks based on their design but instead on the cropping sizes utilized for various picture sections.

As a result, CNN and its progeny NNs (VGGNet, R-CNN, and so on) play an important role in object detection and classification applications. However, their structures are highly complicated, and it is challenging to determine the optimal structure based on image dimension. It necessitates extensive calibration of the network parameters. As a result, the training pace is quite sluggish. The Mean Average Precision (mAP) of CNN and CNN family NNs is 76.4, whereas the Frame Per Second (FPS) of Faster R-CNN stays 5 to 18, significantly slower than the real-time effect.

Furthermore, the CNN does not exhibit rotational invariance. Making it rotational invariant adds complexity to the NN. As a result, increasing speed is critical for real-time object detection algorithms like human recognition from ear images. YOLO [27], [28], [29], [30] has the quickest FPS of 155 and the highest accuracy mAP of 78.6. Furthermore, it may offer the optimal inner border of the object and the confidence level of classification in the probability value. As a result, we integrate YOLO as a deep learning-based ML model in this work to identify the individual in real-time from an ear image.

3. DATASETS AND FEATURE SETUP

The EarVN1.0 dataset, which was generated in 2018, is the most extensive open-source ear image collection. It contains 28,412 ear images from 98 male and 66 female sources. The ear images were then cropped from the facial shots based on differences in location, size, and brightness.

To train the network, we choose 1000 random images from ten random people. All training images are renamed with numerical numbers such as 1.jpg, 2.jpg, 3.jpg,..., 1000.jpg, beginning with 10-100 number images from person 1, 101-200 number images from person 2, 201-300 number images from person 3, 301-400 number images from person 4, and so on, until 901-1000 number images from person 10.To use the YOLO algorithm, we must do the following particular setting. A class.txt file is created for every 100 images of a person, and each image name is defined in a train.txt file for training purposes. Following that, all images were trained in Darknet, an open-source neural network framework. Fig.1 shows a few examples of training images.

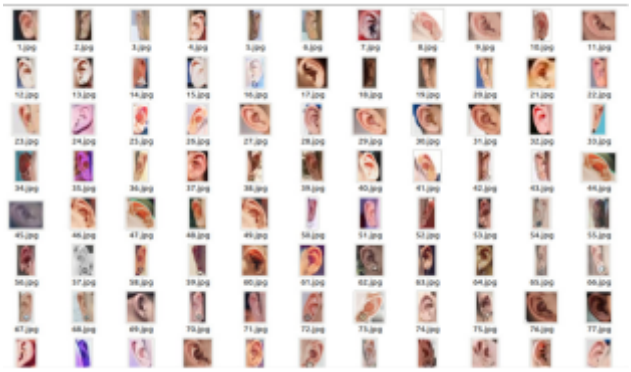

Figure 1. Training images dataset

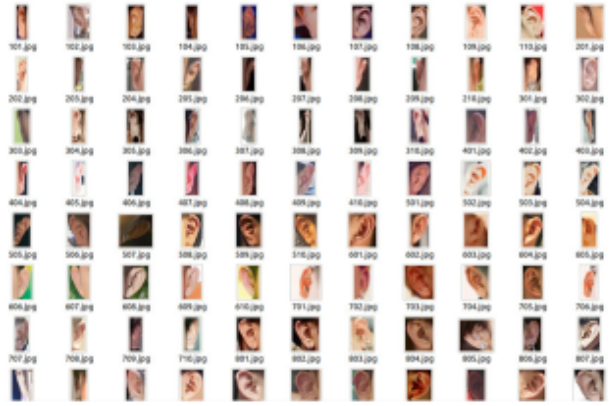

Figure 2. Testing images dataset

Darknet was created using the C and CUDA programming languages. It is simple to set up and supports both CPU and GPU calculations. After that, the Makefile is modified to include GPU=1, CUDA=1, and OpenCV=1 for GPU-based Darknet training. The yolov3.cfg file within the cfg folder is renamed yolov3 custom.cfg, and the image width, height, and color channel numbers (RGB=3) are updated to customize the YOLO according to the inputs. Any value divisible by 32 can be used for the width and height. The filter size is given by the equation (B x (5+C), where B is the yolov3 bounding box (BB) number of 3. Each BB possesses (5+C) properties. The letter C represents the number of output classes. In our dataset, the value of C is 10. Adding BB's height, weight, Center x, Center y, and confidence values together yields the value 5. In our situation, each value is one, and their aggregate is (1+1+1+1+1) =5. Consequently, the kernel size for our dataset is 1 x 1 x 45 (15x3). This kernel creates feature maps with distinct height, width, and detection attributes. The maximum batch size for each class is 2000, and the overall maximum batch size is total class number x 200. After setting the Google Colaboratory, the training begins using pre-trained weights. Following that, we run a test on our PC using the trained network weights from the Google Colaboratory. One hundred additional random images are acquired for testing purposes, with ten images obtained from every ten persons. This time, image numbers 101-110 represent person number one, 201-210 represent person number two, 301-310 represent person number three, and so on until 1001-1010 indicate person number ten. Figure 2 shows a few test images.

4. METHODOLOGY

Image classification and object detections are two different concepts or two different problem scenarios. Image classification uses ML approaches to assign an image to a predefined label or class. Simple CNN or ANN can be applied to image metrics and selects an output category for the input image. Object detection allows the distinguishing of objects in the image and assigns them into a specific label or class. It uses classification and localization to show the bounding boxes around the objects of an image. Based on the description above, our proposed work falls into the object detection category, and to appropriately identify the objects, and two alternative techniques are considered: non-deep ML models and deep learning-based ML models. A single-class object detection algorithm can be applied in a dataset with only one output label or class. Although in our ear images, we have only one object, however, our task is to identify different people from the ear objects, and thus we use multi-class object detection ML algorithms.

A. Non-Deep ML Models

All the non-deep ML models are written in the Python programming language. The required libraries are glob, PIL, OS, sklearn, and CV2. Glob is used to match specified input patterns with files/pathnames. The OS module provides all the functions to communicate with the operating system. The scikit-learn or sklearn library consists of the required machine learning and statistical modeling modules. CV2 is mainly used for image pre-processing tasks, including face identification (Cascade Classifier), grayscale conversion, and image normalization. PIL is the Python Image Library which contains all the image editing modules. NumPy-Numerical Python is a Python library that includes functions for linear algebra, the Fourier transform, matrices, and arrays.

1) Preprocessing Images:

Converting the RGB image to monochrome, rescaling, and locating the entropy are all examples of image preprocessing. Grayscale can convert a continuous-tone image into a manipulatable computer image. We choose the grayscale image due to the simplicity of their presentation and the much more comfortable applying simple mathematics to image processing operations, namely denoise, brightness, contrast, edges, shape, contour, texture, and so on. Besides, we use the built-in ML modules of python, and they apply grayscale images rather than addressing color. Furthermore, an object detection application barely requires information in the RGB image but not in the grayscale image. However, avoiding RGB images reduces additional complexity. The rescale changes the size of the input image into a specific dimension. We crop the image into 40 x 60 sizes. A histogram of an image is a graphical representation of the number of times each intensity value in the image occurs.

2) Feature Extraction:

The technique of creating the number of attributes or directions to characterize a dataset is known as feature extraction. We employ Independent Component Analysis (ICA) and Principal Component Analysis (PCA) to extract the important, meaningful features from ear images. ICA extracts the individual subcomponents of the multivariant components such as images and stores them into a multidimensional random vector. PCA exacts the principal components of an image and formulates the feature vector according to their weighted contribution. We extract ICA and PCA feature vectors from images and train ML models.

3) Identification:

We choose Gaussian Process Classifier (GPC) as a non-deep learning classifier. It uses Gaussian probability distribution and measures the degree of belongings to an individual class. The highest degree directs the classification class. Kernel transforms the nonlinear separable m-dimensional feature space into linearly separable k-dimensional feature space, where k is more significant than m and is used to predict the class membership probabilities. Although choosing an appropriate kernel for an ML model is challenging. However, we use three (3) standard, functional, and practical covariance functions or kernels, namely, Radial Basis Function (RBF), Rational Quadratic (RQ), and Matern. RBF is used to transform the decision boundary to be curve-shaped. It is straightforward, and it needs to adjust two (2) hyperparameters, including the length scale-how much the decision region is spread, and the variance scale- the average distance of the function away from its mean (the penalty misclassification). RQ combines several RBFs with different length scales. It has additional parameters alpha to define the weight of the length scale. Matern kernel is a generalization of the RBF. It also has an additional parameter to control the smoothness of the RBF function. We use the above three (3) kernels with the GPC model and evaluate their performance in person identification from ear images.

B. Deep Learning ML Models

There are many object detection ML models available. However, they have broadly categorized into two (2) main groups: object detection using the classification and object detection using the regression. Object detection using the classification selects the region of interests (ROI) inside the image and then applies CNN to classify each ROI. On the other side, object detection with the regression uses a bounding box for an object and predicts the class of the object of the given image. Our experiment uses the YOLO model and detects objects using regression models.

Let us discuss a simple CNN architecture and a few general terms affiliated with CNN. In simple neural networks, the output unit interacts with every input unit using matrix multiplication. However, CNN uses convolution in the place of matrix multiplication. Two arguments are placed to the convolution, including input and the kernel. The kernel is a weighted average matrix with more than the recent measurements. Usually, kernel size is much smaller than the input size to optimize operations and storage memory size. Besides, traditional NN multiplies a weight matrix with one input element only once. However, in CNN, the kernel is applied at every input position. The output from the convolution is called a feature map. Features in convolution are shift equivalence but invariant to scale and rotation. A CNN is typically composed of three stages: convolution, detector, and pooling. The convolution stage performs many parallel convolutions (linear activation). The detector stage runs the linear activation through a non-linear activation function, and the pooling stage modifies(down-samples) a specific location of the output feature map from the detector stage, with a statistical operation of the nearby locations. Average and max statistic operations can be applied of a predefined rectangle size of the feature maps and correspondingly produce average and max pooling. Pooling helps to reduce the invariance of scale and rotation on the feature map.

We also implement the YOLO v3 [9] model to identify a person from ear images. The Darknet-53 NN architecture is used to implement the YOLO v3 architecture. The design comprises 106 fully convolution layers, 53 of which are from the Darknet-53 architecture. Fig. 3 depicts the detailed YOLO architecture. Three (3) different input images are predicted by YOLO v3. These detections are made in the 82nd, 94th, and 106th layers, respectively, in Fig. 3. The initial detection yields a feature map with 13 x 13 x 45 from a 416 x 416 input image. This feature map is only responsible for detecting more significant objects. A few convolutional layers process the feature map from layer 79 before being up-sampled by 2x to obtain 26 x 26 dimensions. This up-sampled feature map is concatenated with the layer 61 feature map. The combined feature map is processed through a couple more convolutional layers before arriving at the second detection layer (layer 94). It generates a detection feature map with the dimensions 26 x 26 x 45 that is only responsible for identifying medium-

	Type	Filters	Size	Output
	Convolutional	32	3x3	416 x 416
	Convolutional	64	3x3/2	208 x 208
1 x	Convolutional Convolutional Residual	32 64	1x1 3x3	208 x 208
	Convolutional	128	3 x 3/2	104 x 104
2 x	Convolutional Convolutional Residual	64 128	1x1 3x3	104 x 104
	Convolutional	256	3 x 3/2	52 x 52
8 x	Convolutional Convolutional Residual	128 256	1x1 3x3	52 x 52
	Convolutional	512	3 x 3/2	26 x26
8 x	Convolutional Convolutional Residual	256 512	1x1 3x3	26 x 26
	Convolutional	1024	3 x 3/2	13 x 13
4 x	Convolutional Convolutional Residual	512 1024	1x1 3x3	13x13
	Avgpool Connected Softmax		Global 1000	

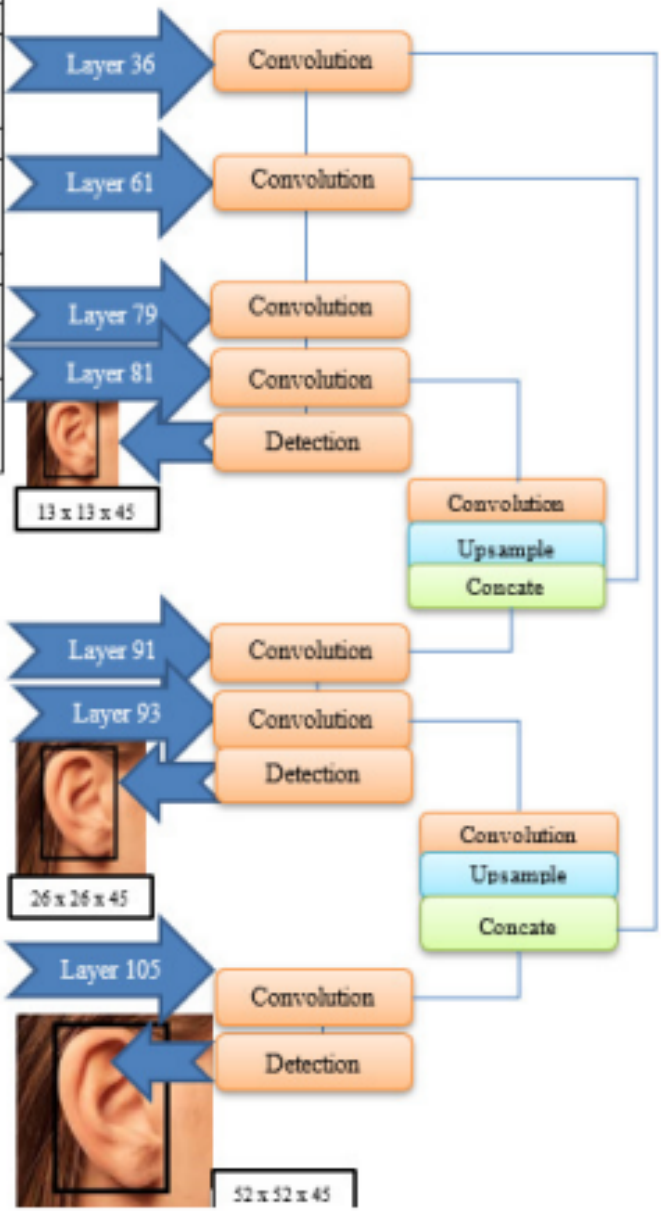

Figure 3. YOLOv3 network architecture [29]

sized objects.

Similarly, the layer 91 feature map is concatenated with a layer 36 feature map after handling a few convolutional layers. The merged feature map then reaches Layer 106, the third detection layer. In order to detect smaller objects, Layer 106 creates a feature map with a size of 52 x 52 x 45.

YOLO reduces an input image to 416x416 pixels and splits it into SxS size cells. The model predicts the number of bounding boxes (B), their confidence scores, and the likelihood of each object class (C). As a result, predictions can be encoded as SxS (Bx(5+C)). YOLO leverages the anchor box concept to solve the overlapping object problem in a single grid. The class-specific confidence scores for each box are calculated by multiplying the individual box confidence score with the conditional class probability. If a box's specific confidence score exceeds a specified threshold, it will be picked and used to locate the object inside the image. The YOLO working technique is depicted in Fig.4. YOLO v3 predicts 10,647 boxes at three distinct scales for the source image. The YOLO evaluation's ultimate loss is calculated by summing the localization, confidence, and classification losses. In addition, each detection feature map is run through a SoftMax activation function to determine the probabilities of the output classes. As a result, the person class in the output is defined by the most considerable probability value. The resulting image shows a bounding box with the person's class number and the probability value. We utilize Batch size 2, learning rate 0.001, epoch 20000, and momentum 0.9 in our implementation.

5. RESULTS AND ANALYSIS

Table I shows the performance of the Gaussian Process Classifier (GPC) with different kernels and features. Overall, ICA features perform better than PCA features. In PCA, we can reduce the number of components according to the variance criteria. However, such an option we do not have for ICA. RBF, RQ, and Matern perform similarly with ICA and PCA. However, Rational Quadratic with PCA performs the best accuracy, 96.2%. The closing training status of the YOLO model at Google Colaboratory GPU-based system is presented in Fig. 5. It shows avg loss is 0.076, and the loss is decreasing by 0.000100 rates. When the average loss surpasses 0.0628, we stop training. However, the model already takes 20000 iterations to reach this convergence. Four random images for every ten persons are selected to measure training accuracy and provide the findings in Table II.

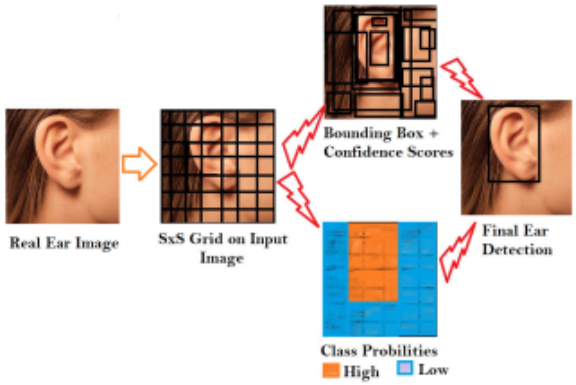

Figure 4. Working tactic of YOLO model

TABLE I. GPC MODEL PERFORMANCE

Function	Method	Accuracy
RBF	ICA	95.93%
RBF	PCA	89.6%
Rational Quadratic	ICA	93.2%
Rational Quadratic	PCA	96.2%
Matern	ICA	93.2%
Matern	PCA	89.6%

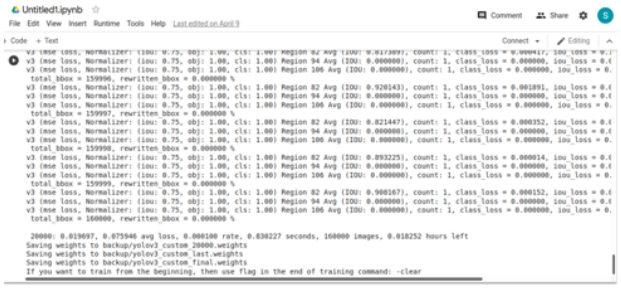

Figure 5. YOLO training status at Google Colaboratory

We choose 12.jpg as image 1, 34.jpg as image 2, 7.jpg as image 3, and 89.jpg as image 4. The table shows their respective YOLO prediction results: Incorrect, correct with confidence score 0.99868, correct with confidence score 0.991991, and correct with confidence score 0. 990771. The same method is followed for the remaining individuals. The accuracy of the model looks to be exceptional. Only seven images out of 40 have an incorrect result (82.5 percent accuracy) due to low image quality. In Fig. 6, we show the accurate identification (person 3) of the YOLO model and its confidence score during the runtime evaluation. Table III also includes the model's testing findings. Testing images are distinct from production images and should not be used in training. Twenty random images (two from every ten people) are picked to generate the recognition results among

all testing images. As per the table, YOLO produces five inaccurate outcomes out of a possible 20 (75% accuracy). Fig. 7 depicts the YOLO model's inaccurate forecast. It recognizes person nine based on the ear picture of person 8. Tables II and III show that our model has difficulty distinguishing amongst Person8 images. Naturally, YOLO fails to recognize little things, and the limited image quality reduces our testing accuracy.

Thus, we train the YOLO model on the complete EarVN1.0 data set (28,412 ear images from 164 (98+66) people) and evaluate the model performance. The training is stopped after specific periods (3 hours). The model training accuracy reaches 98.5%. However, YOLO takes 20,000 epochs to reach that accuracy. For testing, ten random people were chosen among 164. Each people have 100-200 sample images. Thus, we have taken around 1500 images to test the model. Rather than checking 1500, we choose 20 random images (2 random images from every ten persons) to evaluate the model performance. After that, the random testing images are applied to the model to evaluate the test performance. The test accuracy improves to 85% and presents in Table IV. Still, the testing data set of only ten random people may not be good enough. We will add more testing datasets in the future to test the statistical results.

Figure 6. Runtime identification of person 3, the model confidence score is 0.8

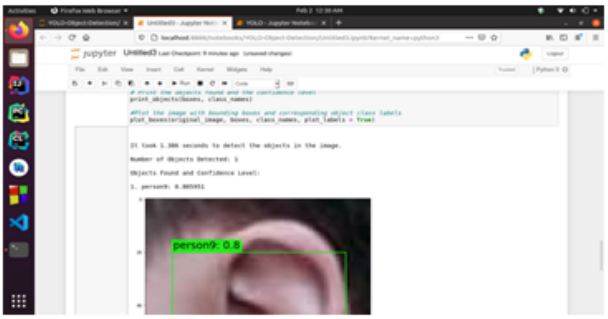

Figure 7. Runtime identification of person 8 but model predicts person 9 (wrong)

TABLE II. TRAINING RESULTS WITH YOLO

Person Number	Image1	Image2	Image3	Image4
1	Incorrect	0.99868	0.991991	0.990771
2	0.998923	0.973116	0.999400	Incorrect
3	0.998736	Incorrect	0.884654	0.999912
4	0.985723	Incorrect	0.424133	0.999705
5	1.00	0.999960	1.00	1.00
6	0.989879	Incorrect	0.999991	0.999444
7	0.995181	0.982481	0.999995	0.999218
8	1.00	0.582485	Incorrect	1.00
9	1.00	Incorrect	0.846152	0.962608
10	0.99999	0.99999	0.999977	0.998898

TABLE III. TESTING RESULTS WITH YOLO

Person Number	Image1	Image2
1	0.999671	0.99994
2	0.588093	0.999129
3	0.924887	Incorrect
4	Incorrect	0.999522
5	1.00	0.914625
6	0.999986	0.999803
7	0.999870	0.995278
8	Incorrect	Incorrect
9	Incorrect	0.999795
10	0.999998	0.974472

TABLE IV. YOLO TESTING RESULTS ON FULL DATA SET

Person Number	Image1	Image2
1	0.9999	0.9735
2	0.9065	0.9992
3	0.9558	0.5067
4	0.9886	0.9892
5	0.9397	0.9831
6	0.9783	0.9545
7	0.8586	0.9997
8	Incorrect	0.9968
9	Incorrect	1.00
10	0.9999	Incorrect

TABLE V. CONVERGENCE TIME AND TESTING ACCURACY OF YOLO AND CNN MODELS

	Training Time to Convergence	Accuracy in Testing
YOLO	20000 epochs take 3.0 hours.	85%
CNN	3000 epochs take 6.0 hours.	90%

On the complete EarVN1.0 dataset, we additionally use CNN and YOLO models. Accuracy level 98.5% or above is the threshold to reach convergence. According to Table V, CNN has a testing accuracy of 90%, whereas YOLO has an accuracy of 85%. Even though YOLO accuracy is not up to the mark; however, the convergence takes around half the time. If we disregard the 5% accuracy differential (incorrect identification of one out of 20), YOLO is far more suited to real-time use than CNN.

6. CONCLUSION

Many algorithms using information from various regions of the human body are utilized to identify a person. Nonetheless, ear identification has a distinct advantage over other biometric modalities since it has a distinct form and shape that does not alter over time. Thus, person identification using ear structure is preferable. In our study, we develop non-deep learning GPC ML models. RBF, RQ, and Matern kernels are applied with the model to improve the classification accuracy. GPC achieved 96% accuracy to classify the person from ear images. Among ICA and PCA, ICA performs better in general. Filtering techniques may help to improve classification performance. We will apply different filtering techniques with our model and evaluate the performance in the future.

On the other hand, numerous deep learning models, like CNN, R-CNN, and others, identify a person based on ear biometrics. Their efficiency is likewise between 90% and 99.9%. However, a quicker detection approach is required for these applications, and YOLO is the quickest object detection algorithm [30]. As a result, the YOLO machine learning approach is utilized and evaluated for the suggested application. It also gives a real-time experimental environment. YOLO shows 98% training accuracy and 85% testing accuracy.

Thus, this paper approaches two ML model implementations and their performance in person identification from ear images. GPC performs better, even better than CNN models. However, we focus on real-time outputs. YOLO can perform faster training and testing than other models. However, the accuracy is lesser. Besides, YOLO is not invariant to scale and rotation, which is the major drawback of CNN implementation. Our future goal is to improve the YOLO performance in real-time classifications. The finding of this article is that YOLO performs faster but has a small amount of accuracy difference from CNN. The accuracy model can be improved by using additional features inherited from non-deep ML models, including ICA or PCA. Overall ICA feature performs well in ear object detection. Thus, we will investigate combining the ICA feature value of the bounding box objects with YOLO prediction to improve the object detection accuracy. Practical implementation of this scheme is possible with low-cost devices supported by a secondary identification method.

REFERENCES

[1] A. Abaza and A. Ross, "Towards understanding the symmetry of human ears: A biometric perspective," in *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 2010, pp. 1–7.

[2] A. Bertillon, *La photographie judiciaire: avec un appendice sur la classification et l'identification anthropométriques*. Paris: Gauthier-Villars, 1890.

[3] R. Imhofer, "Die bedeutung der ohrmuschel fur die feststellung der identitat," *Archiv Kriminol Bd*, vol. 26, pp. 150–163, 1906.

[4] C. Champod, I. W. Evett, and B. Kuchler, "Earmarks as evidence: a critical review," *Journal of Forensic science*, vol. 46, no. 6, pp. 1275–1284, 2001.

[5] L. Meijerman, S. Sholl, F. De Conti, M. Giacon, C. van der Lugt, A. Drusini, P. Vanezis, and G. Maat, "Exploratory study on classification and individualisation of earprints," *Forensic Science International*, vol. 140, no. 1, pp. 91–99, 2004.

[6] P. Singh and R. Purkait, "Observations of external ear—an indian study," *Homo*, vol. 60, no. 5, pp. 461–472, 2009.

[7] A. Pflug and C. Busch, "Ear biometrics: a survey of detection, feature extraction and recognition methods," *IET biometrics*, vol. 1, no. 2, pp. 114–129, 2012.

[8] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[10] V. Truong Hoang, "Earvn1.0"," in *Mendeley Data, V4, doi: 10.17632/yws3v3mwx3.4http://dx.doi.org/10.17632/yws3v3mwx3.4*, 2020.

[11] A. Bertillon and R. W. McClaughry, *Signaletic instructions including the theory and practice of anthropometrical identification*. Werner Company, 1896.

[12] A. Lannarelli, "Ear identification. forensic identification series," 1989.

[13] M. Burge and W. Burger, "Ear biometrics in computer vision," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 2. IEEE, 2000, pp. 826–830.

[14] B. Moreno, A. Sanchez, and J. F. Vélez, "On the use of outer ear images for personal identification in security applications," in *Proceedings IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology (Cat. No. 99CH36303)*. IEEE, 1999, pp. 469–476.

[15] A. Kumar and C. Wu, "Automated human identification using ear imaging," *Pattern Recognition*, vol. 45, no. 3, pp. 956–968, 2012.

[16] Z. Mu, L. Yuan, Z. Xu, D. Xi, and S. Qi, "Shape and structural feature based ear recognition," in *Chinese Conference on Biometric Recognition*. Springer, 2004, pp. 663–670.

[17] T. Yuizono, Y. Wang, K. Satoh, and S. Nakayama, "Study on

individual recognition for ear images by using genetic local search," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 1. IEEE, 2002, pp. 237–242.

[18] H.-J. Zhang, Z.-C. Mu, W. Qu, L.-M. Liu, and C.-Y. Zhang, "A novel approach for ear recognition based on ica and rbf network," in *2005 International Conference on Machine Learning and Cybernetics*, vol. 7. IEEE, 2005, pp. 4511–4515.

[19] A. Abaza, A. Ross, C. Hebert, M. A. F. Harrison, and M. S. Nixon, "A survey on ear biometrics," *ACM computing surveys (CSUR)*, vol. 45, no. 2, pp. 1–35, 2013.

[20] K. Wang and J. Wu, "Machine learning and cybernetics," in *2006 International Conference On. Piscataway: IEEE*. Springer, 2006, pp. 3621–3626.

[21] F. I. Eyiokur, D. Yaman, and H. K. Ekenel, "Domain adaptation for ear recognition using deep convolutional neural networks," *iet Biometrics*, vol. 7, no. 3, pp. 199–206, 2018.

[22] T. Nanaware, S. Nandargi, K. Parag, N. Pote, and C. Rawat, "Ear biometric techniques: A comparative approach," *International Journal of Emerging Technologies and Innovative Research (www. jetir. org— UGC and ISSN Approved), ISSN*, pp. 2349–5162, 2019.

[23] C. Cintas, M. Quinto-Sánchez, V. Acuña, C. Paschetta, S. De Azevedo, C. C. S. de Cerqueira, V. Ramallo, C. Gallo, G. Poletti, M. C. Bortolini *et al.*, "Automatic ear detection and feature extraction using geometric morphometrics and convolutional neural networks," *IET Biometrics*, vol. 6, no. 3, pp. 211–223, 2017.

[24] Y. Zhang and Z. Mu, "Ear detection under uncontrolled conditions with multiple scale faster region-based convolutional neural networks," *Symmetry*, vol. 9, no. 4, p. 53, 2017.

[25] Ž. Emeršič, L. L. Gabriel, V. Štruc, and P. Peer, "Pixel-wise ear detection with convolutional encoder-decoder networks," *arXiv preprint arXiv:1702.00307*, 2017.

[26] I. I. Ganapathi, S. Prakash, I. R. Dave, and S. Bakshi, "Unconstrained ear detection using ensemble-based convolutional neural network model," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 1, p. e5197, 2020.

[27] W. Raveane, P. L. Galdámez, and M. A. González Arrieta, "Ear detection and localization with convolutional neural networks in natural images and videos," *Processes*, vol. 7, no. 7, p. 457, 2019.

[28] J. Du, "Understanding of object detection based on cnn family and yolo," in *Journal of Physics: Conference Series*, vol. 1004, no. 1. IOP Publishing, 2018, p. 012029.

[29] Z.-F. Xu, R.-S. Jia, H.-M. Sun, Q.-M. Liu, and Z. Cui, "Light-yolov3: fast method for detecting green mangoes in complex scenes using picking robots," *Applied Intelligence*, vol. 50, no. 12, pp. 4670–4687, 2020.

[30] S. Geethapriya, N. Duraimurugan, and S. Chokkalingam, "Real-time object detection with yolo," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 3S, 2019.

Shahadat Hossain Shahadat Hossain is an undergraduate student and researcher at the AISIP lab. His research interest includes Deep Learning algorithms and their implementation in real-life applications.

Sanjida Sultana Mitu Sanjida Sultana Mitu is an undergraduate student and researcher at the AISIP lab. Her research interest includes AI, machine learning models, and data mining algorithms..

Sadia Afrin Sadia Afrin is an undergraduate student and researcher at the AISIP lab. Her research interest includes machine learning, data mining, and big-data applications.

Dr. Shamim Akhter Dr. Shamim Akhter received his BSc and MSc in Computer Science degrees from AIUB in 2001 and AIT in 2005, respectively. In 2009, he received his Ph.D. from TITECH. He is currently employed as a Professor at IUBAT. Dr. Akhter has around 50 research articles in prestigious journals and conferences. His research interests include Machine Intelligence, Intelligent Algorithms, and Evolutionary Algorithms and Parallelization Models. He has been a member of IEEE since 2008 and a senior member since 2012. During these 20 years career, he worked on the program committee, as a reviewer, as a session chair, and as an organizer for over twenty IEEE conferences. He participated in a number of R10 regional award distribution events, seminars, and workshops as a volunteer.