User Identification through Iris features via Deep Convolutional Neural Networks

Yerassyl Zhalgasbayev

I. ABSTRACT

Iris recognition stands out among biometric methods for its non-invasive nature and enduring uniqueness over time. Despite its effectiveness, the classic Daugman method, while popular, suffers from a low false match rate. However, ongoing research presents opportunities for enhancement. This paper addresses accuracy improvement by integrating the Daugman method with convolutional neural networks (CNNs) as feature extractors. It establishes an efficient system by leveraging EfficientNetB1 for feature extraction and classification. Utilizing the CASIA-Iris Thousand dataset and preprocessing procedures ensures data readiness for training. As a result, the implementation of EfficientNetB1 yields a notable 97.7% accuracy in closed set recognition.

Table of Content

II. INTRODUCTION

In the era of technology, unlocking a phone with just a fingerprint or face recognition feels like a common thing. However, behind this simple action lies a whole system called biometrics. According to the National Institute of Standards and Technology (NIST), biometrics often refers to automated methods used to identify and confirm physical attributes of humans, such as voice patterns, face patterns, hand measurements, fingerprints, and eye retinas and irises.(Biometrics - Glossary | CSRC, n.d.). Most biometric analyses require some physical action from humans. The less invasive methods of authentication are face and iris detection. The fact that the face is a dynamic, three-dimensional (3-D) entity whose picture changes depending on viewing angle, posture, lighting, attire, and age, as well as that it is a changing social organ showing a range of expressions, makes face identification challenging (Daugman, 2004). The iris verification approach is a more realistic means of identifying individuals as the mature human iris pattern is distinct, unique, and stable (Wildes, 1997). The hypothesis that the iris structure remains consistent for identification as people age is backed by both clinical data and developmental biology (Nilsson et al., 2011). This paper will focus on the method of identifying a person based on the patterns in their irises, specifically utilizing closed set recognition, which seeks to recognize only individuals who have been previously learnt within the system.

The present study employs the Daugman technique as modified by Rathgeb et al. to effectively capture and segment the iris from images of the eye (Rathgeb, 2016). This methodology is a well-known method for iris recognition. In addition, convolutional neural networks (CNN) are used for closed set iris image recognition. CNNs, a form of deep learning model, are extremely good at identifying images because they can automatically learn features from the data (Craig & Awati, 2024).

This work chooses EfficientNet (Tan & Le, 2020) as the CNN architecture, which is well known for its efficiency and effectiveness in image categorization. This option allows us to use advanced techniques to improve performance.

My work begins by reviewing related literature and then progresses to discussing the methodology, detailing the dataset, outlining the recognition pipeline, and establishing the experimental protocol. In the results section, I describe the evaluation metrics used for closed set recognition.

III. RELATED WORK

This paper incorporates a reference to the work by Hafner A., et al. titled "Deep Iris Feature Extraction," published in 2021. In their study they used CASIA-Iris-Thousand dataset, which will be described deeply in the Methodology part of this report. This dataset contains 10 left and 10 right eye images for each of 1000 subjects, amounting to 20,000 images altogether. Given that each iris possesses a distinct pattern, even within the same individual, they considered each iris as a separate class, acquiring a classification problem with 2000 classes. In this work authors define two recognition pipelines: one for closed set recognition and other for open set recognition.

A. Closed set recognition:

Closed set recognition refers to the process where models are designed to identify only those identities that were included in the training dataset. In this pipeline, authors randomly took 1500 classes for closed set evaluation from their 2000 different classes of irises. This pipeline has two main stages, preprocessing and recognition, that solves the given 1500-class classification problem:

1. Preprocessing stage

In this stage authors used the adapted Daugman method for processes such as iris image acquisition and segmentation (Rathgeb, 2016). This method is based on the weighted adaptive Hough transform and ellipsopolar transform (WAHET).

Weighted adaptive Hough transform and ellipsopolar transform (wahet):

The Hough Transform is a computer vision approach for detecting forms in images, such as lines and circles. It turns these patterns into mathematical representations in a parameter space, making them easier to recognize even when they're damaged or obstructed (Bhatt, 2023). Weighted adaptive Hough transform enhances the classic Hough transform by dynamically changing parameters based on image attributes such as illuminance and aperture size, resulting in higher segmentation accuracy via repeated initialization, voting, and weighting steps (Uhl & Wildes, 2012). On the other hand, the ellipsopolar transform is a technique for mapping concentric ellipses to lines parallel to the axes, allowing for more precise boundary selection, particularly inner and outer borders in iris segmentation. Wahet boosts the resilience and reliability of iris recognition systems by boosting edge detection and segmentation accuracy (Uhl & Wildes, 2012).

By using this WAHET technique they segmented iris from input image and normalized the segmented iris image to size 256x64. After, this output is stacked four times on top of each other to get an image of size 256x256 as shown in Figure-1. By doing this they received a square sized output, which is required by deep learning models.

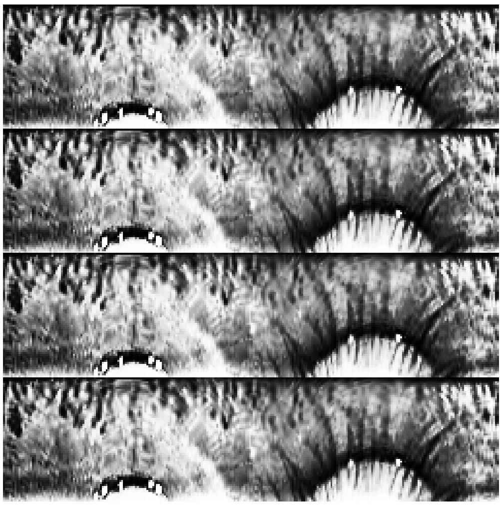

Figure-1.Stacked Iris 256x256

2. *Recognition (Fine-Tuning):*

At this stage, the authors conducted experiments by fine-tuning two distinct pre-trained CNN models, ResNet-101(He et.al, 2016) and DenseNet-201(Huang, et. al, 2017), which were initially trained on the ImageNet dataset (Deng et al., 2009). These architectures are renowned for their deep learning capabilities, particularly in handling complex image recognition tasks. The overall comparison of the two architectures for the ImageNet dataset is given in the following Table-1.

Table-1. Comparison of ResNet-101 and DenseNet-201 models for ImageNet Dataset

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
ResNet101	171	76.4%	92.8%	44.7M	209
DenseNet201	80	77.3%	93.6%	20.2M	402

- *Model: CNN architecture names*
- *Size (MB): Size of the model in Megabytes.*
- *Top-1 Accuracy: This metric reflects the accuracy of each model when only the top prediction is considered.*
- *Top-5 Accuracy: Similar to top-1 accuracy, this metric measures how often the correct answer is within the top 5 predictions made by the model.*
- *Parameters: This indicates the total number of trainable parameters in each model.*
- *Depth: Refers to the topological depth of the network. This includes activation layers, batch normalization layers etc.*

Utilizing pre-trained models accelerates the learning process and enhances accuracy because the initial layers already possess filters that have been trained to identify edges and color blobs - features common in many images. This strategy is particularly advantageous when there is a limited amount of data available, which aligns with their scenario where they have only 10 images for each class (iris).

Since both architectures were pre-trained on the ImageNet dataset, the final layers of these networks have 1000 outputs to correspond with the total number of classes in this dataset. To adapt these networks to their closed dataset with 1500 classes, they removed the final layers of each architecture and added a fully connected layer with 1500 outputs.

Thus, the preprocessed images of the irises were fed into these networks, and the class corresponding to the highest value in the output layer was assigned to each image.

B. Open set recognition:

Open set recognition is a process designed to identify known classes and also has the capability to recognize that an input does not belong to any of the classes seen during training. In other words, it can handle unknown classes that were not part of the training dataset. This type of recognition is more practical in real life situations, since new identities can be enrolled on a daily basis. To circumvent the need for constantly reconfiguring the output layer and retraining the network with each new identity, the authors employed the model trained for closed set recognition as a feature extractor. To assess this pipeline, the remaining 500 out of 2000 iris classes, which were not included in the closed set evaluation, are used.

This pipeline consists of three main stages:

1. Preprocessing

This stage uses exactly the same preprocessing methods used in closed set recognition, WAHET algorithm and stacking of normalized iris images.

2. Enrollment

In this stage the model trained for the closed set recognition is used. New identities (iris images) which were not included in the training of the model are fed through the network and extracted just before the last output (prediction) layer. Then, these outputs are flattened into 2048-dimensional feature vectors and stored in the database.

3. Recognition

During the recognition process, the iris image is passed through the model to extract a feature vector as outlined in the enrollment stage. Subsequently, cosine similarity (1) is computed between this feature vector and every other feature vector stored in the database.

$$cos(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||} \qquad (1)$$

The input iris image is then classified based on the class of the feature vector in the database that exhibits the highest similarity. The overall architecture of their proposed model is shown in Figure-2.(Hafner et. al, 2021)

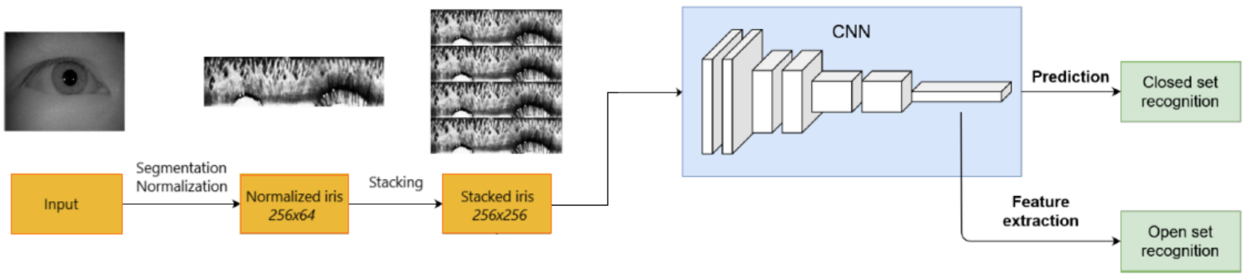

Figure-2. Iris recognition pipeline for open and closed sets proposed

IV. METHODOLOGY

The central emphasis of this project was on closed set recognition. As such, the methodology outlined in this section is dedicated solely to that aspect. I begin with an overview of the dataset utilized for my project, followed by a discussion on preprocessing. After, I delve into the details of the model architecture and elaborate on the training procedure employed in my study.

A. Dataset

The CASIA-Iris Thousand dataset was used in this study. The CASIA-Iris-Thousand dataset, developed by the Chinese Academy of Sciences' Institute of Automation (CASIA), includes 20,000 photos of 1,000 people's iris. The photos were captured using an IKEMB-100 near-infrared (NIR) camera. Each participant submitted ten photographs of their left and ten photographs of their right eyes. Notably, spectacles and mirror reflections appear in several of the images, encouraging variation within each class. Two examples from the dataset are illustrated in Figure-3.

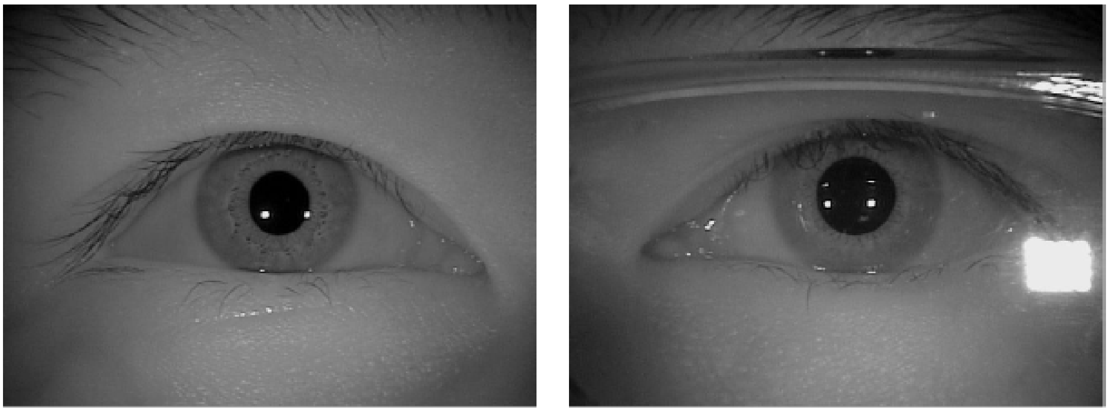

Figure-3. Sample images from the CASIA-Iris-Thousand dataset, featuring the left and right eyes of two distinct subjects. The subject depicted on the right side of the figure is wearing glasses.

B. Preprocessing

Drawing inspiration from the study presented in (Hafner et. al, 2021) I employed the same WAHET (Weighted Adaptive Hough Transform and Ellipsopolar Transform) technique to segment the iris from the eye images. However, contrary to treating each iris as an individual class, I categorized the left and right irises of the same individual as a single class. Consequently, my dataset, CASIA-Iris-Thousand, consists of 1000 classes corresponding to 1000 subjects. Furthermore, rather than duplicating the same segmented and normalized iris image four times in a stack, I opted for a combination of left and right iris images. Given that each individual possesses 10 images for each of their left and right irises, I paired each left iris image with a unique right iris image, making certain that the selected right iris image was not matched with any other left iris image (one to one relationship). So, I stacked two copies of this left iris image on the top, followed by two copies of the right iris image at the bottom. The output of this preprocessing stage is shown in the figure below.

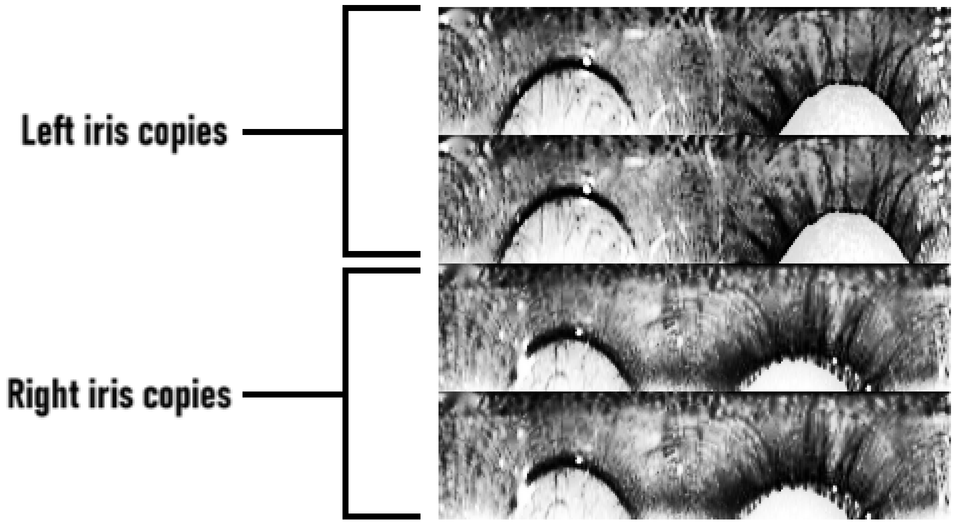

Figure-4. Left and right iris images stack on top of each other giving an output of size *256x256*

C. Model Architecture

For the closed set recognition I also utilized a pre-trained CNN model, particularly EfficientNetB1, which was initially trained on the ImageNet dataset. This model was chosen because of its exceptional balance between efficiency and accuracy. EfficientNetB1 is part of the EfficientNet family, which uses a compound scaling method to uniformly scale the depth, width, and resolution of the network based on a set of fixed scaling coefficients. This approach allows EfficientNetB1 to achieve higher accuracy with fewer parameters and less computational cost compared to other models of similar complexity. Its architecture is designed to optimize performance, making it highly suitable for tasks requiring detailed feature extraction from images, such as iris recognition, even when computational resources are limited or efficiency is a priority. The table below presents a comparative analysis of this model against the two networks selected by (Hafner et. al, 2021) for the Imagenet Dataset, showcasing its effectiveness in terms of both size and accuracy.

Table-2. Comparison of EfficientNetB1 with other two models for Imagenet Dataset

Model	**Size (MB)**	**Top-1 Accuracy**	**Top-5 Accuracy**	**Parameters**	**Depth**
ResNet101	171	76.4%	92.8%	44.7M	209
DenseNet201	80	77.3%	93.6%	20.2M	402
EfficientNetB1	***31***	***79.1%***	***94.4%***	***7.9M***	***186***

For the training and evaluation of my selected model on closed set recognition, I randomly selected 750 subjects from the total of 1000 classes. Consequently, I adjusted the final layer of EfficientNetB1 to have 750 outputs, adapting from its original configuration which, being pre-trained on the ImageNet dataset, had 1000 outputs. Thus, the preprocessed irises are inputted into the network, and the classes that correspond to the maximum value on the output layers are assigned to the input irises. The proposed architecture of my model is shown in the figure-5 below.

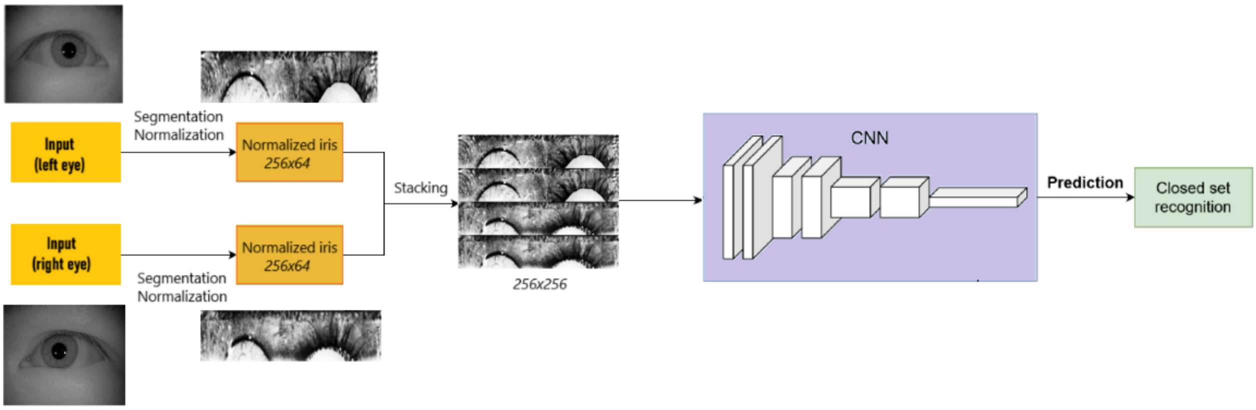

Figure-5. Architecture of our proposed model

D. Training Procedure

In my study, I divided my dataset into two segments. The first segment, consisting of 750 subjects selected at random, was used for closed set recognition evaluation and to train the Convolutional Neural Network (CNN) model. The remaining 250 subjects were reserved for the evaluation of open set recognition for our future work. For the closed set evaluation, I designated 7 images from each class for training, 2 images for validation, and 1 image for testing purposes.

I utilized the Adam optimizer and categorical cross-entropy loss function to train EfficientNetB1. The model underwent training with a learning rate of 0.001 over 5 epochs. This training process was conducted on Google Colaboratory, making use of its free GPU resources. The model, along with its initial weights, was sourced from the PyTorch library (Paszke et.al, 2019). The summary of hyperparameters is presented in the table below.

Table-3. Hyperparameters and their values used in our experiment

Hyperparameter	**Value**
Adam optimizer	True
Learning rate	0.001
Batch size	32
Epochs	5

V. RESULTS AND DISCUSSION

In this part, I specify the metrics that will be used to evaluate the closed set recognition. I then present the findings.

A. Metrics

To evaluate the effectiveness of our closed set recognition algorithm, I employ three metrics: rank-1 (also known as top-1) accuracy, rank-5 (also known as top-5) accuracy, and the average ROC AUC score of all classes in the set.

In closed set recognition, I utilize the softmax function to transform the final linear layer's output into probabilities for each class. The sum of these probabilities for each class equals 1, indicating the level of confidence in the forecast for each class. The Rank 1 accuracy is determined by comparing the ground truth label to the class label with the highest likelihood.

Similarly, rank-n accuracy looks for the presence of the ground truth label among the top n predictions with the highest likelihood. In both cases, the accuracy is calculated by dividing the number of right predictions by the total number of test cases.

B. Results

After training my model for 5 epochs, I observed immediate convergence in both the loss and accuracy metrics. The results of the training are shown in the Figure-6 below.

Figure-6. Loss and Accuracy values for training 5 epochs

The loss graph shows a rapid decline initially, indicating the model's quick adaptation to the data. Similarly, the accuracy graph reveals a sharp increase early on, with both training and validation accuracy stabilizing at high levels, denoting effective learning and generalization capabilities of the model within a few epochs. This efficiency is particularly advantageous when contrasted with the two models referenced in (Hafner et. al, 2021), which required training for 80 epochs.

Afterwards, I evaluated my model for the test set. First, I calculated the ROC AUC score for each class with OvR, One versus Rest, method. OvR is a strategy employed to evaluate multiclass models by comparing each individual class against all remaining classes simultaneously. In this method, one class is designated as the "positive" class, and the rest are collectively labeled as the "negative" class.

This approach effectively simplifies the complexity of a multiclass classification problem to a binary classification framework, allowing us to apply all the known binary classification metrics for evaluation.

For every class in the dataset, this process is replicated, which means that for my dataset with 750 classes, I would compute 750 distinct OvR scores. After computing these scores for each class, they can be averaged to yield a comprehensive OvR score for the model. Fortunately, the task of computing the One vs Rest (OvR) scores is greatly simplified by a built-in function in the scikit-learn library, which automates the process for us:

```
roc_auc = roc_auc_score(labels_ground_truth, labels_probabilities, average='macro', multi_class='ovr')
roc_auc
```

```
0.9999946595460615
```

Figure-7. ROC AUC score function for each class and test set value. The ‘average=macro’ parameter in the above code means to calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.

Based on the ROC AUC score shown in the above Figure-7, which is approximately 0.9999, we can state that my model has an exceptionally high ability to distinguish between the different classes. A score this close to 1.0 indicates that the model has excellent classification performance with a very high True Positive Rate (TPR) and a very low False Positive Rate (FPR) across all classes. This suggests that the model is highly effective at correctly identifying the positive class for each OvR (One versus Rest) comparison and is robust across varied thresholds.

Next, I compared my model’s accuracy with the models indicated in the (Hafner, et. al, 2021). The results of the comparison are shown in the table below.

Table-4. Accuracy of the models for the closed set recognition

Model	**Rank-1 accuracy**	**Rank-5 accuracy**
DenseNet-201	97.3%	99.3%
ResNet-101	96.2%	98.7%
EfficientNetB1 (My)	***97.7%***	***99.86%***

From the table we can see that all CNN architectures are performed well, in both ranks the values are higher than 96%. However, my model, EfficientNetB1, outperformed both DenseNet-201 and ResNet-101 in rank 1 and 5 accuracy with 97.7% and 99.86%.

Finally, for a comprehensive evaluation of my model, I generated a confusion matrix which is shown in Figure-8 on the next page. Given the extensive number of classes in our closed set, which is 750, the resulting confusion matrix is quite large. Nevertheless, it is evident from the matrix that my model has achieved commendable performance in the given task. The prominently illuminated diagonal line represents instances of correct predictions, showing a substantial accuracy rate. The predominantly dark backdrop of the matrix implies minimal misclassifications, as the off-diagonal elements that would indicate incorrect predictions are scarcely noticeable.

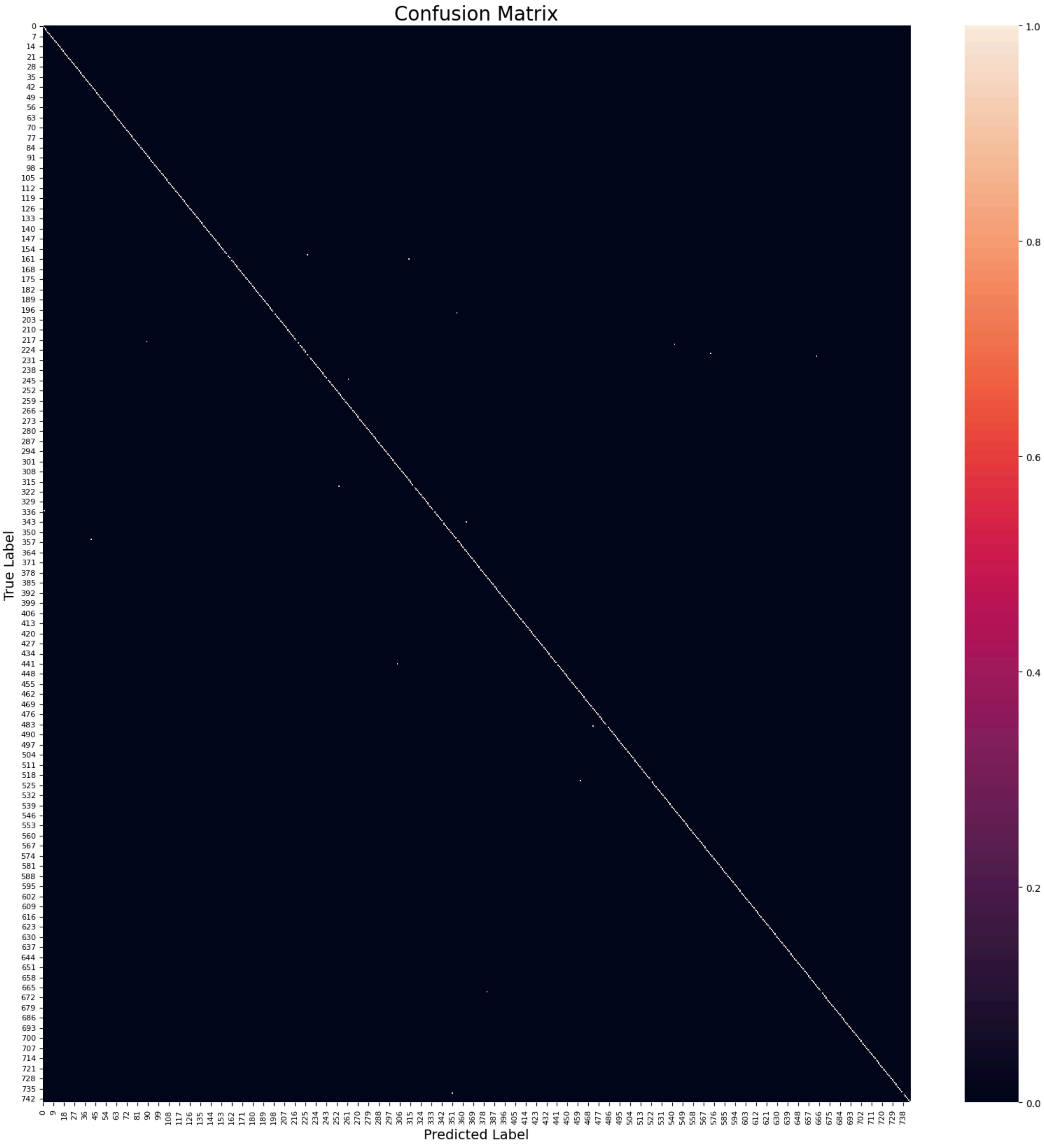

Figure-8. Confusion matrix of test set

VI. CONCLUSION

This paper offers an efficient user identification system that blends traditional biometric techniques with cutting-edge deep learning technology. By merging Daugman's iris segmentation method with EfficientNetB1 for feature extraction and classification, an efficient and accurate system is built. Using the CASIA-Iris Thousand dataset and pre-processing procedures guarantees that the data will be available for training. The emphasis is on closed set recognition, which is accomplished by changing the last layer of EfficientNetB1 to contain fewer classes. The findings indicate that EfficientNetB1 offers size and performance benefits over other models. This research contributes to the development of user identification technology, which has the potential to be employed in security and authentication systems. Future research will focus on enhancing the algorithm and incorporating open set recognition. Overall, this work indicates the viability of merging classic biometric approaches with contemporary deep learning methodology, opening the door for future breakthroughs in user identification technology.

REFERENCE

Bhatt, D. (2023, August 18). A Complete Guide on Hough Transform. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/06/a-complete-guide-on-hough-transform/#:~:text=Hough%20Transform%20is%20a%20computer,they're%20broken%20or%20obscured.

Biometrics - Glossary | CSRC. (n.d.). https://csrc.nist.gov/glossary/term/biometrics

Bowyer, K. W., & Burge, M. J. (2016, July 28). Handbook of Iris Recognition. Springer. http://books.google.ie/books?id=tNLBDAAAQBAJ&printsec=frontcover&dq=Handbook+of+Iris+Recognition&hl=&cd=1&source=gbs_api

Craig, L., & Awati, R. (2024, January 11). convolutional neural network (CNN). Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network#:~:text=A%20convolutional%20neural%20network%20(CNN)%20is%20a%20category%20of%20machine,suited%20to%20analyzing%20visual%20data.

Daugman, J. (2004, January). How Iris Recognition Works. IEEE Transactions on Circuits and Systems for Video Technology, 14(1), 21–30. https://doi.org/10.1109/tcsvt.2003.818350

Deng J., Dong W., Socher R. , L. -J. Li, Kai Li and Li Fei-Fei,(2009) "ImageNet: A large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

Hafner A, Peer P., Emeršič Z. and Vitek M. (2021) "Deep Iris Feature Extraction," *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Jeju Island, Korea (South), 2021, pp. 258-262, doi: 10.1109/ICAIIC51459.2021.9415202.

Huang G., Liu Z.,Van L. Der Maaten and Weinberger K. Q., (2017) "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 2261-2269, doi: 10.1109/CVPR.2017.243

He K., Zhang X., Ren S. and Sun J., (2016) "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

Infographic: biometrics on the silver screen. (2018, April 9). Thales Group. https://www.thalesgroup.com/en/markets/digital-identity-and-security/magazine/infographic-biometrics-silver-screen

Nilsson, S. F. E., Hoeve, J. V., Wu, S., Kaufman, P. L., & Alm, A. (2011, April 4). Adler's Physiology of the Eye. Elsevier Health Sciences. http://books.google.ie/books?id=1oIlvnXnJvEC&printsec=frontcover&dq=Adler%27s+Physiology+of+the+Eye&hl=&cd=1&source=gbs_api

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (1970, January 1). *Pytorch: An imperative style, high-performance deep learning library*. Advances in Neural Information Processing Systems. https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

Tan, M., & Le, Q. V. (2020, September 11). *EfficientNet: Rethinking model scaling for Convolutional Neural Networks*. arXiv.org. https://arxiv.org/abs/1905.11946v5

Rathgeb C., Uhl C, Wild P., and Hofbauer H.,(2016) "Design decisions for an iris recognition sdk," in Handbook of Iris Recognition, second edition ed., ser. Advances in Computer Vision and Pattern Recognition, K. Bowyer and M. J. Burge, Eds. Springer.

Uhl A. & Wild P. (2012) *Weighted Adaptive Hough and Ellipsopolar Transforms for Real-time Iris Segmentation*. Retrieved from: https://www.researchgate.net/publication/267556461_Weighted_Adaptive_Hough_and_Ellipsopolar_Transforms_for_Real-time_Iris_Segmentation

Wildes, R. (1997). Iris recognition: an emerging biometric technology. Proceedings of the IEEE, 85(9), 1348–1363. https://doi.org/10.1109/5.628669