113-2 NTU Computer Vision

Ganzin Project Group 2 你是我的眼

Final Report

Team Members:

陳勇嘉 R13921104

何家祥 R13921072

陳芷寧 R13943177

I. Introduction

Biometric authentication based on iris recognition has emerged as a reliable and secure method due to the unique texture patterns found in the human iris. This report presents a complete iris recognition pipeline—from raw image preprocessing to identity matching—focused on accurately distinguishing between users based on their eye images. The system begins with iris segmentation using RITNet, a deep learning model designed for precise localization of iris and pupil boundaries. Following segmentation, the iris region is normalized using Daugman's rubber sheet model, which transforms the circular iris region into a fixed-size rectangular representation via polar-to-Cartesian unwrapping.

To enable identity comparison, the normalized iris templates are passed through a Siamese neural network built upon a ResNet architecture. This allows the system to learn a robust encoding space where iris images from the same individual are close together, and those from different individuals are far apart. The pipeline is designed to be scalable, generalizable, and suitable for both intra-class (same eye) and inter-class (different user) recognition tasks.

II. Methodology

1. Iris Image Segmentation

A. Segmentation with RITNet

To isolate the iris in eye images, we chose RITNet, a fast and efficient deep neural network originally built for retinal segmentation. We adapted it for iris images because it's great at handling circular structures with clear edges, which is key for accurate iris analysis in biometric applications.

Advantages of using RITNet:

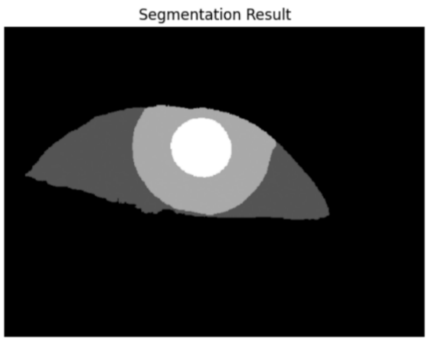
- Sharp boundaries: It captures fine details, ensuring precise iris outlines.
- Lightweight design: Runs quickly, even on basic hardware.
- Flexible across conditions: Handles varying iris sizes and lighting well.

Architecture Overview:

RITNet, is a lightweight encoder-decoder neural network designed for real-time iris segmentation.

- Encoder: Downsamples input iris images (e.g., 640x400) to extract hierarchical features. Uses convolutional layers with increasing filters to capture multi-scale patterns.
- Decoder: Upsamples features to reconstruct a segmentation map. Employs transposed convolutions or upsampling layers to restore spatial resolution.
- Skip Connections: Links encoder and decoder layers, combining low-level texture details with high-level semantic features for precise boundary detection.
- Output: Produces a mask with four labels (0 for background, 1 for eye white, 2 for pupil, 3 for iris) via a softmax layer.

We used a pre-trained RITNet model provided by the original authors on GitHub. As the training code and data preparation were already available, we directly adopted their model weights (best_model.pkl) without retraining from scratch. Therefore, specific training hyperparameters (e.g., epochs, batch size) are not detailed here.



Segmentation result of RITNet

B. Pre-process for segmentation

To prepare the grayscale iris images for RITNet inference, the following preprocessing steps were applied:

- Rescaling – Each image was resized to a fixed height of 400 pixels while maintaining its aspect ratio.
- Normalization – Pixel values were normalized from [0, 255] to [-1, 1] using mean 0.5 and standard deviation 0.5.
- Padding – The width was padded with symmetric reflection to reach 640 pixels, ensuring consistent input dimensions for the model.

These steps ensure compatibility with the RITNet architecture without altering semantic content of the original iris image.

C. Post-process for segmentation

Some eye images suffer from specular reflections caused by eyeglasses, which can lead to segmentation errors where the model fails to correctly identify regions such as the pupil and iris. To detect such cases after generating the initial segmentation labels, we employed two evaluation strategies:

- Pixel Count Threshold: Specular reflection often causes the iris to be partially or entirely missed during segmentation, resulting in a significantly reduced number of pixels labeled as iris (label = 3). We set a threshold, and if the number of iris pixels falls below it, the segmentation is considered invalid.
- Mask Coverage Ratio: Before segmentation, we fit two circular masks to approximate the pupil and iris regions. After segmentation, we calculate the proportion of the predicted iris mask that falls within the predefined iris region. If this coverage is below a certain threshold, the prediction is also considered incorrect.

For images identified as erroneous using the above criteria, we apply image enhancement techniques to improve segmentation quality. Specifically, we increase local contrast using CLAHE (Contrast Limited Adaptive Histogram Equalization) and reduce noise via bilateral filtering. These preprocessed images are then re-evaluated using the RITNet model to obtain more accurate segmentation results.

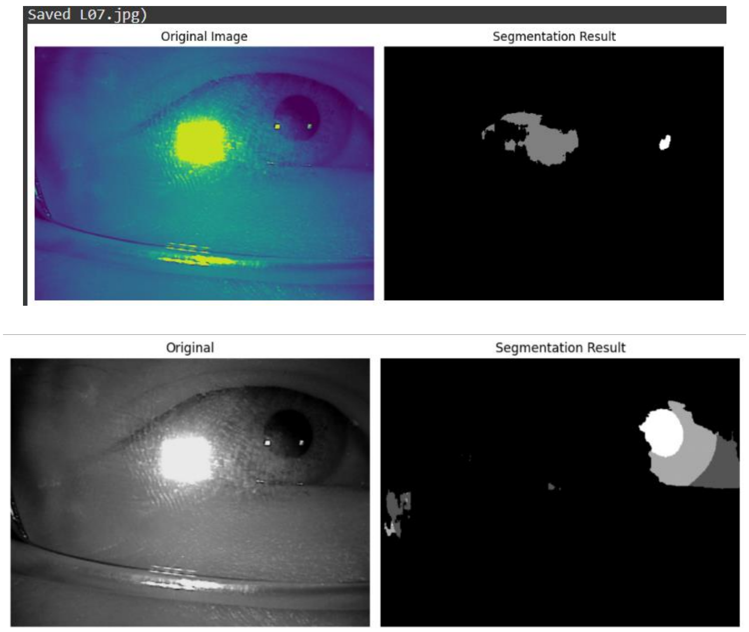


Illustration of the image before and after contrast enhancement.

2. Segmented Image Normalization
After segmenting and localizing the iris and pupil, we applied iris normalization to convert the circular iris region into a standardized rectangular image. This process follows the rubber sheet model proposed by Daugman, enabling consistent representation across different eye images for biometric recognition.

Method Overview:
We used the rubberSheetNormalisation function, which takes the circular boundaries of the pupil and iris and unwraps the annular region into a fixed-size rectangular form. The main steps include:

A. Circle Detection :
   ○ The segmentation mask is processed to extract contours for the pupil (label = 3) and iris (label = 2).
   ○ cv2.minEnclosingCircle is applied to obtain the center coordinates and radii for both the pupil and iris.

B. Polar-to-Cartesian Unwrapping :
   ○ The circular region is sampled along the angular direction
   ○ For each angle, points are interpolated radially from the pupil boundary to the iris boundary.
   ○ Bilinear interpolation is used to extract pixel values from the original image.
   ○ The result is a normalized rectangular image (e.g., 100 × 480 pixels) of the iris.

   Implementation Key Parameters:
   ○ angle_samples = 480: Number of samples along the circular direction
   ○ radius_samples = 100: Number of samples from the pupil to the iris edge
   ○ use_interpolation = True: Enables bilinear interpolation for smoother results

   Invalid Mask Filter:
   ○ If a sampled point is outside the iris region (mask != 2), the value is set to zero to suppress noise.

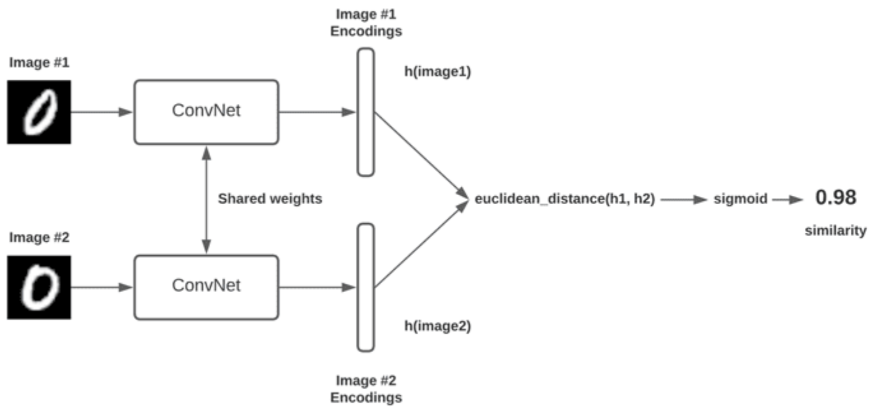
   Advantages:
   ● Standardized output size: The normalized image has a consistent shape and scale, suitable for CNNs or template matching.
   ● Geometric distortion compensation: Transforms the annular iris region into a linear format while preserving spatial relationships.

- Segmentation-aware sampling: By leveraging the mask, only valid iris pixels are retained, improving accuracy.

3. Siamese Neural Network (SNN)

For the Deep Learning Model, we have used the Siamese Neural Network framework to compare and produce the similarity scores for 2 input images.

SNN is made up of 2 identical neural networks, with shared weights between each other. SNN can take in 2 different image inputs and output the vectors of the feature encodings. Then, the network will compare the encodings to measure how similar the 2 inputs are and give it a similarity score.



Architectural design of a Siamese Neural Network

- Feature Extractor CNN: Typically a small CNN like ResNet18, MobileNet, etc. For this project, we have used both ResNet-18 and ResNet-50 as our feature extractor.

- Encodings: A vector, which may be 128-dimensional or 256-dimensional, representing the image.

- Distance Layer: Computes Euclidean or Cosine distance between encodings.

- Loss Function: Trains the network to minimize distance for similar irises and maximize it for different ones.

The feature extraction model that we have used is Resnet. ResNet, short for Residual Network, is a deep convolutional neural network architecture introduced by Microsoft Research in the 2015 paper "Deep Residual Learning for Image Recognition" by Kaiming He et al. It won the ImageNet 2015 competition by a large margin and has since become one of the most influential models in deep learning.

Before ResNet, training very deep neural networks (e.g., more than 20 layers) often resulted in degradation—as more layers were added, accuracy got worse, not better. This was not due to overfitting but to optimization difficulties (such as vanishing gradients).

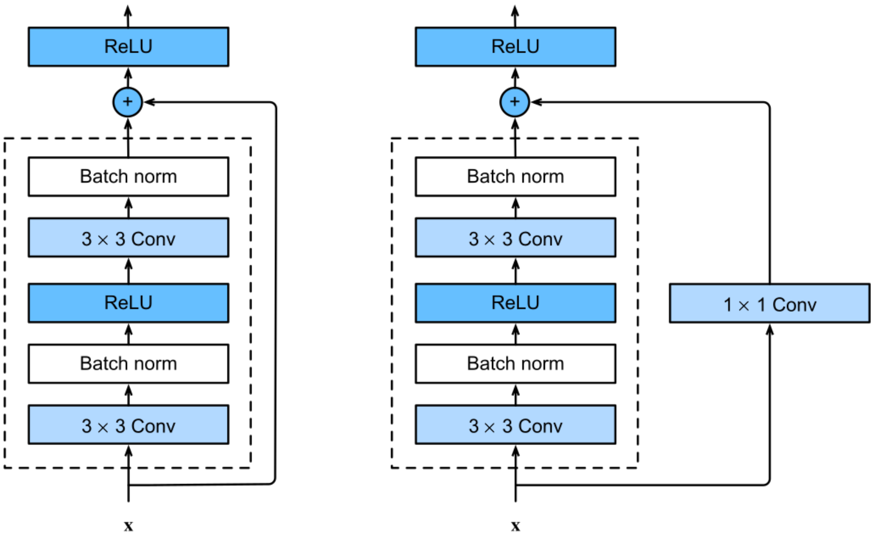
ResNet solves this by introducing residual blocks that use skip connections (also called identity shortcuts).

Instead of learning a direct mapping $H(x)$ from input x to output, the network learns the residual function $F(x) = H(x) - x$, which is rewritten as:

$$H(x) = F(x) + x$$

This formulation allows the layers to learn modifications to the input rather than the entire transformation, making it easier to optimize.

A typical residual block is mainly made up of Convolution layers and Rectified Linear Unit (ReLU) as activation function. A skip connection is also provided which allows upper layers to be referenced whilst the model is still undergoing.

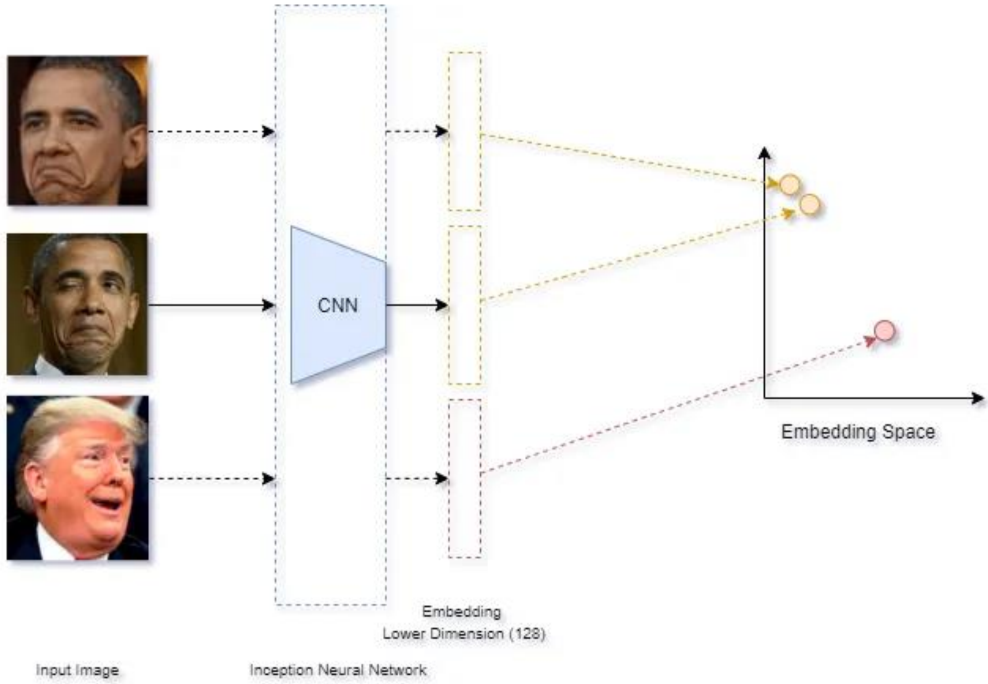


ReLU structure

ResNet's ability to go deep without degradation enables it to extract fine-grained patterns from iris textures. The skip connections help preserve gradient flow, making training stable even on high-resolution biometric data. Pretrained ResNets can be fine-tuned for iris tasks, reducing training time and improving performance. ResNet works well with Siamese networks to compare iris encodings in verification or recognition scenarios.

To train the SNN, we must create positive and negative image pairs, where positive pairs are the normalized images of the eyes coming from the same subject, while the negative pairs can be the normalized images coming from different subjects. In this project, we have used a Pairs Generator Python script (generate_pairs.py) to generate the positive pairs and negative pairs.

For the loss function, we have used Contrastive Loss since it encourages the network to output close encodings for similar pairs and far encodings for dissimilar pairs. This allows us to group similar features obtained from the irises to be grouped together, allowing us to identify the same subject.



Contrastive Loss illustration

As we can learn from the illustration, we are hoping that the features captured by the ResNet model of images of President Obama can be grouped closer, even though the image is taken in different angles, it still can identify if it is the same subject through the common features.

Contrastive loss equation:

$$\mathcal{L}(x_i) = -\log \left[\frac{\exp(s_{i,i}/\tau)}{\sum_{k\neq i} \exp(s_{i,k}/\tau) + \exp(s_{i,i}/\tau)}\right]$$

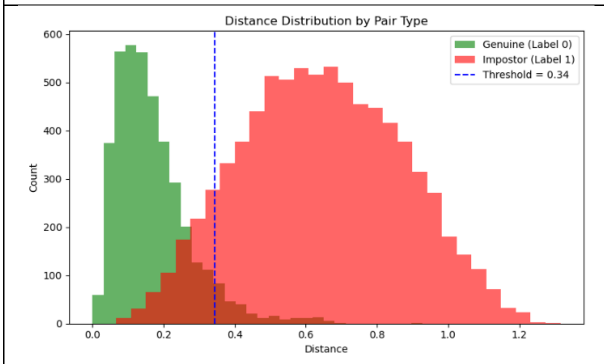
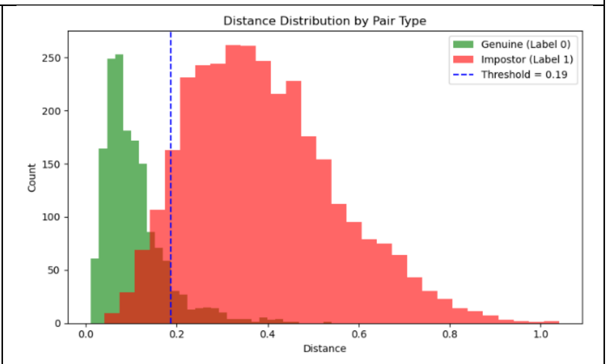
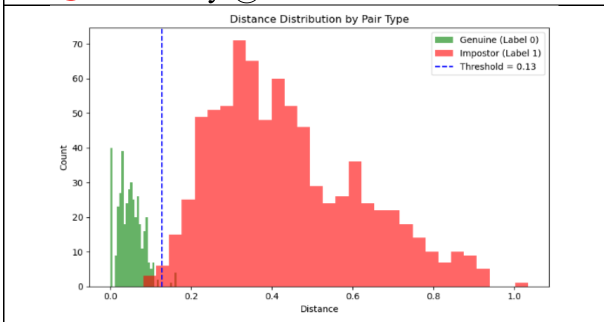
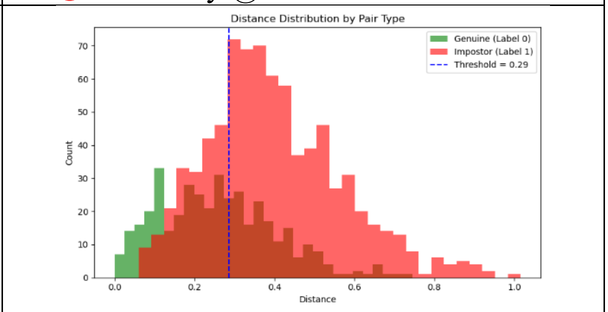
Where, when $\mathcal{L}(x_i) = 0$ would indicate that the 2 images have high similarity, while $\mathcal{L}(x_i) = 1$ would indicate that the 2 images are very different.

After training the model, we are hoping to obtain 2 clusters of data, one at each end in the best-case scenario, indicating that the model can successfully identify the same subject through the normalized rubber sheet images of the irises. Yet, most of the time there would still be a small overlap due to the model not having the confidence that the input images are from the same subject. Thus, a threshold has to be considered in order to successfully segmentate the 2 clusters of data from each other.

SNN is very suitable for our work since it learns by comparing 2 input images, which allows us to avoid large number of classes. SNN is robust enough, as it would learn to ignore irrelevant noise such as reflection of light from some portion of the glasses being worn by the subjects.

4. Model Training and Validation
For model training, we only used the Thousand and Light dataset normalized images. The reason we did not use Gaze dataset in our training is because the Gaze dataset is too small to be significantly useful for our work, we have also discovered that the Gaze dataset would affect with the general accuracy of other datasets, possibly due to the images taken from a slanted angle.

For model training, we have introduced a pairs maker, which helps us to generate positive and negative pairs for our model to be trained on, we initially let the same subject, regardless of the left or right eye, to be paired as positive pairs. Yet, we have discovered that the training results for this arrangement has a much worse result compared to separating the different sides of eyes of a same subject. Hence, in our training arrangement, only the same subject with the same eye is used.

Threshold Selection Results	
	
Thousand 🎯 Accuracy @ threshold 0.34: 0.9135	Lamp 🎯 Accuracy @ threshold 0.19: 0.9135
	
Ganzin 🎯 Accuracy @ threshold 0.13: 0.9925	Bonus 🎯 Accuracy @ threshold 0.29: 0.7008

III. Results

After numerous model trainings, we have found out that using the ResNet50 architecture with only the normalized scores within the range of 0 and 1, allow us to have the best Dscore performance in identifying the same user for the Thousand and Lamp Datasets, while the ResNet101 architecture with binarization using the aforementioned threshold is the most suitable for our Gaze and Bonus datasets.

For our work, we do think that more preprocessing should be done since for some of the images, the iris center detection might not be that accurate, causing the creation of the circles and the rubber sheet process to be not as good as it should be. This has caused problems for images that the subjects are wearing glasses and also for the Gaze dataset with the images of irises taken from a slant angle.

In future works, we do hope to improve on our image preprocessing abilities by including circle augmentation and adding filters so that the features stand out more. We also hope to experiment with more types of models such as TripletNN and also GoogLeNet, and also different loss functions such as Triplet loss, since it might help us to group the irises belonging to the same subject even though they do not have similar features physically.

IV. Conclusion

This report demonstrates an effective iris recognition pipeline that integrates deep learning for segmentation, classic iris normalization via Daugman's model, and modern deep metric learning using a ResNet-based Siamese network. While the system performs well in distinguishing irises from different users, it encounters a notable challenge in differentiating between the left and right eyes of the same individual. Due to anatomical and lighting differences, left and right iris textures can appear significantly dissimilar, making it difficult for the network to associate them with the same person.

Addressing this issue would require further enhancement, such as incorporating spatial alignment techniques, eye-specific encoding, or multi-eye joint modeling. Nevertheless, the approach serves as a solid foundation for real-world iris recognition systems and offers insights into the complexities of biometric identification.

V. Work Distribution

陳勇嘉	Model Training and Validation, Pipeline Construction, Report
何家祥	Iris Segmentation, Report
陳芷寧	Rubber Sheet Normalization, Report