Iris Recognition in Deep Learning

- Implementing CNN in post-preprocessing

Yi-Yang Lin

Traditional Pipeline for Iris Recognition

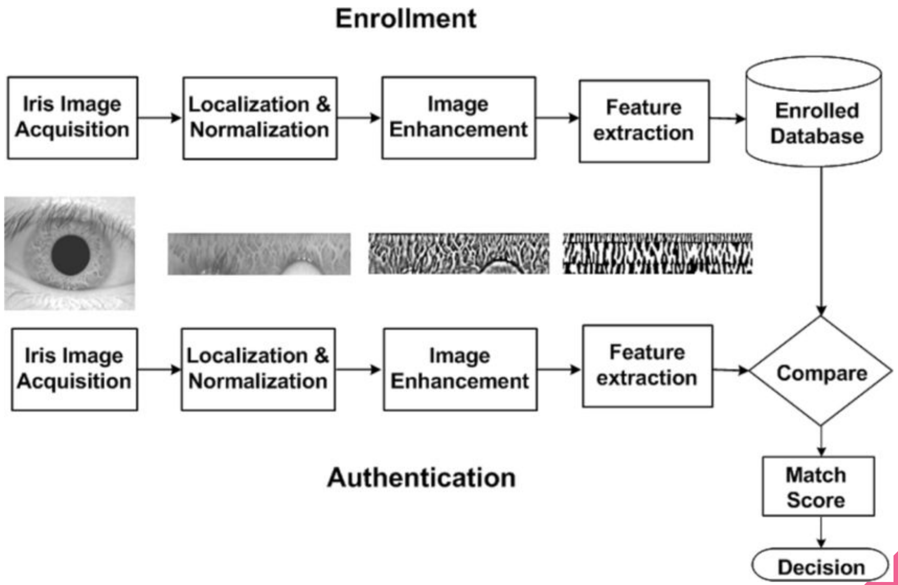
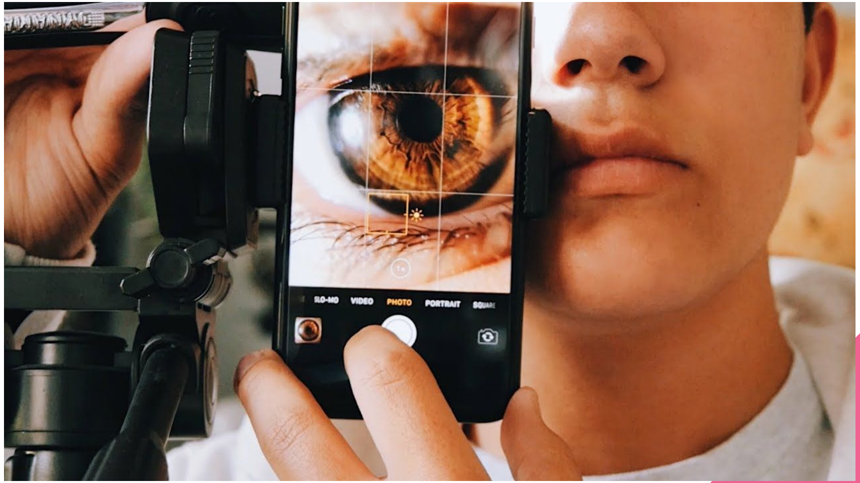
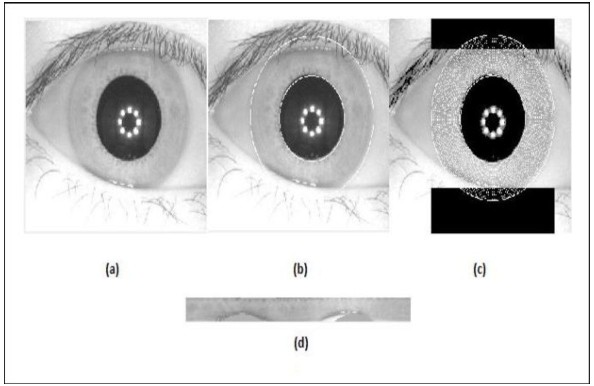


Image Acquisition

- The first step is to capture a high-quality image of the eye.
- This can be done using a specialized iris recognition camera that typically uses near-infrared light to illuminate the eye without causing discomfort.



Preprocessing



The captured image then undergoes preprocessing

to improve the quality for better feature extraction.

- **Localization**: Identifying and isolating the iris region from the rest of the eye, including the sclera (white of the eye) and eyelids.
- **Normalization**: Transforming the iris region to a fixed size and dimensions, usually using a model like Daugman's rubber sheet model, which remaps each point within the iris region to a pair of polar coordinates.

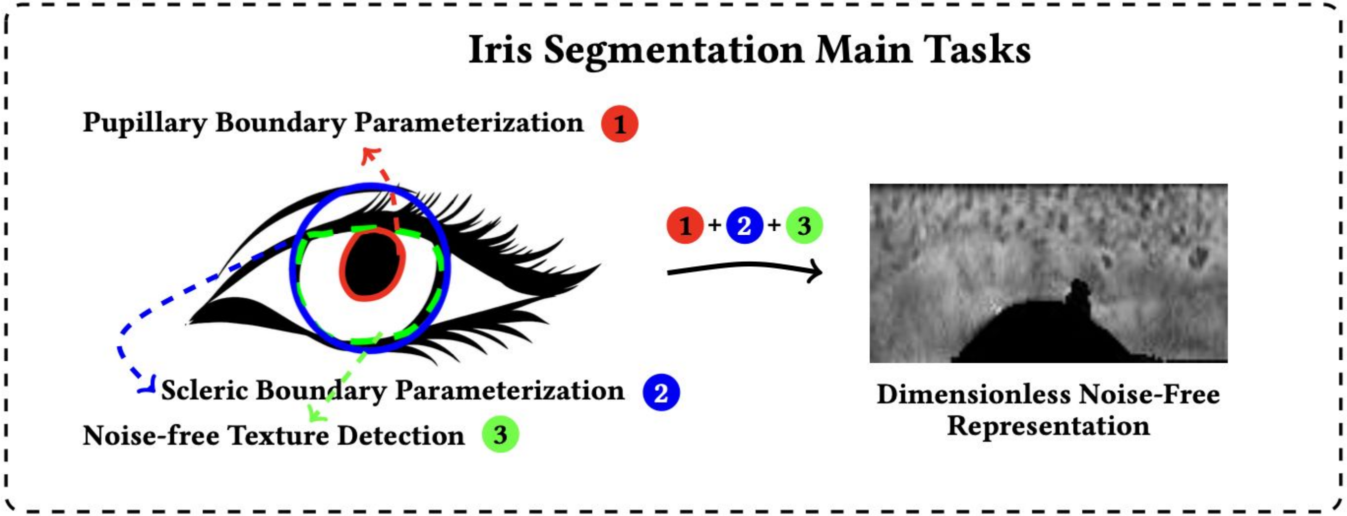
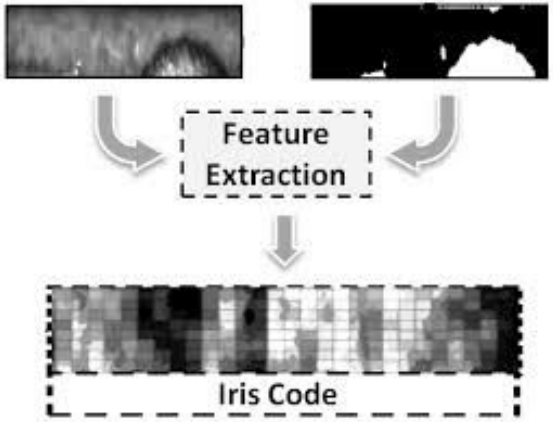


Fig. 1. Three main tasks typically associated to *iris segmentation*: 1) parameterization of the pupillary (inner) boundary; 2) parameterization of the scleric (outer) boundary; and 3) discrimination between the unoccluded (noise-free) and occluded (noisy) regions inside the iris ring. Such pieces of information are further used to obtain dimensionless polar representations of the iris texture, where feature extraction methods typically operate.

Feature Extraction

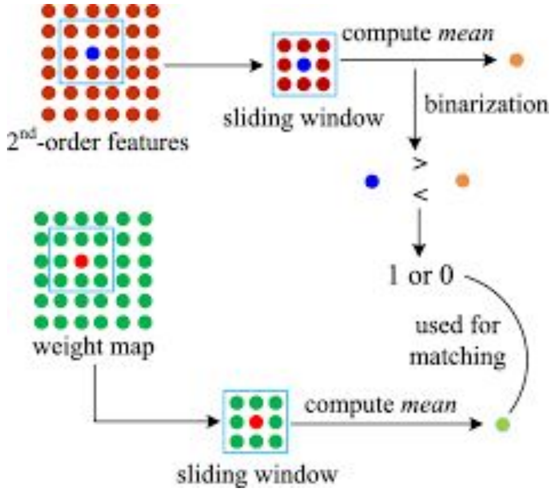


Once the iris is localized and normalized, the next step is to extract unique features from it.

This can involve several techniques:

- **Gabor Filter**: Applying Gabor filters to capture both local and spatial information about the iris pattern.
- **Wavelet Transform**: Using wavelet transforms to analyze the iris texture in different frequency bands.
- **Fourier Transform**: Analyzing the overall iris patterns based on their frequency components.

Feature Encoding



- The extracted features are then encoded into
- a compact biometric template.
- This encoding helps in effectively representing the iris pattern in a way that maximizes the differences between different irises while minimizing the variations due to lighting or other environmental factors.

Matching

- The encoded features are finally used to compare against stored templates in the database during the matching phase.
- A similarity score is computed to determine whether the presented iris matches any of the stored templates, indicating the identity of the individual.

Convolutional Neural Networks (CNNs) in Image Recognition - Basics

- Specialized deep neural networks optimized for data with a grid-like topology, such as images.
- Automatically learn important features from images, eliminating the need for manual feature extraction.
- Core Components
  - **Convolutional Layers**: Apply filters to extract features like edges and textures.
  - **Pooling Layers:** Reduce data dimensions and computational complexity.
  - **Activation Functions**: Introduce non-linear processing (e.g., ReLU).
  - **Fully Connected Layers**: Classify images based on learned features.

What can CNNs do in Iris Recognition Pipeline

- Convolutional Neural Networks (CNNs) are versatile tools that can be applied at various stages of the iris recognition pipeline to enhance performance and accuracy.

Preprocessing

- Quality Enhancement: Adjusting the contrast, brightness, or sharpness to improve image quality.
- Normalization: Transforming the iris region to a standard format, often involving corrections for pupil dilation and contraction.
- Off-axis Gaze Correction: Compensating for the angle of the gaze if the iris is not directly facing the camera, to standardize the image.

- Noise Reduction: CNNs can learn to distinguish noise from useful information, effectively denoising images.
- Quality Enhancement: Adjustments like correcting blurriness or uneven illumination can be automated using CNNs trained on a dataset of images with known good quality.
- Off-axis Gaze Correction: CNNs can estimate and correct the angle of gaze, normalizing images where the iris is not directly facing the camera.

Segmentation

- Boundary Detection: Locating the circular boundaries of both the pupil (inner boundary) and the iris (outer boundary).
- Exclusion of Obstacles: Removing parts of the image that might obscure the iris, such as eyelids, eyelashes, or reflections.
- Objective: The goal is to accurately detect the circular boundary between the pupil (inner boundary) and the iris, and the outer boundary between the iris and the sclera.
- Challenges: Variability in pupil size due to lighting changes, occlusions like eyelids and eyelashes, and reflections.
- Detecting the inner boundary: This boundary separates the pupil (the central part of the eye) from the iris.
- Detecting the outer boundary: This boundary separates the iris from the sclera (the white part of the eye).

- Boundary Detection: Training a CNN to detect the circular boundaries of the iris and the pupil. This can be more robust against variations in image quality or occlusions than traditional methods.
- Occlusion Handling: CNNs can also be trained to identify and mask out occlusions such as eyelashes, eyelids, or specular reflections, which are common issues in eye images.

Normalization

- Adjusts for variations in pupil dilation.
- Compensates for imaging inconsistencies such as tilting and rotation.
- Converts the circular iris region into a rectangular block that is easier for neural networks to analyze.

Encoding (Feature Extraction)

In the encoding phase, the segmented iris is analyzed to extract distinctive features:

- Feature Extraction Using Filters or CNNs: Applying specific techniques (like Gabor filters or Convolutional Neural Networks) to capture relevant features from the iris texture.
- Encoding Into a Template: The extracted features are converted into a digital form, typically a binary code or a vector, which serves as a unique "iris template."

CNNs excel in feature extraction due to their ability to automatically learn the most relevant features from training data:

- Feature Learning: CNNs can be trained to extract rich, descriptive features from the iris texture, which are essential for creating a unique and compact iris template.
- Direct Encoding: Some advanced CNN models can be designed to output a binary code or a feature vector directly, integrating feature extraction and encoding into a single step.

Noise Reduction

includes reducing or excluding noise and occlusions such as eyelashes, eyelids, and reflections:

- Noise identification: Identifying parts of the iris image that may distort the feature extraction due to occlusions or poor image quality.
- Noise exclusion: Techniques are applied to either correct for these distortions or to mask them out before feature extraction.

Matching (Classification)

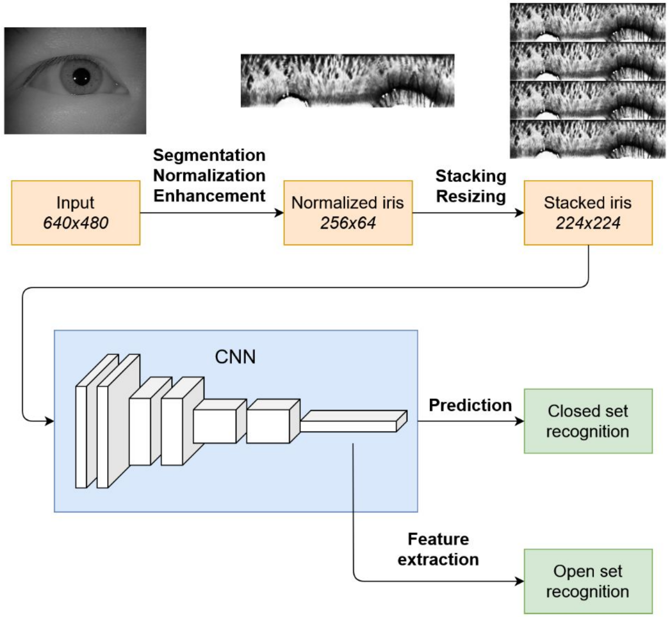
The matching phase, which can be considered a form of classification, involves comparing the encoded features extracted from a presented iris image against those stored in a database.

- Verification: Checking if the presented iris matches the specific iris template stored in the database (one-to-one comparison

- Direct Matching: CNNs can compare feature vectors or templates derived from the iris and perform classification to determine if they match known templates in the database.
- Verification and Identification: More complex CNN architectures can be designed to output probabilities indicating a match or no match, or even identify the individual among multiple classes (identities) directly.

Design a Iris Recognition Pipeline

- Inspired idea from Andrej
- Following Daugman & Libor's work
- Implementing ResNet to stretched polar images



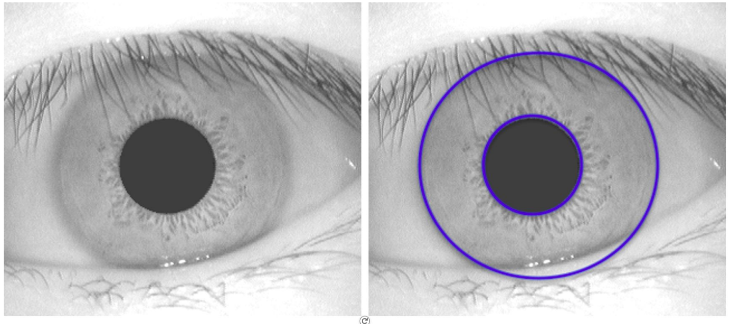
Andrej’s Iris Feature Extraction

- Use a similar approach to traditional iris recognition, first defined by Daugman. Modified method was defined by the authors of and the code was taken from their GitHub repository. Iris is first segmented from the image using weighted adaptive Hough transform and ellipsopolar transform (wahet).
- Normalize the segmented iris to size 256×64. Next, we enhance the normalized image by increasing the contrast between the lighter and darker areas. This accentuates the features of the iris that make it unique. We then **stack copies** of the enhanced image on top of each other four times, getting an image of size 256 × 256. This enables us to get a square image, which is required by the model. Finally we resize the image to the input size of 224 × 224.

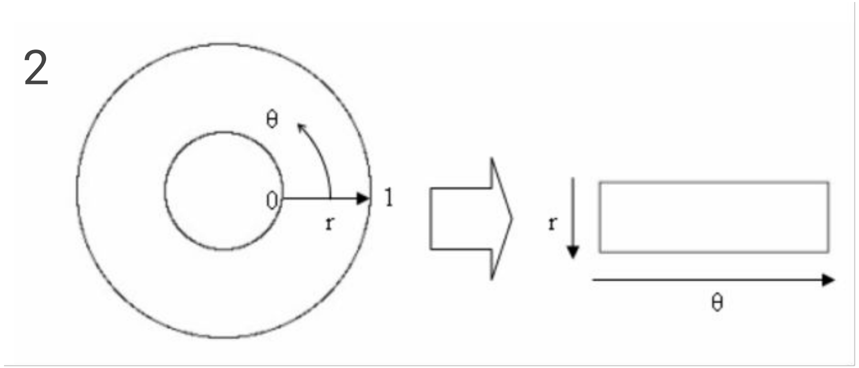
Design a Iris Recognition Pipeline

- Inspired idea from Andrej
- Following Daugman & Libor's work
- Implementing ResNet to stretched polar images

1



2



3

→

ResNet - Residual Network

ResNet revolutionized deep learning by enabling the training of much deeper neural networks through the use of residual blocks and skip connections.

Key Features

- Residual Blocks: Layers learn residual functions with reference to the layer inputs, simplifying the learning process.
- Skip Connections: Facilitate the flow of gradients throughout the network by allowing the output of one layer to feed directly into a later layer, preventing the vanishing gradient problem.
- Deep Architectures: Supports constructing very deep networks with variants like ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152.

Impact & Applications:

- Dominant in tasks requiring complex pattern recognition such as image classification, object detection, and medical image analysis.
- Achieved state-of-the-art results on major benchmarks like ImageNet.
- Significance: Pioneered a new way of building deeper neural networks,

  influencing subsequent designs in the field of artificial intelligence.

Code Demo

- ResNet in MNIST handwritten digits
- ResNet in stretched iris images

Challenges

- Primary Challenge: Data Scarcity
  - Limited data per individual hampers model accuracy and increases overfitting risks.
- Mitigation Techniques:
  - ROI Checks: Focus training on critical iris features.
  - Image Enhancement: Boost input image quality for better model training.
- Future Directions:
  - Hybrid Approaches: Integrate traditional image processing with CNNs.
  - Advanced Data Augmentation: Enrich training datasets to improve generalization.

Reference

Deep Learning for Iris Recognition: A Survey

DeepIris: Iris Recognition Using A Deep Learning Approach

Iris Feature Extraction Using Convolutional Neural Networks

Recognition of Human Iris Patterns for Biometric Identification

Towards More Accurate Iris Recognition Using Deeply Learned Spatially Corresponding Features

Tufts CS152 DS153 Sp’24 Class 18: Convolutional Neural Networks