Wavelets Course Project

Identification, Authentication & Verification of Biometrics
by

Ayush Anant (20d070019)
G Kamalesh (20d070029)

under the guidance of

Prof. V M Gadre

Department of Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai 400 076

Contents

List of Figures

1 Introduction

2 Theory

2.1 Shearlets

Shearlets provide the advantage of directionality unlike the traditional wavelets and are associated with the scaling parameter (a), shearing parameter (s), and the translation parameter (t).

The Continuous Theory
The continuous shearlets are defined by

$$\Psi_{\mathrm{a,s,t}}(\mathrm{x}) = \mathrm{a}^{-3/4}\Psi\left(\left(\mathrm{D}_{\mathrm{a,s}}^{-1}(\mathrm{x}-\mathrm{t})\right),\ \text{where } \mathrm{D}_{\mathrm{a,s}} = \left[\mathrm{a}, -\mathrm{a}^{1/2}\ \mathrm{s}; 0, \mathrm{a}^{1/2}\right]\right.$$

The mother shearlet function ψ is defined by

$$\psi\left(\xi_1, \xi_2\right) = \psi_1\left(\xi_1\right)\psi_2\left(\xi_2/\xi_1\right)$$

where ψ_1 is a wavelet and ψ_2 is a bump function.
The associated continuous shearlet transform is defined by

$$SH_f(\mathrm{a,s,t}) = \langle \mathrm{f}, \Psi_{\mathrm{a,s,t}} \rangle$$

This transform can also be regarded as matrix coefficients of the unitary representation

$$(\sigma(\mathrm{a,s,t})\psi)(\mathrm{x}) = \psi_{\mathrm{a,s,t}}(\mathrm{x}) = \mathrm{a}^{-3/4}\Psi\left(\left(\mathrm{D}_{\mathrm{a,s}^{-1}}(\mathrm{x}-\mathrm{t})\right)\right.$$

The Discrete Theory
To obtain the discrete shearlets, we sample the three parameters as

$$\begin{aligned} a_j &= 2^j\ (j \in \mathbb{Z}) \\ s_{j,k} &= k a_j^{1/2} = k 2^{j/2}\ (j \in \mathbb{Z}) \\ t_{j,k,m} &= D_{a_j, s_j, k}\left(m \in \mathbb{Z}^2\right) \end{aligned}$$

We choose the mother shearlet function ψ similarly as in the continuous case, i.e., we now choose ψ_1 to be a discrete wavelet and Ψ_2 to be a bump function with certain weak additional properties. The tiling of the frequency plane is illustrated in the figure below.

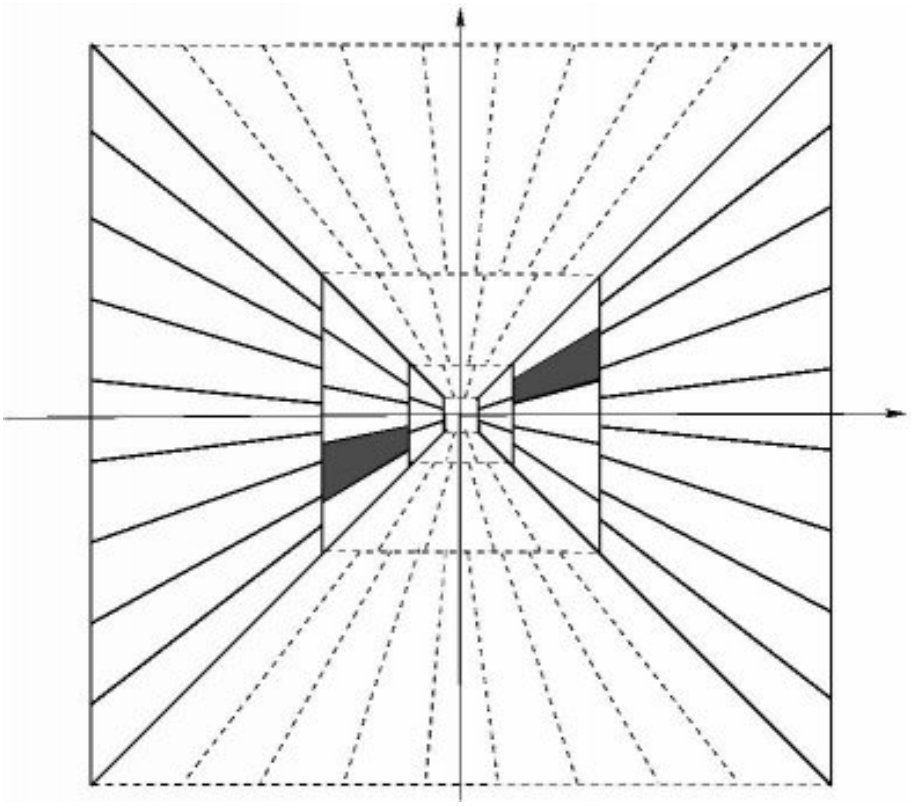

Figure 1: Frequency plane tiling

This system forms a Parseval frame for $\mathrm{L}^2(R)$, and they are optimally sparse. Furthermore, they are associated with a generalized MRA structure, where the scaling space is not only translation invariant but also invariant under the shear operator.

2.2 Scattering Network

The scattering wavelet transform computes a translation invariant representation that is stable to deformation. It cascades wavelet transform convolutions with non-linear modulus and averaging operators. Modulus is a contraction operator and provides stability but loses phase while averaging provides translation invariance and improves stability against time-warping deformations but loses high-frequency content. The second-order coefficients are obtained by convolving with a wavelet of a particular scale at different orientations thereby preserving the information lost in the first stage. The flow of energy represents the flow of information.

$$A(\omega) = |\hat{\phi}(\omega)|^2 + \sum_{\lambda \in \Lambda} \left|\hat{\psi}_\lambda(\omega)\right|^2, \forall \omega \in \mathbb{R}$$

where Λ is the set of all center frequencies. This is Littlewood-Paley summation, the sum of all filters' energies, which satisfies

$$1 - \alpha \leq A(\omega) \leq 1,$$

where $\alpha < 1$. The wavelet transform can be represented as:

$$Wx = (x \star \phi, x \star \psi_\lambda)_{t \in R, \lambda \in \Lambda}$$

Applying Parseval-Plancherel's theorem:

$$(1 - \alpha)\|x\|^2 \leq \|Wx\|^2 \leq \|x\|^2$$

If $\alpha = 0$, the filter bank is a tight frame implying input energy is conserved exactly.

This transform uses a **Morlet wavelet** and covers up the polar frequency plane with its radial dilations and angular translations as depicted in the picture below:

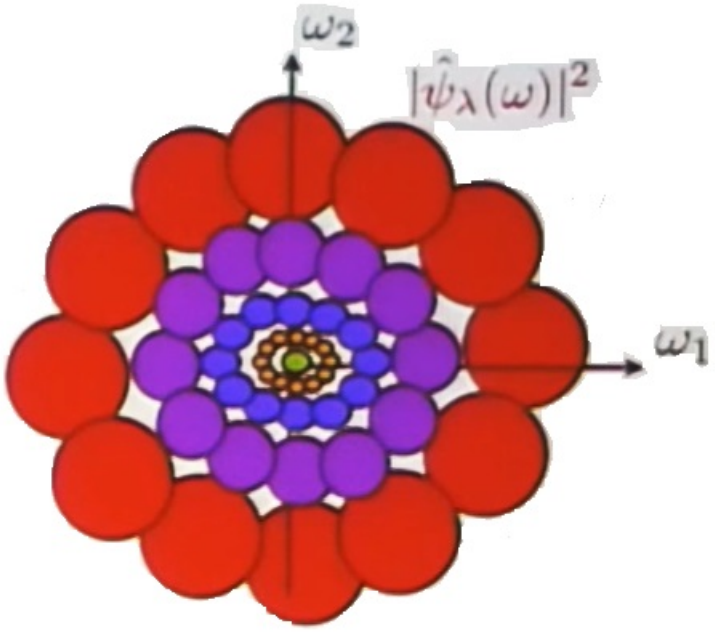

Figure 2: Frequency plane tiling

The scattering transform mainly has two parameters, namely scale(J) and the number of angles(L). Generally, the scattering coefficients die out after 2 stages as most of the energy gets absorbed in the low-frequency coefficient outputted at each stage.

The following image gives a visual understanding of what the scattering transform does when applied to an MNIST image.

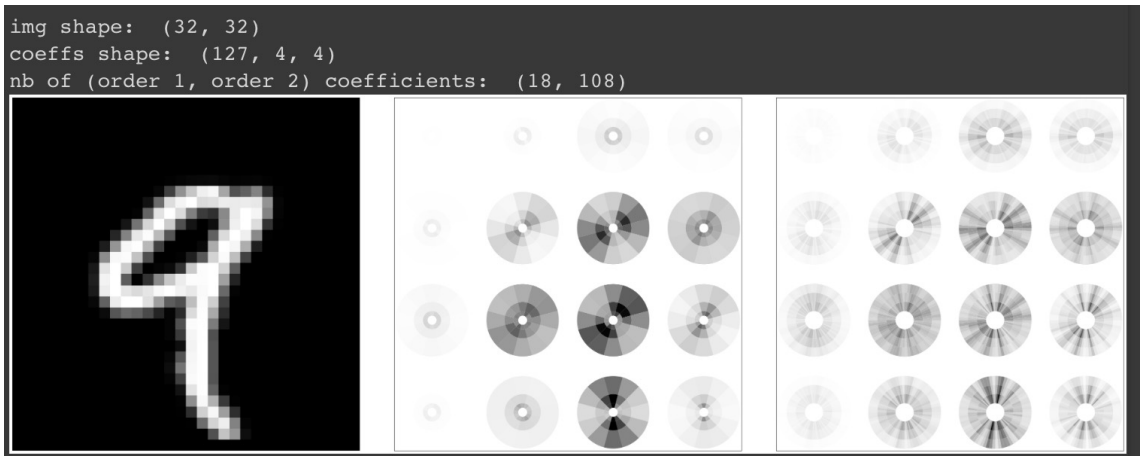

Figure 3: Circular disk visualization of Scattering Coefficients

The scattering coefficients are obtained by choosing 3 different scales and 6 angular orientations on $[0,\pi]$. Hence, each coefficient covers a pixel block of size 8×8 due to downsampling and each disc is divided into 12 parts. The shaded region denotes the value of the coefficient with black indicative of high value and white indicative of low value.

3 General Approach

The most commonly used approach for image classification is to use a Convolutional Neural Network(CNN) wherein various features are learned by the convolutional filters. However, there are a few disadvantages to this approach such as:

- number of layers to use
- interpretability of convolutional layers
- usage of pooling layers
- CNNs require large data to train

Hence, we take a wavelet-based approach that tries to answer most of these questions.

To classify an image, we need to learn features that eliminate intra-class variability while preserving interclass variability which boils down to learning rotational and translation-invariant feature representation. Once we extract such features, the goal is to learn some kind of metric to classify similar images and dissimilar images. We tried using the following metrics:

- **Cosine Similarity**: This technique converts the 2-dimensional image into a 1-dimensional vector and computes the similarity between the two vectors by calculating the cosine of the angle between them:

  $$\cos(\theta) = \frac{A \cdot B}{\|A\|_2 \|B\|_2}$$

  However, we **weren't** able to obtain a proper threshold that can distinguish between similar and dissimilar images because cosine similarity is subject to variations in the pixel intensities due to noise and struggles to discriminate between images that have subtle changes due to lack of geometric information.

- **SSIM (Structural Similarity Index**: Although, this is a better metric that can discriminate between similar and dissimilar images, finding a threshold that works for a large dataset could still be a problem.

  $$(x, y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

- **Metric Learning using Triplet Siamese Network**: This network is used to transform the extracted feature coefficients into a feature embedding space where the distance between the embeddings is a measure of similarity or dissimilarity between the corresponding input samples. The triplet siamese network is trained on triplets of samples: an anchor, a positive sample, and a negative sample where the goal is to minimize the distance between anchor and positive embedding while maximizing the distance between anchor and negative embedding. This results in a feature embedding space where similar samples are close and dissimilar samples are far apart. The problem of **identification, authentication, and verification** can be addressed as follows:

  - **Identification**: The user is assigned a unique identifier based on their biometric feature embedding that can be stored in the database when the user registers for the first time.
  - **Authentication**: The process of comparing the biometric feature embedding of the presented biometric with the stored embedding of the authorized user.

– **Verification**: The process of comparing whether a presented biometric sample matches the claimed identity. This can be done by giving a similarity score which could simply be the Euclidean distance between the feature embeddings.

4 Fingerprint

Dataset−Contactless 2D to Contact-based 2D Fingerprint Dataset
IIT Bombay Fingerprint Dataset

4.1 Preprocessing

The general idea behind the fingerprint image enhancement is to first segment the fingerprint image into ridge and valley regions. The segmented image is then used to estimate the local orientation of the ridges. Once the local orientation of the ridges is known, the ridge frequency image can be calculated. The ridge frequency image is then used to enhance the fingerprint image.

The following is a more detailed explanation of each step:

- **Ridge segmentation:** This step identifies the ridge regions of the fingerprint image. This is done by identifying image regions with dark deviation. The standard deviation measures how spread out the intensity values in a region are. Regions with a high standard deviation are more likely to be ridge regions, as ridges are typically darker than valleys.
- **Ridge orientation estimation:** This step estimates the local orientation of the ridges in the fingerprint image. This is done by computing the image gradients and then using a weighted summation of the gradients to find the principal direction. The principal direction is the direction in which the ridges are most aligned.
- **Ridge frequency estimation:** This step calculates a ridge frequency image. This image shows the number of ridges per unit length in each region of the fingerprint image. The ridge frequency image is used to enhance the fingerprint image.
- **Fingerprint image enhancement:** This step enhances the fingerprint image by increasing the intensity of the ridge regions. This is done by applying oriented Gabor filters to increase the intensity of ridge regions.

(a) Original fingerprint

(b) Pre-processed fingerprint

Figure 4: Fingerprint

4.2 Feature Extraction Techniques

We look at two feature extraction techniques:

- Shearlets
- Scattering network

4.3 Shearlets

In the context of fingerprint image enhancement, shearlets can be used to improve the detection of ridges and valleys, which are the key features used to identify fingerprints. This is done by first decomposing the image into shearlet coefficients. The following interpretations can be drawn from the shearlet coefficients:

- The coefficients detect singularities along various directions.
- They capture curved patterns better than the separable approach.
- Shearlets possess the unique ability to preserve the singularity location and direction simultaneously. Scattering transform also possesses this ability but it loses spatial precision due to its translational invariance
- **Shearlet decomposition:**

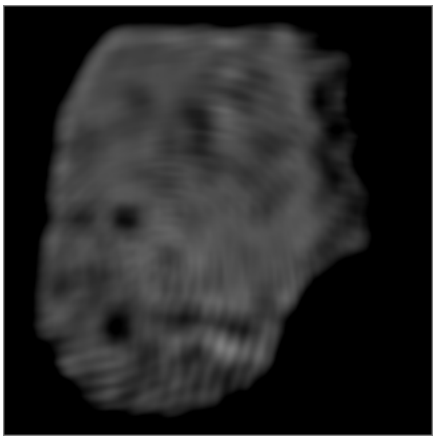

Figure 5: Lowpass Scattering Coefficient

The shearlet coefficients with 4 scales and shear levels [1 1 2 2] can be visualized as images by normalizing them in the range [0,255].

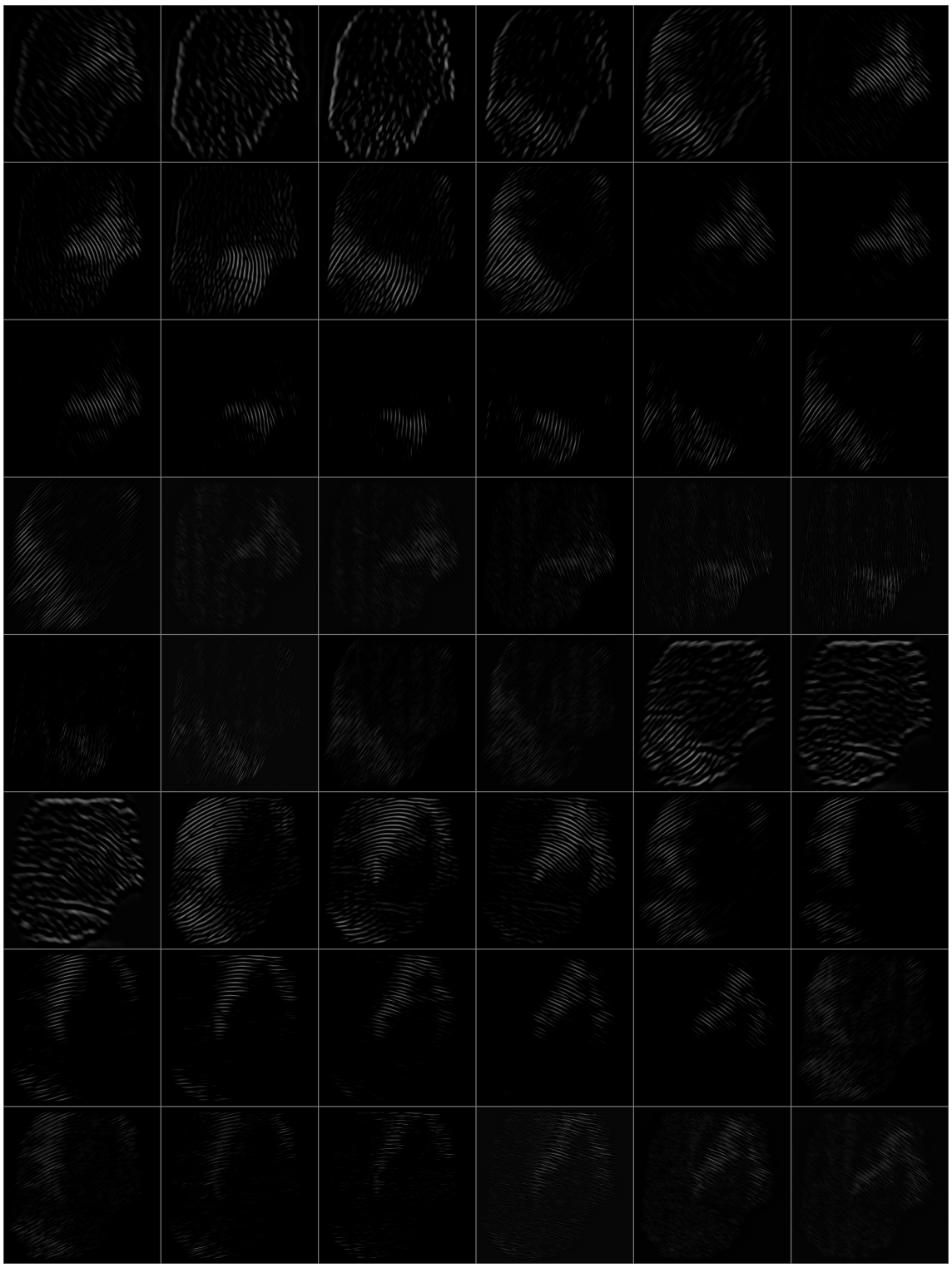

Figure 6: Shearlet Coefficient Visualised

NOTE: Please brighten the display to view the edges properly
The shearlet coefficients capture the edges and their orientation sharply.

4.4 Scattering Network

Now, we apply the scattering transform on the pre-processed fingerprint images, and the coefficients for one such fingerprint are visualized as an image attached below:

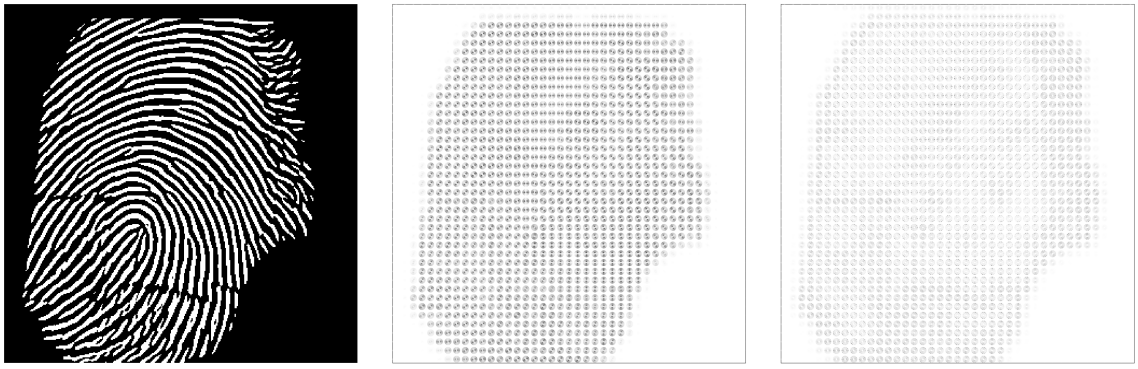

Figure 7: First and Second order Scattering coefficients of fingerprint image visualized as circular disks

On zooming, one can notice that the scattering coefficients are indicative of the edge direction. The scattering coefficients can also be visualized as images by normalizing them in the range [0,255].

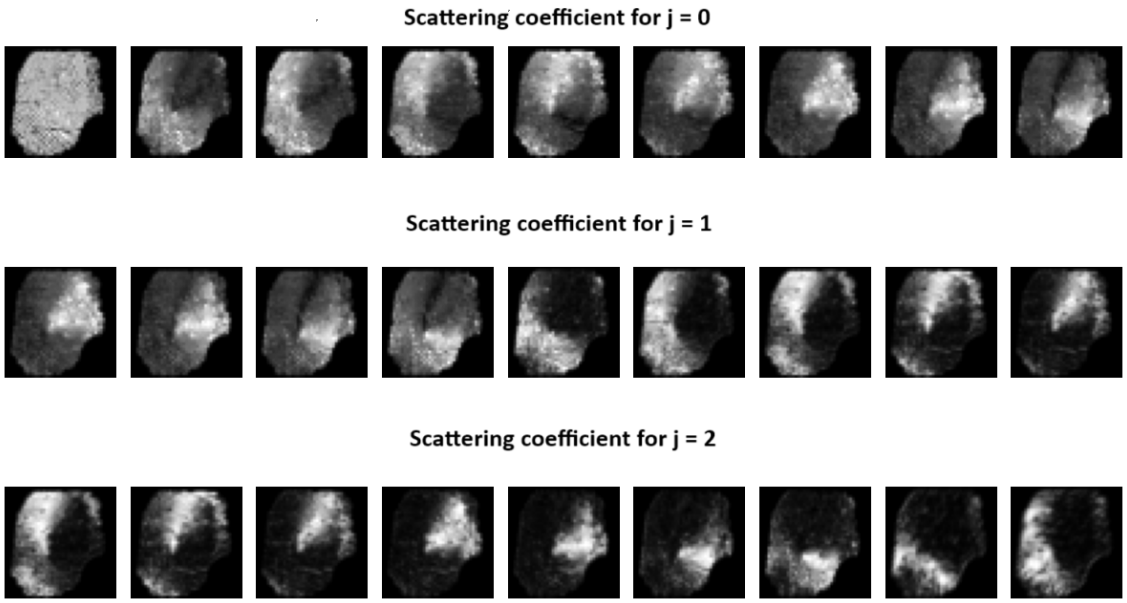

Figure 8: First order Scattering coefficients at different scales

We proceed with the scattering network for feature extraction as it provides a compact representation without losing the **interpretability** of the coefficients.

4.5 Architecture

The pre-processed image is passed through a scattering network with scale(J) = 4 and angles(L) = 6 resulting in 25 scattering coefficients including a lowpass coefficient. The lowpass coefficient is eliminated as it does not capture the interclass variability and the remaining 24 coefficients are passed through a couple of CNN

layers which further improve translational invariance and capture hierarchical features. This is followed by the addition of fully connected layers to move to a lower dimensional space which can be used as an embedding vector that serves as a compact representation(identifier) of the input image. The architecture of the triplet siamese model is attached below:

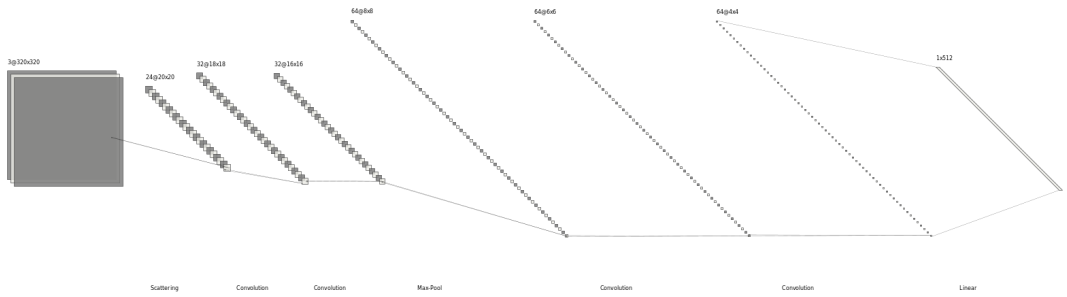

Figure 9: Triplet Siamese Scattering network

4.6 Results

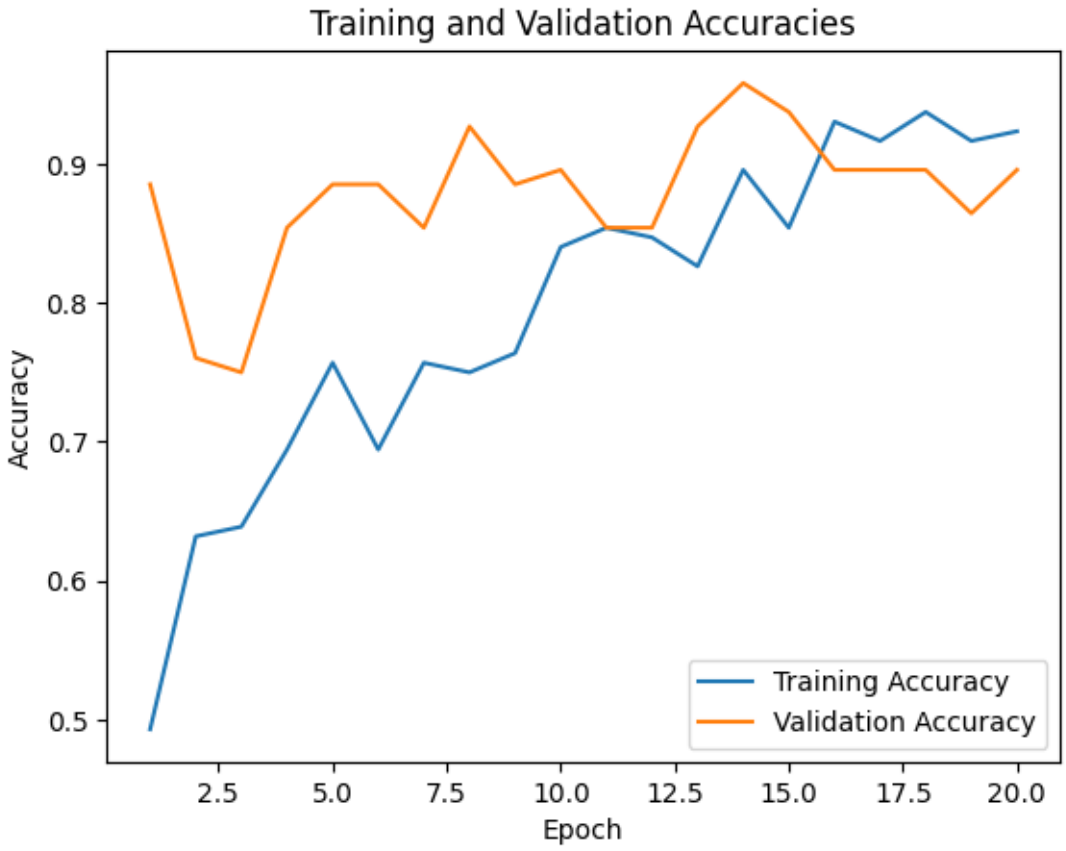

Figure 10: Training accuracy vs epochs

The training accuracy is calculated by

$$L(a, p, n) = \max\{d(a_i, p_i) - d(a_i, n_i) + margin, 0\}$$

$$\mathrm{d}(\mathrm{x}_i, y_i) = ||x_i - y_i||_2$$

where a_i, p_i, and n_i represent the scattering coefficients of anchor, positive and negative image respectively.

Figure 11: Training Loss vs epochs

The triplet margin loss function is used for computing the training loss. It penalizes the model when the difference between the distance of the (positive, anchor) and (negative, anchor) pair falls below a certain margin. The margin can be chosen to control the **robustness** of the model.

Figure 12: Distance annotated between (positive, anchor) and (anchor, negative)

4.7 Comparison with Convolutional Triplet Siamese network

The **computational cost** has been reduced greatly by using the scattering network as a feature extractor instead of using learnable filters that involve a lot of convolutional layers and hence, more parameters.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 316, 316]	832
ReLU-2	[-1, 32, 316, 316]	0
MaxPool2d-3	[-1, 32, 158, 158]	0
Conv2d-4	[-1, 64, 156, 156]	18,496
ReLU-5	[-1, 64, 156, 156]	0
MaxPool2d-6	[-1, 64, 78, 78]	0
Conv2d-7	[-1, 64, 76, 76]	36,928
ReLU-8	[-1, 64, 76, 76]	0
MaxPool2d-9	[-1, 64, 38, 38]	0
Conv2d-10	[-1, 128, 36, 36]	73,856
ReLU-11	[-1, 128, 36, 36]	0
MaxPool2d-12	[-1, 128, 18, 18]	0
Conv2d-13	[-1, 128, 16, 16]	147,584
ReLU-14	[-1, 128, 16, 16]	0
MaxPool2d-15	[-1, 128, 8, 8]	0
Flatten-16	[-1, 8192]	0
Linear-17	[-1, 512]	4,194,816
Linear-18	[-1, 256]	131,328
Linear-19	[-1, 128]	32,896
Conv2d-20	[-1, 32, 316, 316]	832
ReLU-21	[-1, 32, 316, 316]	0
MaxPool2d-22	[-1, 32, 158, 158]	0
Conv2d-23	[-1, 64, 156, 156]	18,496
ReLU-24	[-1, 64, 156, 156]	0
MaxPool2d-25	[-1, 64, 78, 78]	0
Conv2d-26	[-1, 64, 76, 76]	36,928
ReLU-27	[-1, 64, 76, 76]	0
MaxPool2d-28	[-1, 64, 38, 38]	0
Conv2d-29	[-1, 128, 36, 36]	73,856
ReLU-30	[-1, 128, 36, 36]	0
MaxPool2d-31	[-1, 128, 18, 18]	0
Conv2d-32	[-1, 128, 16, 16]	147,584
ReLU-33	[-1, 128, 16, 16]	0
MaxPool2d-34	[-1, 128, 8, 8]	0
Flatten-35	[-1, 8192]	0
Linear-36	[-1, 512]	4,194,816
Linear-37	[-1, 256]	131,328
Linear-38	[-1, 128]	32,896
Conv2d-39	[-1, 32, 316, 316]	832
ReLU-40	[-1, 32, 316, 316]	0
MaxPool2d-41	[-1, 32, 158, 158]	0
Conv2d-42	[-1, 64, 156, 156]	18,496
ReLU-43	[-1, 64, 156, 156]	0
MaxPool2d-44	[-1, 64, 78, 78]	0
Conv2d-45	[-1, 64, 76, 76]	36,928
ReLU-46	[-1, 64, 76, 76]	0
MaxPool2d-47	[-1, 64, 38, 38]	0
Conv2d-48	[-1, 128, 36, 36]	73,856
ReLU-49	[-1, 128, 36, 36]	0
MaxPool2d-50	[-1, 128, 18, 18]	0
Conv2d-51	[-1, 128, 16, 16]	147,584
ReLU-52	[-1, 128, 16, 16]	0
MaxPool2d-53	[-1, 128, 8, 8]	0
Flatten-54	[-1, 8192]	0
Linear-55	[-1, 512]	4,194,816
Linear-56	[-1, 256]	131,328
Linear-57	[-1, 128]	32,896

Total params: 13,910,208
Trainable params: 13,910,208
Non-trainable params: 0

Input size (MB): 4096000000.00
Forward/backward pass size (MB): 274.24
Params size (MB): 53.06
Estimated Total Size (MB): 4096000327.31

(a) Convolutional Siamese

Layer (type)	Output Shape	Param #
BatchNorm2d-1	[-1, 24, 20, 20]	48
Conv2d-2	[-1, 32, 18, 18]	6,944
BatchNorm2d-3	[-1, 32, 18, 18]	64
Conv2d-4	[-1, 32, 16, 16]	9,248
BatchNorm2d-5	[-1, 32, 16, 16]	64
MaxPool2d-6	[-1, 32, 8, 8]	0
BatchNorm2d-7	[-1, 32, 8, 8]	64
Conv2d-8	[-1, 64, 6, 6]	18,496
BatchNorm2d-9	[-1, 64, 6, 6]	128
Conv2d-10	[-1, 64, 4, 4]	36,928
BatchNorm2d-11	[-1, 64, 4, 4]	128
Linear-12	[-1, 512]	524,800
EmbeddingModel-13	[-1, 512]	0
BatchNorm2d-14	[-1, 24, 20, 20]	48
Conv2d-15	[-1, 32, 18, 18]	6,944
BatchNorm2d-16	[-1, 32, 18, 18]	64
Conv2d-17	[-1, 32, 16, 16]	9,248
BatchNorm2d-18	[-1, 32, 16, 16]	64
MaxPool2d-19	[-1, 32, 8, 8]	0
BatchNorm2d-20	[-1, 32, 8, 8]	64
Conv2d-21	[-1, 64, 6, 6]	18,496
BatchNorm2d-22	[-1, 64, 6, 6]	128
Conv2d-23	[-1, 64, 4, 4]	36,928
BatchNorm2d-24	[-1, 64, 4, 4]	128
Linear-25	[-1, 512]	524,800
EmbeddingModel-26	[-1, 512]	0
BatchNorm2d-27	[-1, 24, 20, 20]	48
Conv2d-28	[-1, 32, 18, 18]	6,944
BatchNorm2d-29	[-1, 32, 18, 18]	64
Conv2d-30	[-1, 32, 16, 16]	9,248
BatchNorm2d-31	[-1, 32, 16, 16]	64
MaxPool2d-32	[-1, 32, 8, 8]	0
BatchNorm2d-33	[-1, 32, 8, 8]	64
Conv2d-34	[-1, 64, 6, 6]	18,496
BatchNorm2d-35	[-1, 64, 6, 6]	128
Conv2d-36	[-1, 64, 4, 4]	36,928
BatchNorm2d-37	[-1, 64, 4, 4]	128
Linear-38	[-1, 512]	524,800
EmbeddingModel-39	[-1, 512]	0

Total params: 1,790,736
Trainable params: 1,790,736
Non-trainable params: 0

Input size (MB): 3375000.00
Forward/backward pass size (MB): 1.34
Params size (MB): 6.83
Estimated Total Size (MB): 3375008.17

(b) Scattering Siamese

Figure 13: Model Summary

The model summary shows that the convolutional siamese network consumes **54MB** for parameters while the scattering siamese network consumes only **1.34MB** for parameters.

5 Iris

5.1 Wavelet based approach

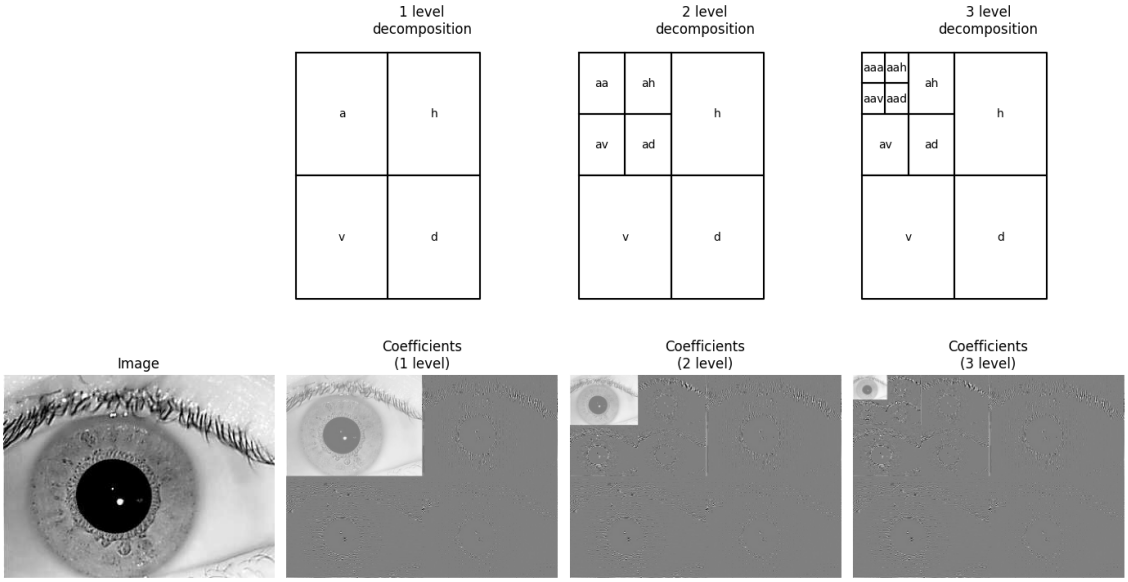

Figure 14: Iris Wavelet Decomposition

We applied cosine similarity on the wavelet decomposition coefficients but the margin is too small to distinguish between the similar (~ 0.99) and dissimilar pairs (~ 0.98).

5.2 Wavelet based approach

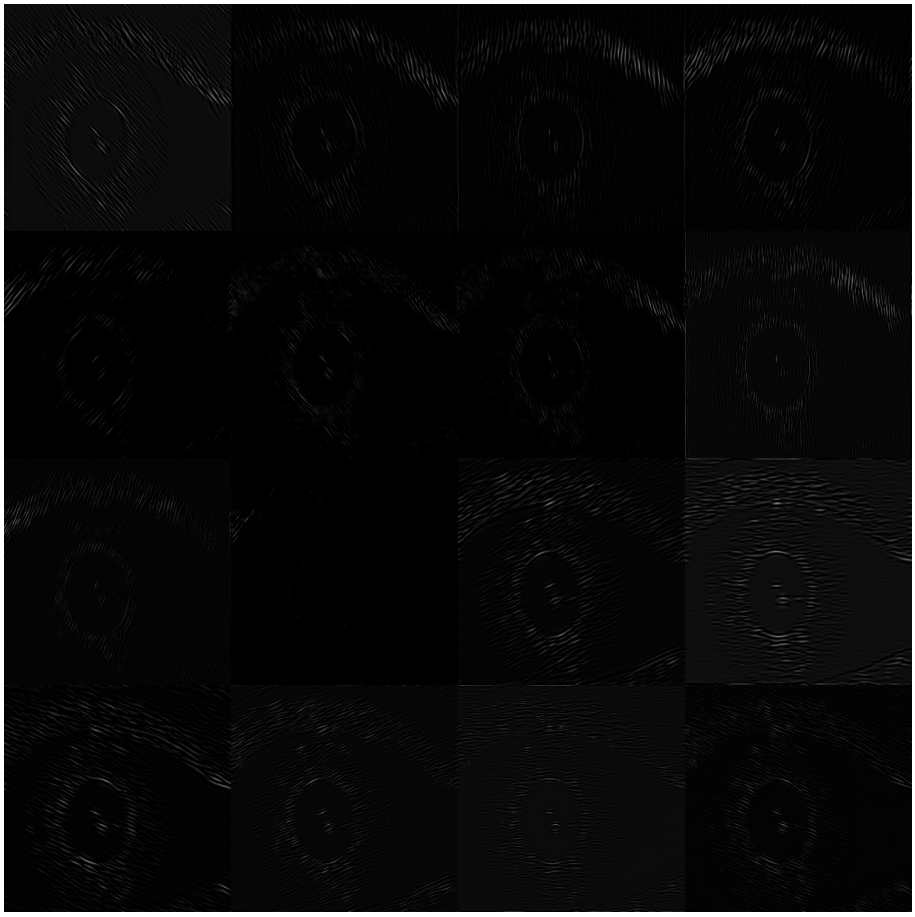

Figure 15: Iris Shearlet Coefficients Visualized

5.3 Scattering Wavelet Transform

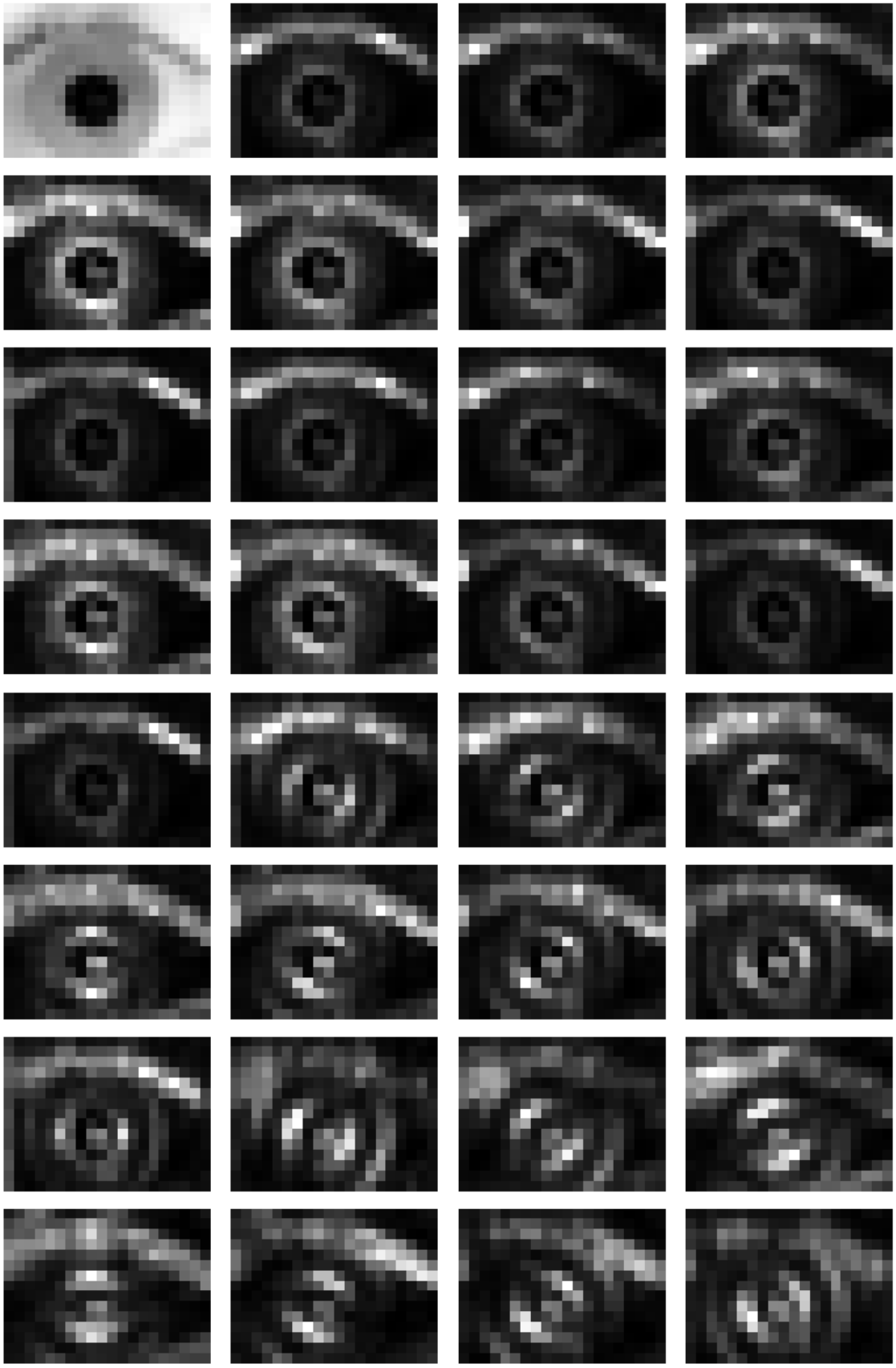

Figure 16: Iris Scattering Coefficients

5.4 Feature Extraction Techniques

We look at two feature extraction techniques:

- Haar
- Scattering network

5.5 Haar

The Haar wavelet's mother wavelet function ψ(t) can be described as

$$\psi(t) = \begin{cases} 1 & 0 \leq t \leq \frac{1}{2} \\ -1 & \frac{1}{2} \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Its scaling function ϕ (t) can be described as

$$\psi(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Given a signal or an image, the Haar wavelet decomposition involves the following steps:

- Approximation Coefficients (LL): Obtained by low-pass filtering (convolution with the scaling function).
- Detail Coefficients (HL, HH, LH): Obtained by high-pass filtering (convolution with the wavelet function).
- This process is repeated iteratively to obtain multiple levels of approximation and detail coefficients.

5.6 Architecture

```
----------------------------------------------------------------
       Layer (type)               Output Shape         Param #
================================================================
      BatchNorm2d-1          [-1, 32, 15, 20]              64
           Conv2d-2          [-1, 64, 13, 18]          18,496
      BatchNorm2d-3          [-1, 64, 13, 18]             128
           Conv2d-4          [-1, 64, 11, 16]          36,928
      BatchNorm2d-5          [-1, 64, 11, 16]             128
        MaxPool2d-6            [-1, 64, 5, 8]               0
           Linear-7                [-1, 1028]       2,632,708
           Linear-8                 [-1, 512]         526,848
   EmbeddingModel-9                 [-1, 512]               0
     BatchNorm2d-10          [-1, 32, 15, 20]              64
          Conv2d-11          [-1, 64, 13, 18]          18,496
     BatchNorm2d-12          [-1, 64, 13, 18]             128
          Conv2d-13          [-1, 64, 11, 16]          36,928
     BatchNorm2d-14          [-1, 64, 11, 16]             128
       MaxPool2d-15            [-1, 64, 5, 8]               0
          Linear-16                [-1, 1028]       2,632,708
          Linear-17                 [-1, 512]         526,848
   EmbeddingModel-18                 [-1, 512]               0
     BatchNorm2d-19          [-1, 32, 15, 20]              64
          Conv2d-20          [-1, 64, 13, 18]          18,496
     BatchNorm2d-21          [-1, 64, 13, 18]             128
          Conv2d-22          [-1, 64, 11, 16]          36,928
     BatchNorm2d-23          [-1, 64, 11, 16]             128
       MaxPool2d-24            [-1, 64, 5, 8]               0
          Linear-25                [-1, 1028]       2,632,708
          Linear-26                 [-1, 512]         526,848
   EmbeddingModel-27                 [-1, 512]               0
================================================================
Total params: 9,645,900
Trainable params: 9,645,900
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 3375000.00
Forward/backward pass size (MB): 1.53
Params size (MB): 36.80
Estimated Total Size (MB): 3375038.32
----------------------------------------------------------------
```

Figure 17: Iris Scattering Siamese Model Summary

5.7 Results

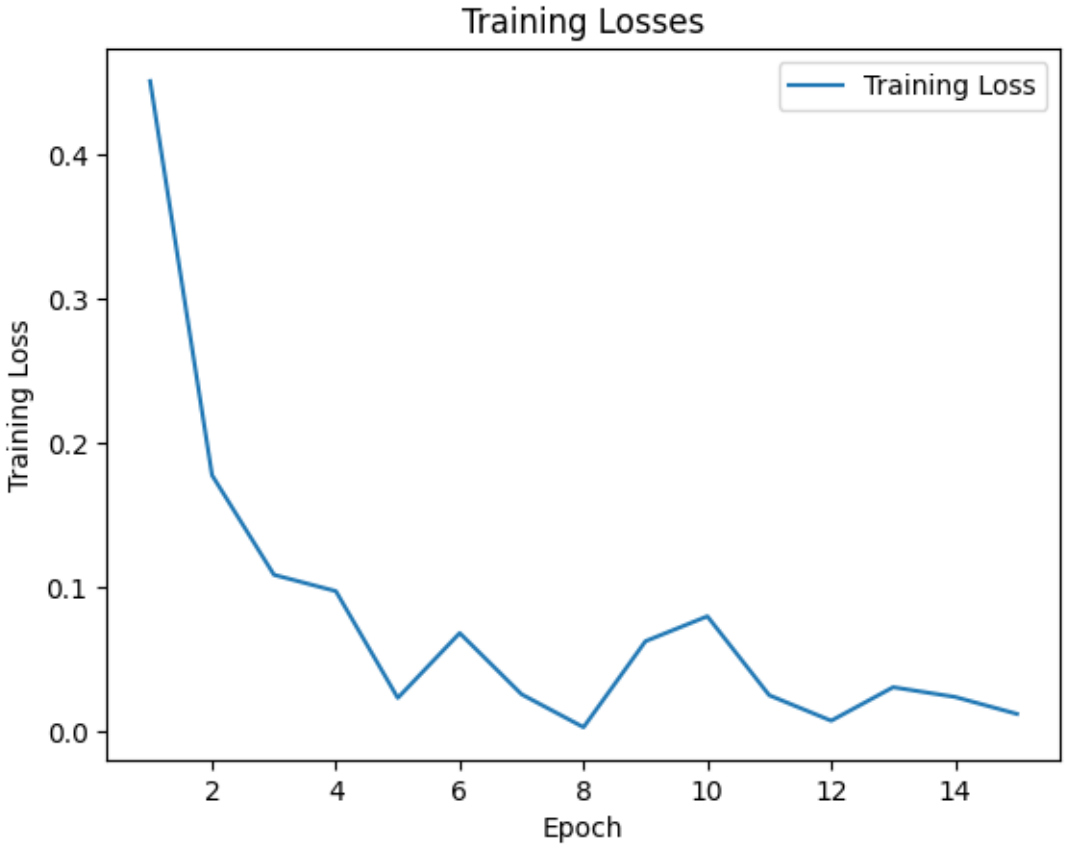

Figure 18: Iris Training loss

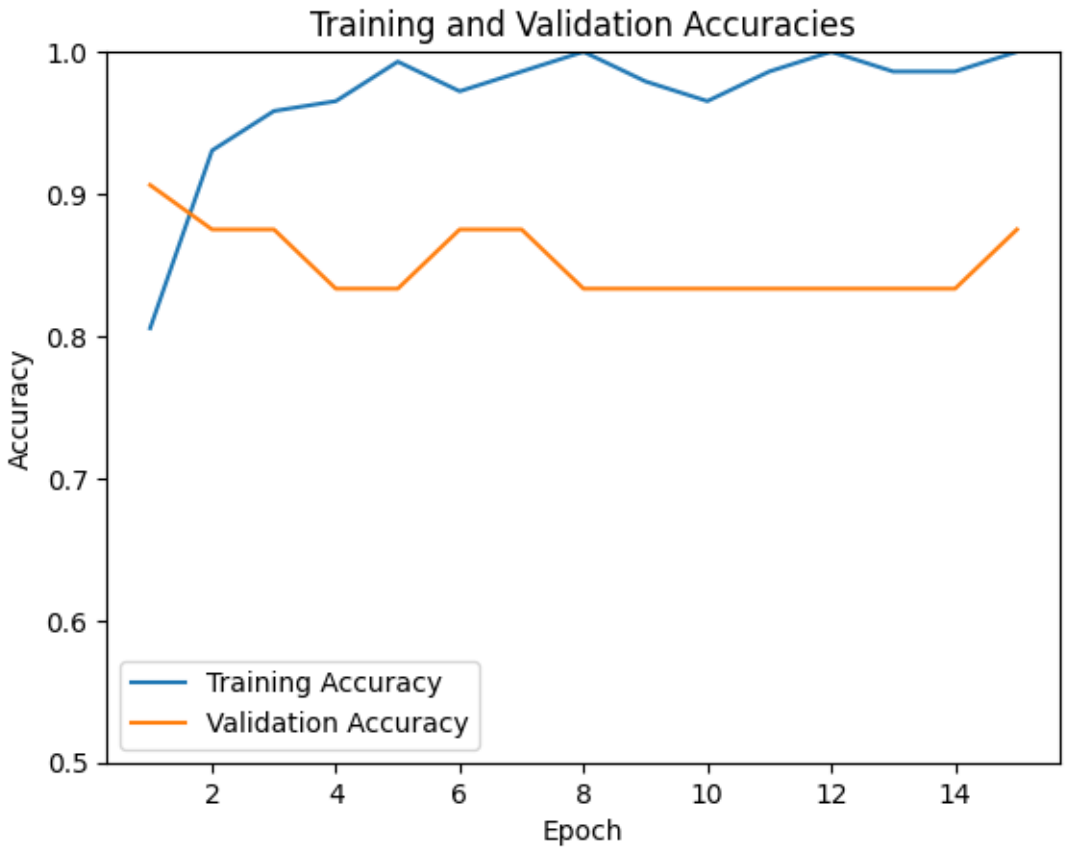

Figure 19

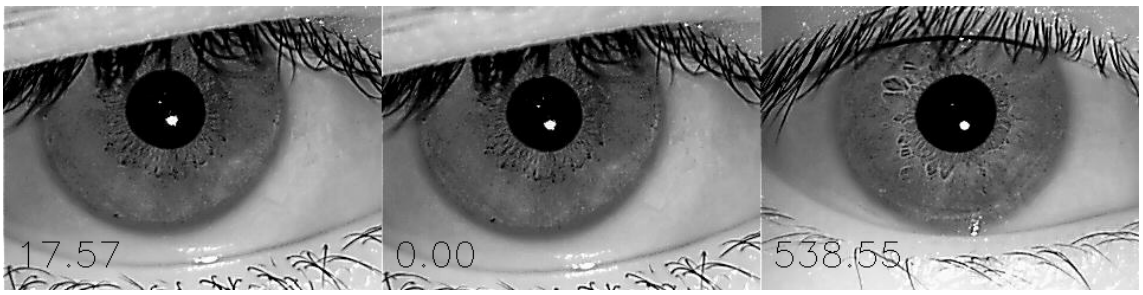

Figure 20: Distance annotated between (positive, anchor) and (anchor, negative)

6 Ear

6.1 Feature Extraction Techniques

We look at two feature extraction techniques:

- Haar
- Scattering network

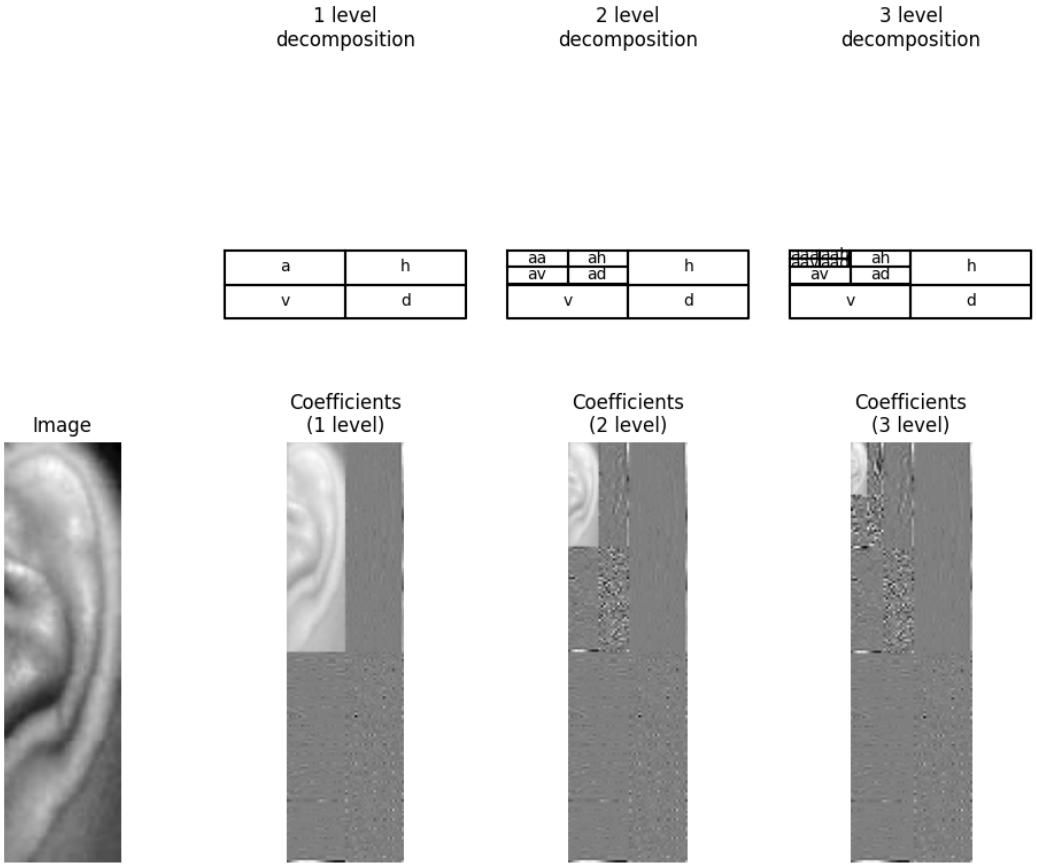

Figure 21: Ear Wavelet Decomposition

6.2 Results

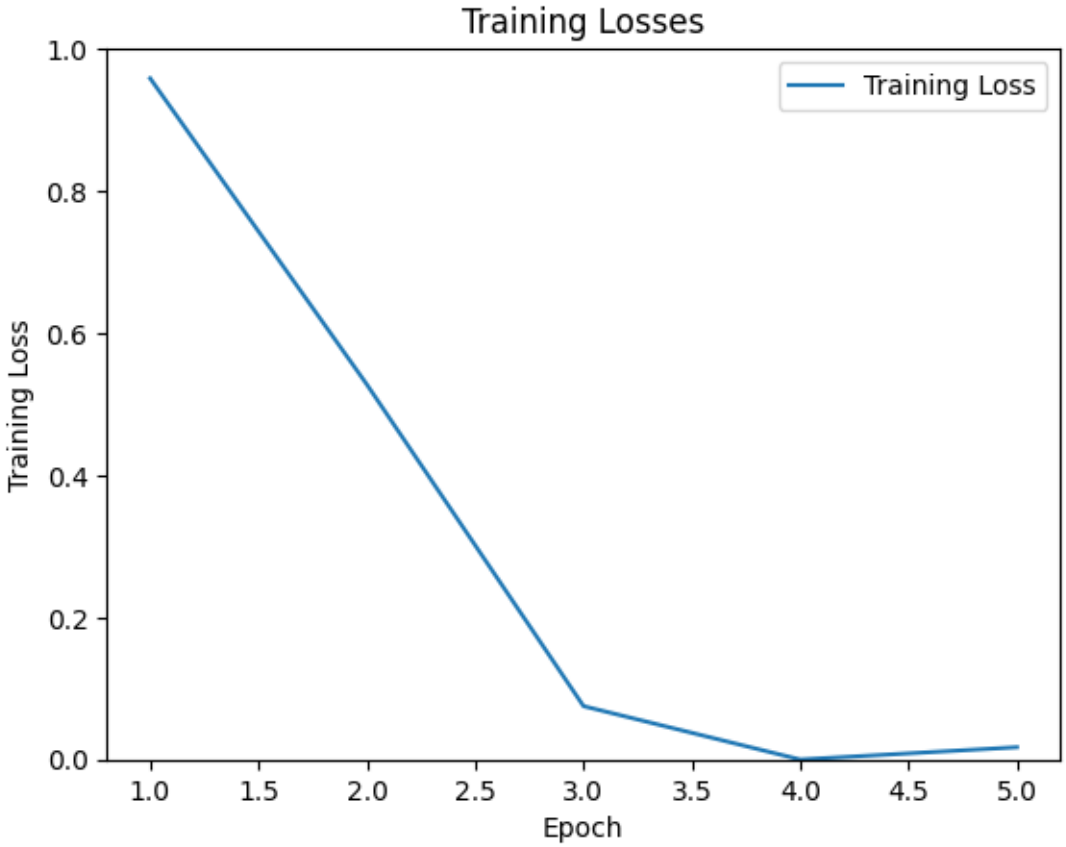

Figure 22: Ear Training Loss

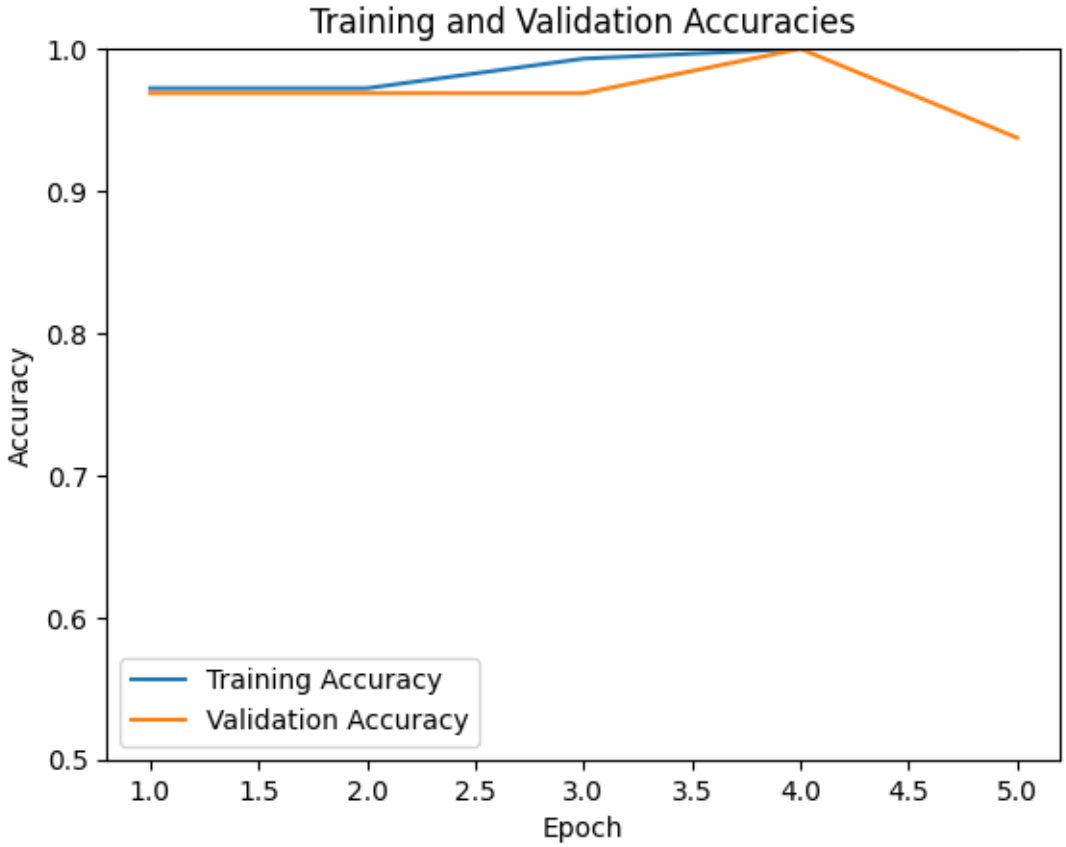

Figure 23: Ear Training and Validation accuracies

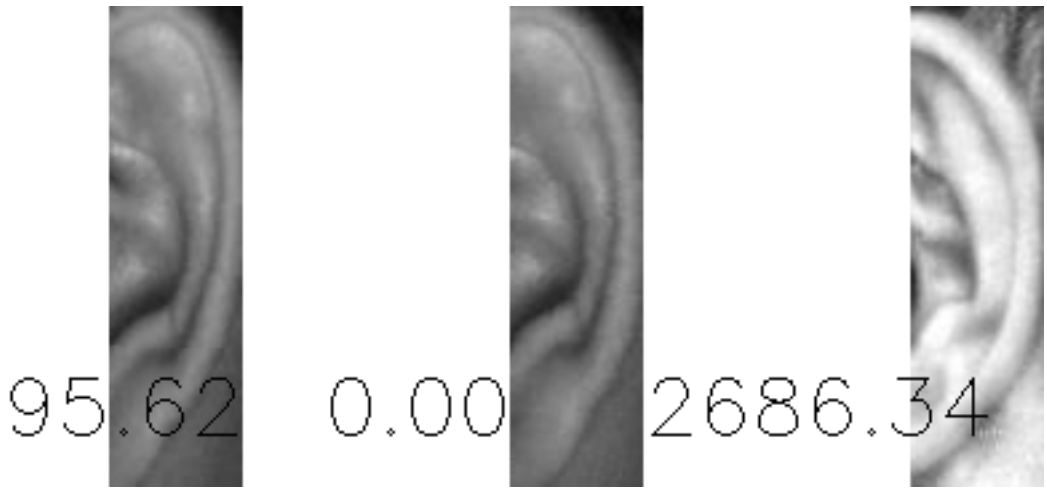

Figure 24: Distance annotated between (positive, anchor) and (anchor, negative)

We face many unexpected issues in different fields: o The number of samples is limited to samples per individual; 3-to 4 samples for learning and 2 samples for testing; which are not enough for machine learning. o The bad quality of the fingerprint images database, which contains all types of challenges. o The high displacements of fingerprint images, in which 80our selected database is out of image borders, it feels like we work in partial fingerprint images. o The high and different image rotations, in which the rotation angle ranges from 0-360 degrees

7 Interpretations

- The features extracted are easy to interpret and provide rotational and translational invariance.
- In the training and validation curves, one can observe that the model starts training at a very high accuracy. This can be attributed to the use of wavelet-based feature extraction techniques.
- The model converges faster in comparison to the convolutional Siamese network and also trains faster due to fewer parameters.

References

[1] B. Fang, H. Wen, R. -Z. Liu and Y. -Y. Tang, "A New Fingerprint Thinning Algorithm," 2010 Chinese Conference on Pattern Recognition (CCPR), Chongqing, China, 2010, pp. 1-4, doi: 10.1109/CCPR.2010.5659273.

[2] Shin Fujieda, Kohei Takayama, Toshiya Hachisuka. "Wavelet Convolutional Neural Networks for Texture Classification" arXiv:1707.07394

[3] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pp. 674-693, July 1989, doi: 10.1109/34.192463.