EE 678:Wavelets

Course Project: Final Report
Applications of Wavelets and Deep Learning in Fingerprints Matching

Badal Varshney (19D070015)
Bhavishya (19D070017)

November 14, 2022

Faculty Mentor
Prof. Vikram M Gadre
Department of Electrical Engineering
Indian Institute of Technology Bombay

Contents

1 Abstract

Scattering Transforms (or ScatterNets) introduced by Mallat are a promising start into creating a well-defined feature extractor to use for pattern recognition and image classification tasks. They are of particular interest due to their architectural similarity to Neural Networks, while requiring no parameter learning and still performing very well (particularly in constrained classification tasks). We are developing a technique to use these Scattering Networks for Biometric FingerPrints for the identification or verification of a person in Army Camps.

2 Introduction

Scattering transforms, or ScatterNets, have recently gained much attention and use due to their ability to extract generic and descriptive features in a well-defined way. They can be used as unsupervised feature extractors for image classification and texture classification or in combination with supervised methods such as Convolutional Neural Networks (CNNs) to make the latter learn quicker and in a more stable way.
Fingerprints are one of the most important biometric characteristics due to their known uniqueness and persistence properties. The idea is to use touchless fingerprint recognition to solve the problem by extracting the features from fingerprint images and comparing them with the model, which uses signal processing, multiresolution, and machine learning-based techniques. The verification is done by matching with another image of the same person by setting a high threshold value. Identification is done by matching the query image with all the images in the database and giving the top five best matches it.

3 The Scattering Transform

The Scattering Transform, or ScatterNet, is a cascade of complex wavelet transforms and modulus non-linearities (throwing away the phase of the complex wavelet coefficients). At a chosen scale, averaging filters provide invariance to nuisance variations such as shift and deformation (and potentially rotations). Due to the non-expansive nature of the wavelet transform and the modulus operation, this transform is stable to deformations.
Typical implementations of the ScatterNet are limited to two 'orders. In addition to scattering order, we also have the scale of invariance, J. This is the number of band-pass coefficients output from a wavelet filter bank (FB), and defines the cut-off frequency for the final low-pass output: 2^{-J}*(fs/2) (fs is the sampling frequency of the signal). Finally, we call the number of oriented wavelet coefficients used L. These are the three main hyperparameters of the scattering transform and must be set ahead of time. We describe a system with scale parameter J = 4, order m = 2, and with L = 6 orientations (L is fixed to 6 for the DTCWT but is flexible for the FFT-based Morlet wavelets).
Consider an input signal x(u), u ϵ R^2. The zeroth order **scatter** coefficient is the lowpass output of a J level FB:

$$S_0 x(u) = (x * \phi_J)(u) \tag{1}$$

This is invariant to translations of up to 2^J pixels. In exchange for gaining invariance, the S_0 coefficients have lost a lot of information (contained in the rest of the frequency space). The

remaining energy of x is contained within the first order wavelet coefficients:

$$W_1 x(u, j_1, \theta_1) = x * \psi_{j_1, \theta_1} \tag{2}$$

for $j_1 \epsilon$ (1, 2,...., J), $\theta_1 \epsilon$ (1, 2,.....,L). We will want to retain this information in these coefficients to build a useful classifier.
Let us call the set of available scales and orientations $\Lambda 1$ and use $\lambda 1$ to index it. For both Morlet and DTCWT implementations, ψ is complex-valued, i.e., $\psi = \psi^r + \mathrm{j}\ \psi^i$ with ψ^r and ψ^i forming a Hilbert Pair, resulting in an analytic ψ. This analyticity provides a source of invariance — small input shifts in x result in a phase rotation (but little magnitude change) of the complex wavelet coefficients.
Taking the magnitude of W1 gives us the first order propagated signals:

$$U_1 x(u, \lambda_1) = |x * \psi_\lambda| = \sqrt{(x * \psi^r_{\lambda_1})^2 + (x * \psi^i_{\lambda 1})^2)} \tag{3}$$

The first order scattering coefficient makes U_1 invariant up to our scale J by averaging it:

$$S_1 x(u, \lambda_1) = |x * \psi_\lambda| * \phi_J \tag{4}$$

If we define $U_0 = \mathrm{x}$, then we can iteratively define:

$$W_m = U_{m-1} * \psi_{\lambda m} \tag{5}$$

$$U_m = |W_m| \tag{6}$$

$$S_m = U_m * \phi_J \tag{7}$$

We repeat this for higher orders, for natural images, we get diminishing returns after m = 2. The output of our ScatterNet is then:

$$Sx = (S_0 x, S_1 x, S_2 x) \tag{8}$$

4 Discrete Wavelet Transform (DWT)

First, we apply a one-level, one dimensional DWT along the rows of the image. Second, we apply a one-level, one-dimensional DWT along the columns of the transformed image from the first step. As depicted in Figure 1, the result of these two sets of operations is a transformed image with four distinct bands: (a) LL, (b) LH, (c) HL and (d) HH. Here, L stands for low-pass filtering, and H stands for high-pass filtering. The LL band corresponds roughly to a down-sampled (by a factor of two) version of the original image. The LH band tends to preserve localized horizontal features, while the HL band tends to preserve localized vertical features in the original image. Finally, the HH band tends to isolate localized high-frequency point features in the image.

As in the one-dimensional case, we do not necessarily want to stop there, since the one-level, two-dimensional DWT extracts only the highest frequencies in the image. Additional levels of decomposition can extract lower frequency features in the image; these additional levels are applied only to the LL band of the transformed image at the previous level. Figure 2 for example illustrates the multi-level, two-dimensional DWT on a sample image.

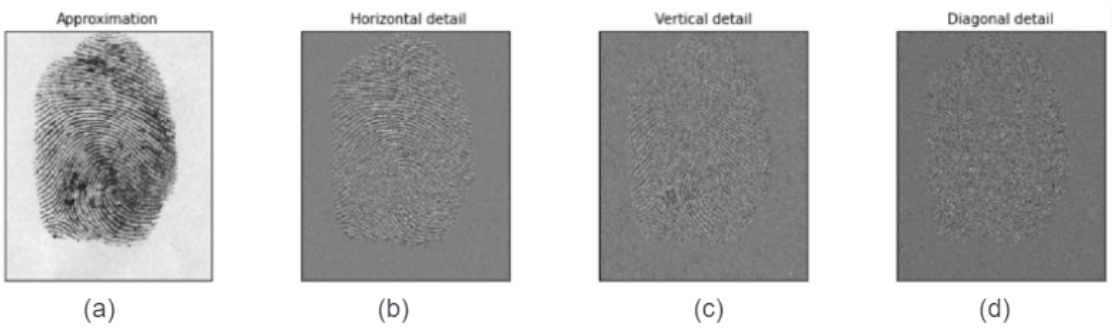


Figure 1: 2D DWT of a Fingerprint Images

This decomposition is implemented for n-dimensional data by wavedecn() and the inverse by waverecn(). The top row indicates the coefficient names as used by wavedec2() after each level of decomposition. The bottom row shows wavelet coefficients for the fingerprint image (with each subband independently normalized for easier visualization).
Here, It can be seen that many of the coefficients are near zero (gray). This ability of the wavelet transform to sparsely represent natural images is a key property that makes it desirable in applications such as image compression and restoration.

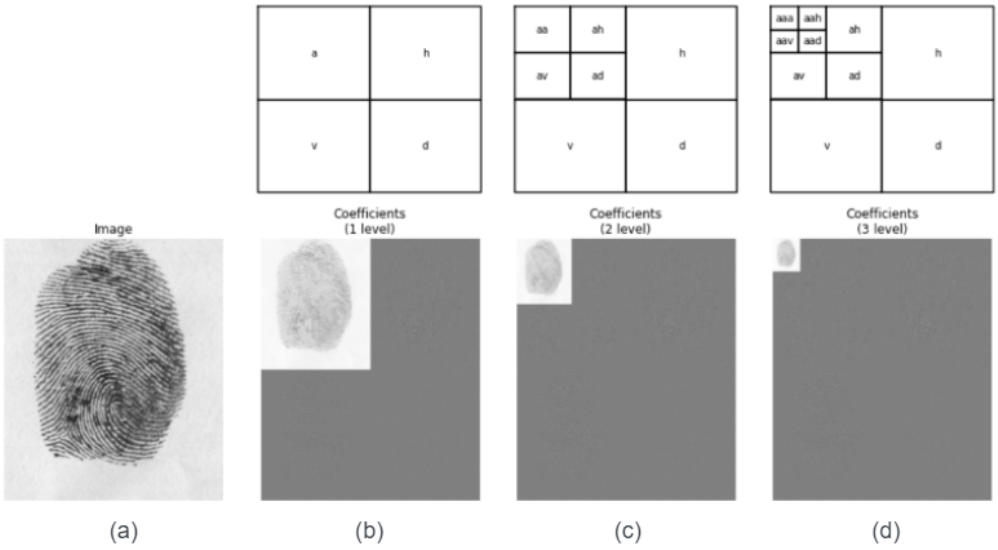


Figure 2: Multilevel DWT of a Fingerprint Images

5 Continuous Wavelet Transform (CWT)

Continuous wavelet transform (CWT) is defined as adding all the time signals and multiplying by the shift version of the wavelet. The output of the continuous wavelet transform gives the wavelet coefficients as the output.

A wavelet is a waveform of effectively limited duration with an average zero value. It is defined as:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a})\ a,b\ \epsilon\ Real \tag{9}$$

Here, a and b are called dilation and translation parameters, respectively. Here we uses the morlet wavelet which equation as follows:

$$\psi(t) = e^{jw_0 t}e^{-\frac{t^2}{2}} \tag{10}$$

The CWT of a signal f(t) is given by the equation below:

$$CWT_{(a,b)} = (f,\psi_{a,b}) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} f(t).\psi * \frac{(t-b)}{a}\,dt \tag{11}$$

Here, (f,$\psi_{a,b}$) is the inner product. The results of the CWT are many wavelet coefficients that are a function of a and b. For different shapes of the wavelet, we compute the coefficients and plot them on the magnitude axis. Here we uses the morlet wavelet for computing the cwt coefficients-

Figure 3: CWT of a Fingerprint Images

6 First and Second-order Scattering Coefficients

Extract the scattering coefficients from the image up to the 2nd order (2nd layers of scatnet) with the help of the kymatio Scattering2D module. For each u = (2^J)(n1, n2) with $1 \leq$ ni ≤ 4, the scattering vector SJ[p]X(u) is displayed for paths of length m = 1 and m = 2 in figure 4

7 First Layer With Spatial Wavelets

This section defines the first wavelet modulus operator $\widetilde{W_1}$ which computes S_0x and U_1x from the input image x. Locally invariant translation and rotation coefficients are first computed by averaging the image x with a rotation invariant low pass filter $\phi_J(u) = 2^{-2J}\phi(2^{-J}u)$:

$$S_0x(u) = x * \phi_J(u) = \sum_v x(v)\phi_J(u-v) \tag{12}$$

where ϕ is a rotationally invariant gaussian.
The averaged image S_0x is nearly invariant to rotations and translations up to 2^J pixels, but it has

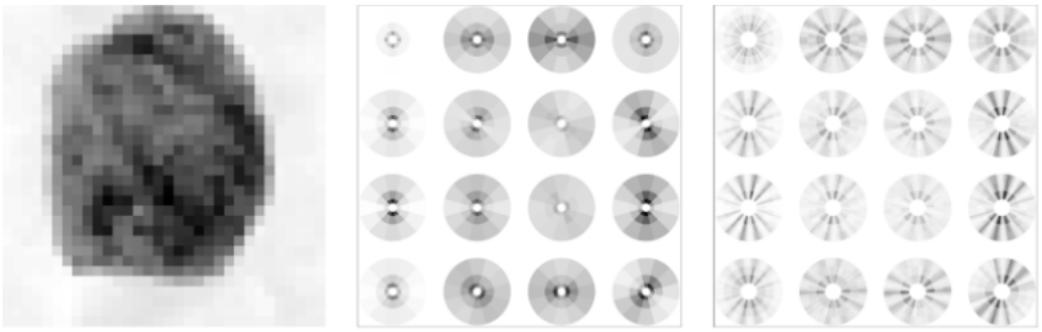


Figure 4: (a): Image X(u) of a Fingerprint Images. (b): Array of scattering vectors $S_J[p]X(u)$, for m = 1 and u sampled at intervals $2^J = 8$. (c): Scattering vectors $S_J[p]X(u)$, for m = 2.

lost the high frequencies of x. These high frequencies are recovered by convolution with high pass wavelet filters. To obtain rotation covariant coefficients, we rotate a wavelet ψ by several angles θ and dilate it by 2^j:

$$\psi_{\theta,j}(u) \;=\; 2^{-2J}\psi(2^{-J}r_{-\theta}u) \tag{13}$$

The resulting wavelet transform W_1 computes

$$W_1 x \;=\; (x * \phi_J(u),\; x * \psi_{\theta,j}(u))_{u,\theta,j} \tag{14}$$

Wavelets are designed so that W_1 is contractive and potentially unitary. We use complex wavelets whose real and imaginary parts have a quadrature phase. The complex phase of $x * \psi_{j,\theta}$ then varies linearly with small translations of x. Removing this phase with a modulus operator yields a regular envelop which is more insensitive to translations:

$$U_1 x(p_1) \;=\; |x * \psi_{\theta_1,j_1}(u)| \; with \; p_1 \;=\; (u, \theta_1, j_1) \tag{15}$$

The vector of coefficients $U_1 x$ is computed with spatial convolutions and is thus covariant to a translation of x. It is also covariant to rotations $r_{\theta'}x(u) = x(r_{-\theta'}u)$. With a change of variable we indeed verify that

$$U_1(r_\theta x)(u, j_1, \theta_1) \;=\; U_1 x(r_{-\theta}u, j_1, \theta_1 - \theta), \tag{16}$$

which defines the action of r_θ on $U_1 x$. The resulting wavelet-modulus operator is

$$\widetilde{W_1} x \;=\; (x * \phi_J,\; \{|x * \psi_{\theta,j}|\}_{\theta,j}) \;=\; (S_0 x, U_1 x) \tag{17}$$

The non-linear operator $\widetilde{W_1}$ is contractive because the wavelet transform W_1 is contractive and a modulus is also contractive. The norm of $\widetilde{W_1}x$ is equal to the norm of x if W_1 is unitary. One can prove that $\widetilde{W_1}$ is stable to deformations because wavelets are regular localized functions. For one fingerprint, it is calculated with complex Morlet wavelets displayed in Figure 5. They are equal to Gabor functions whose mean is set to zero by subtracting a Gaussian.

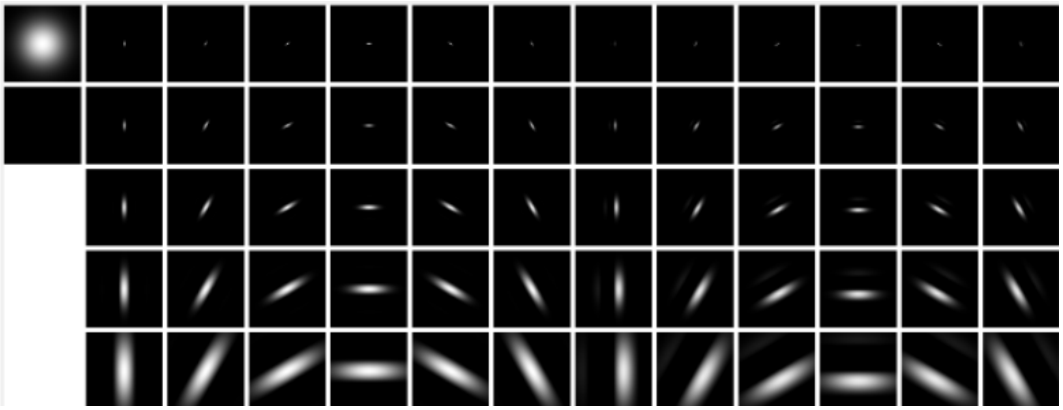


Figure 5: Quadrature phase complex Morlet wavelets $\psi_{j,\theta}$, dilated (along rows) and rotated (along columns). Their real and imaginary parts are shown on the left and on the right, respectively.

8 Testing using multiple finger (three)

This is very important part of the problem, knowing that our images are not good and messy, we try to increase accuracy by using multiple fingerprints of the same person. "Multiple" here means taking the prints of three of the fingers of our subject at once (to correctly identify, even in the case of bad print of one of the finger).

8.1 Algorithm

We first take the tuple of three fingerprints, and then calculate the scattering coefficients of level 2, which will be of the form 108*4*4, we then calculate the euclidean distance, of each of the subject's fingerprints with the corresponding class to verify and calculate the score. Lesser the distance, more our confidence will be.

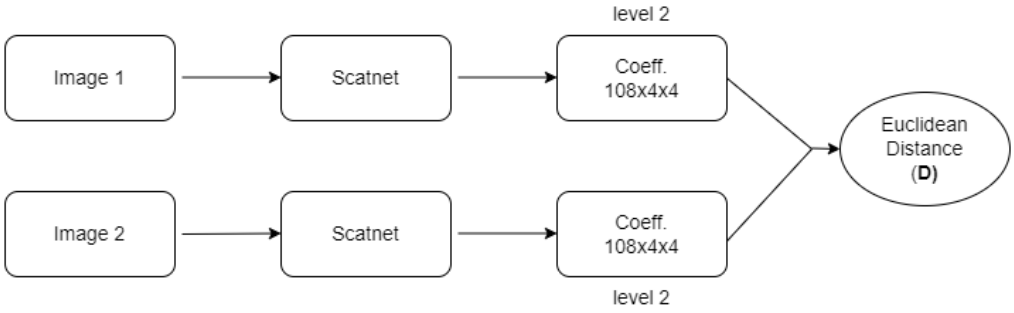


Figure 6: Matching Algorithm

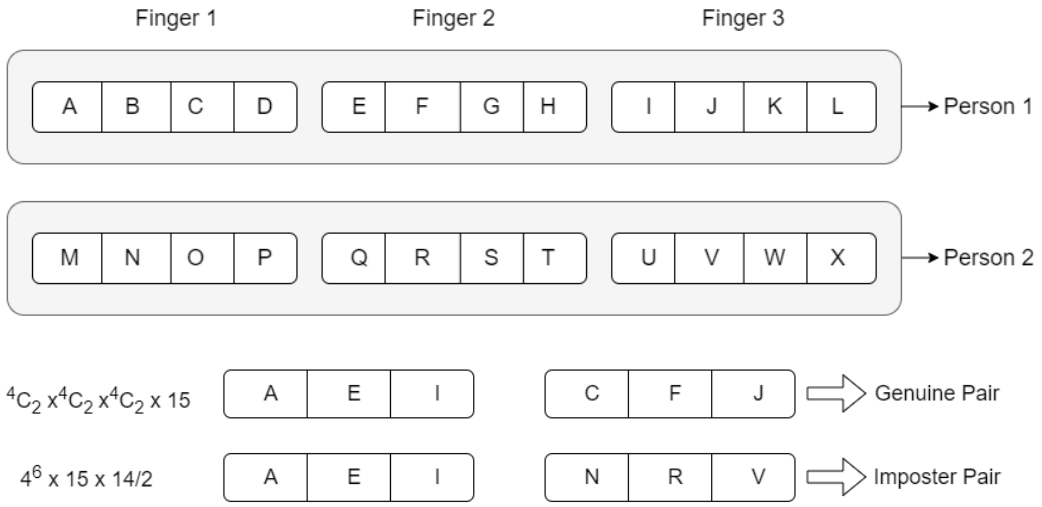
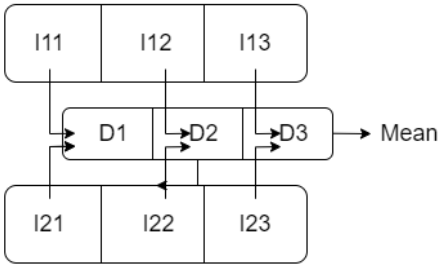


Figure 7: Genuine and Imposter Pairs

8.2 Performance

8.2.1 Genuine and Imposter Pairs and their Matching Efficiency

But now comes the question of deciding the threshold. To tackle this issue, we make genuine and imposter pairs for the tuple of three images. Genuine tuples being the one, of which all the three prints match. Imposters are the one, prints of which doesn't come from the fingers of the same person. When calculating, we processed 3240 genuine pairs and 430080 imposter pairs of tuples of three prints. We then found the average inter-genuine pairs distance and average inter-imposter pairs to be 76 and 95 units respectively. Next step is to decide threshold, which is set to be the mid value of 75 and 95, i.e. 85. Then, we calculated the credibility of this threshold rule, we found that in around 80 percent cases, it gives the correct answer, i.e. 20.3 percent FP and 21.8 percent FN.



8.3 Demo for the Verification-Identification

A total of 15 classes are taken, each containing three sub-classes representing three different fingers. Furthermore, each sub-class contains four images, three of which are used for the Identification purposes and last one as the test image. It was found out that 12/15 correct prediction with rank 1, and 2/15 with the rank 2.

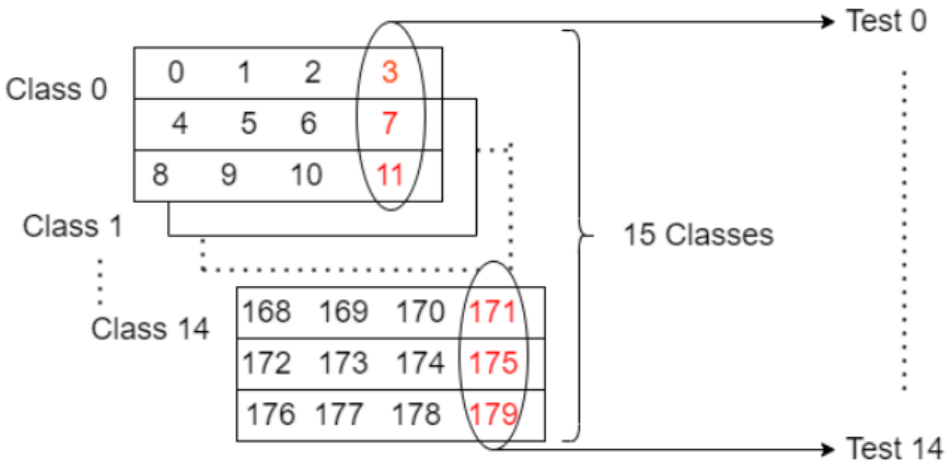


Figure 8: Identification and Verification

Test	Rank 1	Rank 2	Rank 3	Rank 4
Test 0	0	5	9	14
Test 1	1	12	4	8
Test 2	4	6	1	8
Test 3	3	11	6	4
Test 4	4	6	11	8
Test 5	5	4	11	9
Test 6	6	4	5	3

Test 7	7	4	8	5
Test 8	4	8	5	9
Test 9	9	4	6	14
Test 10	10	4	7	5
Test 11	11	5	4	7
Test 12	8	12	4	11
Test 13	13	6	4	5
Test 14	14	6	9	4

Figure 9: Results

9 Some Ongoing Efforts and Future Plans

- **Siamese Network**: We can also use siamese-type network to make 128*1 1D embeddings of our 108*4*4 coefficient channels and train them using triplet loss to find similarity between the two fingerprints.
- **Fusion with the Minutiae Technique**: A minutiae approach for this same problem statement is also developed side by side which is also giving approx 80 percent results. A hybrid approach can be developed by the suitable fusion of the results of the both techniques.

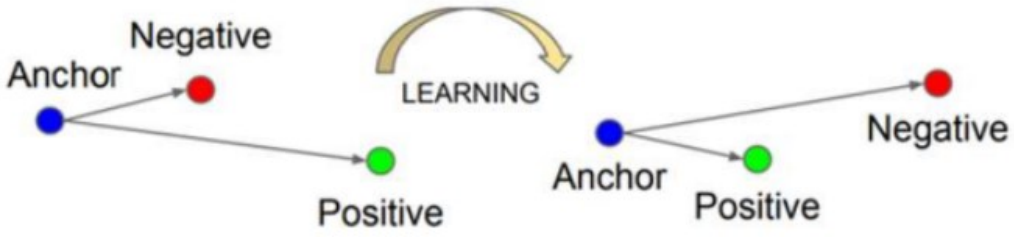


Figure 10: Triplet Loss

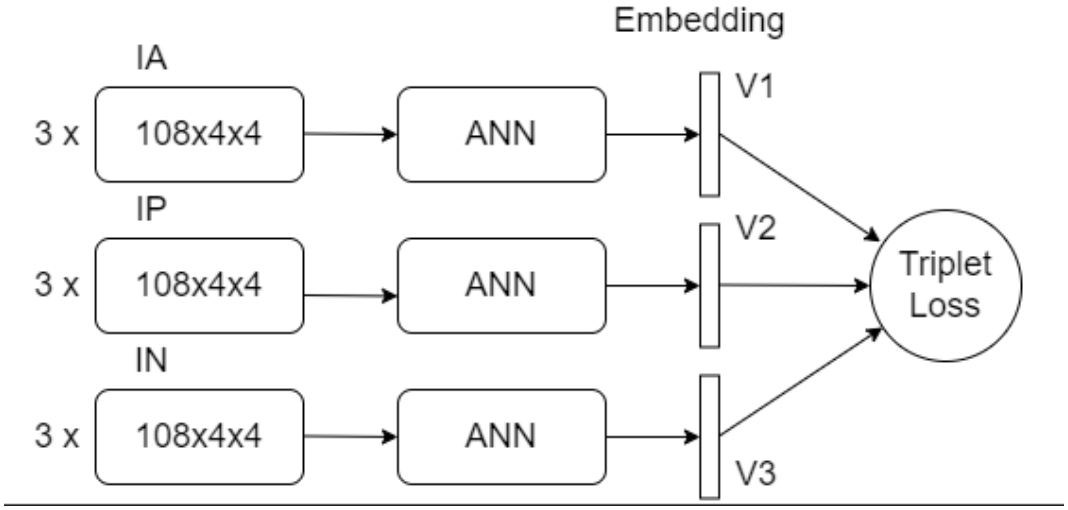


Figure 11: Siamese Network

10 References

References

[1] J. Bruna and S. Mallat, "Invariant scattering convolution networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1872–1886, 2013.

[2] S. Mallat, A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way, 3rd ed. USA: Academic Press, Inc., 2008.

[3] F. Cotter and N. Kingsbury, "A Learnable Scatternet: Locally Invariant Convolutional Layers," *IEEE International Conference on Image Processing (ICIP)*, pp. 350-354, 2019, doi: 10.1109/ICIP.2019.8802977.

[4] Birajdar Parmeshwar, Haria Meet, Kulkarni Pranav, Gupta Shubham, Joshi Prasad, Singh Brijesh, and Gadre Vikram, "Towards Smartphone-based Touchless Fingerprint Recognition," *Sādhanā*, Vol. 44, No. 7, pp. 1 - 15, 2019.

[5] Cotter, Fergal, and Nick G. Kingsbury. "Deep Learning in the Wavelet Domain," ArXiv abs/1811.06115, 2018.

[6] S. A. Grosz, J. J. Engelsma, E. Liu, and A. K. Jain, "C2CL: Contact to Contactless Fingerprint Matching," *IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 196-210, 2022, doi: 10.1109/TIFS.2021.3134867.

[7] A. Malhotra, A. Sankaran, M. Vatsa, and R. Singh, "On Matching Finger-Selfies using Deep Scattering Networks," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, Vol. 2, No. 4, pp. 350-362, Oct. 2020, doi: 10.1109/TBIOM.2020.2999850.

[8] Jannis Priesnitz , Rolf Huesmann , Christian Rathgeb , Nicolas Buchmann , Christoph Busch, "Mobile Touchless Fingerprint Recognition: Implementation, Performance and Usability Aspects," *Sensors*, Vol. 22, No. 3, pp. 792, 2022.

[9] Stéphane Mallat. "Group Invariant Scattering". en. In: Communications on Pure and Applied Mathematics 65.10 (Oct.2012), pp. 1331–1398.

[10] Edouard Oyallon and Stephane Mallat, "Deep Roto-Translation Scattering for Object Classification," 2015, pp. 2865–2873.