Metric-Learning Methods for One-shot Fingerprint Verification

Tanguy Dieudonné
EPFL
Research conducted at the Neuro-Machine Augmented Intelligence Laboratory of KAIST
한국과학기술원
Under the supervision of **Prof. Sungho Jo and Hochang Lee**
Daejeon, South Korea, Summer 2023

Abstract

Learning the important features from a fingerprint can be challenging and computationally demanding in the case where we only have one image per finger. In this research, we explore the potential of siamese vision transformers for image verification and metric learning tasks using pairs of fingerprints from the SOCOFing(1) dataset. Transformer models have recently gained attention in computer vision since we can also apply for images the attention-based transformer models initially introduced for natural language processing purposes. However, most existing studies focus on the application of Vision Transformer models to representation learning, e.g. image classification or dense predictions. We compare this new model with already existing *siamese neural networks* and show that both approaches yield powerful discriminative abilities that generalize to entirely new fingerprints from different distributions. Furthermore, for the task of measuring image similarity, we compare the performance of our Vision Siamese Transformer (ViST) with Neural Network models to show the superiority of ViST. The code is available at `https://github.com/Godgiven75/KAIST-2023`.

1 Introduction

The COVID-19 epidemic compelled widespread mask-wearing, causing notable limitations for face-recognition technologies due to obscured facial features. Consequently, fingerprint recognition has remained the predominant method for secure identification purposes.

Fingerprint authentication poses a unique challenge known as one-shot learning, wherein we must work with a limited training set to compare images against. In our study, we are constrained to just one scan per finger, resulting in scarce data. Our goal is to determine the authenticity of a person's identity based on a single image. We encounter additional constraints as users scan their fingers in various conditions. There can be multiple types of alterations such as sweating, wetness or damage from burns, further complicating the authentication process.

We present a complete pipeline for fingerprint verification. The image restoration process won't be detailed in this work. Initially, we employ a U-Net combined with a Siamese Neural Network to achieve the restoration and verification tasks. We extensively evaluate and compare multiple architectures and learning techniques to achieve our objectives. In the second stage, we introduce novel architectures incorporating state-of-the-art technique. For fingerprint image reconstruction, we leverage a U-Net enhanced with Residual Connections while for image verification, we use Vision Transformers. The proposed pipeline demonstrates promising results in both tasks.

2 Methodology

We now detail both the structure and the specificities of the learning algorithms used in our experiments. The methodology adopted for image reconstruction is not detailed here.

2.1 On image verification and metric-learning tasks

Research into one-shot learning algorithms is still in a nascent stage and has yet to gather significant attention within the machine learning community. Nevertheless, several seminal investigations have paved the way for our current study.

2.1.1 Siamese Neural Network

We make use in a first stage of a modified version of the Deep Siamese Neural Network (SNN) for image recognition introduced by Koch et al(2). We indeed simply want it to learn a similarity metric for two matching fingerprints. We input two fingerprints $\mathbf{x}_i$ and $\mathbf{x}_j$ to two identical networks with shared weights and architecture. Each fingerprint belongs to a pair (x_i, x_j) associated to a label $y_{i,j} \in \{0, 1\}$ that equals **1** if it is a matching pair and **0** otherwise. Each network is a convolutional neural network (CNN) that consists of several convolutional layers, fully connected ones and activation functions (detailed in 3). After the processing of both inputs by each network, the extracted features representation outputted by the two networks are compared using a similarity metric. Our goal is to learn a function $f_\theta(.) : \mathcal{X} \to \mathbb{R}^d$ that encodes a sample x_i into an embedding vector so that pairs of non-matching fingerprints are as distant as possible (very different embeddings) and that matching fingerprints have a distance close to zero (very similar embeddings). Possible similarity metrics are the L1 (Manhattan) distance, the cosine distance and the Euclidean distance. We chose the latter in our experiments. As for the loss function, popular ones are the Triplet Loss (Schroff *et al.* (4)) and the Contrastive Loss (Chopra *et al.* (3)) that can be used when there is no sigmoidal layer at the end of our network. The contrastive loss is of the form:

$$\mathcal{L}_{cont}(\mathbf{x}_i, \mathbf{x}_j, \theta) = \mathbf{y}_{i,j} \| f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j) \|_2^2 + (1 - \mathbf{y}_{i,j}) \max(0, \epsilon - \| f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j) \|_2)^2$$

and minimizes the embedding distance when it is a pair of matching fingerprints. Note that ϵ is a hyperparameter defining the lower bound distance between non-matching fingerprints.

Another possible choice is the Binary Cross Entropy (BCE) Loss, commonly used when we link the outputs of each twin network to a sigmoidal layer, resulting in a unique embedding vector $s(x_i, x_j) = \sigma(\sum_j \alpha_j |\mathbf{h}_{1,L-1}^{(j)} - \mathbf{h}_{2,L-1}^{(j)}|)$, where $\mathbf{h}_{1,L-1}^{(j)} = f_\theta(x_i)$ and $\mathbf{h}_{2,L-1}^{(j)} = f_\theta(x_j)$ are the hidden vectors of the penultimate layer of our network and σ the sigmoid activation function:

$$\mathcal{L}_{BCE}(\mathbf{x}_i, \mathbf{x}_j, \theta) = -\mathbf{y}_{i,j} \log(s(x_i, x_j)) - (1 - \mathbf{y}_{i,j}) \log(1 - s(x_i, x_j))$$

We discuss in 5.3.2 our final choice.

The final output of the network isn't the class of the support image with the highest probability like in (2) but rather a distance d between the two images. To decide if we indeed have a matching pair of fingerprints, we decide to put a threshold $T = 0.5$ such that if $d > 0.5$, we say that it is a matching pair of fingerprints, and the contrary when $d \leq 0.5$.

2.1.2 Multi-Head Self-Attention Mechanism

The attention mechanism is used in Transformers first introduced by Vaswani *et al.*(5) in 2017 for machine translation. They have since become the state of the art in NLP-related tasks. This mechanism is a part of our second stage compromised of more advanced methods. It is used in the encoder block later mentionned in 4.2. The input consists of queries and keys of dimension d_k, and values of dimension d_v. The attention scaled dot-product is formulated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

where $Q \in \mathbb{R}^{hw \times d_k}$ is the query matrix, $K \in \mathbb{R}^{hw \times d_k}$ is the key matrix and $V \in \mathbb{R}^{hw \times d_v}$ is the value matrix with h and w the height and width of the query, key and value feature maps. The scaling by $\frac{1}{\sqrt{d_k}}$ counteracts possible extremely small gradients. To capture multiple visual features of interest for our one-shot verification, we linearly project the queries, keys and values H times with different learned linear projections to d_k, d_k and d_v dimensions respectively in order to perform the attention mechanism in parallel. The model thus identifies different regions of interest and collects important information from different subspaces:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V),$$
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_H)W^O,$$

where $W_i^Q \in \mathbb{R}^{d \times d_k}$, $W_i^K \in \mathbb{R}^{d \times d_k}$, $W_i^V \in \mathbb{R}^{d \times d_v}$ and $W_i^O \in \mathbb{R}^{hd_v \times d}$ are parameter matrices and H is the number of heads. Note that $Q = K = V$ in the MSA in the encoder.

3 Initial approach: U-Net & Siamese Neural Network

For coherent comparison, we design neural networks with a similar number of weights (approx. 11,500,000).

3.1 On image verification

3.1.1 Siamese Neural Network (SiaNet-18)

We firsts make use of a 18-layer plain neural network being the ResNet-18 with no residual or skip connections and a slightly modified fully connected layer. The latter now has a ReLU activation function and two linear layers instead of one, eventually outputting a vector of resolution (1, 1).

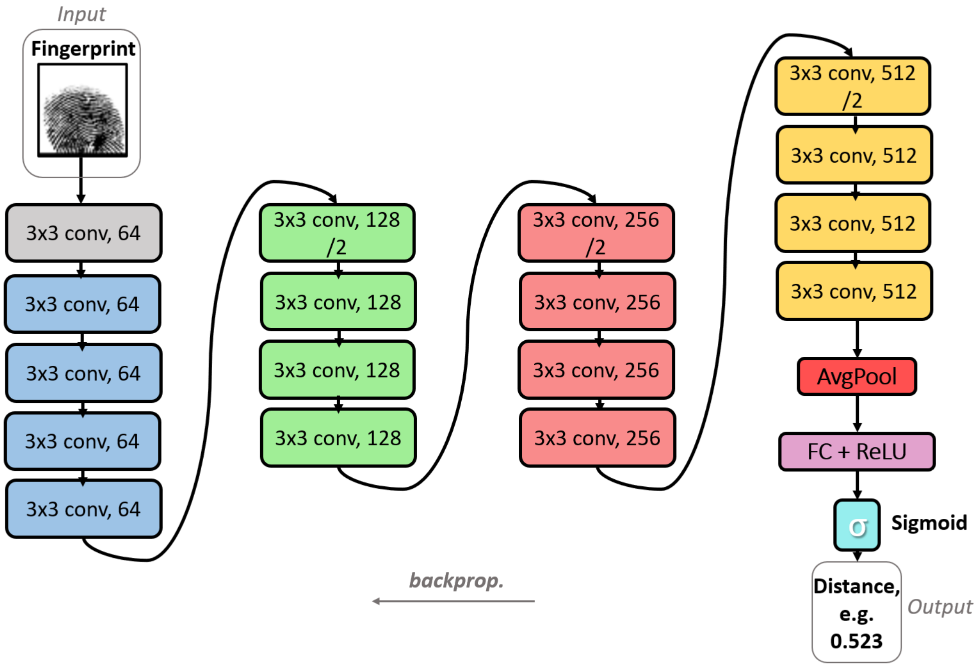

Figure 1: the Net-18 being used in the SiaNet-18

Although our model is tested on images from categories (ie. fingers) it has never seen during training, it still manages to tell for new fingerprints whether they are the same pairs or not. This ability is due to the assumption that the learned embedding can be generalized to be used even for measuring the distance between unknown fingerprints. This is the same type of assumption made when using transfer learning: via the adoption of a pre-trained model, we can benefit from it as long as the new task doesn't diverge from the original task the model was trained on.

3.1.2 Siamese Residual Neural Network (SiaResNet-18)

The main difference with the previous architecture lays in the presence of residual and skip connections as introduced by He *et al.*(6). They reformulate layers as learning residual functions with reference to the initial inputs instead of learning unreferenced functions.

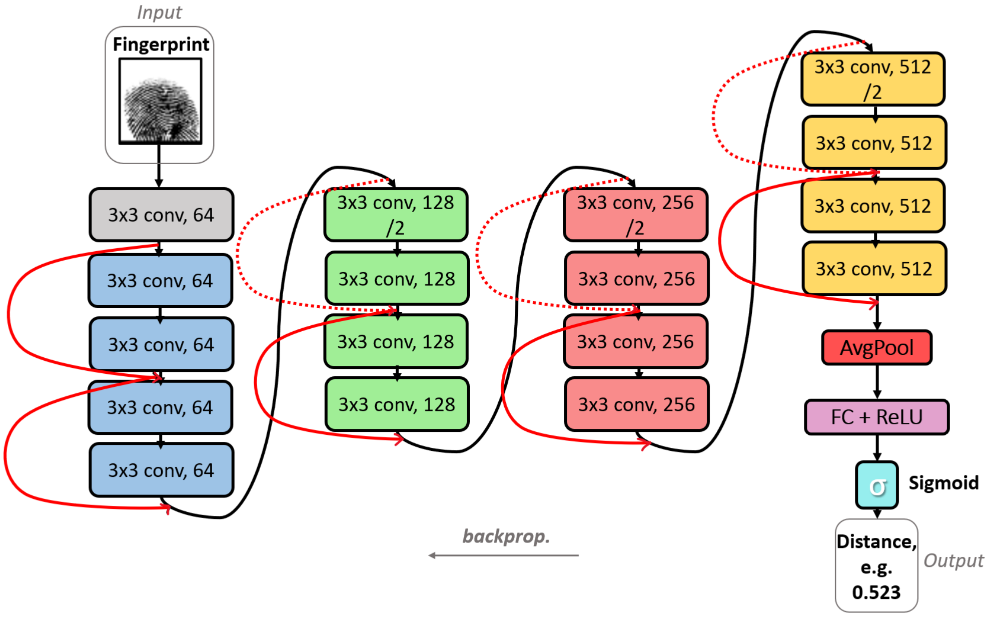

Figure 2: Overview of the modified ResNet18 used in the SiaResNet-18. In plain red lines are the residual connections and in dashed red lines the skip connections.

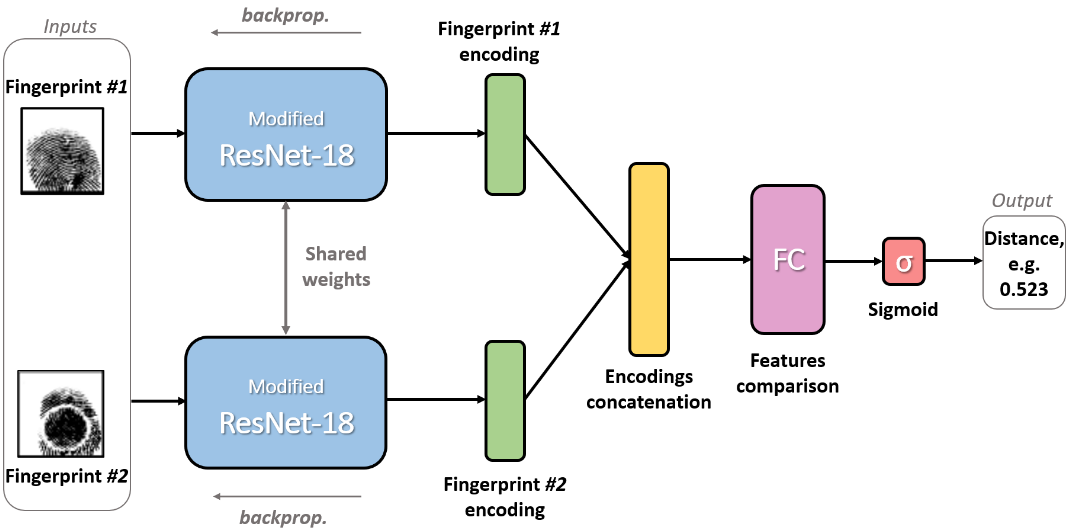

Figure 3: SiaResNet-18 overview.

We make use of the ResNet-18 first introduced in (6). We replace its last layer[1] with linear layers to compare between the features of the two images.

[1] the linear layer right after the average pooling layer

4 Better approach: Residual U-Net model & Siamese Vision Transformer

4.1 Learning

4.1.1 Weight Initialization.

To achieve good generalization error, we use the He (or Kaiming) Normal Initialization (9) such that $W \sim N(0, Var(W))$ with $Var(W) = \sqrt{\frac{2}{n_{in}}}$ where n_{in} is the number of nodes in the previous layer.

4.2 ViST: a Vision Siamese Transformer architecture

The Vision Transformer (ViT) Encoder block follows the original work from Dosovitskiy *et al.*(7) closely. To deal with the scenario of One-Shot fingerprint verification, we propose using the ViT encoder module to better explore visual feature representations and further encode them in the resulting features maps upon which a fully-connected layer operates. Experiments (cf. 5.3.3) confirm that it is a more efficient way of locating the regions of interest in a fingerprint.

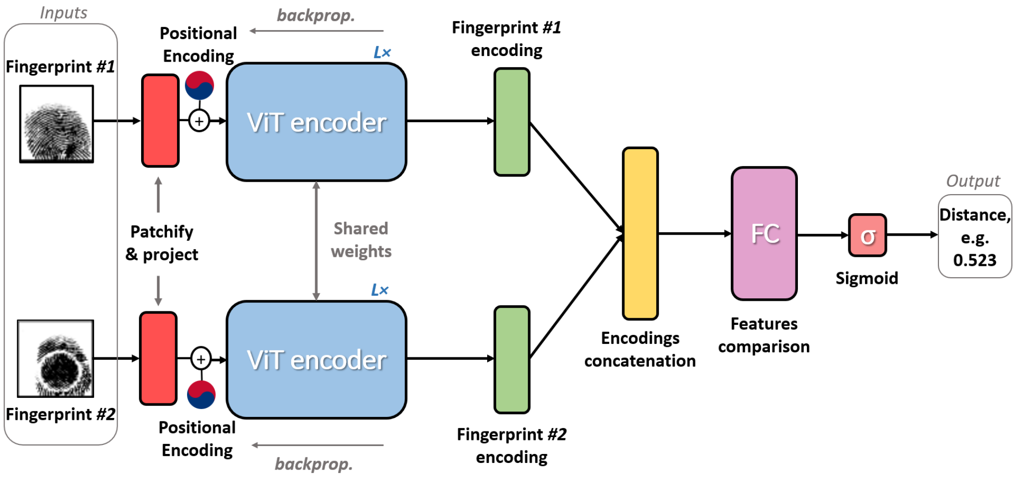

Figure 4: ViST model overview. We split each image into fixed-size patches, linearly embed each of them, add positional embeddings and feed the resulting sequence of vectors to a standard ViT encoder of L stacked encoding layers.

A model overview is depicted in Figure 4. Our Siamese Vision Transformer receives as input two images $x_i, x_j \in \mathbb{R}^{H \times W \times C}$. Each image is first patchified into a sequence of N 2D flattened patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$ where $N = HW/P^2$ with P the size of a patch. In the case of our fingerprints of resolution $(95, 95)$, we obtain 25 patches of resolution $(19, 19)$. We then linearly project each patch into an embedding of constant latent vector size D, without using the classification token of the original paper as we do image verification. The resulting vectors are finally fed to the ViT encoder. The Transformer encoder consists of alternating multi-head self-attention (MSA) layers and MLP blocks. We selected 12 heads per encoder block for a total of 12 blocks and $D = 36$ for the latent vector size. We apply LayerNorm (LN) before each layer and residual connections after every two pairs of LN and MLP Block/MSA layer. We stick to the *BCELoss* and reuse the fully-connected layers and sigmoidal layer used for the SNN to output the final distance. Note that, thus, our ViST directly outputs at once the distance instead of working in a auto-regressive fashion in the likes of usual Transformer models.

To decide if we indeed have a matching pair of fingerprints, we decide to put the same threshold as for the SiaResNet-18, meaning $T = 0.5$ such that again, if $d > 0.5$, we say that it is a matching pair of fingerprints, and the contrary when $d \leq 0.5$.

Overall, the ViST architectures contains about 1,498,365 parameters.

5 Experiments

We now evaluate the metric-learning capabilities of the SNN and ViST. We use no pretraining for every model.

5.1 Setup

Dataset. We use SOCOFing(1) made up of 6,000 fingerprint images from 600 subjects. The training set corresponds to 480 randomly picked individuals, and the remaining 120 subjects are used for testing. Some fingerprints can only be distinguished by minors details, making this dataset challenging. Furthermore, it also contains altered versions of these fingerprints. They are regrouped in three categories: "altered-easy", "altered-medium" and "altered-hard". Each category contains three types of damage: a central rotation, an obliteration or a Z-cut.

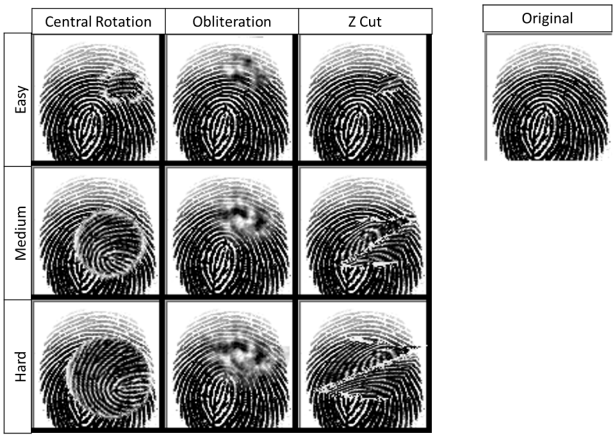

Figure 5: Dataset illustration.

We train our model by creating pairs in a fashion described earlier (2.1.1). We decided to feed all possible types of pairs at once in the following way: matching and non-matching fingerprint pairs each represent 50% of the total existing pairs. Independently of whether they are matching or not, we create three possible pairings: a real fingerprint with one of the three possible types of alterations, each representing a third of all the generated pairs. We detail the reason for such a choice in section 5.3.1. This represents indeed a One-Shot image verification task: we only have one scan per finger.

Computation. We use a single Nvidia RTX 3090.

5.2 Implementation details

The training takes 5 epochs and uses the AdamW optimizer otherwise specified. Batch size is 64 for SNNs and 32 for ViST. The chosen learning rate is 1×10^{-5} and we include a learning rate scheduler with $\gamma = 0.7$ and $step = 1$ after noticing some issues during the learning process. The loss function output would stagnate at some point, requiring a more tailored approach for the choice of the learning rate at a specific step of training. Furthermore, we also decided to use AMSGrad(8) to correct that possible convergence issue. AMSGrad is a stochastic optimization method aimed at addressing a convergence issue observed in Adam-based optimizers. Instead of updating the parameters using the exponential moving average of past squared gradients, it adopts the maximum of past squared gradients. Finally, we use PyTorch's *BCEWithLogitsLoss* function that combines a *Sigmoid* layer and the *BCELoss* for more numerical stability.

5.3 Findings

5.3.1 Training methods

We tested two ways of feeding data to the network. The first way consists of the following: we first train our model only on pairs of real fingerprints and altered-easy ones. We then transfer the weights of that trained model to the next one which is then trained only on pairs of real fingerprints and altered-medium ones. We finally do the same with pairs of real fingerprints and altered-hard ones.

Table 1: SiaNet-18 performance compared to that of SiaResNet-18.

Name	SiaNet-18	SiaResNet-18	ViST
Total	90.39%	99.00%	99.78%
Altered-Easy	90.51%	99.03%	99.80%
Altered-Medium	90.50%	98.95%	99.86%
Altered-Hard	90.12%	99.02%	99.65%

The second way consists of pairing a real fingerprint with one of all three types of alterations and inputting them to the network with equal probability. The first approach resulted in a 83.85% versus 99% accuracy thus giving the latter approach our preference for the remaining experiments. This could be explained by the fact that learning with all types of alterations at once helped the model generalize better. Finally, we also use data augmentation as it yields even more generic models.

5.3.2 Loss function

We tested both functions described in section 2.1.1 and eventually chose the *BCELoss* as it provided a better accuracy on the SiaNet-18 (83.88% against 90.39%) compared to the same SNN but with a contrastive loss.

5.3.3 Comparing the SiaNet-18, the SiaResNet-18 and the ViST

Please refer to Table 1. The SNNs evoked in sections 3.1.1 and 3.1.2 show a clear hierarchy in performance: the ViST achieves notably better accuracy on every type of alteration. The SiaResNet-18 also beats the SiaNet-18 by a comfortable margin thanks to the addition of residual connections that better recall the model of features of a fingerprint.

6 Conclusion

We have presented a strategy for reconstructing damaged fingerprints and performing one-shot fingerprint identification using different neural network and Transformers based learning methods. We outlined new results comparing the performance of each approach on the SOCOFing dataset. Our Siamese Vision Transformer outperforms every other neural network approach to image verification.

In this research, we have only considered fingerprints from one dataset. We could extend the work to few-shot learning problems with multiple angle fingerprint scans and improve fingerprint reconstruction using diffusion models. The latter is currently being explored.

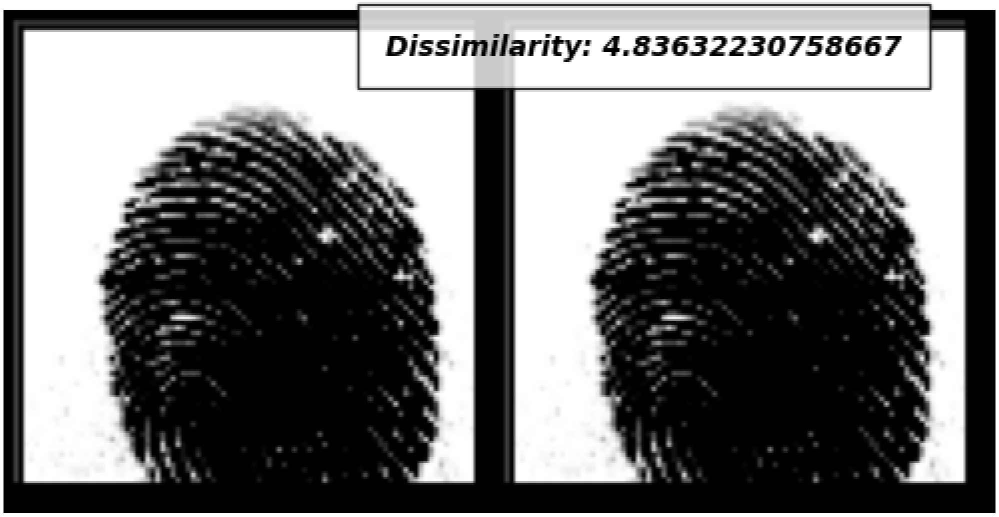

Figure 6: Distance outputted by the ViST. Those are two identical fingerprints.

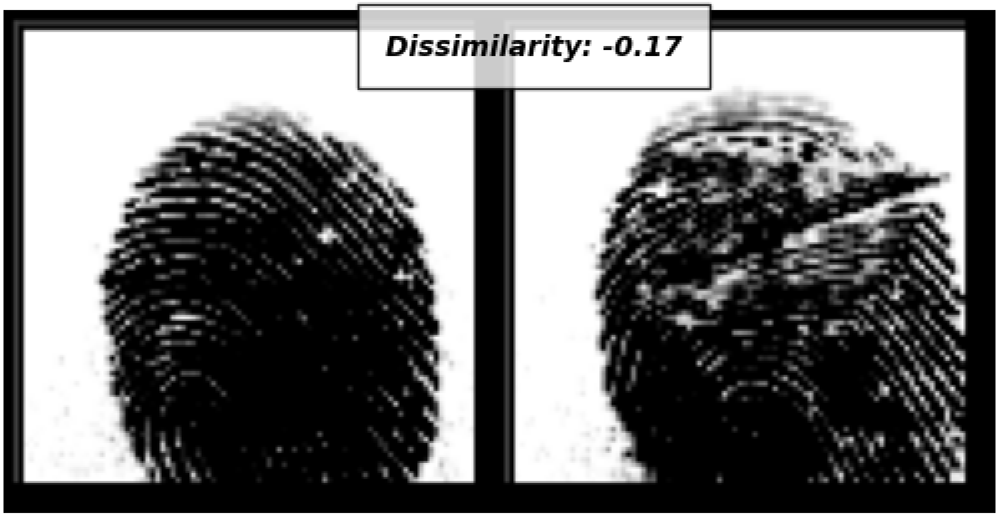

Figure 7: Distance outputted by the ViST. Those are two almost identical fingerprints: their distance is indeed smaller than 0.5.

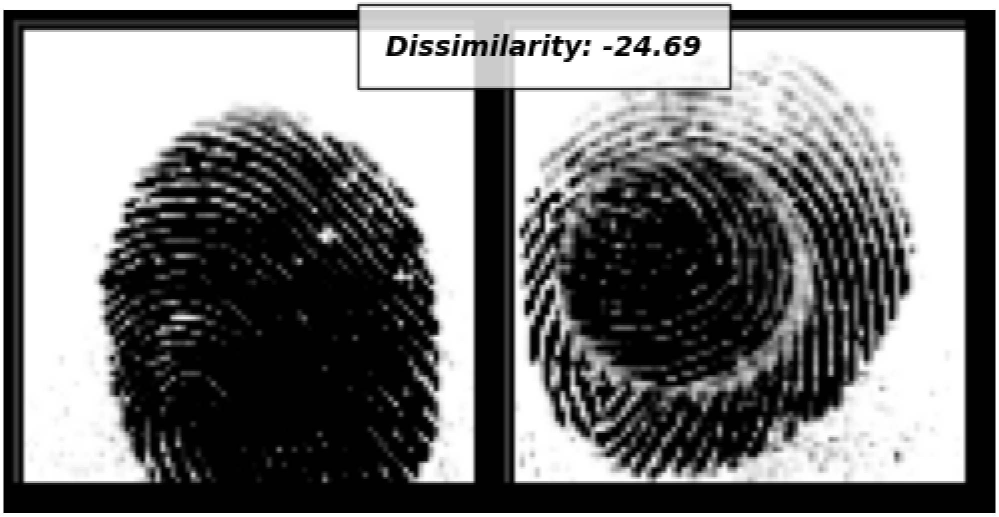

Figure 8: Distance outputted by the ViST. Those are very different fingerprints: their distance is much smaller than 0.5.

Acknowlegdments

The work was performed at the NMAIL Laboratory of KAIST. I would like to thank Lee Hochang (NMAIL) for his valuable feedback and help throughout the whole work and Prof. Jo Sungho's Neuro-Machine Augmented Intelligence laboratory for welcoming me this summer to conduct this research.

References

[1] Yahaya Isah Shehu, Ariel Ruiz-Garcia, Vasile Palade & Anne James. (2018). Sokoto Coventry Fingerprint Dataset.

[2] Koch, Zemmel & Salakhutdinov. Siamese neural networks for one-shot image recognition. ICML Deep Learning workshop, vol.2, 2015.

[3] Chopra, S., Hadsell, R. & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (pp. 539-546 vol. 1).

[4] Florian Schroff, Dmitry Kalenichenko & James Philbin (2015). FaceNet: A unified embedding for face recognition and clustering. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, pages 5998–6008, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren & Jian Sun. (2015). Deep Residual Learning for Image Recognition.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit & Neil Houlsby. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

[8] Sashank J. Reddi, Satyen Kale, Sanjiv Kumar. (2019). On the Convergence of Adam and Beyond.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren & Jian Sun. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.