Transformer based Fingerprint Feature Extraction

Saraansh Tandon
International Institute of Information and Technology,
Hyderabad, India 500032
Email: saraansh.tandon@research.iiit.ac.in

Anoop Namboodiri
International Institute of Information and Technology,
Hyderabad, India 500032
Email: anoop@iiit.ac.in

Abstract—Fingerprint feature extraction is a task that is solved using either a global or a local representation. State-of-the-art global approaches use heavy deep learning models to process the full fingerprint image at once, which makes the corresponding approach memory intensive. On the other hand, local approaches involve minutiae based patch extraction, multiple feature extraction steps and an expensive matching stage, which make the corresponding approach time intensive. However, both these approaches provide useful and sometimes exclusive insights for solving the problem. Using both approaches together for extracting fingerprint representations is semantically useful but quite inefficient. Our convolutional transformer based approach with an in-built minutiae extractor provides a time and memory efficient solution to extract a global as well as a local representation of the fingerprint. The use of these representations along with a smart matching process gives us state-of-the-art performance across multiple databases. The project page can be found at https://saraansh1999.github.io/global-plus-local-fp-transformer.

I. INTRODUCTION

From daily usage in smartphones to granting access to highly confidential resources, fingerprints have proven to be a reliable source of biometric identity. Hence it comes as no surprise that this field has been a very active domain of research since many decades. Most fingerprint feature extraction algorithms today are built upon the domain knowledge of global features like ridge-flow or local features like minutiae and pores. Knowledge of at least one of these, possibly in combination with others, is used to process an input fingerprint image and output a representation. This representation acts as a proxy of the image and is used to determine the identity of the owner of the fingerprint by matching it with a database.

Global approaches for extracting fixed-length fingerprint representations function by processing the full image as shown in Figure 2a. One of the first such attempts was made by Jain et al. in Fingercode [1] where a set of Gabor filters were used to produce a 640 byte long representation. Later as the deep learning era emerged, [2] and [3] made use of convolutional networks for the same. Recently, Engelsma et al. proposed DeepPrint[4] which uses a multi-task approach to incorporate domain knowledge into deep learning of fingerprint representations and obtains state-of-the-art performance among fixed-length global approaches.

Local approaches function by breaking down the input image into multiple local patches based on some local features. The most popular of these local features of interest are the minutiae points, because of their studied permanence [5] and uniqueness [6] among humans. [7], [8], and [9] proposed methods that led to fixed-length representations, but local approaches may also lead to variable-length representations whose size depends on the number of minutiae identified in the input image. Cao et al. proposed such an end-to-end method called LatentAFIS[10] as shown in Figure 2b. It treats the entire set of the predicted minutiae and their corresponding embeddings as the representation of the full fingerprint.

Since DeepPrint[4] and LatentAFIS[10] obtain state-of-the-art performances in their respective domains, we treat them as baselines for our approach. On analyzing the mutually exclusive failure cases of [4] and [10] we observe that the global and local perspectives can compensate for the semantic shortcomings of each other. Figure 1a shows a pair of fingerprints originating from two different identities, but due to a similar ridge-flow structure, [4] marks it as a true pair. Whereas [10] is able to focus on the subtle differences existing in the minutiae and mark it accurately as a false pair. Figure 1b shows a pair of impressions from the same finger, but [11] claims that distortion is able to fool [10] which misses a lot of minutiae matches as the corresponding patches look significantly different. [4] on the other hand is able to look at the big picture similarities at the center of the fingerprint to make the correct prediction. Note that [4] is inferior in overall performance as compared to [10] as the latter is much more strongly tied to the domain knowledge of fingerprints.

Fig. 1: Mutually exclusive failure cases of [4] and [10] respectively where the other method works well.

Hence we posit that using both global and local approaches together can help eliminate a lot of failure cases and consequently boost performance. But a naive combination poses issues including: 1) Slow feature extraction of [10] due to...
Fig. 2: Approaches for Fingerprint representation extraction: (a) A global approach that outputs a fixed-length representation. (b) A local approach that first extracts a variable number of minutiae points and then extracts representations corresponding to each of them. (c) Proposed methodology extracts a global fixed length representation, minutiae points and representations corresponding to all minutiae points in an end-to-end fashion.

expensive minutiae detection and multiple feature extraction steps. 2) Huge memory consumption of [4] due to use of big models. 3) Expensive graph matching of [10] as it evaluates all possible minutiae pairs between two images.

To facilitate the semantic fusion of global and local approaches while avoiding any of the aforementioned issues, we propose a novel methodology as shown in Figure 2c. We can list out the contributions of our paper as:

- A novel architecture which, provided a fingerprint image, can in a single forward pass generate: 1) a global representation 2) a set of minutiae points 3) a local representation for each of the predicted minutiae points. Whereas other state-of-the-art methods usually use a separate model for each of those steps.
- A methodology that:
  - is 57.93% smaller in size than [4].
  - does 54.41% faster feature extraction and 78.47% faster feature matching as compared to [10].
  - obtains a state-of-the-art FRR = %@FAR = 0.1% of 1.45 over 6 standard matching databases as compared to 5.65 of [4] and 2.32 of [10].
- First work to the best of our knowledge which makes use of a transformer model for fingerprint feature extraction.

Our approach makes use of a Convolutional Transformer based architecture because 1) the image classification capability of transformer models, as demonstrated in [12], make them suitable for global feature extraction. 2) the ability of transformers to deal with sequences and object localisation, as demonstrated in [13], make them suitable for multiple local feature extraction. 3) it has been shown by [2], [4], [8], [9] [10] that convolutional networks are effective in fingerprint representation extraction. 4) the utility of transformers for dealing with fingerprint images has been shown in [14].

II. METHODOLOGY

We will first describe the architecture of the model used in our approach, followed by details of training and inference.

A. Architecture

We use an encoder-decoder style architecture, as shown in Figure 3, to “translate” an input fingerprint image into the required representations. Let the input be a fingerprint image 

\[ X \subseteq R^{C \times H \times W} \]

where \( C \) is the number of channels and \( H, W \) represent the size of the image.

1) Encoder: We use a CvT-13 [15] model \( f_{enc} \) as our encoder which consists of 3 stages of Convolutional Transformer Blocks. On passing \( X \) through \( f_{enc} \) we obtain a \( CLS \subseteq R^{d_{enc}} \) token along with \( E_{out} \subseteq R^{L_{enc} \times d_{enc}} \) which is a sequence of \( L_{enc} \) embeddings of size \( d_{enc} \). Similar to traditional classification tasks we employ the \( CLS \) token to capture a global essence of the input and hence it is passed through a linear layer \( f_g \) to produce the final global representation of our approach \( G \subseteq R^{d_g} \) where \( d_g \) is the dimensionality of said representation. \( E_{out} \) tokens are usually ignored for classification tasks, but similar to [16], [17] we posit that these tokens can be used to extract useful auxiliary information about the input. Hence these tokens act as a “summary” of the input image and are passed on to the decoder as its “memory” input.
2) Decoder: We design a 6 layer decoder \( f_{dec} \) where we consider \( E_{out} \) as the “memory” input and use a placeholder “target” sequence of length \( L \) similar to the object queries defined by [13]. \( f_{dec} \) computes token embeddings \( D_{out} \subseteq R^{L \times d_{dec}} \) where \( d_{dec} \) refers to the dimensionality of the decoder. Each of the \( L \) tokens of \( D_{out} \) is representative of one minutia of the fingerprint and hence \( D_{out} \) is passed through:

- Pos-Ori MLP \( (f_{po}) \): This 3 layer MLP outputs \( M_{po} \subseteq R^{L \times 3} \) which represents the 2D coordinates and orientation of the \( L \) minutiae points predicted in the image.
- MinEmb MLP \( (f_e) \): This 1 layer MLP outputs \( M_e \subseteq R^{L \times d_m} \) which is the local representation of the \( L \) minutiae points of the image as characterized by \( M_{po} \). Here \( d_m \) represents the dimensionality of the local minutiae representations.

Together \( M_{po} \) and \( M_e \) form the final local representation of the input fingerprint. Here \( L \) is a hyperparameter determined by the average number of minutiae in the training data.

Algorithm 1 Forward pass of the proposed model. \( (f_{model}) \)

**Input:** \( X \subseteq R^{C \times H \times W} \)

**Output:** \( G, M_{po}, M_e \)

1. \( CLS, E_{out} \leftarrow f_{enc}(X) \)
2. \( G \leftarrow f_3(CLS) \)
3. \( D_{out} \leftarrow f_{dec}(E_{out}) \)
4. \( M_{po} \leftarrow f_{po}(D_{out}) \)
5. \( M_e \leftarrow f_e(D_{out}) \)

B. Training Stage

We pass the input through the model as described in Algorithm 1 and train using a multi-task learning framework so as to facilitate the learning of both global and local perspectives. Global Loss \( L_g \) adopts a teacher-student framework for training the model to generate good global representations for fingerprints. We use [4] as our teacher model and use it to generate ground truth embeddings \( (G' \subseteq R^{192}) \) for our training images.

\[
L_g = MSE(G, G') \tag{1}
\]

We use a COTS minutiae extractor called Verifinger to generate ground truth minutiae locations and orientations \( (M'_{po} \subseteq R^{L \times 3}) \) of minutiae points for each image in our training dataset. Since our model predicts a fixed number \( (L) \) of minutiae per image, we pick the top \( L \) candidates based on confidence scores. Next, we obtain the ground truth of the local representations corresponding to the minutiae in \( M'_{po} \) similar to [11]. This gives us ground truth local representations \( M'_e \subseteq R^{L \times 64} \).

To establish correspondences between the predicted and ground truth minutiae points in an image (Figure 4), we make use of a Hungarian matcher[18] that scores each correspondence using a linear combination of the L2 distances between the minutiae locations, orientations and representations. Once these correspondences are predicted, we reorder \( M'_{po} \) and \( M'_e \) to \( M''_{po} \) and \( M''_e \) respectively and use:

\[
L_{po} = MSE(M'_{po}, M''_{po}) \tag{2}
\]

\[
L_e = MSE(M_e, M''_e) \tag{3}
\]

We also include supervision over the outputs of the intermediate layers of the decoder. For this we use Eqn. 4 and 5 where \( M_{po} \) and \( M_e \) are the outputs of \( f_{po} \) and \( f_e \) respectively, when applied on the output of the decoder’s \( i^{th} \) layer. \( M''_{po} \) and \( M''_e \) are the re-orderings of the ground truths \( (M'_{po} \) and \( M'_e \)\) based on Hungarian matching with \( M_{po} \) and \( M_e \).

\[
L_{po}^{inter} = \sum_{i=1}^{5} MSE(M_{po}, M''_{po}) \tag{4}
\]

\[
L_{e}^{inter} = \sum_{i=1}^{5} MSE(M_e, M''_e) \tag{5}
\]

Overall this multi-task, teacher-student training paradigm is guided using the following loss function:

\[
L_{tot} = \lambda_g L_g + \lambda_{po} L_{po}^{inter} + \lambda_e L_{e}^{inter} \tag{6}
\]

C. Inference Algorithm

Given two images, the inference process of our approach has multiple steps as shown in Algorithm 2.

Feature extraction stage is executed in steps 1 and 2 which simply includes running the forward pass of the model.

Global Matching (step 3) stage involves a dot product between \( G_o \) and \( G_b \) as they are fixed-length global representations.

Although our transformer based approach is able to parallelly extract global and local feature representations, to reap the benefits of all that information we need to perform global and local matching both. Due to the expensive nature of minutiae matchers(\( \text{MinMatch} \)), this will lead to an inefficient inference algorithm. To counter this we introduce a Thresholding stage which determines the cases where local matching is not expected to add significant value to the approach. Since the calculation of \( s_q \) is pretty fast, Thresholding uses it to filter out the cases where solely the global representation leads to a confident inference. Specifically, step 4 represents a highly confident genuine pair prediction and step 6 represents a highly confident impostor pair prediction. In such cases we only use
Algorithm 2 Inference using proposed methodology.

**Input:** $X_a, X_b \subseteq \mathbb{R}^C \times H \times W$

**Output:** $s$

1: $G_a, M_{poa}, M_{ea} \leftarrow f_{model}(X_a)$
2: $G_b, M_{pob}, M_{eb} \leftarrow f_{model}(X_b)$
3: $s_g \leftarrow Norm(G_a \cdot G_b)$ \quad \triangleright \text{Global matching score}
4: if $s_g > \theta_l$ then
5: \quad $s_l \leftarrow 1$
6: else if $s_g < \theta_f$ then
7: \quad $s_l \leftarrow 0$
8: else
9: \quad $s_l \leftarrow Norm(M_{minMatch}(M_{poa}, M_{ea}, M_{pob}, M_{eb}))$
10: \quad \triangleright \text{Local matching score}
11: end if
12: $s \leftarrow (s_g + s_l)/2$ \quad \triangleright \text{Final matching score}

$s_g$ to make the final inference by assigning trivial constants to $s_l$ (steps 5 and 7).

On the other hand if $\theta_f \leq s_g \leq \theta_l$ , we assume that the global representation is not good enough for inference and hence only in these cases Thresholding permits the execution of Local Matching to obtain $s_l$ (step 9).

Dot products used for Global Matching produce scores in the range $[0, 1]$, whereas $MinMatch$ used of Local Matching produces unbounded scores greater than 0 (sum of cosine similarities of all the matched minutiae). On combining the raw scores, local matching will dominate and any possible advantages of the global perspective would fade out. Therefore, before combining these scores we need to align their distributions using a Score Normalization function ($Norm$). We make use of the Double Sigmoid\cite{19} for this purpose as it is able to bring both distributions in the range $[0, 1]$ while being robust and highly efficient\cite{20}.

The final scores are obtained using Mean-Fusion (step 12) to combine the normalized scores. Other score normalization and fusion approaches have been explored in Section V-C.

Any inference hyperparameters can be obtained using a hold-out dataset readily available in most real-world scenarios. Score distributions across various steps when Algorithm 2 is operated on all pairs of a dataset are shown in Figure 5. The utility of fusion is clearly represented by the reduced overlapping regions in Figure 5d as compared to those in Figure 5a.

III. DATASETS

For training and validation purposes we have used an in-house dataset made up of fingerprint images captured using CrossMatch and DigitalPersona optical sensors. The training and validation splits consist of 130K and 38K images respectively.

For evaluation of fingerprint matching we report on different sensors of FVC 2000 \cite{21}, FVC 2002 \cite{22}, FVC 2004 \cite{23} and FVC 2006 \cite{24} datasets following the official protocols. Each sensor of the FVC 2000, 2002 and 2004 datasets contains 100 subjects with 8 impressions per subject, leading to 2800 genuine ($= 100 \times 8 C_2$) and 4950 impostor ($= 100 C_2$) comparisons, while each sensor within the FVC 2006 dataset contains 140 subjects with 12 impressions per subject leading to 9240 ($= 140 \times 12 C_2$) genuine and 9730 ($= 140 C_2$) impostor comparisons.

IV. EXPERIMENTS AND RESULTS

We compare our proposed methodology with \cite{4} and \cite{10} as they are the current state-of-the-art approaches for global and local matching respectively. We also compare with the performance obtained by using a straightforward combination of \cite{4} and \cite{10} called Cascade. In this, feature extraction and matching are performed by running both approaches independently to completion and fusing the final matching scores as described in Algorithm 2. This is done to provide insight into how our methodology helps in parallelizing aspects of global and local approaches. We do not make comparisons with commercial matchers like VeriFinger as their proprietary nature prevents the evaluation of their complexity and we want to focus on generating an efficient approach.

Since we train using a private dataset, to make our work reproducible we present the experimentation and details of a model trained using public datasets in the supplementary.

A. Matching

Since our training dataset consists of optical images, in this experiment we evaluate on the various optical sensors of FVC datasets to assess the same-sensor performance of our approach. The minutiae predicted by our methodology obtain a Goodness Index and average positional error\cite{25} of 0.31 and 5.51 pixels respectively (with a distance threshold of 20 pixels) which is comparable to 0.20 and 5.29 pixels obtained by the open source minutiae extractor \cite{26}. This indicates the utility of our approach for the task of minutiae detection.

In Table I we report the metric $FRR\%@FAR = 0.1\%$ and...
the ROC curves for [23] can be found in the supplementary. Our methodology clearly beats both of the current state-of-the-art methods on 4 and the Cascade baseline on 5 datasets individually. Moreover we beat all three baselines on an average of 6 datasets. Hence this proves the state-of-the-art nature of our approach for the task of matching.

B. Time and Memory

We can observe in Figure 6a that our method is 54.41% faster than [10] and has almost the same speed as [4] in the feature extraction stage. This leads to a 68.72% speed-up as compared to Cascade for the same.

Figure 6b shows that our method achieves a significant speed-up in the feature matching stage as compared to [10] which leads to a 78.88% speed-up as compared to Cascade.

Finally, Figure 6c shows that our model is 57.93% smaller than the model used in [4], which is a significant advantage to compensate for the difference in matching time of the two approaches. Moreover, it leads to 61.05% more memory efficiency as compared to Cascade. Although we show that our model contains more parameters than [10], it is important to note that here we have excluded the parameters required by [10] to extract minutiae during inference. Verifinger, a black-box operator, is used for that purpose and hence its complexity cannot be evaluated. Whereas our model includes an in-built minutiae detector.

This shows that our methodology is able to bring together the semantic advantages of the two state-of-the-art approaches more efficiently than Cascade by improving upon the bottleneck factors including the time taken by [10] and the memory consumption of [4].

C. Generalization

To evaluate the cross-sensor generalization performance of our approach we evaluate on the non-optical sensors of FVC datasets as shown in Table II. Note that our training data consists of only optical images and hence thermal, synthetic and capacitive sensors are unseen domains for our model. We observe that our approach is able to obtain state-of-the-art performance across three sensors belonging to different domains.

V. ABLATION STUDY

A. Loss function

In this study we try to play around with the supervision provided to train the model by changing the values of the hyperparameters in Eqn. 6 and observe the differences in performance on FVC 2004 DB1A[23]. The magnitude of these hyperparameters was chosen based on the magnitude of loss values on the validation set.

It is clear that a model trained with a combination of local and global supervision is superior to a model trained solely for one of them. Moreover we also observe a significant benefit of including the intermediate loss terms.

TABLE I: Comparison of the matching performance ($FRR@FAR = 0.1\%$) on FVC datasets' optical sensors

| Method          | 2006 DB2A | 2004 DB1A | 2002 DB2A | 2000 DB1A | Average† |
|-----------------|-----------|-----------|-----------|-----------|----------|
| DeepPrint [4]   | 0.30      | 3.86      | 9.96      | 7.25      | 3.96     | 5.14     | 5.65     |
| LatentAFIS [10] | **0.00**  | 5.54      | 4.02      | **0.79**  | **0.36** | 1.39     | 1.79     | 2.32     |
| Cascade         | 0.01      | 3.39      | 2.62      | 1.65      | 0.93     | 1.32     | 1.50     | 1.90     |
| Our             | 0.12      | **2.00**  | **2.73**  | 1.11      | 0.86     | **1.04** | **0.93** | **1.45** |

Best and second best results are in bold and underline respectively.

† FVC 2006 DB2A has not been included in the average as it is used as a hold-out dataset in Sections V-B and V-C.

TABLE II: Comparison of the matching performance ($FRR@FAR = 1.0\%$) on FVC datasets’ non-optical sensors

| Method          | Thermal 2006DB3A | Synthetic 2006DB4A | Capacitive 2000DB2A | Average |
|-----------------|------------------|--------------------|---------------------|---------|
| DeepPrint [4]   | 5.17             | 14.58              | 1.14                | 6.96    |
| LatentAFIS [10] | 1.20             | 2.42               | 0.14                | 1.25    |
| Cascade         | 1.01             | 1.63               | 0.11                | 0.92    |
| Our             | 1.27             | 1.18               | 0.11                | 0.85    |

TABLE III: $FRR@FAR = 0.1\%$ for FVC 2004 DB1A[23] using various loss functions

| $\lambda_g$ | $\lambda_{po}$ | $\lambda_e$ | $\lambda_{inter}^{po}$ | $\lambda_{inter}^e$ | $FRR\%$ |
|-------------|----------------|-------------|-------------------------|---------------------|---------|
| 1           | 0              | 0           | 0                       | 0                   | 5.32    |
| 10          | 1              | 1           | 0                       | 0                   | 6.89    |
| 60          | 1              | 1           | 1                       | 1                   | 2.79    |
| 60          | 1              | 1           | 1                       | 1                   | 2.00    |

B. Inference parameters

In Algorithm 2 we have made use of $\theta_t$ and $\theta_f$ for Thresholding. In practice, their values can be obtained using some hold-out representative dataset. Here we make use of the FVC 2006 DB2A[24] dataset to experiment with different combinations of values for these hyperparameters. Our methodology retains all of its utility except for feature matching speed-up when trivial constants are used for $\theta_t$ and $\theta_f$. Such a case is shown in the first row of Table IV. Subsequently we observe that as the gap between $\theta_t$ and $\theta_f$ decreases, the matching time decreases too, as the number of cases for which Thresholding permits local matching decreases. But after a while it leads to a decrease in performance as the advantage of fusing local scores with global ones fades out. For our results presented in Section IV we have used
\( \theta_t = 0.75 \) and \( \theta_f = 0.15 \) as this provides the best trade-off between matching time and performance as shown in Table IV.

**TABLE IV:** \( FRR@FAR = 0.1\% \) for FVC 2006 DB2A[24] using various Thresholding hyperparameter values.

| \( \theta_t \) | \( \theta_f \) | \% times when \( \theta_f \leq s_g \leq \theta_t \) | Matching Time (sec / 1e6 pairs) | FRR\% |
|---|---|---|---|---|
| > 1 | < 0 | 100.00 | 161.48 | 0.11 |
| 0.8 | 0.1 | 25.03 | 42.74 | 0.11 |
| 0.75 | 0.15 | 19.56 | 34.08 | 0.12 |
| 0.7 | 0.2 | 12.97 | 23.63 | 0.83 |

**C. Score Normalization and Fusion**

In Table V we set \( \theta_t > 1 \) and \( \theta_f < 0 \) so that we can explore the various score normalization and fusion techniques[20] independent of Thresholding. We again use FVC2006 DB2A[24] as a hold-out dataset to determine any hyperparameters required for normalization.

Double Sigmoid[19], being the most robust and efficient technique, gives the best performance among normalizations. Although Max-Fusion gives a better performance with Double Sigmoid[19] than Mean-Fusion, it is much more sensitive to normalization and Thresholding.

**TABLE V:** Average \( FRR@FAR = 0.1\% \) over optical FVC datasets(as in Table I) w.r.t. score normalization and fusion.

| Normalization | Mean-Fusion | Max-Fusion | Min-Fusion |
|---|---|---|---|
| None | 2.22 | 2.51 | 5.75 |
| Min-max | 1.79 | 5.82 | 2.51 |
| z-score | 1.77 | 4.57 | 2.42 |
| Median and MAD | 1.91 | 2.51 | 3.93 |
| Modified tanh[27] | 1.77 | 4.57 | 2.42 |
| Double Sigmoid[19] | 1.68 | 1.46 | 3.70 |

**D. Amount of Local Information**

In this study we set \( \theta_t > 1 \) and \( \theta_f < 0 \) and alter the amount of local information available for fusing by controlling the number of minutiae points used for Local Matching during inference. Figure 7 clearly shows that a smaller subset leads to faster matching speeds but worse performances. This introduces a trade-off, but it is interesting to note that the deterioration in performance of our method is significantly less as compared to [10] due to the constant advantage of fusion with global scores. Hence this approach of selecting a random minutiae subset and merging local with global scores in all cases can be used as an alternative to Thresholding to speed-up feature matching while maintaining performance.

**VI. CONCLUSIONS AND FUTURE WORK**

We presented a novel methodology for fingerprint feature extraction where a single end-to-end transformer model can be used to parallelly and efficiently obtain global representations, minutiae predictions as well as local representations from fingerprint images instead of employing a separate module for each of them. To prove the utility of combining global and local perspectives, we also present an efficient feature matching process over these representations to achieve state-of-the-art performances on multiple datasets.

A possible future extension of this work can be incorporating a filtering mechanism for spurious minutiae (as shown in Figure 4) so that we can use different number of minutiae points for different images instead of a constant \( L \). This is important as this eliminates false minutiae matches and any potential attacks exploiting a fixed number of local features. We are also working on merging this work with [11] to integrate spoof detection in this methodology and eliminate the vulnerability against presentations attacks.
