SAPIENZA
UNIVERSITÀ DI ROMA

Sapienza University of Rome

Faculty of Information Engineering, Computer Science, and Statistics

A Contrastive Learning Framework for Closed-Set Finger Photo Identification

Submitted in fulfillment of the course "**Biometric Systems**"
MS in Computer Science

by

Luca Zhou, Matteo Concutelli, Simone Ciferri

Professor: **Maria De Marsico**

January 2024

1. Introduction

In the era of advanced biometric technologies, the quest for secure and efficient identification methods has led to the forefront of fingerprint recognition. Fingerprint identification stands as one of the most reliable and widely adopted biometric techniques, offering a unique and distinctive signature for each individual. This project embarks on the fascinating realm of finger photo identification, where cutting-edge technologies converge to leverage the intricate patterns and details captured in mobile-resolution finger images. By harnessing the power of image processing and machine learning, our endeavor aims to develop a robust system capable of accurately identifying individual fingers based on the intricate and unique features encapsulated in their images. A Colab demo of the system can be found ***here***.

1.1 Motivation

Our project is motivated by the compelling advantages offered by a finger photo identification system. In comparison to other forms of biometric traits, finger photos present several favorable characteristics:

- They can be captured with **standard equipment** like a modern mobile camera.
- The process of capturing them is **non-intrusive**, allowing anyone to capture them in their preferred environment.
- **Integration** with existing systems is viable. For example, with appropriate image processing, they can be matched against traditional fingerprints.

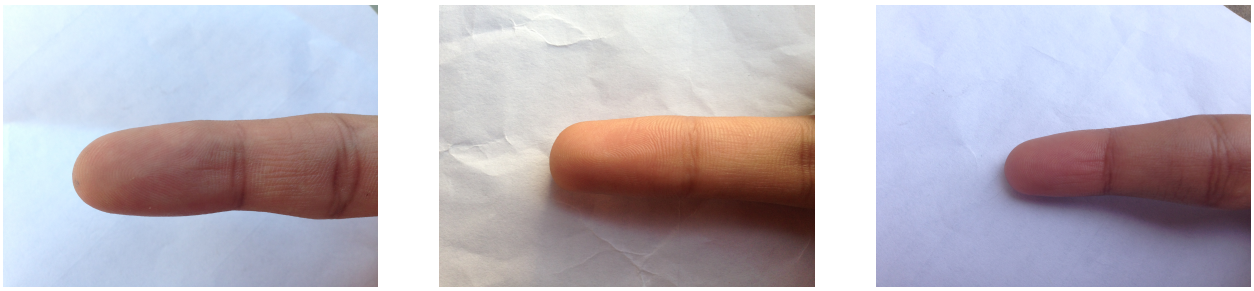
1.2 Dataset

We develop and test our system with the ***IIITD Mobile Finger Photo*** dataset, encompassing 1k+ fingerprints and 4k+ finger photos taken from 64 individuals, each providing photos of 2 different fingers. The images are categorized into:

- **Natural**: with random background;
- **White**: with white background;
- **Indoor**: under indoor conditions;
- **Outdoor**: under outdoor conditions;
- **Live Scans**: scanned with a dedicated device.

For our particular project, we exclusively utilized **outdoor finger photos with a white background**, totaling 1024 images. All finger photos originally have a size of 3264x2448. While we

acknowledge that incorporating more data could potentially enhance subsequent performance, our primary objective is to assess and validate our approach.



3 examples of finger photos used

1.3 Project Description

Within this project, we create a **finger photo identification** system. When provided with a gallery of images and a probe image, the system is designed to provide the gallery image with the highest matching score. We assume that the gallery consistently includes at least one correct match for each probe, making the task **closed-set** identification. Nevertheless, the framework can readily be expanded to accommodate open-set identification with minor adjustments.

On a high level, our project follows the following pipeline:

1. **Image preprocessing**
    - Background removal
    - Orientation alignment
    - Cropping fingertips
    - *CLAHE* enhancement
    - Thresholding
    - Ridge Thinning
2. **Contrastive learning**
3. **Gallery evaluation**

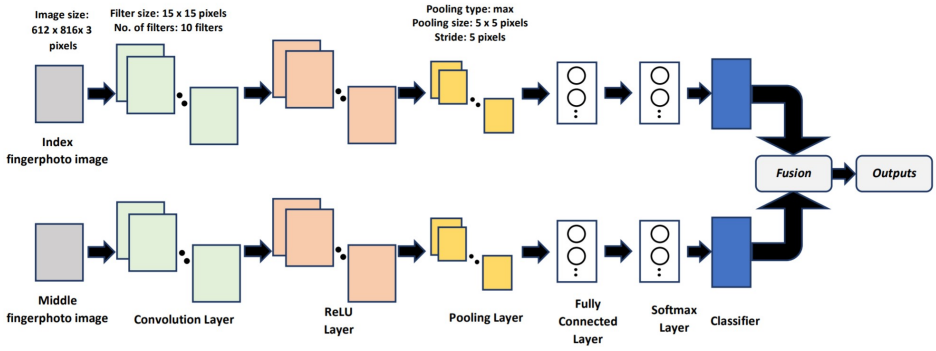
The full project source code is accessible ***here***. The rest of this report will cover some related works, our pipeline in detail, the experimental settings, and evaluation outcomes.

2. Related Work

Despite recognizing the potential advantages of utilizing finger photos for identification, our review of the existing literature revealed limited prior work in this specific domain. Although there have been some previous attempts, we identified certain limitations and sought to address them through the implementation of this project.

2.1 Dual Deep Fingerphoto Learning (DDFL)

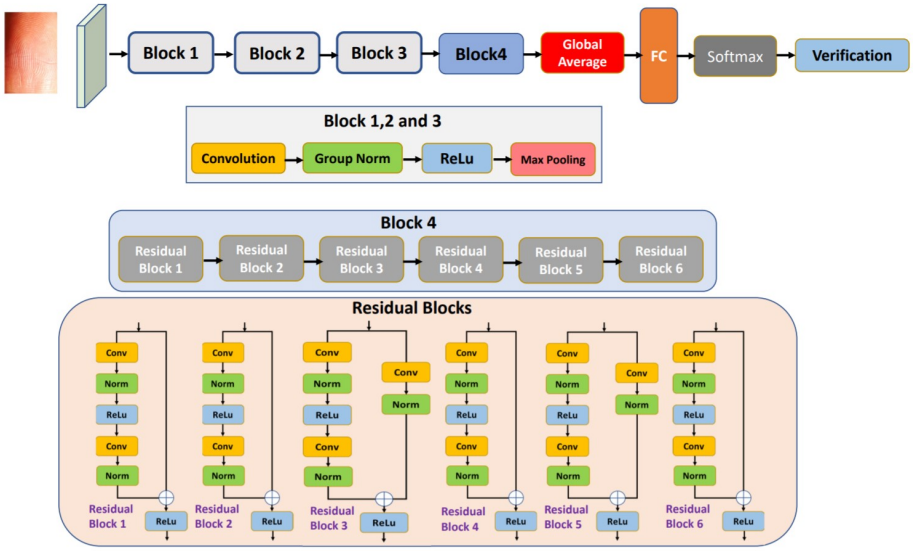
The authors in [5] endeavored to employ two distinct finger photos for the collective recognition of an individual. Their approach involves training two parallel deep neural networks, with one specialized for the index finger and the other for the middle finger. The ultimate decision is made by combining the outputs from both networks. In contrast, our methodology does not rely on classification and operates at the finger level rather than the person level.



DDFL architecture

2.2 Finger-NestNet

In the investigation conducted by [6], the authors tackle the task of finger photo verification employing a sophisticated model that integrates *convolutional layers* and *nested residual blocks*. The network is structured as a classifier; however, an inherent limitation of this approach arises as the system faces challenges in classifying finger photos that were not part of the training dataset, given the one-output-neuron-per-class nature of classification.



Finger-nestNet architecture

2.3 Contrastive Learning

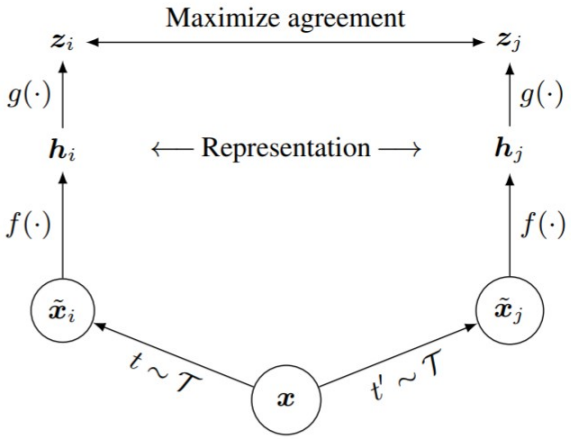
Self-supervised learning represents a machine learning paradigm in which a model autonomously learns to make predictions about specific aspects of its input data without relying on explicit human-provided labels. Instead of using labeled datasets, the model generates its own supervision signal by creating surrogate tasks derived directly from the data.

In our specific scenario, despite having access to labeled data, we found value in adopting this approach. Notably, **contrastive learning** stands out as one of the self-supervised paradigms that endeavor to train models by maximizing the similarity between positive pairs (similar samples) and minimizing the similarity between negative pairs (dissimilar samples). The core concept involves embedding data points in a manner that brings similar samples closer in the feature space while pushing dissimilar samples apart.

In our case, a positive pair consists of two images of the same finger, whereas a negative pair can be any combination of two photos featuring different fingers.

2.3.1 SimCLR

The *contrastive learning* architecture employed in this project is ***SimCLR*** [2], with just a few nuances. Originally, *SimCLR* works as follows:



SimCLR framework

An image undergoes augmentation to create two versions, both of which are processed by a backbone model and transformed into two embedding vectors h. These embedding vectors are then projected through a linear layer to obtain z and are subsequently compared. The loss function is structured in a way that encourages bringing two augmented versions of the same original image closer together in the feature space, while pushing apart augmentations from two different images.

In our specific scenario, we do not augment our data. Instead, a positive pair simply consists of two distinct images of the same finger. These two images pass through the backbone feature extractor network (pretrained ***ResNet18*** [3]) and are projected by a linear layer, resulting in two embedding vectors.

3. Method

This chapter comprehensively outlines the intricacies of our pipeline. In summary, the process involves three key steps: image preprocessing, contrastive representation learning, and evaluation.

3.1 Image Preprocessing

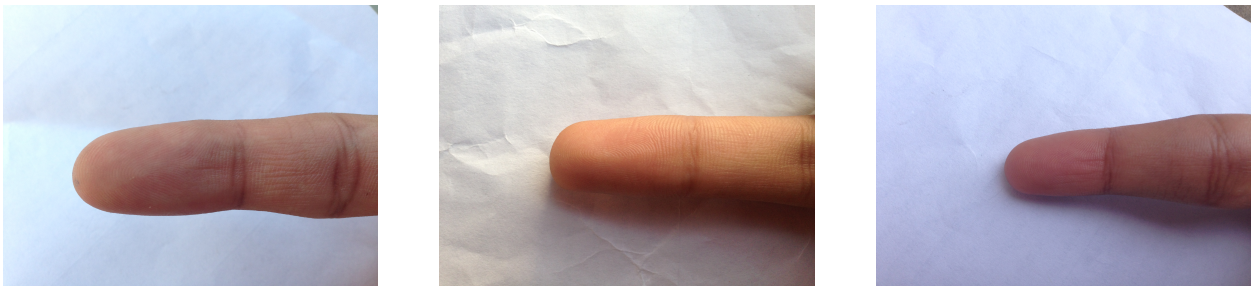
The objective of this stage is to transform the original images into a format that is readily digestible by a machine-learning model. This stage is structured sequentially, with each step building upon the previous. The steps are as follows:

1. Background removal
2. Orientation alignment
3. Cropping fingertips
4. *CLAHE* enhancement
5. Thresholding
6. Ridge Thinning

Some operations are directly executed by leveraging the ***OpenCV*** [1] library while others are by hand-crafted functions. Next, we delve into each step.

3.1.1 Background Removal

The background of the original finger photos is often noisy, containing irrelevant artifacts that might sway the feature extractor's behavior. Thus, removing the background is a must step, and we exploit the *rembg* library that implements the paper [4]. In short, this algorithm estimates foreground elements through color and opacity (alpha) considerations, using a variable-sized search window for unknown pixels. The algorithm is robust across different input trimaps. After initial alpha matte estimation, they employ a *fully connected conditional random field* (CRF) for pixel-level correction.

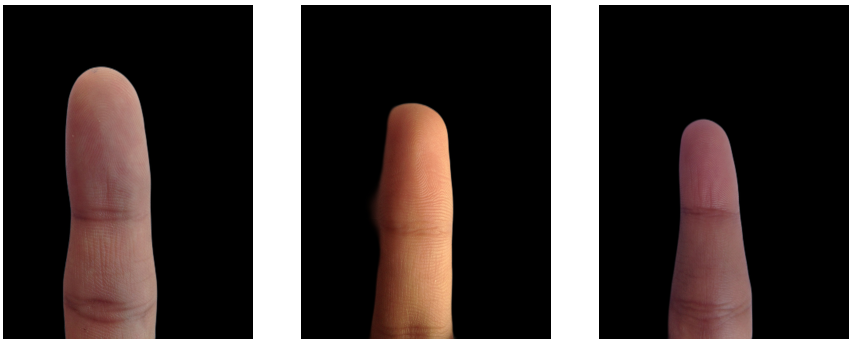


Examples of original photos

Outputs of rembg

3.1.2 Orientation Alignment

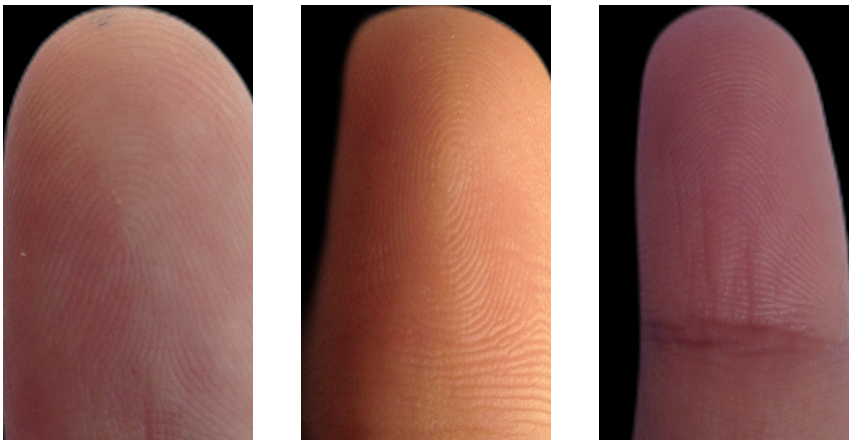
As the images in the dataset exhibit various orientations, we standardize their alignment by rotating all images such that the fingers are directed upwards. This alignment process involves computing the average grayscale values along the four extremes of an image and rotating the side with the darkest values to the bottom. This decision is based on the assumption that the darkest extreme corresponds to the direction from which the finger originates.



Rotated images

3.1.3 Fingertip Cropping

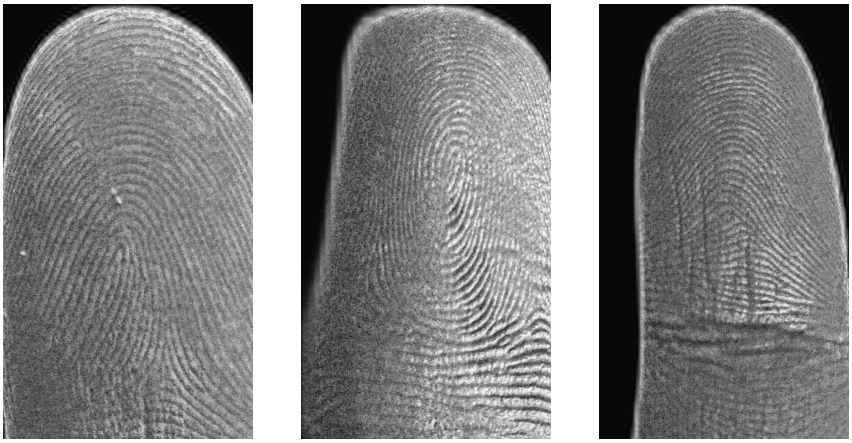
To enable the feature extractor model to focus solely on pertinent features, we selectively crop the images to include only the region enclosing the fingertip. This process involves identifying the highest pixel with a non-zero grey value. Given the removal of the background and the vertical rotation of the fingers, this pixel is expected to be situated at the top of the fingertip. Once this pixel is determined, we draw a constant-size bounding box downward from it and crop the enclosed region.



Cropped fingertips

3.1.4 CLAHE Enhancement

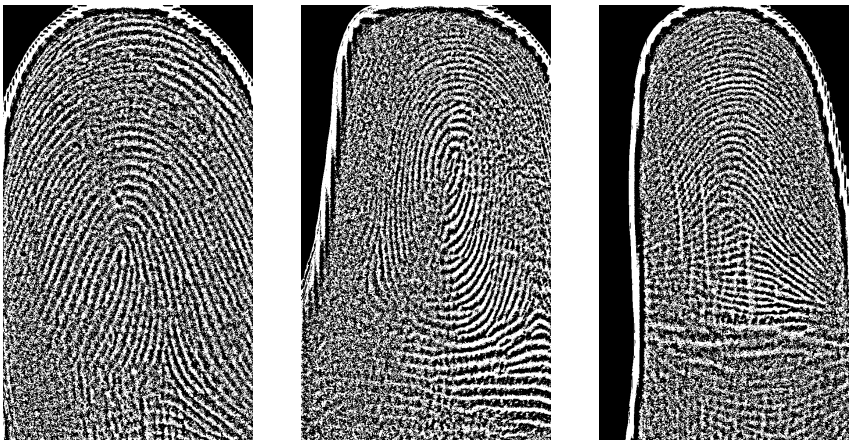
We observe that oftentimes, one side of the finger photo is darker due to the lighting conditions. To rule out this inconvenience, we apply ***CLAHE*** (Contrast Limited Adaptive Histogram Equalization). This image processing technique proves especially beneficial in mitigating the impact of uneven lighting on fingerprint images. By locally enhancing the contrast in small regions, CLAHE helps reveal intricate details in both well-lit and shadowed areas of the fingerprint, ensuring a more uniform and enhanced representation. For us, it contributes to improved visibility and feature extraction.



CLAHE-enhanced images

3.1.5 Thresholding and Thinning

To streamline the task for the feature extractor, we simplify the enhanced images by thresholding them into binary images. This is achieved using OpenCV's ***adaptive thresholding*** method, employing a neighborhood block size of 27 and a mean-based type of thresholding.



Thresholded images

As an improvement step before thinning, we cleanse the noise by sliding a 3x3 local filter across the image and blacking out a white pixel if the majority of the surrounding 8 neighbors are black.

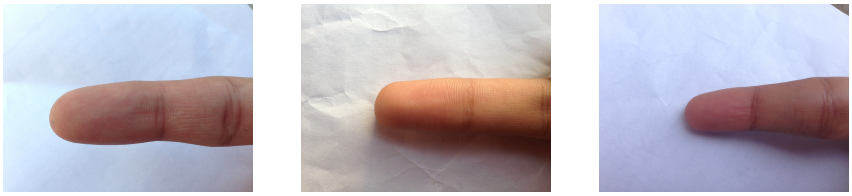
Cleansed images

Finally, we apply *thinning* to make the ridges 1 pixel wide. Specifically, we employ OpenCV's ***ximgproc.thinning*** method to reduce the ridges into one-pixel-wide skeletons.

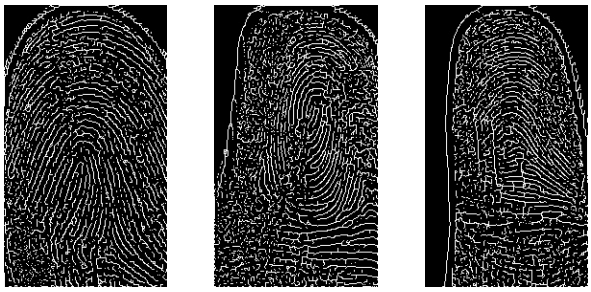
Thinned images

3.1.6 Overall Transformation

That is it, the image preprocessing stage. Let us now have a holistic look at the whole image transformation.



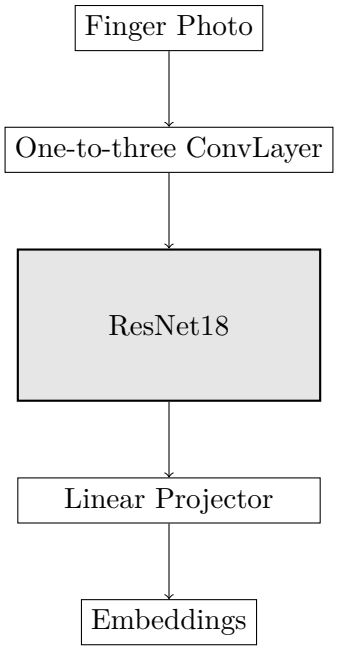
Original images



Preprocessed images

3.2 Contrastive Learning Setup

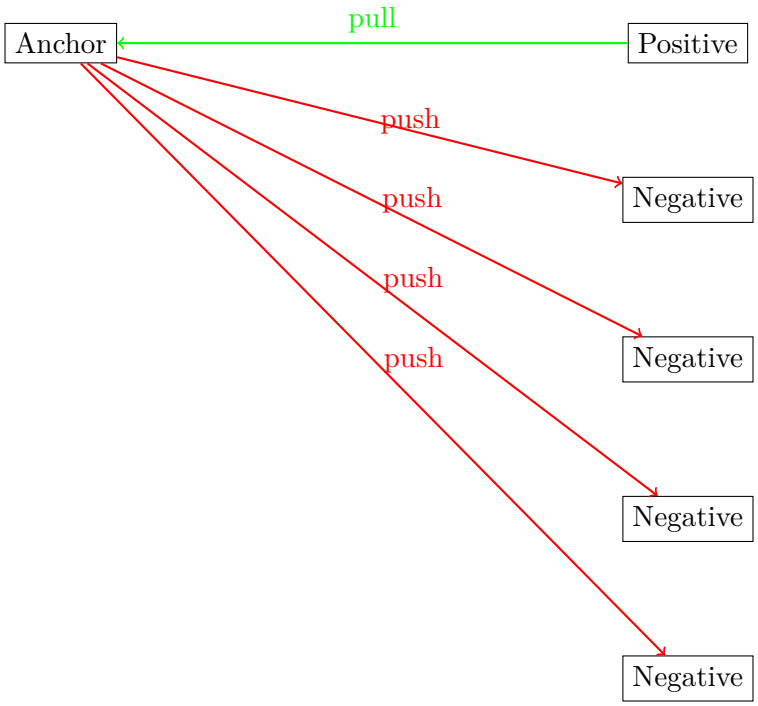
For training our backbone feature extractor, we adopt the ***contrastive learning*** framework on a ***ResNet18*** network pretrained as a classifier on *ImageNet*. Prior to it, we introduce a ***one-to-three convolutional layer*** to handle single-channel images. Following the ResNet18, a ***linear projector*** layer of size 128 is added to facilitate specialization.



ResNet18 backbone with one-to-three convLayer and linear projector

The specific framework employed adheres to ***SimCLR***, as detailed in the introduction. To reiterate briefly, the goal of *SimCLR* is to train a model to generate contrastive embeddings, where embeddings of the same entity are brought closer in the latent space, while embeddings of different entities are pushed farther apart. In our context, the objective is to have a feature extractor that produces contrastive finger photo embeddings.

To achieve this, we structure our data batches as sets of finger photos. In each batch, we designate one image as the anchor, include another image of the same finger as the positive example, and include the rest as negative images from different fingers.



The goal of contrastive learning on a batch

The entire model is trained with the **InfoNCE** loss function as proposed in the original paper of *SimCLR*.

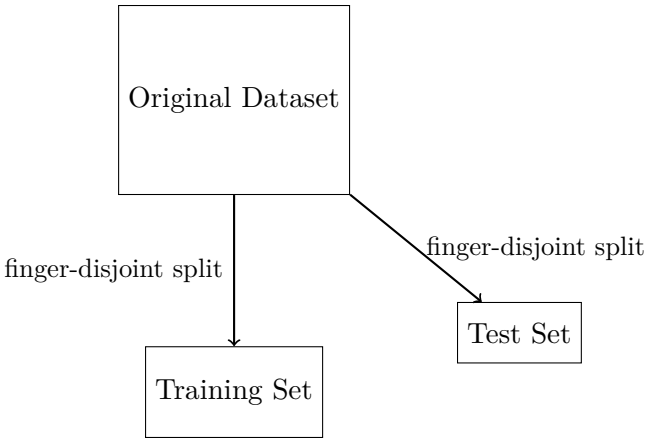
$$\text{InfoNCE Loss} = -\log\left(\frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_i^+))}{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_i^+)) + \sum_{j=1}^{K}\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j^-))}\right)$$

Where:

- $\mathbf{z}_i$: Representation of the anchor sample.
- $\mathbf{z}_i^+$: Representation of a positive sample.
- $\mathbf{z}_j^-$: Represents negative samples.
- $\text{sim}(\mathbf{a}, \mathbf{b})$: Similarity function between $\mathbf{a}$ and $\mathbf{b}$.
- K: Number of negative samples.

3.2.1 Training Setup

Our dataset comprises 8 finger photos for each of the 128 fingers, resulting in a total of 1024 images. We partition it into a training set and a test set, ensuring that the set of fingers common to both sets is empty. Specifically, we reserve all images of 12 fingers (192 images) exclusively for the test set. This choice is motivated by the consideration that, during deployment, the system should not have been previously trained on images of the incoming fingers.



Train-test split

Details about the training are omitted in this report and can be found in the code.

3.3 Evaluation Method

Following the fine-tuning of our feature extractor, we proceed with the system evaluation. Given a gallery of finger photo templates (embeddings) and a probe (finger photo), the backbone extracts a feature vector from the probe and compares it to the gallery embeddings. The gallery template with the highest matching score is then returned as the identified one.

For efficient comparisons, we input all test images into the feature extractor to generate the 192x192 **all-against-all** matrix. In this symmetric matrix, the entry (i, j) contains the matching score between template i and template j. Each template is utilized as a probe once. To determine the identified finger in the gallery, we retrieve the second argmax in each row, as the first argmax corresponds trivially to the probe itself.

As for the scoring function, we investigated the ***euclidean distance*** as well as the ***cosine similarity*** and observed no clear winner.

Using the all-against-all matrix, we simulate multiple galleries by randomly partitioning the embeddings into probes and gallery templates. To simulate a gallery, we sample a few rows as probes and some columns as gallery templates. A crucial constraint is that the gallery must include at least one template from the probe fingers, as we are dealing with closed-set identification, ensuring the presence of a correct template in the gallery.

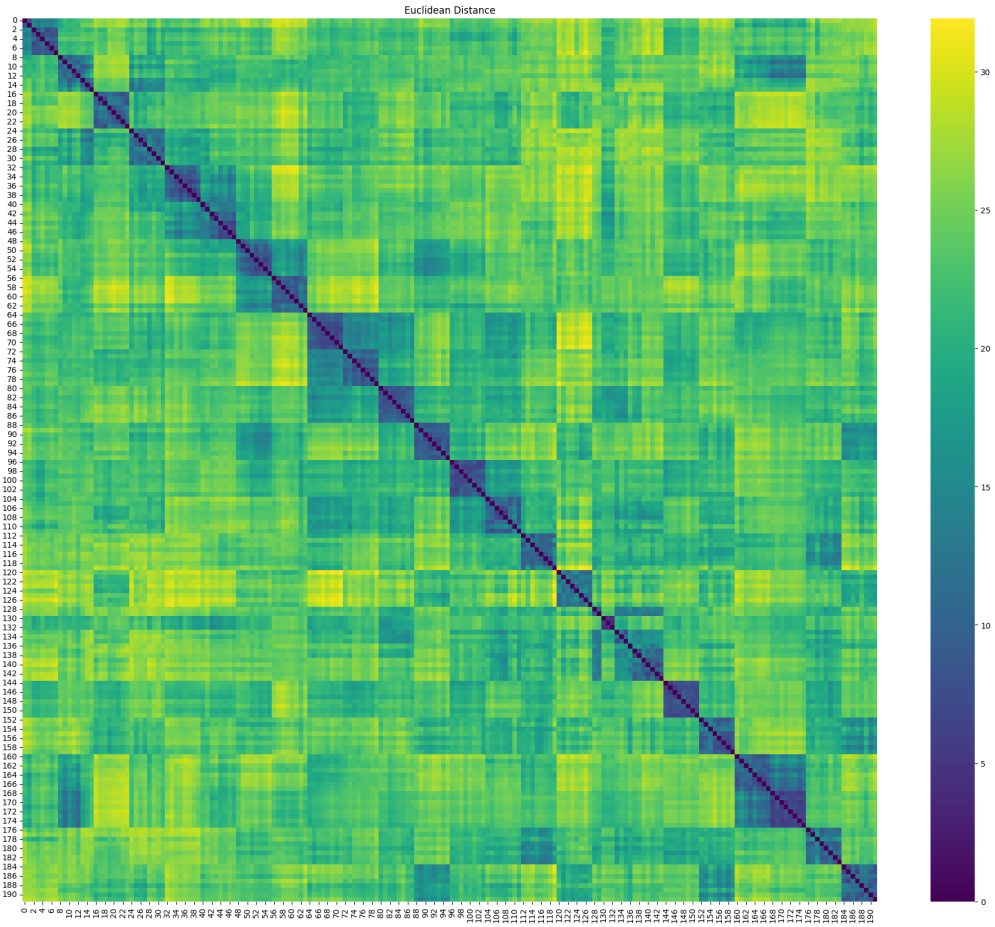
For each of these galleries, we calculate the **cumulative match score** at various ranks. Additionally, we assess the system's performance on the complete score matrix, computing a generalized metric we define as the k^{th}**-highest matching rate** for $k \in [1, 2, 3, 4, 5, 6, 7]$, considering there are 8 templates per finger. For a generic k, the k^{th}-highest identification is correct if the k^{th} highest match returns a correct template. The **identification rate** is the specific case when $k = 1$. To the extreme, if the correctness extends to the 7^{th} highest match, the system demonstrates a notable degree of robustness.

4. Results

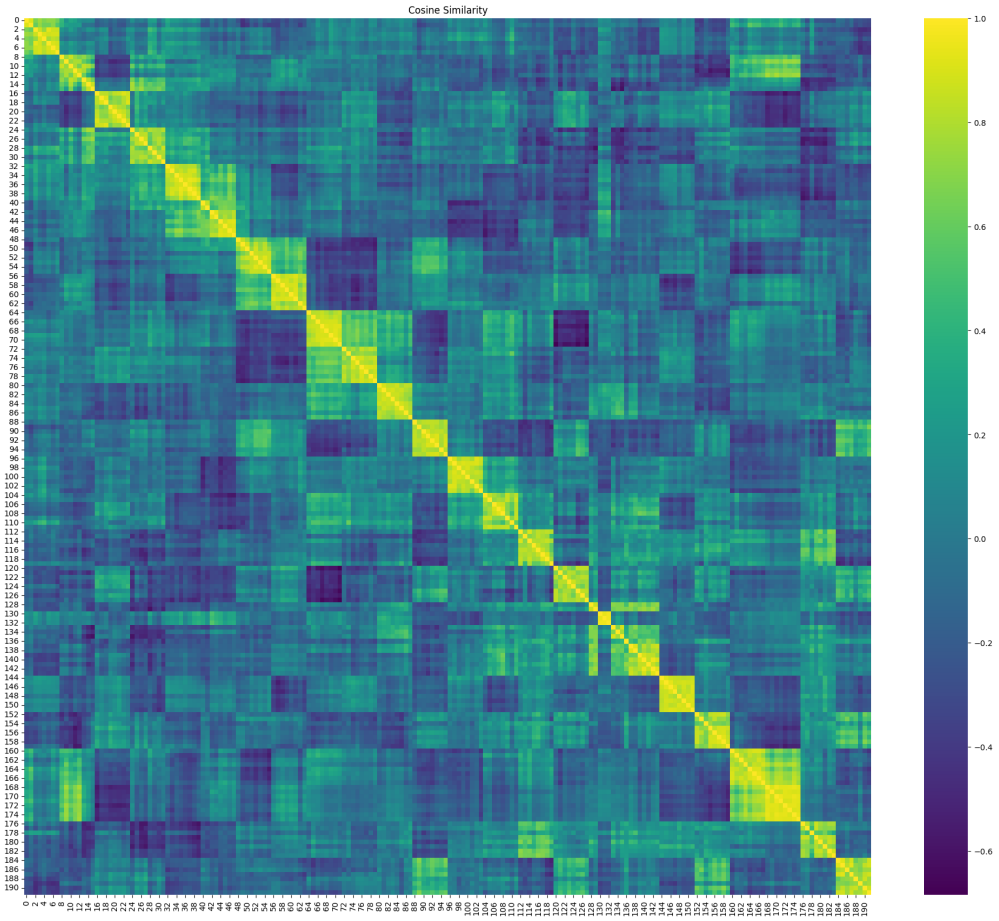
The results of the experiments are reported below. We begin by inspecting the results on the complete 192x192 matrix and then discuss the averaged outcome on random gallery simulations.

4.1 Results on Complete Matrix

The Euclidean distance matrix and the cosine similarity matrix are visually depicted below.



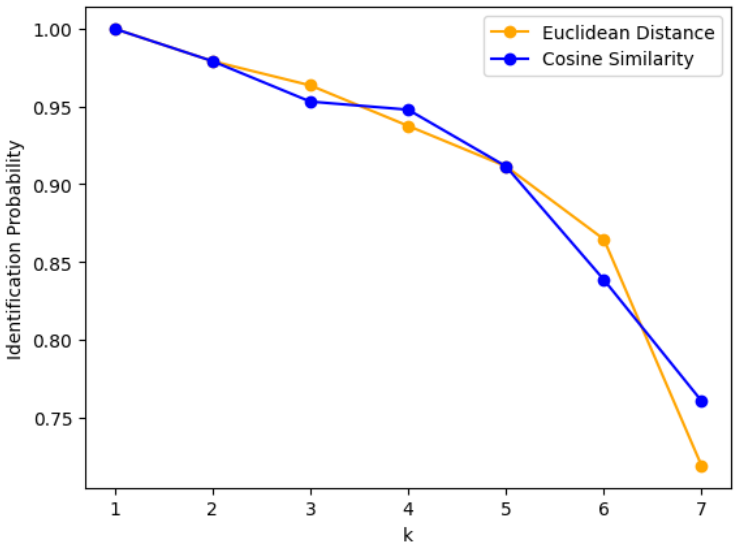
Euclidean distance all-against-all matrix



Cosine similarity all-against-all matrix

The matrices are structured in such a way that 8 consecutive rows or columns represent 8 different templates of the same finger. Notably, along the diagonal, squares of size 8 are distinctly discernible. This observation is promising as consecutive 8 templates are expected to be noticeably close in the latent space, in contrast to negative templates.

Regarding the metrics, we present a graph specifically depicting the k^{th}-highest matching rate as k ranges from 1 to 7. The identification rate is inherently captured when $k = 1$, hence a separate discussion is not provided.



k^{th}-highest matching rate

Remarkably, the identification probability ($k = 1$) for both score functions attains a perfect 1, and as anticipated, the probability gradually decreases as k increases. Most surprisingly, even when $k = 7$, the system maintains an identification rate exceeding 0.7 with both score functions. In the subsequent section, our experiments will be built upon the *euclidean distance* matrix, rather than the *cosine similarity* one.

4.2 Results on Random Galleries

To further assess the performance of our system, we simulate random use cases. In each case, we construct a gallery as a random subset of test images and reserve 30 probes for querying, while still adhering to the closed-set requirement.

We aggregate the ***cumulative match scores*** at different ranks across all 1000 simulations and calculate the average statistics.

In detail, we simulate 1000 gallery scenarios of 30 probe queries, varying the number of guaranteed positives per probe in the gallery. Beyond the guaranteed positive images, we randomly include 120 images in the gallery, potentially incorporating additional positives. Subsequently, we collect the mean *cumulative match scores* at rank k for $k \in [1, 2, 3, 4, 5, 6, 7]$.

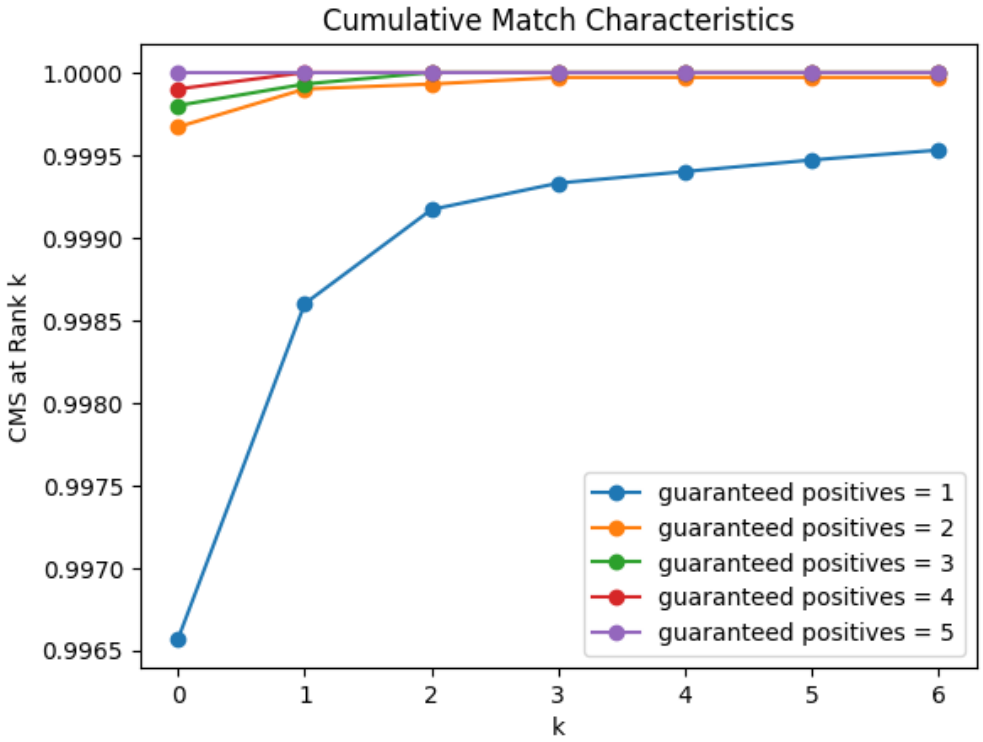
The results are reported in the table below.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
guaranteed positives = 1	0.99657	0.9986	0.99917	0.99933	0.9994	0.99947	0.99953
guaranteed positives = 2	0.99967	0.9999	0.99993	0.99997	0.99997	0.99997	0.99997
guaranteed positives = 3	0.9998	0.99993	1.0	1.0	1.0	1.0	1.0
guaranteed positives = 4	0.9999	1.0	1.0	1.0	1.0	1.0	1.0
guaranteed positives = 5	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Mean *CMS* at rank k for different configurations

For a better visual grasp, we also show the ***cumulative match characteristics*** curve, which is

the graph of *CMS* as the rank k varies.



CMC Curve

Note the observed increase in the *cumulative match scores (CMS)* as the number of guaranteed positive images in the gallery rises. This is a logical trend, as a greater number of positives provides more correct identification options for each probe.

5. Conclusion

In light of the promising applications of finger photo identification, our project revolves around a *contrastive learning* framework for extracting and matching finger photo features. The pipeline extensively employs image processing techniques to simplify feature extraction complexity, followed by the utilization of a pretrained backbone model for image embedding, with contrastive fine-tuning inspired by *SimCLR*. The obtained results underscore the effectiveness of this approach, warranting further exploration. Potential future work involves extending the framework to open-set identification, exploring alternative backbone models, and training with more data.

In conclusion, this project provided valuable hands-on experience, allowing us to delve into traditional image processing techniques, deep convolutional networks, transfer learning, and various training paradigms, many of which wound up in failures.

Bibliography

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[4] Fang-Ju Lin and Jen-Hui Chuang. Alpha matting using robust color sampling and fully connected conditional random fields. *Multimedia Tools and Applications*, 77, 06 2018.

[5] Raid Rafi, Saba Hasan, and Sahar Mahmood. Exploiting the deep learning with fingerphotos to recognize people. pages 13035–13046, 04 2020.

[6] Raghavendra Ramachandra and Hailin Li. Finger-nestnet: Interpretable fingerphoto verification on smartphone using deep nested residual network, 2022.