






## Research Article

# Improving Sensor Interoperability between Contactless and Contact-Based Fingerprints Using Pose Correction and Unwarping

Laurenz Ruzicka <sup>1</sup>, Dominik Söllinger <sup>2</sup>, Bernhard Kohn,<sup>1</sup> Clemens Heitzinger <sup>3</sup>,  
Andreas Uhl <sup>2</sup> and Bernhard Strobl <sup>1</sup>

<sup>1</sup>DSS, Austrian Institute of Technology (AIT), Vienna, Austria

<sup>2</sup>Artificial Intelligence and Human Interaction, Paris Lodron Universität Salzburg, Salzburg, Austria

<sup>3</sup>Mathematics and Geoinformation, TU Wien, Vienna, Austria

Correspondence should be addressed to Laurenz Ruzicka; [laurenz.ruzicka@ait.ac.at](mailto:laurenz.ruzicka@ait.ac.at)

Received 23 May 2023; Revised 20 November 2023; Accepted 25 November 2023; Published 18 December 2023

Academic Editor: Vincenzo Conti

Copyright © 2023 Laurenz Ruzicka et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Current fingerprint identification systems face significant challenges in achieving interoperability between contact-based and contactless fingerprint sensors. In contrast to existing literature, we propose a novel approach that can combine pose correction with further enhancement operations. It uses deep learning models to steer the correction of the viewing angle, therefore enhancing the matching features of contactless fingerprints. The proposed approach was tested on real data of 78 participants (37,162 contactless fingerprints) acquired by national police officers using both contact-based and contactless sensors. The study found that the effectiveness of pose correction and unwarping varied significantly based on the individual characteristics of each fingerprint. However, when the various extension methods were combined on a finger-wise basis, an average decrease of 36.9% in equal error rates (EERs) was observed. Additionally, the combined impact of pose correction and bidirectional unwarping led to an average increase of 3.72% in NFIQ 2 scores across all fingers, coupled with a 6.4% decrease in EERs relative to the baseline. The addition of deep learning techniques presents a promising approach for achieving high-quality fingerprint acquisition using contactless sensors, enhancing recognition accuracy in various domains.

## 1. Introduction

Fingerprints have been a reliable means of identification and verification for over a century [1]. They are unique to each individual and can provide a highly accurate and reliable means of identification [2–4]. With the advancement of technology, contactless fingerprint sensors have emerged as a more hygienic and convenient means of collecting fingerprint data for authentication and identification [5]. However, for security purposes, it is important that the data collected by these new sensors is interoperable with the large datasets of contact-based fingerprints that exist in the possession of law enforcement and security agencies. These datasets have been collected over many years and are an essential tool for police and security agencies in identifying criminals and verifying identities. A lack of interoperability can affect the

accuracy of template comparison and result in higher false rejections or false positive rates.

Contactless fingerprint sensors produce images with different characteristics than contact-based ones. Examples of this are a higher variability in image contrast throughout the image or a switch of ridges and valleys through lighting polarity inversion, which can lead to lower template comparison scores for cross-matching than for purely contact-based to contact-based template comparison [6]. Additionally, contactless fingerprint sensors are affected by various external factors, such as lighting conditions and viewing angles, which can further reduce the quality of the collected images and negatively impact template comparison accuracy. Traditional image enhancement techniques for contactless fingerprint sensors, such as denoising and contrast adjustment alone, are, therefore, not sufficient to produce images with

matching features similar to those of contact-based fingerprints, highlighting the need for more advanced methods.

There has been progress in the development of methods for improving the interoperability of contact-based and contactless fingerprint sensors in the last few years [7]. Previous research has focused on new image enhancement techniques to improve the quality of contactless fingerprint images [8–11], therefore enhancing their matching features, as well as on developing algorithms to make contactless recorded images more similar to their contact-based counterparts [12–16]. Additionally, new ways of comparing contactless and contact-based images have been the topic of research [10, 17–19, 20]. However, the task of improving interoperability still remains an ongoing challenge [21].

This paper builds on the insights gained from our previous study [22] about the acquisition of contactless fingerprints. There, we described our data acquisition process as well as the effect of enhancement and circular unwarping on the matching rate. The data presented in this study originates from the acquisition during the previous study. The data was acquired by national police officers on real data with two sensors: our contactless fingerprint sensor prototype as well as an established contact-based one. Since we observed satisfaction with the acquisition process, our next step was to work on further improving the sensor’s interoperability and, therefore, the match rates.

In this paper, we aim to improve sensor interoperability between contact-based and contactless fingerprint sensors by extending the processing pipeline by pose correction [16] and more advanced, parametric unwarping methods. The pose correction is built on top of three deep-learning neural networks using an extended and modified U-Net [23] architecture. The networks are responsible for subtasks in the pose correction step, which are detecting the fingers, segmenting the fingers, and localizing the fingers’ core. The code for training the fingertip segmentation and core detection models can be found in the study of Ruzicka [24]. The pose correction itself is accomplished by a sequence of rotations and warpings.

In the realm of pose correction for fingerprint images, two distinct categories of approaches have emerged, each with its own unique methodology and underlying principles. The first category, as proposed by Tan and Kumar [16], focuses on employing a series of rotations and warpings based on an ellipsoid finger shape to transform a set of minutiae. This approach leverages geometric transformations to rectify fingerprint poses, aiming to improve sensor interoperability between contact-based and contactless fingerprint sensors. Note that this approach is limited since it can not be applied to the image itself.

In contrast, the second category, as introduced by Dabouei et al. [25], employs a learned robust-thin-plate spline model to correct the pose of the fingerprint image. However, it is essential to note a fundamental distinction between these two categories: the approach by Tan and Kumar [16] is rooted in geometric principles and relies on an ellipsoid finger shape model, whereas the approach by

Dabouei et al. [25] is based on a learned model, lacking a clear mathematical mapping function.

The reliance on a mathematical model in pose correction is a critical aspect to consider, as it can have a substantial impact on the results obtained. Mathematical models provide a well-defined framework for transformation, offering predictable and consistent correction.

The combination of deep learning and classical techniques presents a promising approach for achieving high-quality fingerprint acquisition using contactless sensors. This would enable authentication systems based on contactless sensors to access the vast dataset of contact-based fingerprints, providing a more hygienic and convenient means of identification. An added benefit of using classical algorithms for the rotation and warping inside the pose correction step is the improved explainability and traceability of the operation. This is particularly important for real-world applications in the field of law enforcement, where transparency and accountability are essential. By leveraging the strengths of both deep learning and classical techniques, we can create a more robust and reliable system for contactless fingerprint acquisition and authentication.

This development is not only a significant step towards improved recognition accuracy in various domains, but it also provides a more user-friendly and convenient way of authentication, which is essential in today’s fast-paced world.

*1.1. Contribution of Work.* We propose a new processing pipeline for contactless fingerprint recordings that improves sensor interoperability between contactless and contact-based fingerprint sensors and requires only a single camera. Key contributions from this paper can be summarized as follows:

- (i) In this paper, we propose a novel deep-learning neural network architecture based on the U-Net [23] and MobileNet V2 [26], which is used to segment the fingertip and also to detect a reference point based on the fingerprint core. We employ transfer learning to aid the training process. The model is compared to two other contemporary U-Net-based model architectures, the EfficientUNet++ [27] and Squeeze U-Net [28]. The code for training the model for both image segmentation, as well as core detection, is shared publicly in the study of Ruzicka [24].
- (ii) Furthermore, we introduce a new pose correction step in the form of a horizontal rotation before lateral rotation, which is based on the center line of the segmented fingertip. This aligns the finger axis with the lateral rotation axis and, therefore, improves the effect of the rotation.
- (iii) Unlike earlier work [16], we developed the pose correction to work on the contactless recorded image and not only its set of minutiae. This allows us to apply further enhancements to the outcome of the pose correction step.

- (iv) Finally, we combine all of the steps above with advanced parametrized unwarping methods [29] to create a novel processing pipeline that reduces the differences between contactless and contact-based fingerprint recording, increasing sensor interoperability.

To assess the efficacy of our novel approach, we calculated the equal error rates (EERs) for a random selection of 78 participants from our previous study [22], consisting of 37,162 contactless fingerprint images. The selection was necessary due to time and computational constraints.

*1.2. Structure of Paper.* The paper is organized as follows: First, the methods Section 2 begins by describing the recording hardware in Section 2.1. This is followed by the preprocessing in Section 2.2 and pose correction in Section 2.3. Sections 2.3.1, 2.3.2, and 2.3.3 provide detailed explanations of different aspects of the pose correction. Section 2.3.1 also contains a description of our custom U-Net architecture and a comparison to other contemporary model architectures. Further enhancement steps, such as unwarping in Section 2.4, are explained next.

Section 3.1 briefly describes the template comparison paradigm, while the last three Sections 3.2.1, 3.2.2, and 3.2.3, provide details on how the datasets for training the models and evaluating match scores were constructed.

In Section 4, we present the results, beginning with the results for the fingertip segmentation task in Section 4.1. This is followed by the results for core detection in Section 4.2 and, finally, the template comparison scores in Section 4.3. Section 5 discusses the findings, and Section 6 provides the final conclusion.

## 2. Contactless Fingerprint Acquisition and Processing

*2.1. Capturing Device.* In this study, we refer to the dataset that was collected with our contactless fingerprint sensor prototype as described in the study of Weissenfeld et al. [22]. The prototype used in this study consists of three main components: a grayscale camera sensor, a liquid lens integrated with a time-of-flight (TOF) sensor, and illumination LEDs. The camera sensor captures images at a high resolution of  $3,072 \times 2,048$  pixels, which are corrected for lens distortion. Additionally, the sensor is calibrated using flat-field correction to account for variations in the sensitivity of individual pixels in the sensor, resulting in a more accurate and reliable image. The final images have a resolution of  $3,052 \times 2,015$  pixels. This results in clear, high-quality images that are suitable for fingerprint recognition.

The liquid lens integrated with the TOF sensor is a unique feature of the contactless fingerprint sensor. It allows the system to adjust the focal plane of the camera sensor based on the distance of the hand from the TOF sensor. This ensures that the fingerprint images remain sharp and in focus, regardless of the distance between the hand and the sensor. The liquid lens is capable of adjusting the focal plane in less than 5 ms, making it a fast and reliable component of the system. One caveat of the liquid lens is its small depth of

field. This allows us to assume equidistance from the camera to the hand of all sharp parts of the image. On the other hand, small deviations in the measured distance lead to blurred images.

The illumination LEDs are another important component of the contactless fingerprint sensor. They provide uniform illumination of the hand, which is critical for accurate fingerprint recognition. The three stripes of LEDs are positioned around the camera sensor, creating a ring of light that surrounds the hand. The LEDs also change color to provide feedback to the user, indicating when the hand is outside the acceptable range or when all fingers have been successfully recorded.

The contactless fingerprint sensor produces a continuous stream of images at a rate of around 10 frames per second, resulting in a video of fingerprint recordings. This frame rate is high enough to enable live identification in just a few seconds, making the system suitable for real-time monitoring and analysis of fingerprint data in applications such as security, access control, and identification. The system's ability to capture high-quality video in real-time makes it a valuable tool for biometric authentication in a variety of settings.

*2.2. Live Preprocessing.* The contactless fingerprint sensor system employs several preprocessing steps during the recording process to ensure that the captured fingerprint images are of high quality and suitable for use in biometric authentication. A depiction of those steps can be seen on the left side of Figure 1. The system utilizes an object-detection model based on a quantized Mobile Net V2 [26] to detect the fingertips in the captured images [12]. This model allows the system to quickly and accurately identify the regions of interest in the images, which are then used to extract the fingerprint images. The detected bounding boxes are also used to infer which finger is which, aiding in the identification process. This identification is based on the detection of the longest finger, which is assumed to be the middle finger. If two more fingers (i.e., bounding boxes) are found to the left of the middle finger, we can infer that the other fingers are the ring and little fingers based on their location.

Once the fingerprint images have been extracted, they are subjected to several enhancement operations to improve their quality. First, the images are rotated upright and resized to match the FBI standard of 500 DPI [30]. This step is critical in ensuring that the images contain sufficient detail for accurate identification. Fingerprint template comparison algorithms are very sensitive to the DPI resolution of the images, and deviations from the standard resolution can significantly impact the accuracy of the template comparison process. By adhering to the FBI standard, the system ensures that the images are of sufficient resolution to capture the necessary level of detail for accurate identification. Next, the image enhancement algorithm from Kauba et al. [12] is applied. It consists of a gray-value inversion, a sharpness-improving and noise-removing bilateral filter, and a contrast-limited adaptive histogram equalization (CLAHE) filter [31] with a clip limit of 4 and a tile grid size of 15 by 15

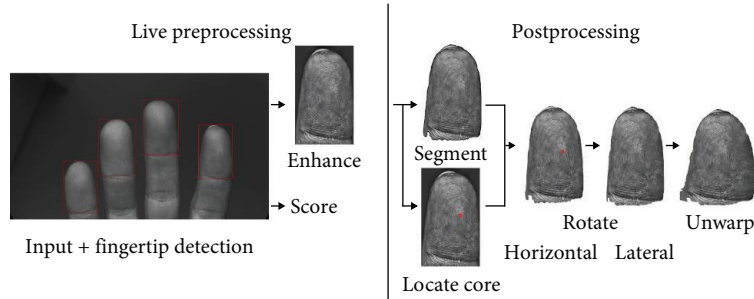


FIGURE 1: Preprocessing and postprocessing pipeline. The input is the grayscale image from the camera sensor. It is rotated and then processed by the object-detection network to find the fingertips. Those are cut out from the image and enhanced. Additionally, the fingertip’s sharpness is scored. All of this happens live as preprocessing. The results are saved and later used for the postprocessing. In postprocessing, the fingertip’s core is localized, and the finger contour is segmented. Those are used to correct the horizontal rotation of the finger, followed by a lateral rotation correction. Finally, the image is unwarped before the template comparison.

pixels. The bilateral filter uses a pixel neighborhood diameter of 2, a filter sigma in the gray level space of 4, and a filter sigma in the coordinate space of 1. The CLAHE filter enhances the contrast of the image by adjusting the distribution of its intensity values, while the gray-value inversion ensures that the appearance of the images is consistent with contact-based recordings. Both the enhanced image and the nonenhanced image are passed on to the next step.

In addition to these steps, a sharpness score is calculated for each fingerprint image during preprocessing. It is based on the Canny edge detection algorithm [32], which is applied to a ring-shaped area in the center of the finger. The number of edge pixels is counted and normalized to produce the final sharpness score. This step helps to ensure that only the sharpest and most detailed images are used for identification. Later on in the template comparison process, we use the score information of the finger to rank different recordings of the same finger. This allows us to select only the top five sharpest images for each finger.

**2.3. Pose Correction.** After the fingerprint images have been captured and preprocessed in real-time, a series of postprocessing steps are applied to further improve their quality and prepare them for use in biometric authentication. These steps include pose correction, unwarping, and possible future additional image enhancements. They are depicted on the right side of Figure 1.

The pose correction step of the contactless fingerprint sensor system is a critical component in ensuring the accuracy and reliability of biometric authentication. This step involves a series of rotations that correct the pose of the captured fingertip images to improve their 3D alignment for template comparison. In contrast to other existing pose correction frameworks, our system not only transforms the output features, i.e., the minutiae [16] but also applies both horizontal and lateral corrections to the fingertip’s pose. Additionally, it uses only one camera [14] to estimate the finger shape. The transformation of the whole image allows us to gradually improve the recording quality by adding future improvements to the pipeline.

The operations used for pose correction will be explained in the order in which they are applied. Therefore, the first

part of Section 2.3.1 is about image segmentation and horizontal rotation, the second part of Section 2.3.2 is about core detection, and the final part of Section 2.3.3 is about lateral rotation. The following pose correction is unwarping, which is explained in Section 2.4.

**2.3.1. Segmentation and Horizontal Rotation.** The first step in the pose correction process is segmentation, which involves separating the fingertip from the background of the captured image. This is achieved using a custom model based on the U-Net architecture [23], which accurately identifies the regions of interest in the image.

Other examples of contemporary neural network architectures that build on the idea of Ronnebergers U-Net are Squeeze U-Net [28] and EfficientUNet++ [27]. Both reduce the number of compute required to run the model, while improving on the performance when compared to the original U-Net. Similar to the custom U-Net extension introduced in this paper, the EfficientUNet++ builds on a pretrained decoder, in this case, the EfficientNet [33]. Compared to our implementation, the EfficientUNet++ can only work with input images of the shape  $224 \times 224 \times 3$ , since it does not resize the input image in the model. Therefore, for comparing EfficientUNet++ with the other models, we had to train and test the model on  $224 \times 224 \times 3$  sized images instead of  $448 \times 448 \times 1$  sized images, which are used for the other two models. Furthermore, all three models are designed to work as efficient as possible, leading to low numbers of trainable parameters. This is especially obvious when compared to another segmentation model like Vision Transformer [34]. Vision Transformer has 86 million parameters in the base version and 632 million parameters in the largest version. Our custom U-Net extension is the largest of the three U-Net-based models and has 10.2 million trainable parameters. Next is EfficientUNet++ with 6.3 million, followed by Squeeze U-Net with 2.5 million trainable parameters. A comparison of the three models regarding model scores can be found in Sections 4.1 and 4.2. Note that the custom U-Net introduced in this paper is used in the final processing pipeline for pose correction.

The custom U-Net architecture is a modification of the original design proposed by Ronneberger et al. [23] to



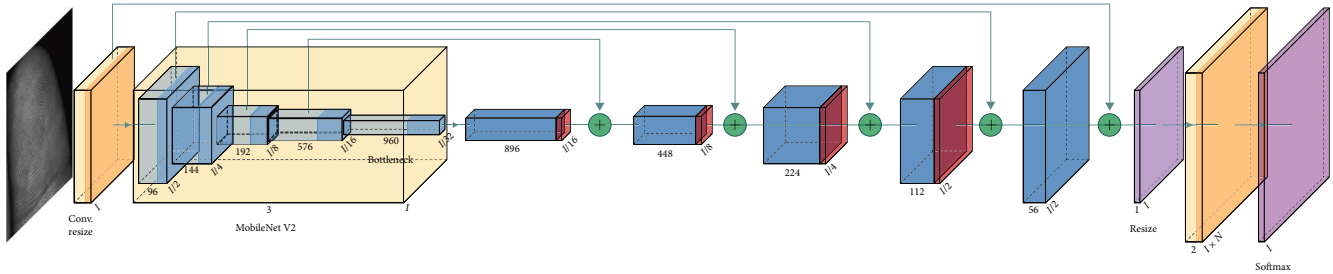


FIGURE 2: The figure shows the architecture of a custom U-Net model used for both finger segmentation and core detection. The model consists of several convolutional layers with leaky ReLU activations (right-banded orange), a pretrained MobileNet V2 (light orange), and several transpose convolutional layers (blue). The intermediate outputs of the MobileNet V2 (right-banded blue) are used for skip connections, and concatenation is denoted by green spheres with a “+” symbol. Batch normalization layers are depicted as thin red layers following the transpose convolution blocks. The first violet layer is a resizing layer, and the second violet layer represents the Sigmoid/Softmax output activation.

enhance its performance. One modification includes the use of batch normalization [35], which stabilizes the training of neural networks and reduces the internal covariate shift, preventing the vanishing or exploding gradient problem. Additionally, to prevent gradients from potentially disappearing, a leaky ReLU activation function [36] is introduced.

The pretrained neural network, MobileNet V2 [26], is incorporated into our custom U-Net architecture in this study to improve its performance in semantic segmentation. MobileNet V2 is a highly efficient architecture designed for resource-constrained devices, such as mobile phones and embedded systems. It utilizes depthwise separable convolutions and linear bottleneck layers to reduce the computational cost of the network while maintaining high accuracy. Furthermore, MobileNet V2 has been pretrained on a large-scale dataset, making it an ideal choice for transfer learning in various computer vision tasks. The U-Net architecture itself is widely used in semantic segmentation tasks due to its ability to capture high-resolution features while maintaining a large receptive field.

Figure 2 shows our custom U-Net model architecture. It consists of a downsampling path, followed by an upsampling path after the bottleneck layer. The two paths are connected through skip connections between equally shaped layers. In contrast to the original publication and traditional residual connections [37], we do not use a linear layer to equalize the dimensions but instead, concatenate the output of the previous layer to the input of the current layer. This is depicted by the green spheres with a “+” sign in Figure 2.

The downsampling path of our U-Net architecture consists of the MobileNet V2 model, which is used to extract low-level features from the input images. We freeze the weights of the MobileNet V2 model during training to prevent overfitting and improve the generalization of the model.

To ensure that the pretrained MobileNet V2 [26] model can process the input images, we first resize the images to multiples of its input shape, e.g.,  $224 \times N \times 224 \times N$  pixels. We use a convolutional layer with three filters and a stride that is adapted to the task-specific image dimensions to convert the output to  $224 \times 224$  pixels. This is depicted by the first right-banded orange box in Figure 2. Its kernel size is calculated as being the stride times two plus one. This initial convolutional

layer helps the model learn more complex features from the input images and allows it to be more flexible in handling different input sizes.

The upsampling path of our U-Net architecture consists of a series of upsampling blocks that gradually increase the resolution of the feature maps. Each upsampling block consists of a transposed convolutional layer, followed by batch normalization [35] and the leaky ReLU activation function ( $\alpha = 0.3$ ) [36]. The transposed convolutional layer has a kernel size of 3 and a stride of 1, while the number of filters in each layer gradually decreases as we move further up the upsampling path.

After the final upsampling block, we leave out the batch normalization. By an interpolating resizing, the output feature maps are reshaped to match the dimensions of the input images. We then apply a final convolutional layer with a kernel size of 3 and a number of filters equal to the number of desired output channels. If the number of output channels is greater than two, we apply a Softmax activation function to each pixel of the output map to obtain a probability map for each class. If there are only two output channels, we use a Sigmoid activation function for binary classification. This last layer produces the final segmentation output of our U-Net model.

Our custom U-Net model can also be quantized to reduce its memory footprint and increase inference speed, which is especially important for embedded sensor devices where resources are limited. We can use quantization techniques, such as posttraining quantization and quantization-aware training, to convert the model’s weights and activations to lower-precision formats.

Since the training data is sparse, we added two data augmentation methods: flip horizontal and random brightness. These augmentations can effectively double the size of the training data by creating a mirrored version of each image and randomly adjusting the brightness of the image, respectively. This can help reduce overfitting and improve the model’s generalization ability, as well as enhance its ability to accurately identify objects and boundaries under varying lighting conditions.

For the training itself, we used a batch size of 8 and trained for a maximum of 300 epochs, but early stopping

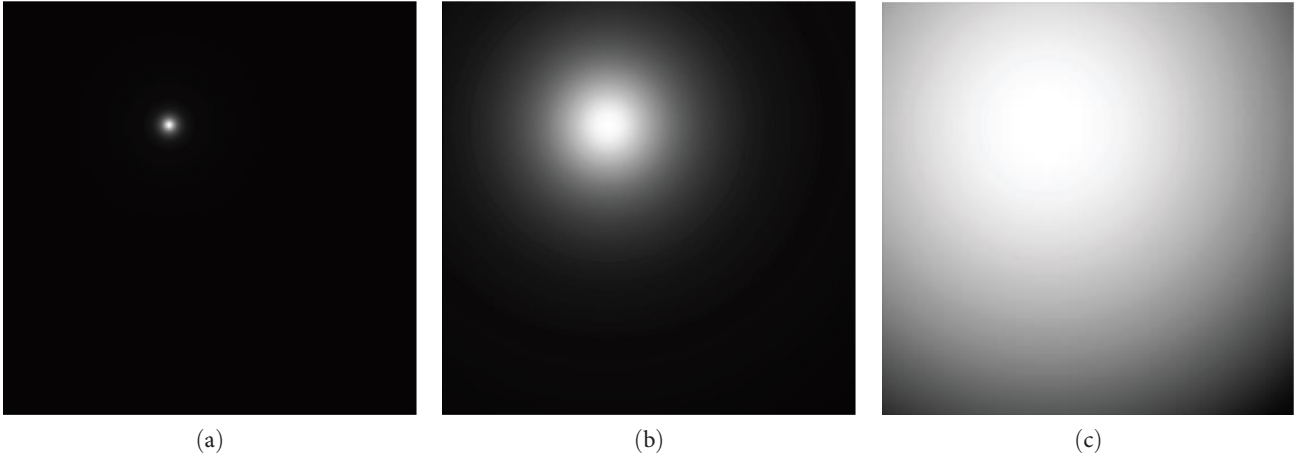


FIGURE 3: Annotated core position for different fall-off rates: (a) 0.1; (b)  $10^{-3}$ ; (c)  $10^{-5}$ .

ended the training early for all tested models. A binary cross-entropy loss is used.

Once the fingertip is segmented, the correction process usually involves fitting a box around the finger and correcting for the rotation of the box to ensure that the center line of the finger is vertical [38]. However, this approach is more sensitive to outliers in the segmentation, which can lead to inaccurate results. That is why we used our new approach, which involves calculating the slope of the center line by fitting a linear function over the middle of the points on the finger’s contour at the same height. This method reduces the effect of outliers in the contour detection and allows for accurate rotation of the image. This step can be seen as the first pose correction in the pose correction process.

**2.3.2. Reference Point Detection.** The next step in the pose correction process involves detecting a reference point that indicates the center of the fingerprint image. Around 98.5% of the population have a fingerprint type that contains at least one singular point called core [39], which is defined in ISO/IEC 19794-1:2011 3.33 as being the topmost point of the innermost ridge line. We assume that this core point is close to the fingerprint center [40] and use it as a reference point. When there is no core point in the presented fingerprint, we use the ridge orientation to manually set the reference point to the central fingerprint position for the testing data and exclude the sample for the automatically generated training data. When two cores are present, we choose the one closer to the image center.

Unlike traditional methods that rely on ridge orientation for core detection [41–43], we use once more our custom U-Net architecture to directly localize the reference point in the fingertip image. Additionally, the U-Net approach is better suited for images with low depth of field, where some parts of the image may not be in sharp focus. Similar to Section 3.2.2, we train and test in addition to our proposed model architecture also a Squeeze U-Net and EfficientUNet++. To train these U-Net models, a large dataset of fingerprint images was created. The dataset was constructed using a semi-automatic

annotation process, where the position of the core for the training was generated primarily by a commercial fingerprint matcher (IDKit from Innovatrix [44]) and partially manually corrected. During the correction procedure, the reference point annotation for fingerprint types without a core was set according to the generation of testing samples, i.e., we used the ridge orientation to manually set the reference point to the center of the fingerprint. For fingerprint samples with a core, the reference point is set to the core point position. The validation and test data were annotated entirely by hand to ensure high accuracy.

The dataset comprises a diverse range of fingerprint images, covering populations of different ages and skin types. This ensures that our model can accurately detect the core location in various scenarios. A detailed description of the dataset can be found in Section 3.2.1.

For the reference point detection task, we treated the problem as a segmentation problem. The model output is a probability map predicting the position of the reference point, i.e., the core for most fingerprints. We used  $448 \times 448$ -pixel images as input and normalized the pixel values. Only for the EfficientUNet++, we had to downscale the images to  $224 \times 224$  pixels. The models were trained for up to 300 epochs, but we used early stopping and monitored the validation loss to prevent overfitting. We used a batch size of 28 and a binary cross-entropy loss with two custom metrics: (i) the average Euclidean pixel distance from the annotated core and (ii) the average number of separate, detected regions. Our aim was to train a model with a low average pixel distance and an average number of detected regions of one.

The position of the reference point in each image was annotated using a circle that has a radial fall-off, which can be seen in Figure 3. This annotation method allowed for more leniency when the reference point was not detected perfectly, as the circle would still partially overlap with the actual annotated reference point location. The use of this annotation method also improved the behavior of the loss function during training. Specifically, the cross-entropy loss

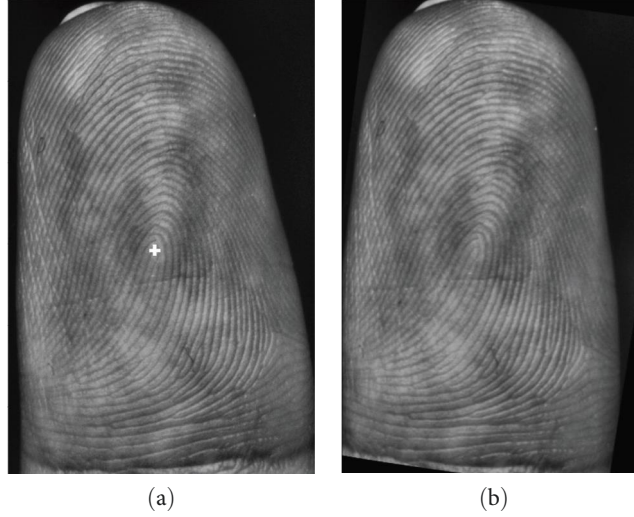


FIGURE 4: Detected core and following pose correction for a segmented and rotated ring finger with Kauba et al. [12] enhancement: (a) core; (b) correction.

used for training would otherwise treat a close miss and a completely wrong detection equally, as only the pixel values are compared.

In order to investigate the effect of different fall-off rates on the accuracy of the model, we run an ablation study with three experiments with the following fall-off rates: 0.1,  $10^{-3}$ , and  $10^{-5}$ . Figure 3 shows the annotated reference point position for all three cases.

**2.3.3. Lateral Rotation.** Once the fingerprint core has been detected, the system assumes that the finger has an elliptic cross-section. This assumption is based on the work of Tan and Kumar [16], who introduced a fingerprint model with an elliptical cross-section and used an ellipse with a major–minor axis ratio of 1.2 for lateral rotation correction.

To calculate the finger’s offset from the center, we use the detected core as a reference point. We assume that most of the finger’s interesting regions lie close to the core and that the core correlates with the intersection of the rotated minor axis of the ellipse with itself, i.e., the flat region of the finger.

Similar to the study of Tan and Kumar [16], we use the offset of the core position from the left finger edge ( $P'l$ ) visible in the rotated image to the right finger edge ( $P'r$ ) visible in the rotated image to implicitly define the rotation angle ( $\alpha$ , i.e., the negative viewing angle) as follows:

$$\begin{aligned} \eta &:= \arctan\left(\frac{\sin \alpha}{1.2 \cos \alpha}\right) \\ R &:= \frac{P'lP'c}{P'cP'r} = \frac{1 + \sin(\eta)}{1 - \sin(\eta)}. \end{aligned} \quad (1)$$

Equation (1) can be solved analytically, and for our choice of major-to-minor axis ratio of 1.2, we find as follows:

$$\begin{aligned} A &:= (R - 1)^2 \\ B &:= (R^2 - 2R + 1) \\ C &:= \frac{-10\sqrt{R(9R^2 + 7R + 9)}A + B(9R^2 + 32R + 9)}{9AB}, \\ \alpha &= 2 \operatorname{sign}(R - 1) \operatorname{atan}\sqrt{C}, \end{aligned} \quad (2)$$

which allows us to calculate an estimated rotation angle from the offset of the core position.

After finding the rotation angle, we use our ellipsoid model of the fingertip again. The width of the elliptic cross-section for each slice in the model is determined using the measured contour width, allowing us to rotate the finger to the correct orientation while taking into account its estimated 3D shape. We then remove any regions that would not have been visible to a front-facing camera. The result is a corrected and aligned image where the core sits in the center.

Figure 4 provides an example of the pose correction process. Figure 4 shows a ring finger that has been enhanced using Kauba’s enhancement technique, which is described in more detail in Section 2.2. In Figure 4(a), the input to the pose correction process is depicted, where the detected core is highlighted using a white cross. The position of the core, as well as the segmentation mask, are utilized to determine the viewing angle of the finger. Additionally, the segmentation mask is used to generate a 3D model of the finger shape, based on elliptic cross-sections with a major-to-minor axis ratio of 1.2. The final output of the pose correction process is illustrated in Figure 4(b).

**2.4. Unwarping.** In the postprocessing steps for contactless fingerprints, one critical task is unwarping the acquired images. Unlike contact-based fingerprint sensors, where the finger is in direct contact with the sensor surface, contactless

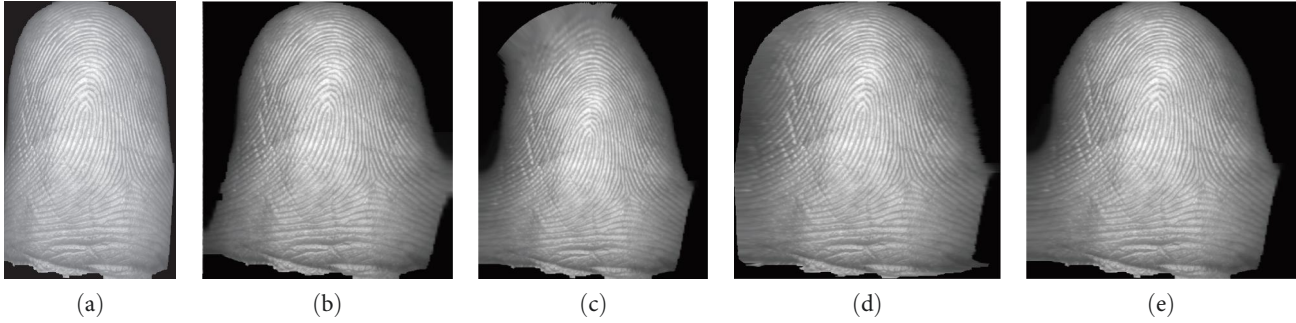


FIGURE 5: Effect of the different unwarpings on a middle finger: (a) input; (b) elliptic; (c) bidirectional; (d) circular-adaptive; (e) circular-fixed.

sensors capture the fingerprint image without physical contact. As a result, contactless images can be distorted due to the curvature of the fingertip. Unwarping is necessary to correct this and to align the fingerprint image to a standard reference frame. It enables accurate template comparison with contact-based fingerprints.

To correct for the distortion caused by the nonplanar surface of the fingertip in contactless fingerprint images, we explore three types of unwarping: circular, elliptic, and bidirectional, which are described in more detail in the study of Sollinger and Uhl [29]. The circular method assumes the finger has a cylindrical shape, while the elliptic method models the finger cross-section as an ellipse. The bidirectional method takes into account the curvature of the fingertip. This is accomplished by combining the circular unwarping with another circular unwarping for the rounded tip region of the finger.

To test the effectiveness of the circular method, we evaluate two different approaches: one that adapts the radius of the cylinder to the contour of the finger in each image and another one that uses a static radius based on the image width. The adaptive choice of the radius is made possible by the segmentation of the fingerprint image, which allows us to precisely identify the contour of the finger and adjust the finger width accordingly.

Figure 5 shows the effect of the unwarpings on a contactless middle finger recording.

### 3. Experiments

**3.1. Template Comparison.** We employed a template comparison procedure to ensure accurate imposter and genuine scores for our dataset and calculate the EER. Our approach involved selecting the top five sharpest contactless recordings for each finger from a particular recording session based on the scoring mechanism in the preprocessing pipeline (see Section 2.2). Using IDKit from Innovatrix [44], we calculated templates for each of these five finger images and compared them to all contact-based recording templates. This methodology enabled us to account for variations in the recording quality, ensuring the reliability of our results.

By selecting only the top five sharpest images for each finger, we not only improved the reliability of our results but also significantly reduced the number of required comparisons. This reduction in required comparisons is especially

crucial for larger datasets where the number of comparisons can become prohibitively large, leading to longer computation times. Our template comparison procedure thus struck a balance between result quality and efficiency.

To assess the variability of our results, we randomly sampled 80% of the genuine scores and 80% of the imposter scores and calculated the EER. This process of subsampling and calculating the EER was repeated 100 times, and we report the mean and standard deviation of the results.

### 3.2. Data

**3.2.1. Core Detection Dataset.** To train and validate our core detection U-Net algorithm, we created a dataset from a subset of the dataset used in our previous study [22]. We utilized a semiautomatic annotation approach that involved both a commercial fingerprint matcher and manual correction. We utilized the IDKit software from Innovatrix [44] to locate the core position in contactless fingerprint images. The matcher was run over both enhanced and original images to ensure accuracy. Although the matcher was designed for contact-based fingerprints, this semiautomatic approach allowed us to annotate a large amount of training data efficiently. However, to ensure higher quality annotations, we manually corrected parts of our training data to minimize the impact of annotation errors on the neural network’s performance.

For our validation and test data, we chose to manually create the entire dataset to ensure the highest possible quality. This was a time-consuming process, but it allowed us to have complete control over the quality and accuracy of the annotations.

In addition to the semiautomatic annotation approach and manual correction, we took an additional step to ensure the quality of our dataset: we filtered the images beforehand, only using images that had a sharpness over a finger-dependent threshold. Specifically, we found in extensive empirical experimentation that different fingers had different optimal thresholds for sharpness. For example, we found that the optimal sharpness threshold for the index and little fingers was 0.16, while the optimal threshold for the middle and ring fingers was 0.22. By tailoring our image filtering approach to the specific finger being analyzed, we were able to ensure that our dataset only contained high-quality images that were well-suited for training and validating our neural network.



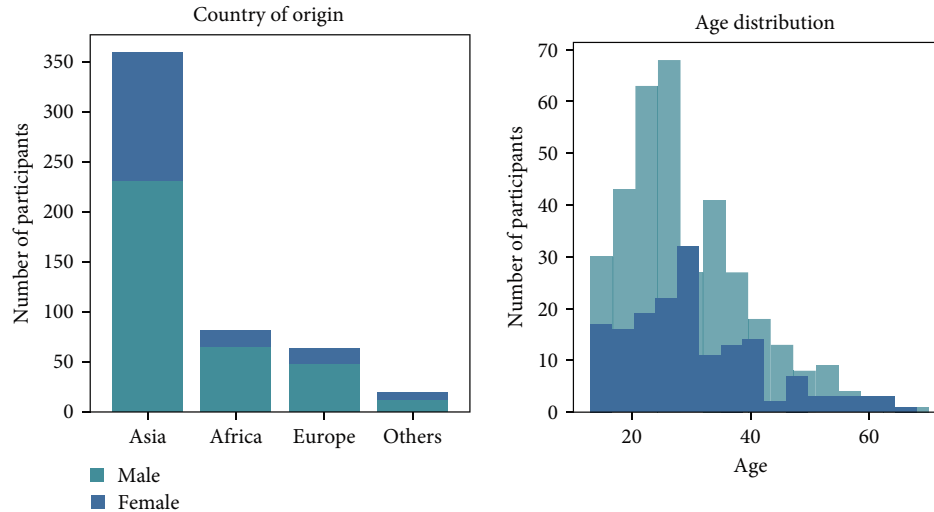


FIGURE 6: Age and country of origin distribution separated by gender for the core detection dataset.

In total, we were able to use 119,993 images from 535 different users to create our dataset.

As shown in Figure 6, our participant population was diverse in terms of their country of origin, age, and gender. While our dataset contained more male users (356) than female users (166), we still believe that the overall diversity of our user population allowed us to capture a wide range of fingerprints and ensured that our dataset was representative of a broad range of users. Note that one participant has an unidentified gender. In order to strengthen the anonymization of the data and to avoid identification using a quasi-identifier, we excluded this participant from Figure 6.

**3.2.2. Finger Segmentation Dataset.** To train a neural network for segmenting fingers in contactless fingerprint images, a carefully curated and annotated dataset is crucial. However, manual annotation of each image can be a time-consuming process. To speed up the annotation process, we utilized an existing finger segmentation dataset with preannotated images for a different recording setup [12] and combined it with newly created, manual annotations from our dataset collected in the study of Weissenfeld et al. [22]. We carefully cropped the preexisting dataset to ensure consistency with our new dataset in terms of image ratios.

This hybrid approach allowed us to significantly reduce the overall workload required to create our dataset while also potentially increasing the diversity of images in our training set, which could, in turn, improve the performance of our model.

To ensure that the model is trained and evaluated on data from previously unseen users, we carefully split our dataset into train, validation, and test sets on a user basis. This means that no user appears in two sets, which encourages the model to learn more generalizable features that can be applied to new users. Moreover, this approach provides a more realistic assessment of the model’s performance in real-world scenarios where the model must perform well on users it has never encountered before.

Our final dataset contained a total of 5,828 images, 5,146 of which were preannotated, and 1,457 and 1,822 images were used for validation and testing, respectively.

**3.2.3. Template Comparison Dataset.** To assess the efficacy of our developed methods, we conducted a random selection of a representative subset of 78 participants from our previous study [22], which contains 37,162 contactless fingerprint images. This was necessary due to time and computational constraints. Both contactless and contact-based fingerprint recordings were obtained from a law enforcement agency, providing a realistic sample of real-world scenarios. The contactless recordings were acquired using our sensor prototype (see Section 2.1), while the contact-based recordings were collected using the commercially available Idemia TP 5300 (<https://www.idemia.com/palmprint-scanner>). The contact-based recordings consisted of rolled fingerprint images with a resolution of 1,000 DPI.

To ensure the quality of our contactless dataset, we again applied the sharpness-filtering approach from Section 3.2.1. By customizing our image filtering approach to the specific finger under analysis, we could minimize the number of blurred images included in our dataset.

**3.2.4. Demographics.** For calculating conclusive match performances of contact-based and contactless fingerprint enhancement methods, it’s important to have a diverse dataset. This allows for evaluating models’ robustness to variations in lighting, image quality, and finger placement, among other factors, and ensures that results are representative of a wide range of people and not biased toward a particular population.

Figure 7 illustrates only 75 of the 78 participants, because we do not know the gender and origin of three participants. Of the 75 participants, we had 42 identified as male and 33 as female. The distribution of participants’ countries of origin was diverse, with Asia being the most represented continent (24 male, 26 female), followed by Africa (10 male, no female) and Europe (five male, no female).

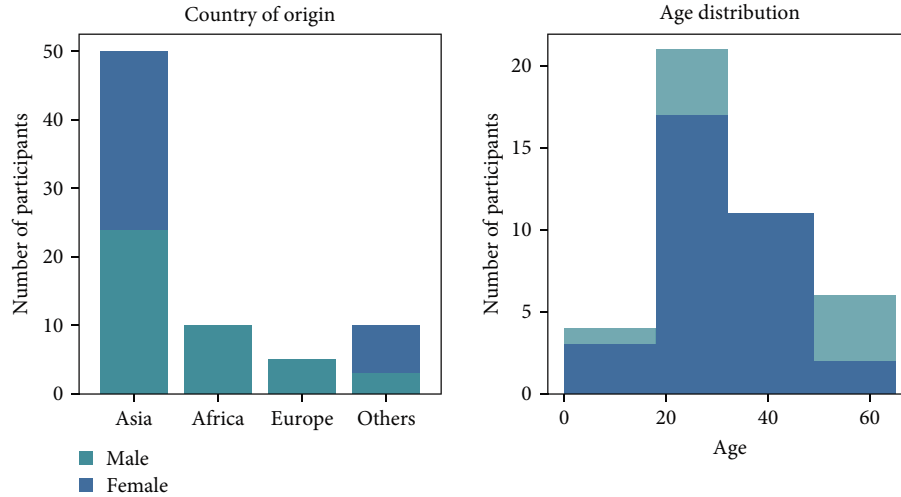


FIGURE 7: Age and country of origin distribution separated by gender for the template comparison of contact-based and contactless images dataset.

TABLE 1: Mean NFIQ 2 scores with standard deviation for the contact-based recordings collected using the Idemia TP 5300 sensor as well as the contactless recordings after the preprocessing stage.

Recording mode	All	Thumb	Index	Middle	Ring	Little
Contact-based	$50.0 \pm 18.9$	$55.4 \pm 13.4$	$54.6 \pm 18.2$	$50.6 \pm 16.5$	$46.0 \pm 18.8$	$43.5 \pm 23.3$
Contactless	$32.3 \pm 13.5$	$38.8 \pm 10.1$	$31.7 \pm 13.5$	$35.7 \pm 12.8$	$30.9 \pm 13.2$	$23.6 \pm 12.5$

The majority of Asian participants were from Syria (36), followed by Afghanistan (five) and China (three), and we had participants from a total of eight different Asian countries. The majority of African participants were from Somalia (four), followed by Nigeria (two), and we had participants from a total of nine different African countries. The majority of European participants were from Ukraine (three), followed by Türkiye (two), and we had participants from a total of five different European countries.

**3.2.5. Quality Analysis.** Participants in the study had their fingerprints captured by the law enforcement agency using both an optical fingerprint scanner, specifically the Idemia TP 5300 which creates 1,000 DPI images, as well as our contactless fingerprint sensor. The collected contact-based recordings are rolled fingerprints. To assess the quality of the contact-based dataset, the NFIQ 2 score was calculated for each fingerprint image, and the results are presented in Table 1. Developed by NIST, the NFIQ 2 score is a metric that scores the quality of a recorded fingerprint image on a scale from 0 to 100, where a higher score indicates better quality [45]. Although designed for contact-based images with a DPI of 500, it can also provide valuable insights for higher resolution images and also partially for contactless images [46]. For comparison, the NFIQ 2 scores of the fingerprint images recorded with our contactless fingerprint sensor can also be seen in Table 1. These scores are calculated for the enhanced images resulting from the live preprocessing pipeline, which includes all images above the sharpness threshold and not just the five sharpest images. It is important to note that the NFIQ 2 score could not be calculated for

3,464 images mostly because the recognized fingerprint area was too small.

We motivate the use of NFIQ 2 for analyzing the dataset quality as a means to compare fingerprint feature quality for different databases, although the NFIQ 2 score alone is not indicative of template comparison performance, given, for example, as EER [38].

Our analysis showed that the contact-based recordings using the Idemia TP 5300 sensor produced higher mean NFIQ 2 scores than the contactless sensor. Specifically, the mean NFIQ 2 score for all fingers was  $50.0 \pm 18.9$  for the Idemia TP 5300 sensor, compared to  $32.3 \pm 13.5$  for the contactless sensor.

We also examined the distribution of NFIQ 2 scores for each sensor. Our analysis revealed that the contact-based recordings had a wider range of scores, with a minimum of 2 and a maximum of 93, compared to the contactless sensor, with a minimum of 0 and a maximum of 91. Additionally, the contact-based recordings had a higher interquartile range (IQR), as evidenced by the  $[q0.25, q0.75]$  range of  $[37, 63]$ , compared to  $[22, 42]$  for the contactless sensor.

Comparing the NFIQ 2 scores of our database with scores from publicly available sources, we find the scores of the contact-based part of our database ( $50 \pm 18$ ) are high when compared to MCYT [47] (dp:  $38 \pm 15$ , pb:  $33 \pm 14$ ), FVC06 [48] ( $36 \pm 9$ ) or PolyU [6] ( $47 \pm 12$ ) and the scores of the contactless images ( $32 \pm 14$ ) are low. [38]

**3.2.6. Cleaning.** During our reanalysis of the mapping from contactless to contact-based recordings, we discovered that finger identification based on the longest finger was error-

prone, resulting in inaccuracies. This was especially prominent for certain users, where the identification would change for a handful of frames in the recording stack. To address this issue, we developed a three-level correction and detection framework to improve the accuracy of the dataset.

Our examination of the contact-based and contactless fingerprint template comparison dataset revealed two indications of finger identification errors: first, the order of the identified names changed relative to their detection indices. Second, the genuine score for a finger was separated into two separate groups during one recording, with the highest imposter score clearly suggesting a match with the counterpart finger (e.g., the index finger with the little finger from the same hand).

In the first level of correction, we identified recordings where finger identification was very likely to have been switched during the recording process. To do so, we looked for three indications: first, all images considered had their naming order changed; second, the genuine scores separated into two distinct groups, with the average genuine score of the first group at least double the genuine score of the inconsistent naming group; and third, the highest imposter match was the match with the same hand of the same user but another finger. We also ensured that each image in the inconsistent naming group had a genuine score of below 40 and an imposter score of above 70 with the corresponding counterpart finger. Applying this level of correction allowed us to identify and correct 80 finger identification errors in the dataset, thus significantly improving the accuracy of our analysis.

In the second level of correction, we identified recordings that were highly likely to have been misidentified. We applied the same criterion of splitting into two genuine score groups, with the average genuine score of the correctly identified group being one and a half times the average genuine score of the misidentified group. Additionally, the highest imposter match score was for the same hand of the same user but with its counterpart finger. However, at this level, we did not detect any incoherent naming order. We were able to correct 16 images in level 2.

Finally, in the last level of correction, we examined images that exhibited naming irregularities but did not meet the imposter score criteria. As before, the mean genuine score of the correctly identified group was one and a half times the average score of the misidentified group. In level 3, we identified and corrected six images.

In addition to recordings with a correctable error in the pipeline, we also excluded 25 individual fingers, where the assignment between contactless and contact-based recordings could not be made.

## 4. Experimental Results

*4.1. Segmentation.* In the task of segmenting the finger from the background, our model achieved the highest accuracy value of 0.979, which can be seen in Table 2. The high-accuracy score suggests that the model predicted the correct class for nearly 98% of the pixels in the image. Moreover, we

TABLE 2: Accuracy and mean intersection over union (MIoU) metrics of the image segmentation approach were evaluated on the test set in addition to the inference speed (Inf. speed) measured running on a GTX 3090.

Model	Accuracy (%)	MIoU (%)	Inf. speed (s)
Custom	<b>97.85</b>	<b>91.38</b>	<b>0.07</b>
Efficient	87.56	49.72	0.12
Squeeze	96.42	85.73	0.08

*Note:* Bold signifies the best performing values.

evaluated the segmentation performance using the commonly used mean intersection over union (MIoU) metric, and the model achieved again the highest score of 0.914 on the test set. This result indicates that the model accurately identified and separated the finger from the background in the image with a high level of precision, achieving a significant overlap between the predicted and ground truth segmentation maps. Additionally, all three models are in the same ballpark regarding inference speed on a Nvidia GTX 3090, with our custom U-Net having a slight speed advantage. The inference speed was calculated by averaging over 50 predictions.

These results demonstrate the effectiveness of our model in accurately segmenting the finger from the background, which is a critical step in many computer vision applications. Furthermore, the potential of the model to perform well in other related segmentation tasks is suggested by its performance in this task.

To visually demonstrate the accuracy of our segmentation model, Figure 8 shows a sample segmentation result. The image on the left is the original input image, and the image on the right is the corresponding segmentation map produced by our model.

*4.2. Reference Point Detection.* Table 3 displays the average distance for a segmentation fall-off rate of  $10^{-5}$  as well as the inference speed running on a GTX 3090 for all three tested models. Similar to segmentation, our custom U-Net model performs the best with an average distance of 9.49 compared to 9.63 for the Squeeze U-Net and 11.03 for the EfficientU-Net++. Furthermore, all three models are in the same ballpark regarding inference speed, with the Squeeze U-Net being the fastest, with 100 ms for an average prediction.

As an ablation study, we tested the impact different fall-off rates have on the performance. The fall-off rate determines the size of the region in which the gradient can be updated based on the loss function. We selected the best-performing model at a fall-off rate of  $10^{-5}$  and restarted training with two other fall-off rate settings. Once with a fall-off rate of  $10^{-3}$  and once with a fall-off rate of  $10^{-1}$ . The results are presented in Table 4. Our analysis indicates that a slower fall-off rate ( $10^{-5}$ ) yielded the best average distance score of 9.49 compared to faster rates of  $10^{-3}$  and  $10^{-1}$ , with scores of 23.46 and 38.47, respectively.

Additionally, the evaluation of the average number of separated regions reveals that all three models generate a single connected prediction region, with an average of one region per model. This observation indicates that the model

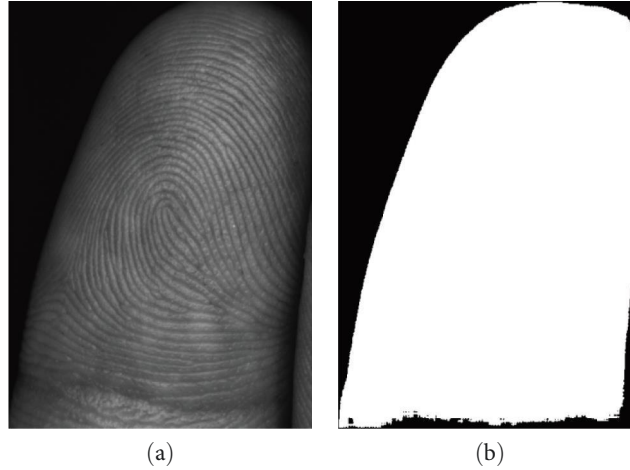


FIGURE 8: Segmentation of a nonenhanced index finger: (a) input; (b) mask.

TABLE 3: Average distance (Avg. distance) and inference speed (Inf. speed) running on a GTX 3090 with a fall-off rate of  $10^{-5}$  were measured for all three models.

Model	Avg. distance (px)	Inf. speed (s)
Custom	<b>9.49</b>	0.16
Efficient	11.03	0.12
Squeeze	9.63	<b>0.10</b>

Note: Bold signifies the best performing values.

TABLE 4: Average distance and average number of separated regions for different fall-off rates for the core detection problem for our custom U-Net.

Fall-off rate	Avg. distance	Avg. regions
$10^{-5}$	<b>9.49</b>	1
$10^{-3}$	23.46	1
$10^{-1}$	38.47	1

Note: Bold signify the best performing value.

successfully identifies the entire reference point region as a single entity without generating any false positive regions.

In conclusion, our findings suggest that the custom U-Net model introduced in this paper is well-suited for both image segmentation as well as reference point detection. Additionally, we found that the model’s performance is significantly improved by utilizing a slower fall-off rate, which enables a larger gradient update region. Moreover, the model generates a single connected prediction region with high accuracy, indicating its robustness and effectiveness for the given task.

**4.3. Recognition Accuracy.** Table 5 displays the EERs were obtained for contactless to contact-based fingerprint template comparison using various unwarping techniques. We calculated the EERs following the same approach described in Section 3.1, whereby we randomly subsampled 80% of the genuine scores and 80% of the imposter scores to calculate the EER. This process was repeated 100 times to obtain a mean and standard deviation of the EERs, which are reported in Table 5.

Table 5 includes the EERs for all fingers combined as well as for each individual finger (Thumb, Index, Middle, Ring, and Little). The first row, labeled as “B” represents the EERs for the contactless to contact-based template comparison without any postprocessing. The following rows represent the EERs for various postprocessing techniques, including circular with fixed (CF) and adaptive (CA) finger width, elliptic unwarping (EI), and bidirectional unwarping (Bi) with a fixed finger width. The rows after the middle row show the EERs for the same postprocessing techniques applied to images after pose correction (PC).

The best EERs of all improvement combinations for each finger are highlighted in bold, and the best EERs for one improvement combination are italicized. For example, in the middle finger category, the best EER is achieved with both the pose correction and circular unwarping with adaptive finger width, as well as with elliptic unwarping only.

The largest improvement in average EERs for the non-pose corrected enhancements compared to the baseline is given by elliptic unwarping. Here, the difference in score is 0.22, which is an improvement of 14.0%. For the pose corrected case, the largest, average improvement is found using bidirectional unwarping. Here, the difference in score is 0.10, which is an improvement of 6.4%. However, the results vary strongly on a finger basis.

This allowed us to combine different unwarpings and the inclusion of pose correction for each finger, which can be seen in Table 6. Specifically, pose correction and elliptic unwarping were used for the thumbs, elliptic unwarping was used for the index and middle fingers, no pose correction or unwarping was applied to the ring finger, and bidirectional unwarping was used for the little finger. This finger-wise combined enhancement method reached an improvement of 0.58 in comparison to the baseline. This is an improvement of 36.9%.

The results show that the elliptic enhancement technique performed best for the thumb, index, and middle finger, while other techniques were more effective for the ring and little fingers. Furthermore, the influence of pose correction as a postprocessing step varied strongly on a finger-to-finger



TABLE 5: Equal error rates (EERs) are given in percentages for contactless to contact-based template comparison.

Enhancement	All (%)	Thumb (%)	Index (%)	Middle (%)	Ring (%)	Little (%)
B	1.57 ± 0.11	2.38 ± 0.58	1.12 ± 0.13	1.28 ± 0.21	<b>0.85 ± 0.23</b>	2.97 ± 0.33
CF	1.46 ± 0.08	<i>0.47 ± 0.02</i>	1.39 ± 0.14	0.89 ± 0.20	1.12 ± 0.19	3.04 ± 0.34
CA	3.30 ± 0.19	1.51 ± 0.20	1.69 ± 0.57	2.28 ± 0.35	<i>0.85 ± 0.50</i>	8.67 ± 0.84
El	<b>1.35 ± 0.10</b>	0.82 ± 0.22	<b>0.94 ± 0.16</b>	<i>0.66 ± 0.15</i>	0.98 ± 0.15	3.43 ± 0.34
Bi	1.37 ± 0.10	1.52 ± 0.22	1.33 ± 0.31	<i>1.00 ± 0.18</i>	1.31 ± 0.17	<b>1.77 ± 0.15</b>
PC	1.61 ± 0.10	<i>0.78 ± 0.40</i>	2.03 ± 0.39	1.68 ± 0.37	0.94 ± 0.22	2.04 ± 0.21
PC + CF	1.62 ± 0.08	<i>0.68 ± 0.05</i>	2.00 ± 0.19	0.80 ± 0.19	1.45 ± 0.21	2.15 ± 0.13
PC + CA	2.27 ± 0.14	<i>0.52 ± 0.06</i>	3.02 ± 0.19	<b>0.63 ± 0.10</b>	2.16 ± 0.28	3.12 ± 0.14
PC + El	1.48 ± 0.11	<b>0.41 ± 0.16</b>	1.45 ± 0.26	0.89 ± 0.22	1.06 ± 0.16	2.60 ± 0.15
PC + Bi	1.47 ± 0.08	<i>0.57 ± 0.16</i>	1.76 ± 0.29	0.79 ± 0.10	1.96 ± 0.18	1.98 ± 0.20

All comparisons are made using Kauba et al. [12] enhancement for the contactless images. Depicted are the mean and standard deviation of EERs for 100 random subsamples containing 80% of the template comparison scores. The baseline was calculated for the enhanced images after the preprocessing pipeline without any further postprocessing steps. Results for the pose correction (PC) run were calculated by applying pose correction as a postprocessing step to the results of the baseline (B). For the next runs, additional unwarping was added: circular unwarping with a fixed finger width (CF), circular unwarping with an adaptive finger width (CA), elliptic unwarping with a fixed finger width (El), and finally, bidirectional unwarping with a fixed finger width (Bi). Bold entries highlight the best score for this column and italicized entries the best score for the row. Ties are broken by smaller standard deviations.

TABLE 6: Average equal error rate for a finger-wise combination of enhancement methods.

EER	Thumb	Index	Middle	Ring	Little
0.99 ± 0.07	P.C. + Elliptic	Elliptic	Elliptic	None	Bidirectional

TABLE 7: NFIQ 2 scores for the contactless recordings after the postprocessing stage with bidirectional unwarping.

—	All	Thumb	Index	Middle	Ring	Little
Mean ± Std.	33.5 ± 10.9	41.6 ± 7.0	33.0 ± 10.8	35.0 ± 10.2	31.3 ± 10.8	27.2 ± 9.8
[Min, Max]	[0, 71]	[10, 71]	[1, 71]	[2, 64]	[1, 60]	[0, 69]
[ $q_{0.25}$ , $q_{0.75}$ ]	[26, 42]	[38, 46]	[27, 41]	[29, 42]	[24, 39]	[20, 34]
[ $q_{0.10}$ , $q_{0.90}$ ]	[18, 46]	[32, 49]	[17, 45]	[20, 46]	[16, 44]	[14, 40]

basis. Overall, the results demonstrate that the combination of different enhancement techniques can significantly improve the accuracy of contactless to contact-based fingerprint template comparison, with some techniques being more effective for certain fingers.

**4.4. Image Quality Analysis.** Table 7 presents the NFIQ 2 scores for the contactless recordings exemplarily enhanced with bidirectional unwarping and Kauba’s enhancement.

Overall, the mean NFIQ 2 score for all fingers combined was 33.5, with a standard deviation of 10.9. The mean NFIQ 2 scores for individual fingers ranged from  $27.2 \pm 9.8$  for the little finger to  $41.6 \pm 7.0$  for the thumb. The minimum and maximum NFIQ 2 scores for all fingers were 0 and 71, respectively. The IQR was 16, indicating that 50% of the NFIQ 2 scores were between 26 and 42. The range of the NFIQ 2 scores from the 10th to the 90th percentile ( $q_{0.10}$ – $q_{0.90}$ ) was 28, indicating that the majority of the scores were distributed between 18 and 46.

Compared to the NFIQ 2 scores of Table 1, which contains the results for the images without pose correction and unwarping, we had an improvement of average score from

32.3 to 33.5, which is an increase of 3.72%. Also, the volatility in the form of standard deviation was reduced from 13.5 to 10.9, which is a reduction of 19.26%. For the thumbs, the improvement was above the average with a change from 38.8 to 41.6, which is an increase of 7.22%. Here, the reduction in standard deviation was even stronger, from 10.1 to 7.0, which is a decrease of 30.69%. For the other fingers, the relative improvements of the average NFIQ 2 scores were 4.1% (index),  $-1.96\%$  (middle), 1.29% (ring), and 15.25% (little). Note that the middle finger was the only finger where pose correction and unwarping led to a reduction of the NFIQ 2 score. The reduction in variability of the standard deviation was given for all fingers. The relative reductions were 20.00% (index), 20.31% (middle), 18.18% (ring), and 21.60% (little).

## 5. Discussion

**5.1. Segmentation and Reference Point Detection.** The results of the segmentation task demonstrate that our custom U-Net model accurately predicted the class for nearly 98% of the pixels in the image and effectively minimized the difference

between the predicted and ground truth segmentation maps during training. The model achieved an MIoU score of 0.914, indicating its high precision in identifying and separating the finger from the background in the image. The significant overlap between the predicted and ground truth segmentation maps further confirms the model's effectiveness in accurately segmenting the finger from the background. Furthermore, for the task of reference point detection, our custom U-Net model performed better than both the Squeeze U-Net and EfficientUNet++ U-Net adaptations. This achievement is crucial for many computer vision applications, indicating the model's potential for performing well in other related segmentation tasks.

From the other two tested models, Squeeze U-Net also performed well on both tasks, and further research could focus on the possibility of quantizing this model, allowing it to run on embedded inference hardware. On the other hand, the performance of EfficientUNet++ could not hold up with the performance of the other two models. We think that this is mostly an effect of the different image resolutions used to train and test the models, since the EfficientUNet++ was only able to work with images of size  $224 \times 224$  pixels.

These results suggest that it might be beneficial to combine both tasks into one step. This means that both the finger contour, as well as the reference point position can be found by a single model. One approach could be to use two separate the reference point detecting and finger detection heads or to train on a multiclass segmentation problem where one label represents the finger contour, and one represents the reference point position. Both approaches have their own merits. Two separate heads offer the best model performance, as the final segmentation steps remain separate for both problems. However, this means that similar calculations need to be repeated. The approach with a multiclass segmentation dataset could introduce a larger efficiency gain, as it shares not only the down-sampling stack of layers but also the last segmentation layers. The decision to choose one approach over the other should take into account the hardware constraints of the sensing environment and require further research to determine the optimal balance.

**5.2. Pose Correction.** The results of the study on the different postprocessing techniques for contactless to contact-based fingerprint template comparison are quite interesting. The study showed that combining different enhancement techniques, such as pose correction, elliptic, and bidirectional unwarping, can significantly improve the accuracy of the fingerprint template comparison system. Moreover, the results varied on a finger-to-finger basis, with certain postprocessing techniques being more effective for certain fingers.

One possible reason is that the shape and size of the fingers differ from one individual to another, which can affect the quality of the captured images, as well as the functioning of the following enhancement steps. Moreover, the influence of pose correction on the template comparison accuracy varied on a finger-to-finger basis. The reason for this can be attributed to the different shapes of the fingers and their orientation during the scanning process. In some

cases, pose correction may have a significant impact on improving the template comparison accuracy, while in other cases, the improvement may be minimal or negligible. Another factor that may contribute to the variations in results is the number of participants. A larger cohort reduces the statistical fluctuations.

The study also showed that the largest improvement in average EERs for the nonpose corrected enhancements was given by elliptic unwarping. The largest, average improvement for the pose-corrected case was found using bidirectional unwarping. However, the results vary strongly on a finger basis, which allowed for the combination of different unwarpings and the inclusion of pose correction for each finger. A finger-wise combined enhancement method, shown in Table 6, reached an average improvement of 0.58 points in comparison to the baseline. This is a relative decrease in EER of 36.9%.

Moreover, since elliptic unwarping outperformed circular unwarping in both pose-corrected and nonpose-corrected cases, it may be worthwhile to explore the possibility of extending the bidirectional unwarping method to work with ellipsoids in future research.

Overall, the results highlight the importance of using a combination of different enhancement techniques, tailored to the specific characteristics of each finger, to achieve the best possible accuracy in contactless to contact-based fingerprint template comparison. The use of deep-learning neural networks and advanced unwarping techniques can help to overcome the limitations of contactless fingerprint scanning and improve the overall reliability and security of biometric systems. Future research could focus on a detailed analysis of deviations in the finger shape from the assumed ellipsoid with a major-to-minor axis ratio of 1.2.

## 6. Conclusion

The aim of our study was to investigate whether implementing novel steps in the processing pipeline for contactless fingerprint sensors, specifically pose correction and unwarping, would improve the accuracy of the system as measured by the NFIQ 2 score and EERs. In order to implement pose correction, we developed a novel deep-learning network architecture closely related to U-Net [23]. It was able to solve both subtasks for pose correction: Image segmentation (MIoU of 91.4%) and reference point localization (average distance of 9.49) and outperformed both Squeeze U-Net as well as EfficientUNet++, which are both contemporary image segmentation networks. Additionally, the structure of the model allows for quantization using the established TensorFlow Lite framework.

We assessed the efficacy of our novel approach by calculating the EERs for 78 participants, totaling 37,162 recorded contactless fingerprint images. In conclusion of the results, we found that the changes in template comparison scores were highly finger-specific, indicating that the effectiveness of the extensions varied based on the characteristics of individual fingerprints. A finger-wise combination of different extension methods leads to an average, relative, finger-wise

decrease of EER of 36.9% compared to the baseline. The analysis of pose correction and bidirectional unwarping combined showed a relative increase of NFIQ 2 scores of 3.72% averaged over all fingers and a relative decrease of 6.4% in EER compared to the baseline, indicating that the extensions enhanced the performance of the system.

Overall, our findings suggest that pose correction and unwarping can be valuable additions to the processing pipeline for contactless fingerprint sensors, potentially leading to improved accuracy and efficiency in fingerprint recognition systems. This finding has important implications for applications that rely on accurate fingerprint recognition, such as forensic investigations and biometric authentication systems. Further studies are needed to explore the generalizability of these results to different populations and recording conditions, as well as to investigate the variability of finger shapes.

### Data Availability

The contactless and contact-based fingerprint data used to support the findings of this study were supplied by the Ministry of the Interior under license and so cannot be made freely available. Requests for access to these data should be made to the Bundeskriminalamt, BMI-II-BK-SPOC@bmi.gv.at.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

This work was partially supported by the AIT Strategic Research Program 2022. We gratefully acknowledge the continuous support of BMI during the recording sessions and the whole study. The research was performed as a part of employment by the Austrian Institute of Technology.

### References

- [1] S. M. Stigler, "Galton and identification by fingerprints," *Genetics*, vol. 140, no. 3, pp. 857–860, 1995.
- [2] Y. Han, C. Ryu, J. Moon, H. Kim, and H. Choi, "A study on evaluating the uniqueness of fingerprints using statistical analysis," in *Information Security and Cryptology—ICISC 2004*, vol. 3506 of *Lecture Notes in Computer Science*, pp. 467–477, Springer, Berlin, Heidelberg, 2005.
- [3] S. S. Wood and C. L. Wilson, *Studies of Plain-to-Rolled Fingerprint Matching Using the NIST Algorithmic Test Bed (ATB)*, National Institute of Standards and Technology, Gaithersburg, MD, 2004.
- [4] R. Kessler, O. Henniger, and C. Busch, "Fingerprints, forever young?" in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8647–8654, IEEE, Milan, Italy.
- [5] K. Okerefor, I. Ekong, I. O. Markson, and K. Enwere, "Fingerprint biometric system hygiene and the risk of COVID-19 transmission," *JMIR Biomedical Engineering*, vol. 5, no. 1, Article ID e19623, 2020.
- [6] C. Lin and A. Kumar, "Matching contactless and contact-based conventional fingerprint images for biometrics identification," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2008–2021, 2018.
- [7] J. Priesnitz, C. Rathgeb, N. Buchmann, C. Busch, and M. Margraf, "An overview of touchless 2D fingerprint recognition," *EURASIP Journal on Image and Video Processing*, vol. 2021, Article ID 8, 2021.
- [8] X. Liu, M. Pedersen, C. Charrier, F. A. Cheikh, and P. Bours, "An improved 3-step contactless fingerprint image enhancement approach for minutiae detection," in *2016 6th European Workshop on Visual Information Processing (EUVIP)*, pp. 1–6, IEEE, Marseille, France, 2016.
- [9] Y. Tang, L. Jiang, Y. Hou, and R. Wang, "Contactless fingerprint image enhancement algorithm based on Hessian matrix and STFT," in *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)*, pp. 156–160, IEEE, Wuhan, China, 2017.
- [10] S. A. Grosz, J. J. Engelsma, E. Liu, and A. K. Jain, "C2CL: contact to contactless fingerprint matching," 2021, arXiv: 2104.02811[cs, eess]. Available from: [Accessed on: 2022 Oct 5], <http://arxiv.org/abs/2104.02811>.
- [11] P. Schuch, S. Schulz, and C. Busch, "Survey on the impact of fingerprint image enhancement," *IET Biometrics*, vol. 7, no. 2, pp. 102–115, 2018.
- [12] C. Kauba, D. Söllinger, S. Kirchgasser et al., "Towards using police officers' business smartphones for contactless fingerprint acquisition and enabling fingerprint comparison against contact-based datasets," *Sensors*, vol. 21, no. 7, Article ID 2248, 2021.
- [13] R. Donida Labati, A. Genovese, V. Piuri, and F. Scotti, "Toward unconstrained fingerprint recognition: a fully touchless 3-D system based on two views on the move," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, pp. 202–219, 2016.
- [14] R. D. Labati, A. Genovese, V. Piuri, and F. Scotti, "Contactless fingerprint recognition: a neural approach for perspective and rotation effects reduction," in *2013 IEEE Symposium on Computational Intelligence in Biometrics and Identity Management (CIBIM)*, pp. 22–30, IEEE, Singapore, 2013.
- [15] A. Ross and R. Nadgir, "A calibration model for fingerprint sensor interoperability," in *Proceedings of SPIE Conference on Biometric Technology for Human Identification III*, vol. 6202, pp. 6202B-1–6202B-12, Orlando, USA, 2006.
- [16] H. Tan and A. Kumar, "Towards more accurate contactless fingerprint minutiae extraction and pose-invariant matching," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3924–3927, 2020.
- [17] J. J. Engelsma, K. Cao, and A. K. Jain, "Learning a fixed-length fingerprint representation," 2019, arXiv: 1909.09901[cs]. Available from: <http://arxiv.org/abs/1909.09901>.
- [18] D. Deb, T. Chugh, J. Engelsma et al., "Matching fingerphotos to slap fingerprint images," 2018.
- [19] J. Priesnitz, C. Rathgeb, N. Buchmann et al., "Deep learning-based semantic segmentation for touchless fingerprint recognition," in *Pattern Recognition. ICPR International Workshops and Challenges, Lecture Notes in Computer Science*, pp. 154–168, Springer International Publishing, Cham, 2021.
- [20] Y. Shi, Z. Zhang, S. Liu, and M. Liu, "Towards more accurate matching of contactless fingerprints with a deep geometric graph convolutional network," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 5, no. 1, pp. 29–38, 2023.
- [21] J. Libert, J. Grantham, B. Bandini et al., *Guidance for Evaluating Contactless Fingerprint Acquisition Devices*,

- National Institute of Standards and Technology, Gaithersburg, MD, 2018.
- [22] A. Weissenfeld, R. Schmid, B. Kohn, B. Strobl, and G. F. Dominguez, "Case study of the acquisition of contactless fingerprints in a real police setting," in *2022 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, IEEE, Darmstadt, Germany, 2022.
- [23] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds., vol. 9351 of *Lecture Notes in Computer Science*, pp. 234–241, Springer International Publishing, Cham, 2015.
- [24] L. Ruzicka, "Znerual/BioSig2022 GitHub," 2023, <https://github.com/Znerual/BioSig2022> [Accessed on: 2023 Jul 21].
- [25] A. Dabouei, S. Soleymani, J. Dawson, and N. M. Nasrabadi, "Deep contactless fingerprint unwarping," in *2019 International Conference on Biometrics (ICB)*, pp. 1–8, IEEE, Crete, Greece, 2019.
- [26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," 2019, doi: 10.48550/arXiv.1801.04381.arXiv: 1801.04381[cs]. Available from: [Accessed on: 2023 Mar 2] <http://arxiv.org/abs/1801.04381>.
- [27] J. Silva, M. Nobre Menezes, T. Rodrigues, B. Silva, F. Pinto, and A. Oliveira, "Encoder–decoder architectures for clinically relevant coronary artery segmentation," 2021.
- [28] N. Beheshti and L. Johnsson, "Squeeze U-Net: a memory and energy efficient image segmentation network," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1495–1504, IEEE, Seattle, WA, USA, 2020.
- [29] D. Sollinger and A. Uhl, "Optimizing contactless to contact-based fingerprint comparison using simple parametric warping models," in *2021 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–7, IEEE, Shenzhen, China, 2021.
- [30] B. J. Wing, "Data format for the interchange of fingerprint, facial and other biometric information ANSI/NIST-ITL 1-2011 NIST special publication 500-290 Edition 2," National Institute of Standards and Technology, 2013, Report Number: NIST SP 500-290e2 Aug: NIST SP 500-290e2. doi: 10.6028/NIST.SP.500-290e2. Available from: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-290e2.pdf>.
- [31] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics Gems*, pp. 474–485, Elsevier, 1994.
- [32] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [33] M. Tan and Q. V. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," 2020, doi: 10.48550/arXiv.1905.11946.arXiv: 1905.11946[cs,stat]. Available from: 2023-07-21, <http://arxiv.org/abs/1905.11946>.
- [34] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: transformers for image recognition at scale," 2021, doi: 10.48550/arXiv.2010.11929. arXiv: 2010.11929[cs]. Available from: 2023-07-21, <http://arxiv.org/abs/2010.11929>.
- [35] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, doi: 10.48550/arXiv.1502.03167. arXiv: 1502.03167[cs]. Available from: [Accessed on: 2023 Mar 6], <http://arxiv.org/abs/1502.03167>.
- [36] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, arXiv: 1505.00853[cs,stat]. Available from: [Accessed on: 2023 Mar 20], <http://arxiv.org/abs/1505.00853>.
- [37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, doi: 10.48550/arXiv.1512.03385. arXiv: 1512.03385[cs]. Available from: [Accessed on: 2023 Mar 26], <http://arxiv.org/abs/1512.03385>.
- [38] J. Priesnitz, R. Huesmann, C. Rathgeb, N. Buchmann, and C. Busch, "Mobile contactless fingerprint recognition: implementation, performance and usability aspects," *Sensors*, vol. 22, no. 3, Article ID 792, 2022.
- [39] H. D. Bansal, A. D. Badiye, and N. S. Kapoor, "Distribution of fingerprint patterns in an Indian population," 2014.
- [40] M. Liu, X. Jiang, and A. C. Kot, "Fingerprint reference-point detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, Article ID 634350, 2005.
- [41] H. B. Kekre and V. A. Bharadi, "Fingerprint core point detection algorithm using orientation field based multiple features," *International Journal of Computer Applications*, vol. 1, no. 15, pp. 97–103, 2010.
- [42] N. K. Johal and A. Kamra, "A novel method for fingerprint core point detection," vol. 2, 2011.
- [43] A. Julasayvake and S. Choomchuay, "An algorithm for fingerprint core point detection," in *2007 9th International Symposium on Signal Processing and Its Applications*, pp. 1–4, IEEE, Sharjah, United Arab Emirates, 2007.
- [44] IDKit SDK, "Innovatrics," [Accessed on: 2023 Mar 6], <https://support.innovatrics.com/support/solutions/folders/5000267908>.
- [45] NFIQ 2, "NIST," 2011, Last Modifed: 2021-11-12T07:00:00. Available from: [Accessed on: 2023 Mar 13], <https://www.nist.gov/services-resources/software/n.q-2>.
- [46] J. Priesnitz, C. Rathgeb, N. Buchmann, and C. Busch, "Touchless fingerprint sample quality: prerequisites for the applicability of NFIQ2.0." in *BIOSIG 2020-Proceedings of the 19th International Conference of the Biometrics Special Interest Group*, pp. 45–56, Gesellschaft für Informatik e.V., 2020.
- [47] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon et al., "MCYT baseline corpus: a bimodal biometric database," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 6, pp. 395–401, 2003.
- [48] R. Cappelli, M. Ferrara, A. Franco, and D. Maltoni, "Fingerprint verification competition 2006," *Biometric Technology Today*, vol. 15, no. 7-8, pp. 7–9, 2007.