

Literature Review for Contactless-to-Contact Fingerprint Matching

1. Executive Context and Problem Landscape

1.1 The Paradigm Shift in Biometric Identity Management

The global landscape of biometric identity management is currently undergoing a foundational transformation, migrating from dedicated, hardware-dependent acquisition modalities toward ubiquitous, software-defined solutions. For decades, the gold standard for fingerprint acquisition has been the optical or capacitive scanner, devices that rely on physical contact to capture the friction ridge patterns of the human finger. These systems, utilizing Frustrated Total Internal Reflection (FTIR), have underpinned the world's largest identity databases, including the Federal Bureau of Investigation's Next Generation Identification (NGI) system and the Unique Identification Authority of India (UIDAI) Aadhaar ecosystem.¹

However, the operational rigidities of contact-based systems have become increasingly apparent. The reliance on specialized hardware limits scalability, particularly in rural or resource-constrained environments where distributing certified scanners is logistically burdensome. Furthermore, the global COVID-19 pandemic catalyzed a permanent shift in user expectations regarding hygiene, rendering shared contact surfaces less desirable.³ Consequently, the industry is pivoting toward **contactless fingerprint acquisition**, leveraging the high-resolution RGB cameras embedded in commodity smartphones to capture "finger photos."

This transition, while operationally advantageous, introduces a profound technical schism known as the **interoperability gap**. The legacy databases (galleries) against which new users must be authenticated are populated almost exclusively with contact-based images. A smartphone camera captures a fundamentally different representation of the finger—a 3D, perspective-distorted, diffusely reflected image—compared to the 2D, flattened, frustration-contrast image of a contact scanner. The challenge, therefore, is not merely to capture a fingerprint, but to algorithmically translate the contactless modality into a representation that is mathematically indistinguishable from a contact scan by the legacy Automated Fingerprint Identification Systems (AFIS).⁴

1.2 The UIDAI/SITAA Challenge Framework

The specific context for this research is the challenge posed by the UIDAI and the SITAA-Startup-Cohort1, which seeks to develop a solution for **contactless-to-contact (C2C) matching** that runs entirely on the mobile edge.¹ The requirements are rigorous and delineate

the boundaries of the necessary technical architecture:

1. **Legacy Interoperability:** The solution must produce output templates that are fully compatible with UIDAI's existing AFIS. This implies that the system must function as a "virtual scanner," outputting standard ISO/IEC 19794-4 images despite the disparate input modality.¹
2. **Edge-Based Processing:** The entire pipeline—from capture to template generation—must execute on Android devices (version 9 and above) with a strict latency threshold of **≤ 2 seconds**. This constraint effectively disqualifies heavy server-side processing or massive deep learning models that cannot be quantized or optimized for mobile Neural Processing Units (NPUs).¹
3. **ISO Standardization:** The output must not only match visually but must adhere to the **ISO/IEC 19794-4** standard, which dictates a spatial resolution of 500 pixels per inch (PPI), specific grayscale depths, and Wavelet Scalar Quantization (WSQ) compression. This introduces a complex requirement for precise scale estimation, as mobile cameras do not have a fixed focal plane relative to the subject.⁷
4. **Quality Assurance:** The system must incorporate mechanisms for "Readiness Scoring," quantifying metrics such as minutiae count and clarity (NFIQ) to flag acceptable versus rejected captures before they are transmitted for matching.¹

The objective of this report is to deconstruct these requirements into a viable technical architecture, identifying the specific deep learning models and signal processing algorithms capable of bridging the C2C gap within the constraints of mobile edge computing.

2. The Physics and Mathematics of Interoperability

To engineer a solution that allows a mobile photo to match a contact scan, one must first rigorously model the physical and geometric divergences between the two modalities. These differences are not merely cosmetic; they are structural discrepancies that prevent standard minutiae extractors from functioning.

2.1 Photometric Divergence: FTIR vs. Ambient Reflection

The primary barrier to immediate interoperability lies in the physics of image formation.

Frustrated Total Internal Reflection (FTIR):

Legacy contact scanners operate on the principle of FTIR. A glass prism is illuminated from the side. When a finger presses against the platen, the ridges of the skin come into physical contact with the glass. At these points of contact, the refractive index boundary changes, causing the light to be absorbed or scattered rather than reflected. The valleys of the fingerprint, which do not touch the glass, essentially retain the air-glass interface, causing the light to be totally internally reflected into the sensor.

- *Result:* The resulting image is binary-like in its contrast. Ridges appear distinctively dark

(or bright, depending on the sensor polarity), and valleys appear the opposite. The transition is sharp, and the image represents the **physical touch** of the skin, not its visual texture. The background is uniformly white or black, free of noise.²

Diffuse Ambient Reflection (Contactless):

In contrast, a smartphone camera captures light that is diffusely reflected from the skin's surface.

- *Mechanism:* The visibility of the friction ridges is determined by the interaction of ambient light (or the camera flash) with the 3D topography of the finger. Ridges are visible primarily due to shadows cast into the valleys (shading) or subtle differences in skin texture/color.
- *Challenges:*
 - **Low Contrast:** The luminance difference between a ridge and a valley in a photo is minimal compared to the binary contrast of FTIR.
 - **Subsurface Scattering:** Human skin is translucent. Light penetrates the stratum corneum and scatters within the tissue. This "glow" can blur the high-frequency ridge details, especially under strong flash illumination, leading to a "washed out" appearance where ridges are indistinguishable.¹⁰
 - **Noise:** The image includes complex background clutter, sensor noise (especially in low light), and variations in skin pigmentation which are irrelevant to the ridge pattern but can confuse feature extractors.⁵

Implication for Modeling: The solution cannot simply process the RGB image. It must essentially "hallucinate" or mathematically reconstruct the contact contrast. This suggests the need for **domain adaptation** models (like CycleGAN or specialized enhancers) that map the *Texture Domain* to the *Contact Domain*.¹²

2.2 Geometric Distortion and Projection Models

The geometric mismatch is arguably the most significant cause of matching failure in C2C scenarios. A contact print is a 2D unrolling of a 3D surface; a contactless photo is a 2D projective transformation of that same surface.

Perspective Distortion:

A smartphone camera acts as a pinhole projector. Points on the finger that are closer to the lens (the bulb of the finger) appear spatially larger than points at the periphery.

- *Mathematical Model:* If we define the finger surface $S \in \mathbb{R}^3$ and the image plane $I \in \mathbb{R}^2$, the projection P induces a non-uniform scaling factor $\lambda(x, y)$ across the image.
- *Effect:* The ridge frequency (ridges per millimeter) appears to vary across the image. Standard AFIS algorithms, which assume a constant ridge frequency (approx. 0.5mm

period), will fail to extract minutiae correctly in the periphery of a perspective-distorted image.¹³

Cylindrical Deformation vs. Elastic Flattening:

- **Contact (Elastic):** When a finger presses against a scanner, the skin undergoes non-linear elastic deformation. The 3D cylinder is physically flattened, introducing shear and tensile stress that locally stretches the ridge pattern.
- **Contactless (Rigid):** The mobile photo captures the finger in its natural, semi-cylindrical state. The periphery of the finger curves away from the camera, resulting in foreshortening.
- *The "Unrolling" Problem:* To make the contactless image match the contact gallery, the system must mathematically "unroll" this cylinder onto a flat plane. However, simply unrolling it geometrically (assuming a perfect cylinder) is insufficient because it does not account for the **elastic deformation** present in the contact gallery. A purely geometric unroll might produce a "correct" flat map, but the gallery image is "distorted" by pressure. Therefore, the effective solution must mimic the pressure distortion—a process termed **"virtual flattening"**.¹⁴

2.3 Scale Ambiguity and the 500 PPI Imperative

ISO/IEC 19794-4 requires fingerprints to be stored at 500 PPI.⁷ In a contact scanner, the sensor size and optics are fixed; 500 PPI is a hardware constant. In mobile capture, the user moves the phone. A finger captured at 10cm is half the size (in pixels) of a finger captured at 5cm.

- *The Challenge:* The SDK must dynamically estimate the PPI of the captured image and rescale it. Without depth sensors (like LiDAR, which are rare on mid-range Androids), this must be done using **Shape-from-Texture** cues, specifically the ridge frequency itself.⁸

3. Computational Architectures for Segmentation and ROI Extraction

The first stage of the operational pipeline is the isolation of the fingerprint from the chaotic background of a video stream. This is critical for mobile implementations where processing the full high-resolution frame would violate the 2-second latency constraint.

3.1 The Failure of Traditional Computer Vision

Historical approaches utilized classical computer vision techniques:

- **Otsu's Thresholding:** Converting to grayscale and finding a global threshold. This fails dynamically under variable lighting or when the background skin tone (palm) matches the finger.¹⁶
- **Color Space Clustering (HSV/YCbCr):** Segmentation based on skin color. This is

non-robust to diverse demographics (skin tones) and background clutter that mimics skin color (e.g., wood tables).¹

Consequently, the industry has standardized on **Deep Semantic Segmentation**, specifically lightweight Convolutional Neural Networks (CNNs).

3.2 MobileUNetV3: The Optimal Balance

For the specific constraints of Android deployment (low latency, high accuracy), **MobileUNetV3** emerges from the research as the most effective architecture.

Architecture Analysis:

- **Encoder (Backbone):** Instead of the heavy ResNet or VGG encoders used in server-side segmentation, this model uses **MobileNetV3-Small**.
 - *Mechanism:* It employs **Depthwise Separable Convolutions**, which factorize standard convolutions into a depthwise layer (filtering inputs) and a pointwise layer (combining outputs). This reduces the computational cost (FLOPs) by a factor of 8-9x compared to standard convolutions.¹⁷
 - *Squeeze-and-Excitation (SE):* MobileNetV3 introduces lightweight attention modules that reweight channel importance, improving the model's ability to distinguish "finger" from "background" without adding significant latency.
- **Decoder:** A lightweight U-Net style decoder.
 - *Skip Connections:* It retains the U-Net's signature skip connections, which concatenate high-resolution feature maps from the encoder with the upsampled features in the decoder. This is crucial for fingerprints because the **boundary precision** (the exact edge of the fingertip) must be preserved to avoid cutting off minutiae.¹⁸

Performance Metrics:

- **Accuracy:** MobileUNetV3 achieves Dice coefficients > 0.96 and Intersection over Union (IoU) > 0.90 on fingerprint segmentation tasks.¹⁹
- **Latency:** On a standard Android device (Snapdragon 845 equivalent), inference for a 224x224 input is approximately **30-50ms**. This leaves ample room in the 2-second budget for subsequent processing.¹⁸

3.3 TipSegNet: Multi-Scale Robustness

An alternative architecture identified in recent literature is **TipSegNet**.

- **Differentiation:** It utilizes a **Feature Pyramid Network (FPN)**.
- **Mechanism:** FPNs extract features at multiple scales. This is particularly useful for the "Auto-Capture" phase where the user might be moving the phone closer or further away. TipSegNet can robustly detect fingertips at varying distances.²⁰

- **Trade-off:** While offering slightly higher accuracy (mIoU 0.987)²⁰, the backbone (often ResNeXt or heavier variants) is computationally more expensive than MobileNetV3. For the UIDAI requirement of "lightweight" models compatible with lower-end Androids, MobileUNetV3 is the pragmatic recommendation, while TipSegNet represents a high-end alternative.

3.4 Operational Strategy: The "Preview" Loop

To minimize latency, the segmentation model should not run on the full 12MP image.

1. **Preview Stream:** Run the quantized (INT8) MobileUNetV3 on the viewfinder stream (e.g., 224x224 or 320x320 resolution).
2. **Coordinate Mapping:** Once a high-confidence mask is detected, map the bounding box coordinates to the full-resolution sensor space.
3. **Crop:** Perform the high-res capture and crop immediately. This ensures the subsequent costly steps (Unwarping) only process the pixels of interest.⁴

4. Geometric Rectification and Unwarping architectures

Once the finger is segmented, the system faces the central challenge: transforming the curved, perspective-distorted 2D photo into a flat, 500 PPI representation that aligns with the contact gallery. This is the **C2C (Contactless to Contact)** transformation.

4.1 Theoretical Models for Unwarping

Thin Plate Spline (TPS) Models:

TPS is a non-rigid interpolation method. It minimizes a "bending energy" function to map a set of source control points to target control points.

- *Limitations:* TPS requires reliable keypoints (minutiae) to be detected *before* unwarping to define the transformation. However, minutiae detection on the raw distorted photo is unreliable, creating a "chicken-and-egg" problem.¹³

Parametric Cylinder Unrolling:

This assumes the finger is a perfect cylinder.

- *Equation:* $x' = R \cdot \arcsin(x/R)$, where x is the horizontal distance from the center and R is the estimated radius.
- *Limitations:* Fingers are not perfect cylinders; they are tapered and irregular. This method fails to account for the "pitch" (finger pointing up/down) which causes trapezoidal distortion.⁸

4.2 The C2CL Framework: State-of-the-Art Architecture

The most effective solution identified in the research is the **C2CL (Contact to Contactless)** framework proposed by Jain et al.⁴ It bypasses the need for manual geometric modeling by learning the transformation directly from data.

Core Component: Spatial Transformer Network (STN)

The STN is a differentiable module that can be inserted into a neural network to perform spatial manipulations.

- **Localization Network:** This sub-network takes the input feature map U and regresses a set of transformation parameters θ (e.g., the 6 parameters of an affine transformation, or a grid of thin-plate spline control points).
- **Grid Generator:** It generates a sampling grid $T_\theta(G)$ based on the predicted parameters.
- **Sampler:** It samples the input image at the grid points to produce the unwarped output V .
 - *Equation:* $V_i = \sum_n \sum_m U_{nm} \max(0, 1 - |x_i - m|) \max(0, 1 - |y_i - n|)$ (using bilinear interpolation).

Why STN is Superior for Mobile:

1. **Efficiency:** The sampling operation is a highly optimized matrix operation that can be parallelized on the GPU. It is significantly faster than iterative 3D reconstruction methods.
2. **End-to-End Training:** The critical innovation in C2CL is how the STN is trained. It is not trained to just "flatten" the image visually. It is trained to **minimize the biometric feature distance** between the unwarped photo and the ground-truth contact scan.
 - *Loss Function:* $\mathcal{L} = |$

$|D(I_{\text{unwarped}}) - D(I_{\text{contact}})|^2$, where $D(\cdot)$ is a fixed-length fingerprint embedding (like DeepPrint). * *Result:* The network learns to warp the photo specifically to maximize matching performance against the contact gallery, implicitly correcting for both perspective and elastic contact distortion.⁴

4.3 3D Reconstruction: Shape-from-Texture

An alternative, albeit heavier, approach is **Monocular 3D Reconstruction**.

- **Method:** A Deep CNN predicts a surface normal map or a depth map from the single RGB image. The system then mathematically unrolls this 3D mesh.
- **Performance:** This offers high accuracy for extreme poses (rotations > 45 degrees).

- **Mobile Viability:** Models estimating dense depth maps are computationally intensive (typically >500ms). For the "lightweight" and "low latency" constraints of the SITAA challenge, the STN-based 2D warping (C2CL) is the superior architectural choice.¹⁴

4.4 Implementation on Android

To implement the C2CL Unwarper on Android:

- **Model Format:** The STN must be exported to TensorFlow Lite.
- **Custom Operators:** Standard TFLite sometimes lacks native support for the GridSample operator used in STNs.
 - *Solution:* Implement a custom TFLite operator in C++ using the Android NDK, or use the `tf.image.projective_transform` approximation if available in the GPU delegate.
- **Optimization:** The Localization Network (which predicts the warp parameters) can be a very shallow CNN (e.g., 4 layers), as the global shape of the finger is a low-frequency feature. This keeps inference < 100ms.⁴

5. Radiometric Enhancement and Domain Adaptation

After geometric unwarping, the image is geometrically correct but radiometrically wrong. It still looks like a photo (gray skin, low contrast) rather than a scan (black/white ridges). The AFIS expects the latter.

5.1 Ridge Frequency Estimation and Scale Normalization

Before contrast enhancement, the system **must** resolve the scale ambiguity to meet the ISO 19794-4 500 PPI requirement.

Algorithm: Spectral Analysis (FFT)

1. **Block Processing:** Divide the unwarped image into non-overlapping blocks (e.g., 32x32).
2. **Fourier Transform:** Apply 2D FFT to each block.
3. **Peak Detection:** Identify the peak energy frequency f_{peak} . This corresponds to the sine wave of the ridges.
4. **Global Consensus:** Compute the median frequency across all blocks to reject noise.
5. **Scaling:**
 - Standard Ridge Frequency at 500 PPI $\approx 1/9$ to $1/10$ cycles per pixel (ridges are ~9-10 pixels apart).
 - Calculate Resizing Factor $k = f_{target} / f_{peak}$.
 - Resize image by k .⁸

Importance: This step is non-negotiable. If a finger is captured too close, the ridges might be 20 pixels apart. The AFIS feature extractor (designed for 10-pixel separation) will fail to detect

them or misclassify them as noise.

5.2 Contextual Gabor Filtering

The gold standard for fingerprint enhancement is **Contextual Filtering**.

- **Concept:** Ridges flow locally in a specific orientation and frequency. A filter aligned with this flow will amplify the ridge and smooth out orthogonal noise.
- **Gabor Filter Bank:** A set of filters $G(x, y; \theta, f)$ are created for discrete orientations θ (e.g., 0°, 22.5°, 45°...) and the estimated frequency f .
- **Application:** For each pixel, the system estimates the local orientation (using image gradients) and applies the corresponding Gabor filter.
- **Pros:** Mathematically robust, restores broken ridges.
- **Cons:** Computationally expensive (convolution with large kernels).
- **Mobile Optimization:** Implementation should use separable filters or FFT-based convolution in the Android NDK (C++) layer to maintain speed.²⁴

5.3 Generative Style Transfer (CycleGAN)

A more modern approach uses **Generative Adversarial Networks (GANs)** for "Domain Adaptation."

- **Architecture: CycleGAN.** It consists of two mappings: $G : X \rightarrow Y$ (Photo to Scan) and $F : Y \rightarrow X$ (Scan to Photo). It ensures cycle consistency: $F(G(x)) \approx x$.
- **Advantage:** It learns complex, non-linear mappings that Gabor filters miss, such as correcting lighting glare or specific sensor noise patterns.
- **Risk:** GANs are prone to **hallucination**. In fingerprints, generating a "likely" ridge where none exists is a critical security failure (increasing False Accept Rate).
- **Recommendation:** For the SITAA challenge, which prioritizes "Accuracy" and "AFIS Interoperability," standard Gabor filtering or **CLAHE (Contrast Limited Adaptive Histogram Equalization)** is preferred over GANs due to predictability and lower computational overhead. If a GAN is used, it must be heavily constrained (e.g., using a "Pixel-to-Pixel" loss with strong regularization) and used only for texture enhancement, not structure generation.²⁶

5.4 ISO 19794-4 Encoding (WSQ)

The final step in the pipeline is encoding.

- **Format:** The raw pixel buffer (now 500 PPI, enhanced) must be compressed using **WSQ**.
- **Library:** The **NIST Biometric Image Software (NBIS)** is the open-source reference implementation. The cwsq utility from NBIS can be compiled via the Android NDK.
- **Quality Check:** Before encoding, the **NFIQ 2 (NIST Fingerprint Image Quality)**

algorithm should be run. NFIQ 2 returns a score (0-100). The SDK should enforce a threshold (e.g., Score > 40) as per the "Readiness Scoring" requirement.²⁸

6. Feature Extraction and Matching Paradigms

While the primary output is the ISO image, the SDK must often perform local matching (1:1 verification) or quality pre-checks. Two distinct paradigms exist here.

6.1 Minutiae-Based Matching (The ISO Standard)

This is the interoperability layer.

- **Model: VeriFinger** (or similar ISO 19794-2 extractors).
- **Method:** Extracts ridge endings and bifurcations (x, y, theta).
- **Pros:** Fully compatible with Aadhaar AFIS.
- **Cons:** Extremely sensitive to image quality. If the unwarping (Stage 4) is imperfect, the relative positions of minutiae shift, causing match failure.³⁰

6.2 Deep Representation Matching (DeepPrint)

This is the performance layer.

- **Model: DeepPrint.**⁴
- **Architecture:** A deep CNN (e.g., Inception-v4 backbone) trained with a softmax classifier on millions of identities. It outputs a fixed-length vector (e.g., 192 floats).
- **Method:** Matching is a simple Cosine Distance calculation between two vectors.
- **Pros:**
 - **Fast:** Matching takes microseconds.
 - **Robust:** The deep network learns to ignore distortion and sensor noise, often succeeding where minutiae matchers fail on contactless images.
 - **Guidance:** As mentioned in Section 4.2, DeepPrint is crucial for *training* the STN unwarper.
- **Cons:** Not interoperable with legacy AFIS (you can't send a DeepPrint vector to Aadhaar).
- **Integration Strategy:** Use DeepPrint for the **on-device 1:1 demo app** and for internal quality checks. Use the standard ISO image for the formal AFIS submission.

6.3 Fusion Strategies

To maximize accuracy (TAR), the system can employ a **Two-Stage Search** or **Score Fusion**:

1. **Stage 1 (Fast):** Use DeepPrint to rapidly verify if the finger matches the local cache (1:1).
2. **Stage 2 (Precise):** If DeepPrint passes, generate the ISO template and perform minutiae matching (or send to server). Research shows that fusing DeepPrint and Minutiae scores significantly boosts TAR, especially for "ugly" quality latent or contactless prints.²¹

7. Edge Implementation Strategy on Android

The "SITAA-Startup-Cohort1" challenge requires this complex pipeline to run on Android 9+ with < 2s latency. This requires aggressive optimization.

7.1 Android Neural Networks API (NNAPI) & TFLite

Deep learning models (MobileUNet, STN) must be converted to **TensorFlow Lite (TFLite)**.

- **Quantization:** Convert weights from Float32 to **Int8**. This typically reduces model size by 4x and speeds up inference by 2-3x on mobile DSPs/NPUs with negligible accuracy loss (<1%).³⁴
- **Delegates:**
 - Use the **GPU Delegate** for the STN unwarping (grid sampling is parallelizable).
 - Use the **NNAPI Delegate** for MobileUNet (segmentation), which allows the OS to schedule the workload on the DSP or NPU.³⁵

7.2 Native C++ (NDK) for Signal Processing

Do not implement FFT (Scaling) or Gabor Filtering in Java/Kotlin. The overhead of the Java Virtual Machine (JVM) and Garbage Collection (GC) is too high.

- **Strategy:** Use JNI (Java Native Interface) to call optimized C++ code.
- **Libraries:** Use **OpenCV Android SDK** (C++) or **Eigen** for matrix operations (FFT, Filtering). The **NBIS** library (for WSQ/NFIQ) is already written in C and integrates directly.⁴

7.3 Pipeline Orchestration & Memory Management

- **Zero-Copy:** Use ByteBuffer for passing image data between Java (Camera API), TFLite (Models), and C++ (OpenCV). Avoid creating new Bitmap objects at every stage to prevent GC pauses.
- **Threading:**
 - Thread 1 (UI): Camera Preview.
 - Thread 2 (Worker): Deep Learning (TFLite interpreter).
 - Thread 3 (Worker): Signal Processing (C++).
 - This pipelining allows the Capture UI to remain responsive while the "heavy lifting" happens in the background.

8. Comparative Evaluation and Strategic Recommendations

8.1 Comparative Analysis of Solution Architectures

The following table synthesizes the available architectures against the specific SITAA/UIDAI constraints.

Solution Component	Models Evaluated	Pros	Cons	Latency Impact	Recommendation
Segmentation	MobileUNetV3	Ultra-lightweight, high boundary precision (skip connections).	Less robust to extreme clutter than Mask R-CNN.	Low (~30ms)	Adopt (Best balance).
	TipSegNet	Multi-scale handling (FPN). Slightly higher mIoU.	Heavier backbone (ResNeXt). Slower inference.	Medium (~100ms)	Backup (High-end only).
	Otsu/HSV (Classical)	Zero latency.	Fails dynamically. Unreliable.	Negligible	Reject.
Unwarping	C2CL (STN)	End-to-end learnable. Optimizes for <i>matching</i> not just visuals.	Requires paired training data. "Black box" warping.	Medium (~150ms)	Adopt (SOTA approach).
	3D Shape-from-X	Physically accurate 3D unrolling.	Computationally prohibitive for edge. Slow.	High (>500ms)	Reject.
	Parametric (Cylinder)	Fast, simple math.	Fails on tapered fingers or	Low	Reject (Poor accuracy).

			pitch rotation.		
Enhancement	Contextual Gabor	Mathematically grounded. Restores ridge flow.	Computationally heavy (convolutions).	High (~600ms)	Adopt (with C++ opt).
	CycleGAN	Excellent visual style transfer.	Hallucination risk. Heavy model size.	Very High	Reject (Security risk).
Feature Extraction	DeepPrint	Fast matching. Robust to distortion.	Non-standard output. Not AFIS interoperable.	Low	Hybrid Use (Internal check).
	Minutiae (ISO)	AFIS Interoperability. Standards compliant.	Sensitive to quality. Slow extraction.	Medium	Adopt (Required).

8.2 Performance vs. Latency Trade-off

Step	Technique	Android Latency (Est.)
Capture	Camera2 API + YUV	100 ms
Segmentation	MobileUNetV3 (Int8)	50 ms
Unwarping	STN (GPU Delegate)	150 ms
Scaling	FFT (Native C++)	100 ms

Enhancement	Gabor / CLAHE (Native C++)	600 ms
Encoding	WSQ (NBIS C++)	100 ms
Total		~1100 ms (1.1s)

This projected latency of **1.1 seconds** sits comfortably within the **≤ 2 seconds** requirement, leaving a 900ms buffer for UI rendering, OS overhead, and the "Readiness Scoring" (NFIQ 2) computation.⁴

8.3 Final Recommendations for Implementation

1. **Architecture:** Adopt the **C2CL** framework logic. Train the Spatial Transformer Network (STN) offline using a large dataset of paired contact/contactless prints (e.g., PolyU, IIT Bombay databases) using the DeepPrint loss function. Deploy only the inference engine (TFLite) to the mobile.
2. **Standards:** Prioritize **Ridge Frequency Estimation** (FFT) for scale normalization. Without this, the ISO 19794-4 output will be technically valid (file format) but biometrically useless (mismatched ridge density) to the Aadhaar AFIS.
3. **Optimization:** Avoid Python/Java for pixel-level operations. Build a robust **JNI bridge** to a C++ library (OpenCV/NBIS) for the enhancement and encoding stages.
4. **Quality Gate:** Implement a strict "Readiness Score" using **NFIQ 2 Mobile** or a lightweight blur detector (Laplacian variance) to reject poor frames *before* they enter the heavy processing pipeline. This improves the perceived user experience by providing instant feedback ("Hold Steady," "Move Closer").¹

By strictly adhering to this architecture, the proposed solution will not only meet the functional requirements of the SITAA challenge but effectively bridge the gap between the mobile future of identity and the massive legacy infrastructure of national ID systems.

Works cited

1. fingerprint_uid.pdf
2. Techniques for contactless fingerprint recognition: A review | AIP, accessed January 26, 2026, <https://pubs.aip.org/aip/acp/article/3318/1/030030/3356802/Techniques-for-contactless-fingerprint-recognition>
3. Touchless fingerprint recognition: A survey of recent developments and challenges, accessed January 26, 2026, https://www.researchgate.net/publication/387832570_Touchless_fingerprint_recognition_A_survey_of_recent_developments_and_challenges

4. C2CL: Contact to Contactless Fingerprint Matching - Biometrics Research Group, accessed January 26, 2026,
http://biometrics.cse.msu.edu/Publications/Fingerprint/C2CL_Contact_to_Contactless_Fingerprint_Matcher_TIFS_final.pdf
5. Fingerprint Systems: Sensors, Image Acquisition, Interoperability and Challenges - PMC, accessed January 26, 2026,
<https://pmc.ncbi.nlm.nih.gov/articles/PMC10384471/>
6. INVITATION FOR EXPRESSION OF INTEREST FOR Participation in Biometric Authentication Proof of Concept Studies - uidai, accessed January 26, 2026,
https://uidai.gov.in/images/tenders/eoi_for_auth_poc_devices_final.pdf
7. ISO-IEC-19794-4-Fingerprint-Image-Record-Standard - BioEnable Aadhaar Solutions, accessed January 26, 2026,
<https://aadhaar.bioenabletech.com/iso-iec-19794-4-fingerprint-image-record-standard>
8. Guidance for Evaluating Contactless Fingerprint Acquisition Devices - NIST Technical Series Publications, accessed January 26, 2026,
<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-305.pdf>
9. On the interoperability of capture devices in fingerprint presentation attacks detection - CEUR-WS.org, accessed January 26, 2026,
<https://ceur-ws.org/Vol-1816/paper-07.pdf>
10. Contactless Fingerprint Recognition Using Deep Learning—A Systematic Review - MDPI, accessed January 26, 2026, <https://www.mdpi.com/2624-800X/2/3/36>
11. A Robust Contactless Fingerprint Enhancement Algorithm - SciSpace, accessed January 26, 2026,
<https://scispace.com/pdf/a-robust-contactless-fingerprint-enhancement-algorithm-4crr6y5u5i.pdf>
12. Finger-UNet: A U-Net based Multi-Task Architecture for Deep Fingerprint Enhancement - arXiv, accessed January 26, 2026,
<https://arxiv.org/html/2310.00629>
13. Deep Contactless Fingerprint Unwarping | Semantic Scholar, accessed January 26, 2026,
<https://www.semanticscholar.org/paper/Deep-Contactless-Fingerprint-Unwarping-Dabouei-Soleymani/8c582981abc3e47dfd459a064b0c2ceea9d98839>
14. A new approach to unwrap a 3-D fingerprint to a 2-D rolled equivalent fingerprint, accessed January 26, 2026,
https://www.researchgate.net/publication/224083408_A_new_approach_to_unwrap_a_3-D_fingerprint_to_a_2-D_rolled_equivalent_fingerprint
15. 3D to 2D fingerprints: Unrolling and distortion correction - Semantic Scholar, accessed January 26, 2026,
<https://www.semanticscholar.org/paper/3D-to-2D-fingerprints%3A-Unrolling-and-distortion-Zhao-Jain/e12333009b0763c0aa806fefcbdbd85886454484e>
16. TipSegNet: Fingertip Segmentation in Contactless Fingerprint Imaging - PMC - NIH, accessed January 26, 2026,
<https://pmc.ncbi.nlm.nih.gov/articles/PMC11946156/>
17. Understanding and Implementing MobileNetV3 | by Rishabh Singh - Medium,

accessed January 26, 2026,

<https://medium.com/@RobuRishabh/understanding-and-implementing-mobilenet-v3-422bd0bdfb5a>

18. (PDF) MobileUNetV3—A Combined UNet and MobileNetV3 Architecture for Spinal Cord Gray Matter Segmentation - ResearchGate, accessed January 26, 2026, https://www.researchgate.net/publication/362390667_MobileUNetV3-A_Combined_UNet_and_MobileNetV3_Architecture_for_Spinal_Cord_Gray_Matter_Segmentation
19. MobileUNetV3—A Combined UNet and MobileNetV3 Architecture for Spinal Cord Gray Matter Segmentation - MDPI, accessed January 26, 2026, <https://www.mdpi.com/2079-9292/11/15/2388>
20. TipSegNet: Fingertip Segmentation in Contactless Fingerprint Imaging - MDPI, accessed January 26, 2026, <https://www.mdpi.com/1424-8220/25/6/1824>
21. C2CL: Contact to Contactless Fingerprint Matching | Request PDF - ResearchGate, accessed January 26, 2026, https://www.researchgate.net/publication/356953456_C2CL_Contact_to_Contactless_Fingerprint_Matching
22. Monocular 3D Fingerprint Reconstruction and Unwarping, accessed January 26, 2026, https://ivg.au.tsinghua.edu.cn/~jfeng/pubs/Cui_PAMI23_3DFinger.pdf
23. Guidance on Contactless Friction Ridge Image Compression, v1.0 - GovInfo, accessed January 26, 2026, <https://www.govinfo.gov/content/pkg/GOVPUB-C13-98d2e58805dd8b30e1f8bed150272ab2/pdf/GOVPUB-C13-98d2e58805dd8b30e1f8bed150272ab2.pdf>
24. Latent fingerprint enhancement for accurate minutiae detection - arXiv, accessed January 26, 2026, <https://arxiv.org/html/2409.11802>
25. Fingerprint Image Enhancement: Algorithm and Performance Evaluation - Biometrics Research Group - Michigan State University, accessed January 26, 2026, <http://biometrics.cse.msu.edu/Publications/Fingerprint/MSU-CPS-97-35fenhance.pdf>
26. Synthetic Latent Fingerprint Generator - YouTube, accessed January 26, 2026, https://www.youtube.com/watch?v=9w5kD_BfgQU
27. Mobile Contactless Fingerprint Recognition: Implementation, Performance and Usability Aspects - PMC - PubMed Central, accessed January 26, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC8839666/>
28. Compression Guidance for 1000 ppi Friction Ridge Imagery - NIST Technical Series Publications, accessed January 26, 2026, <https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.500-289.pdf>
29. usnistgov/NFIQ2: Optical live-scan and ink fingerprint ... - GitHub, accessed January 26, 2026, <https://github.com/usnistgov/NFIQ2>
30. Towards More Accurate Contactless Fingerprint Minutiae Extraction and Pose-Invariant Matching - PolyU, accessed January 26, 2026, <https://www4.comp.polyu.edu.hk/~csajaykr/myhome/papers/TIFS2020a.pdf>
31. VeriFinger fingerprint recognition technology, algorithm and SDK for PC, smartphones and Web - Neurotechnology, accessed January 26, 2026,

- <https://www.neurotechnology.com/verifinger.html>
32. arxiv.org, accessed January 26, 2026, <https://arxiv.org/abs/2104.02811>
 33. [PDF] Fusion2Print: Deep Flash-Non-Flash Fusion for Contactless, accessed January 26, 2026, <https://www.semanticscholar.org/paper/Fusion2Print%3A-Deep-Flash-Non-Flash-Fusion-for-Sahoo-Namboodiri/40490fdb2cebfdabd50085a0ac68aabd4b8cf5e>
 34. Understanding Latency in Mobile & Android Systems — An End-to-End Perspective (Including Jetpack Compose) | by Subrajit Pandey | Dec, 2025 | Medium, accessed January 26, 2026, <https://medium.com/@subhrajeetpandey2001/understanding-latency-in-mobile-android-systems-an-end-to-end-perspective-including-jetpack-9712af44d8aa>
 35. Profiling MLC-LLM's OpenCL Backend on Android: Performance Insights - Callstack, accessed January 26, 2026, <https://www.callstack.com/blog/profiling-mlc-llms-opencl-backend-on-android-performance-insights>