2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) | 979-8-3315-4090-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/UEMCON62879.2024.10754732

Deepfake Attacks on Biometric Recognition: Evaluation of Resistance to Injection Attacks

Rahul Vijaykumar*
Department of Electrical and Computer Engineering
Clarkson University
Potsdam, NY, USA
*v@clarkson.edu

Sandip Purnapatra†
Department of Electrical and Computer Engineering
Clarkson University
Potsdam, NY, USA
†purnaps@clarkson.edu

Richard Plesh‡
Department of Electrical and Computer Engineering
Clarkson University
Potsdam, NY, USA
‡pleshro@clarkson.edu

Masudul Imtiaz§
Department of Electrical and Computer Engineering
Clarkson University
Potsdam, NY, USA
§mimtiaz@clarkson.edu

Daqing Hou¶
Department of Electrical and Computer Engineering
Clarkson University
Potsdam, NY, USA
¶dhou@clarkson.edu

Stephanie Schuckers‖
Department of Electrical and Computer Engineering
Clarkson University
Potsdam, NY, USA
‖sschucke@clarkson.edu

***Abstract*—Deepfake technologies, despite advancements in facial and voice recognition, pose significant threats by replicating voices or faces, compromising identity verification processes even during live interactions. This creates risks of impersonation and unauthorized access to sensitive data. Addressing these threats is crucial, especially given the ease of creating deepfakes with free open-source apps, which have advanced to the point where distinguishing between real and fake is increasingly challenging. Our developed system enhances biometric security by leveraging existing hardware, specifically screen illumination, to scrutinize light reflections from the user's face. This method seamlessly integrates into current applications, adding a necessary layer of security to differentiate genuine individuals from deepfakes, thereby fortifying the overall biometric security framework without requiring additional algorithms or hardware.**

I. Introduction

The growing accessibility of deepfake creation tools raises significant concerns. Online applications now enable users worldwide to effortlessly generate deepfakes, even without advanced technical skills or in-depth knowledge of the technology. This simplicity poses a considerable threat, allowing individuals to manipulate a small portion of personal data or a snippet of a video available online to create deceptive replicas. This accessibility puts anyone in society at risk of falling victim to such deceptive practices.

Of particular concern is the emergence of applications like DeepFake Labs. These applications not only produce high-quality video outputs but also incorporate advanced facial morphing capabilities, enabling attackers to convincingly impersonate others. As depicted in Figure 1, this deepfake example was created using just two minutes of the target identity, combined with video footage capturing the mannerisms and expressions of the attacker or source. This ease of manipulation poses a serious risk to unsuspecting users.

Moreover, individuals with slightly more technical expertise, capable of hacking into mobile devices or laptops, can directly inject these manipulated videos into the system, as shown in Figure 2. To address these escalating threats, it becomes imperative to augment security measures, including the integration of advanced biometric security protocols where possible, to fortify defenses against such manipulative practices.

This becomes problematic for devices relying on hardware-dependent liveliness or deepfake detection algorithms. By circumventing these security layers, hackers can successfully deceive authentication systems and potentially perpetrate identity fraud. The vulnerability of these technologies underscores the need for heightened security measures and continuous innovation in detecting and preventing the misuse of deepfake technology. There have been instances, including banking scams, where imposters have utilized deepfake technology to impersonate others and commit fraud, leading to significant financial theft [1].

Now that we have grasped the gravity of this threat, the immediate objective is to devise effective countermeasures to secure video communications against deepfakes. A few

979-8-3315-4090-6/24/$31.00 ©2024 IEEE

Authorized licensed use limited to: The University of Alabama. Downloaded on March 15,2025 at 02:51:14 UTC from IEEE Xplore. Restrictions apply.

Fig. 1: An image illustrating the deepfake manipulation, wherein the attacker's face is seamlessly integrated with the target identity, creating a deceptive and convincing replica

of the previously conducted experiments include identifying deep-fake videos by analyzing both visual appearance and behavioral cues, where a technique was developed that could analyze deepfakes using behavioral patterns [2], examining differences between phonemes (speech sounds) and visemes (lip movements). By detecting irregularities in the alignment of lip movements with corresponding speech sounds to detect deepfake [3].

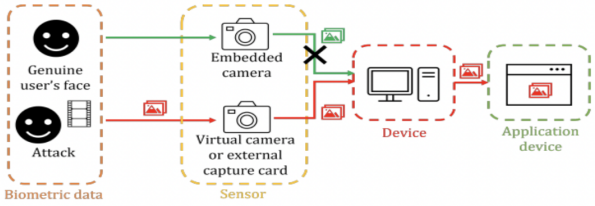


Fig. 2: Injection attack using virtual camera [4]

Employing advanced algorithms to detect expressions and contextual anomalies within images or videos, comparing them with recognized facial features. The algorithm then signals potential forgery or manipulated facial content, assisting in identifying the presence of deepfake technology or facial alterations. [5], Another comparable method prioritizes liveness detection through the front camera of smartphones, leveraging advanced algorithms to analyze real-time facial dynamics and cues. This enables the system to differentiate between genuine live faces and static or altered images, ultimately elevating security measures [6]. These features have been compared in Table I.

However, many of these methods require additional hardware or significant processing power, increasing the overall cost. Moreover, understanding their real-time functionality can be complicated. **Our contribution lies in leveraging the built-in screen illumination of smartphones to ascertain the authenticity of the individual shown during a call**. This approach allows us to determine whether the communication involves a genuine person or if a deepfake image has been maliciously inserted. By projecting light from the smartphone screen, we can illuminate a real person's face, which will be accurately captured by the screen. In contrast, if a deepfake is impersonating the individual, the light will not illuminate the face correctly, as deepfakes are typically computer-generated.

II. Methodology

Numerous studies have explored the creation of real-time deepfakes, employing various algorithms [7]. Among these methods, one of the most efficient approaches involves conducting an injection attack on communication devices like phones, laptops, or desktops, which are connected to the video source. This process manipulates the device's security functions, allowing for the redirection of the camera function. This can be achieved by utilizing a secondary camera or a virtual camera, which serves as a device or software to feed data to the camera application, replacing the live feed from the primary camera. Several open-source applications, including DroidCam [8], iVCam Webcam [9], or VCamera [10], have been utilized for these purposes. It's worth noting that some of these applications may not be readily available on official app stores but can be installed using APK files, which are accessible online. However, the installation of APK files typically requires developer permissions or access to the mobile device. The secondary camera or virtual camera applications leverage specific functionalities of mobile cameras. These applications enable users to load videos from their gallery or connect to external cameras that are not integrated into the mobile device. To achieve this functionality, it becomes essential to gain access to Android features, such as the camera, allowing for manipulation and redirection to the desired location.

Given that many of these applications are not directly available on the Play Store and have faced bans from platforms like the App Store, there is a need to inject them directly into the mobile device. This involves enabling developer access for further development. As we strive to enhance the functionality of these applications, particularly focusing on facial illumination, we plan to build upon this foundation by introducing an illumination feature at the application layer. To root the Google Pixel 3 smartphone using Magisk, we followed the method outlined below [11].

Magisk is a free, open-source software that grants users root access to their Android devices. By using Magisk, users can implement a variety of modifications and customizations, which has made it a favorite among Android enthusiasts. Magisk also includes a built-in app for managing root permissions and installing various modules.

Featuring a systemless approach and modular design, Magisk offers a safe and straightforward way to root a device and add new features and functionalities [12].

1) **Download Factory Image:** On your Windows/Mac computer, download the factory image of the Pixel 3 that matches the QPR2 beta version installed on your device. You can obtain it from the Android Developer download portal.
2) **Unpack Factory Image:** Unpack the factory image using your preferred archive manager. Locate the "image-pixel3-<build number>.zip" file, and extract the 'boot.img' file from it.
3) **Patch Boot Image:** Use Magisk to patch the boot image. Ensure that you have the latest stable version of Magisk.

Authorized licensed use limited to: The University of Alabama. Downloaded on March 15,2025 at 02:51:14 UTC from IEEE Xplore. Restrictions apply.

TABLE I: **Comparison Table: Related Studies vs. Our Approach**

Work Title	Approach	Key Focus	Distinguishing Attacks	Computer Centric	Smartphone Centric
Detecting DeepFake Videos from Appearance and Behavior (Agarwal et al.)	Analyzing appearance and behavior cues	Visual Attributes	2D, rebroadcast, deepfake	Yes	No
Detecting DeepFake Videos from Phoneme-Viseme Mismatches (Agarwal et al.)	Analyzing phoneme-viseme mismatches	Lip Movement	Deepfake	Yes	No
On the Generality of Facial Forgery Detection (Brockschmidt et al.)	Analyzing facial forgery detection generality	Facial forgery detection	Not specified	Yes	No
FaceRevelio: a Face Liveness Detection System for Smartphones with a Single Front Camera (Farrukh et al.)	Utilizing a single front camera and advanced algorithms	Facial liveness detection	Distinguishing real face, 2D, and deepfake attacks	No	Yes
Our Proposed Face Illumination Method on Smartphones	Detecting deepfake injection through virtual cameras	Face Illumination	Distinguishing virtual camera-based deepfake injection	No	Yes

magisk –patch /path/to/boot.img

4) **Copy and Rename:** Copy the Magisk-patched boot image from your phone to your computer and rename it to "`magisk_patched_boot.img`."
5) **Reboot into Bootloader Mode:** If USB debugging is enabled, reboot your Pixel 3 into bootloader mode using the following command: [language=bash] adb reboot bootloader
6) **Flash Patched Boot Image:** Flash the patched boot image and reboot the phone: [language=bash] fastboot flash boot /path/to/magisk_patched_boot.img
7) **Check Root Access:** Open the Magisk app on your Pixel 3, and your device should be recognized as rooted.

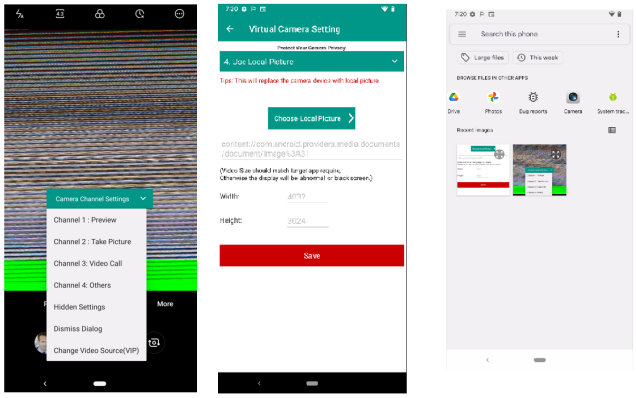


Fig. 3: Screenshots showing how to load a media into the V Camera application.

Following the aforementioned procedures, we successfully rooted the Pixel 3 phone. The V camera application [10], designated for use as a secondary camera, was obtained from the system and subsequently transferred to the mobile device using the Android Debug Bridge (ADB) by establishing a connection between the device and the system. Once the APK file was transferred, it was then installed on the mobile device. Figure 3 illustrates the steps for loading a media file into the camera application. The overall appearance of the application resembles that of a typical camera app, presenting options to switch between front and back cameras. However, in the right-hand corner, there is a hidden selection option that becomes visible only upon interaction. Once selected, this option provides the user with the choice to load a media file, which may include photos or videos stored in the gallery.

Before settling on the Deepfake Lab application for our deepfake creation endeavors, we undertook evaluations of various other applications to determine the most suitable choice. Among the applications scrutinized were Avatarify [13] [14], Synthesia [15], Flawless AI [16], Stability AI [17], Dall-E-2 [18], and Pinscreen [19]. However, these applications fell short of meeting our desired output standards. They either failed to provide the required precision or could not match the quality achieved with Deepfake Lab. The table II below outlines some of the limitations we discovered with these alternative applications.

TABLE II: **Deepfake App Comparison**

Application	General Observations	Limitations Identified
Avatarify	Avatarify excels in replicating realistic facial expressions.	Potential limitations in handling diverse skin tones may impact deepfake quality.
Synthesia	Synthesia exhibits efficient performance in generating lifelike facial animations.	Limited customization options and potential challenges in accurately reproducing intricate facial details may be observed in Synthesia.
Flawless AI	Subpar results, not on par with Deepfake Lab	Users may encounter limitations in terms of flexibility and customization options when using Flawless AI for deepfake creation.
Stability AI	Stability AI demonstrates robust capabilities in refining and enhancing facial features.	Potential for overfitting, limiting generalizability.
Dall-E-2	Output quality not as desired.	Requires significant computational resources.
Pinscreen	Output quality shortcomings.	Not compatible with video.
Deepfake Lab	Superior output quality, preferred choice.	None identified.

In the next phase, we generated deepfake videos using the LivDet 2021 dataset [20], which serves to assess and benchmark the effectiveness of face recognition systems in detecting presentation attacks. The dataset offers a standardized protocol for an independent evaluation of the latest advancements in face liveness detection. Additionally, we utilized the Deepfake

Authorized licensed use limited to: The University of Alabama. Downloaded on March 15,2025 at 02:51:14 UTC from IEEE Xplore. Restrictions apply.

Lab application for this purpose. The process of creating deepfake videos involved the following steps [21]:

1) **Download DeepFaceLab:**
   - Download DeepFaceLab from the official website, ensuring compatibility with your GPU. I used the NVIDIA GeForce RTX 3060 GPU to create the deepfake videos.
2) **Prepare the Workspace Folder:**
   - When DeepFaceLab is installed on the system, it creates a workspace folder that is utilized by the batch scripts, where both the source and destination videos should be added.
3) **Extract Faces from the Source Video:**
   - Name the source video as"data_src" and place it in the workspace folder.
   - Utilize the software to extract images from the source video, preferably in PNG format.
   - Ensure that the extracted faces are saved to the "data_src\aligned" folder.
4) **Train the Model:**
   - Use the software to train the deepfake model.
   - Set parameters such as the number of iterations, resolution, face type, and more.
5) **Use the Interactive Converter:**
   - Use the interactive converter provided by DeepFaceLab.
   - This tool allows you to experiment with different settings before the final conversion.
6) **Export the Final Video:**
   - After converting the frames, export the final deepfake video.
   - Adjust settings for overlay, mask mode, blur, sharpening, and more in the interactive converter.
   - You can preview the result during conversion and ensure it looks correct.

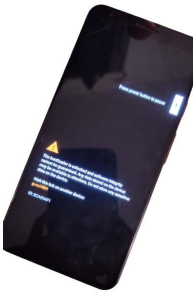


Fig. 4: Image of rooted phone

III. Evaluation and Analysis

The phone was successfully rooted using the outlined steps, granting superuser access to all Android features. The next task involved downloading and installing a secondary camera application. Since this camera application wasn't available on the official app store, it was downloaded to a PC as an APK (Android Package Kit) file and transferred to the mobile device using ADB (Android Debug Bridge). The startup screen of the rooted phone is illustrated in Figure 4.

After installing the application, we tested all functionalities. During testing, we noticed that not all images from the phone's gallery displayed correctly within the camera application. A quality conversion issue caused blurry images, and further analysis revealed that the application could only handle images and videos up to 720 pixels.

The next objective was to create deepfakes using DeepFake Labs. We considered two databases: the META dataset, which includes the casual conversation dataset 2.0 [22], and an in-house recording from the Clarkson dataset, comprising both scripted and unscripted videos of students and others, recorded in a controlled environment to ensure pristine audio quality.

Fig. 5: An illustration of persuasive deepfakes created using Deepfake Labs

Next, we identified a mannerism or expression for the attacker, who would use a target identity to generate a deepfake. Individuals with similar facial features were chosen for better results. Once the source mannerism video and the target identity video were selected, we ran facial extraction algorithms on these videos. The extracted faces were then used to train a model for deepfake generation, converting these frames into a video suitable for loading into the camera application.

Numerous videos were generated, and the most optimal one was selected, as shown in Figure 5. The leftmost image features the source mannerism or attacker, whose expression will be mimicked by the target identity in the middle image. The resulting deepfake is illustrated in the rightmost image. This process was repeated with various source mannerisms and target identities to produce a diverse set of fake videos. Some achieved highly satisfactory results, while others were less successful, yielding easily detectable deepfakes. The quality of the output also depended on features like skin tone and facial structure. Figure 6 shows instances of poor results.

To assess the quality of the generated output, each video or frame was compared with the target identity, examining the matching score to determine their similarity. A face recognition library was employed to calculate the matching score using the Euclidean distance between facial features. The results, as shown in Figure 10, indicate that a lower score corresponds to higher quality.

Authorized licensed use limited to: The University of Alabama. Downloaded on March 15,2025 at 02:51:14 UTC from IEEE Xplore. Restrictions apply.

Fig. 6: An example demonstrating the poor quality of a deepfake generated from DeepFake Labs using two identities with distinct skin tones

In the depicted example, the matching score between the generated deepfake and the target identity is 0.269, a commendable score as anything below 0.5 is considered a good match. The figure also presents a probability histogram illustrating the distribution of these matching scores.

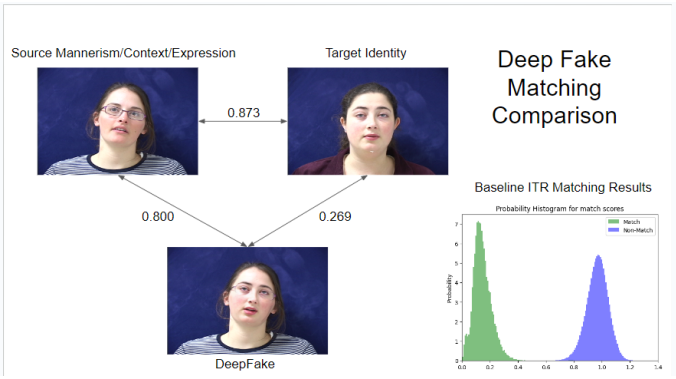


Fig. 7: The figure displays the match scores between the source mannerism, target identity, and the generated deepfake, providing a visual representation of the comparison results.

To implement the face illumination feature, we leveraged an existing camera API that encompasses fundamental Android functionalities. This functionality was seamlessly integrated into an existing camera application, serving as a framework for our feature [23]. Initially, we integrated a basic illumination feature accessible through a single button click, which removed the entire screen's displayed light. This light would then be reflected from users if they were real.

The application's user interface (UI) is divided into two main sections: one dedicated to camera functionality and the other featuring a button for screen illumination, conveniently located on the home screen as shown in Figure 11. In practical scenarios, this feature could be granted to authorized individuals, enabling them to remotely eliminate the screen from their device. However, for demonstration purposes, we integrated this option within the same camera application as part of the UI.

Fig. 8: Figure illustrating the cosine distance between three frames, including Source ID, Destination Action, and the Attack.

Utilizing the gallery option, users can load any previously created deepfake videos. The video is played within the camera interface, simulating a virtual camera and creating the illusion of a real person in front of the camera. This setup has the potential to deceive individuals, as the camera's UI remains unchanged while the display content is sourced from the deepfake video, making it challenging to identify if the data originates from the camera or from a loaded video in the backend.

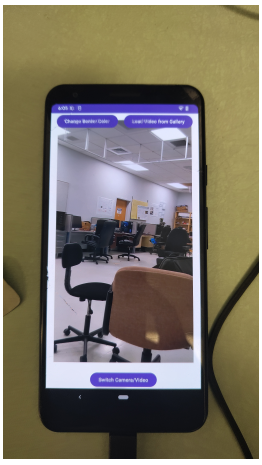


Fig. 11: The developed Android application showcases a home screen that incorporates buttons for loading the gallery and activating the illumination feature.

Authorized licensed use limited to: The University of Alabama. Downloaded on March 15,2025 at 02:51:14 UTC from IEEE Xplore. Restrictions apply.

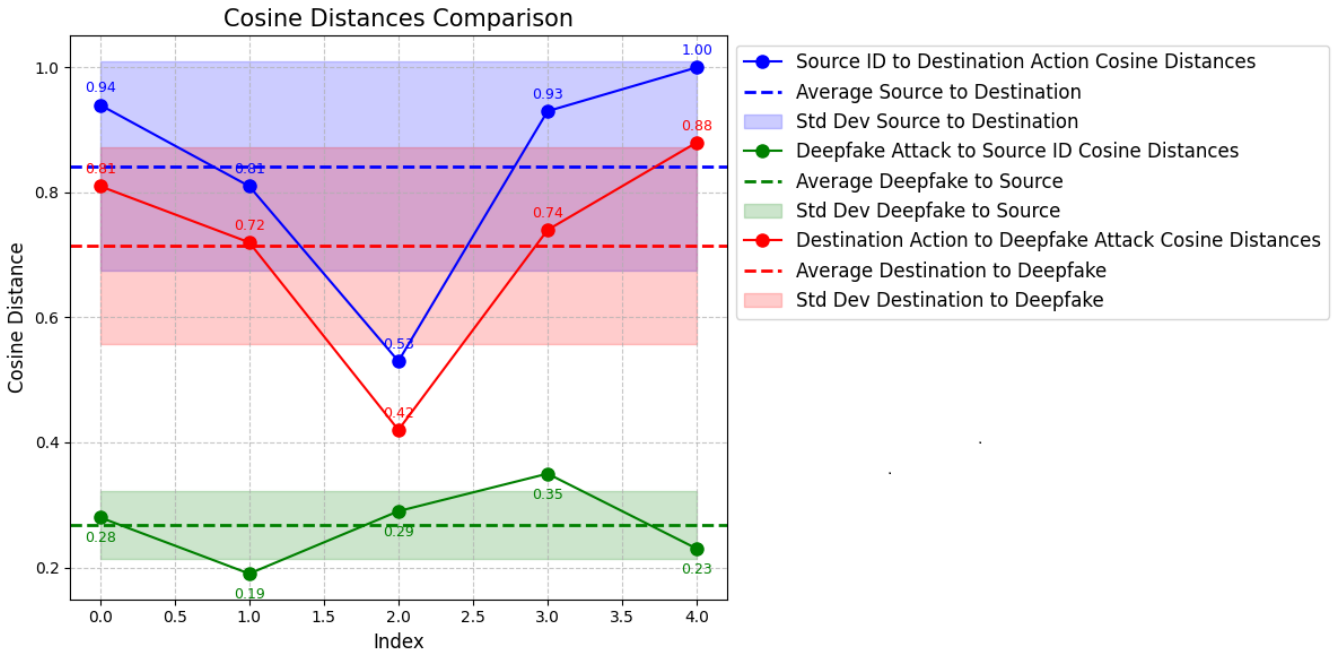


Fig. 9: Figure illustrating the median and average cosine distances of 5 pairs between the source ID and destination, as well as the deepfake created.

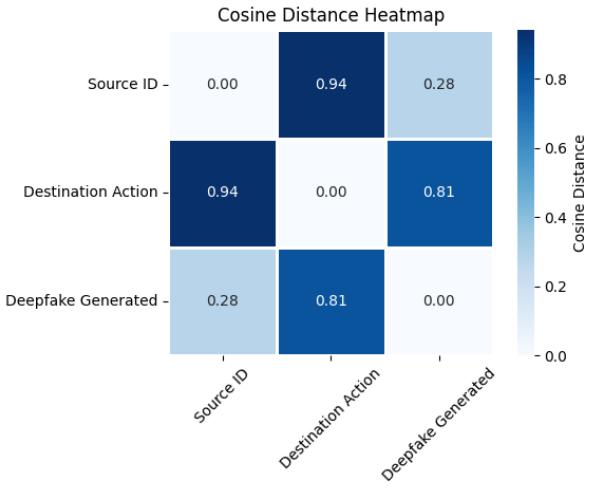


Fig. 10: Cosine Distance Heatmap: This figure shows the cosine distance between the Source, Target, and Deepfake. (Note: Lower cosine distances indicate higher matching accuracy).

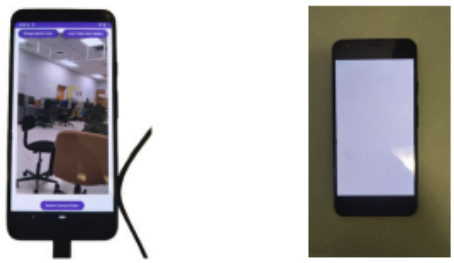


Fig. 12: An illustration of the user interface with and without light illumination. The left side shows the home screen without illumination, and the right side with illumination.

Figure 12 demonstrates the home screen appearance with illumination on and off. Clicking the button in the top-left corner activates the illumination. The impact of illumination on facial recognition is evident in Figure 13. In a dark room without illumination, no light reflects off the face. Conversely, with illumination on, facial reflections occur, a feature not achievable in an injected deepfake video. This distinction aids in identifying a real person versus a deepfake video injected into the device.

IV. Discussion and Future Work

A. Discussion

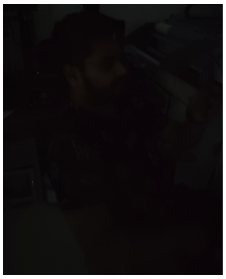
The results clearly indicate that a simple illumination feature, utilizing existing hardware and software, becomes a crucial tool when incorporated into any application. This additional feature proves effective in identifying deep or injection attacks on a device, providing an easy means of detection. This feature utilizes specific pre-existing functionalities from the Android library, ensuring it can be seamlessly integrated into any existing applications.

However, it is essential to note that the effectiveness of this feature may vary based on environmental conditions. In normal light environments, it proves highly effective, but its performance might be compromised in rooms with bright lights or outdoor settings. Nonetheless, it remains a valuable tool for enhancing security in indoor environments with normal lighting conditions.

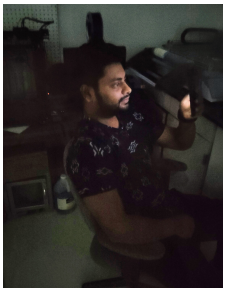
B. Future Work

To advance the technology, we can enhance the existing illumination feature by incorporating multiple colors emitted from the screen in a randomized pattern. This could involve

Authorized licensed use limited to: The University of Alabama. Downloaded on March 15,2025 at 02:51:14 UTC from IEEE Xplore. Restrictions apply.



(a) Image demonstrates how the face appears with the illumination off



(b) Image demonstrates how the face appears with the illumination on

Fig. 13: A figure illustrating the light reflection on the face with the screen illumination off and on.

initiating light emission from one section, such as the bottom part, and progressing to the top. By employing an algorithm that understands the emission pattern, we can integrate deep learning techniques. This allows us to establish and identify the emitted pattern, compare it with the reflected pattern off the face, and significantly elevate security measures. The combination of a random pattern and deep learning integration forms an exceptionally effective tool, offering heightened security against injection attacks or injected deepfakes. This approach proves to be a robust and efficient means of detecting deepfakes using the device's existing hardware.

REFERENCES

[1] A. de Rancourt-Raymond and N. Smaili, "The unethical use of deepfakes," *Journal of Financial Crime*, vol. 30, no. 4, pp. 1066–1077, 2023.

[2] S. Agarwal, H. Farid, T. El-Gaaly, and S.-N. Lim, "Detecting deepfake videos from appearance and behavior," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2020, pp. 1–6.

[3] S. Agarwal, H. Farid, O. Fried, and M. Agrawala, "Detecting deepfake videos from phoneme-viseme mismatches," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2020, pp. 2814–2822.

[4] K. Carta, A. Huynh, S. Mouille, N. El Mrabet, C. Barral, and S. Brangoulo, "How video injection attacks can even challenge state-of-the-art face presentation attack detection systems," in *Proceedings IMCIC-International Multi-Conference on Complexity, Informatics and Cybernetics*, 2023, pp. 105–112.

[5] J. Brockschmidt, J. Shang, and J. Wu, "On the generality of facial forgery detection," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, 2019, pp. 43–47.

[6] H. Farrukh, R. Aburas, S. Cao, and H. Wang, "Facerevelio: A face liveness detection system for smartphones with a single front camera," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2020, pp. 1–13.

[7] G. Frankovits and Y. Mirsky, "Discussion paper: The threat of real time deepfakes," 2023, [Accessed: 14-Apr-2024].

[8] Droidcam. Google Play Store. [Accessed: 25-Jan-2024]. [Online]. Available: https://play.google.com/store/apps/details?id=com.dev47apps.droidcam&hl=en_US&gl=US

[9] ivcam webcam. Google Play Store. [Accessed: 15-Feb-2024]. [Online]. Available: https://play.google.com/store/apps/details?id=com.e2esoft.ivcam&hl=en_US&gl=US

[10] Virtual camera app. Google Play Store. [Accessed: 10-Dec-2023]. [Online]. Available: https://play.google.com/store/apps/details?id=virtual.camera.app&hl=en&gl=US&pli=1

[11] XDA Forums. (2024) Pixel 6 pro easy step-by-step unlock, root, update. [Accessed: 19-Nov-2023]. [Online]. Available: https://xdaforums.com/t/pixel-6-pro-easy-step-by-step-unlock-root-update

[12] Wikipedia, "Magisk (software)," 2024, [Accessed: 16-Jun-2024]. [Online]. Available: https://en.wikipedia.org/wiki/Magisk_(software)

[13] Avatarify. Softonic, [Accessed: 12-Apr-2024]. [Online]. Available: https://avatarify.en.softonic.com/

[14] Avatarify github repository. GitHub, [Accessed: 27-Mar-2024]. [Online]. Available: https://github.com/alievk/avatarify-desktop

[15] Synthesia. [Accessed: 8-May-2024]. [Online]. Available: https://www.synthesia.io/

[16] Flawless ai. [Accessed: 17-Jun-2024]. [Online]. Available: https://www.flawlessai.com/

[17] Stability ai. [Accessed: 28-Nov-2023]. [Online]. Available: https://stability.ai/

[18] Dall-e-2. OpenAI, [Accessed: 5-Jan-2024]. [Online]. Available: https://openai.com/dall-e-2

[19] Pinscreen. [Accessed: 20-Feb-2024]. [Online]. Available: https://www.pinscreen.com/

[20] J. Smith and A. Johnson, "rlivedet 2021: A dataset for liveness detection in facial recognition," Research Data Repository, 2021, [Accessed: 9-Feb-2024].

[21] I. Kim, S. Park, and T. Yoo, "Deepfacelab: Integrated, flexible and extensible face-swapping framework," *arXiv preprint arXiv:2005.05535*, 2020.

[22] Meta AI. Casual conversations v2 dataset. [Accessed: 2-May-2024]. [Online]. Available: https://ai.meta.com/datasets/casual-conversations-v2-dataset/

[23] andvipgroup, "Vcamera," https://github.com/andvipgroup/VCamera, [Accessed: 7-Mar-2024].

Authorized licensed use limited to: The University of Alabama. Downloaded on March 15,2025 at 02:51:14 UTC from IEEE Xplore. Restrictions apply.