PIN-a-Boo: Revealing Smartphone PINs via Segmentation and Hand Skeleton Tracking from Video Feeds

Patrick Weich
Oleg Lobachev
Leibniz-Fachhochschule
Expo Plaza 11, 30539 Hannover, Germany

Deutsche Telekom
Security GmbH
Bonner Talweg 100
53113 Bonn
Germany

Technische Hochschule
Brandenburg
Magdeburger Str. 50
14770 Brandenburg
Germany

Abstract

It is crucial to improve smartphone security, given the prevalence of sensitive information stored on them. This study presents an attack strategy that reveals smartphone PIN entries using computer vision and pattern recognition techniques. By leveraging modern segmentation and hand skeleton tracking, our method accurately identifies and analyzes finger movement patterns, even when partially obscured. We can reliably infer the entered PIN by combining these movement patterns with the smartphone's position and the on-screen keypad layout. This approach significantly enhances shoulder-surfing attacks, requiring only a video recording of the entry process. Our attack requires much less specialized expertise, making it more accessible. We conclude by analyzing the method's potential impact and its implications for public safety.

Keywords
computer vision, pattern recognition, smartphone security, shoulder-surfing, video-based attack, user privacy

1 INTRODUCTION

Smartphones have become indispensable in daily life, offering a plethora of applications while storing vast amounts of sensitive data. This convenience, however, increases security risks, particularly with the growing usage of smartphones. The advancement in technologies, including Artificial Intelligence (AI) (Schneier, 2021), necessitates robust security measures not just at the device level but also in enhancing user awareness regarding potential threats. This paper aims to contribute to this awareness by exploring an attack on smartphone PIN (personal identification number) entries.

Our feasibility study explores the intersection of IT Security and Computer Vision (CV) to develop and implement an attack methodology against PIN entries on smartphones using mostly off-the-shelf components. We aim to determine how advancements in technology have made such attacks more accessible compared to, e.g., Shukla and Phoha (2019). The primary goal is to devise an attack mechanism capable of identifying PIN entries on smartphones through finger movement analysis using modern CV techniques and limited resources. We discuss the feasibility, attractiveness, and profitability of these attacks for potential adversaries. Additionally, we identify defensive measures to mitigate the associated risks, enhancing the overall security of mobile devices (viz., e.g., Harbach et al., 2014).

Our approach is basically predictive analytics of PIN entry from a video feed. The priors are pre-existing knowledge on the shape of a smartphone (for segmentation) and on possible hand skeleton movements (for finger tracking).

1.1 Scenarios for PIN Entry on Smartphones

PIN entry on smartphones is commonly used in multiple scenarios. Upon booting, multifactor authentication involves the SIM card for possession-based authentication and a PIN for knowledge-based authentication, typically starting with a 4-digit PIN that can be customized. For unlocking devices, users can choose PINs, patterns, or biometric methods such as facial or fingerprint recognition. Preferences have shifted towards biometric authentication, though many users still rely on PINs. Even with biometric authentication, a PIN is required as a backup, underscoring its continued importance in smartphone security. Wang et al. (2020) provide a good overview

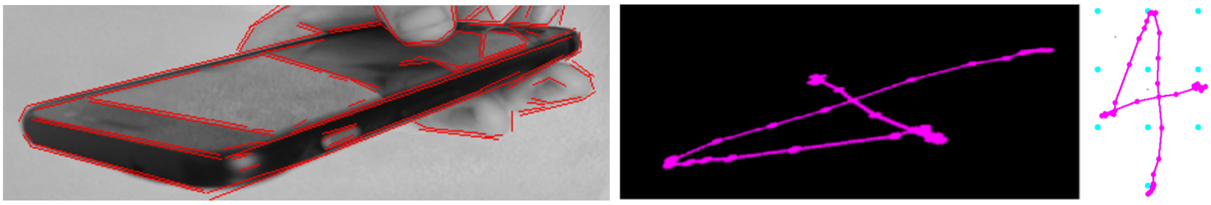


Figure 1: The essence of our method, we guessed the PIN 6 7 2 0 from a video feed of finger movements.

of the topic. Certain applications, particularly banking apps and identity verification services such as German AusweisApp2 (Willomitzer et al., 2016), specifically require PIN authentication, often limited to four or six digits for user convenience.

The selection of a PIN is generally left to the user, with recommendations favoring high-entropy, random-like sequences for security. However, memorability often takes precedence over security in practice, with users opting for significant dates or simple sequences for ease of recall (Markert et al., 2020). This approach leads to commonalities among users' PINs, with the most popular 5–8% of PINs accounting for more than half of all PINs used (see, e.g., Wang et al., 2017). The preference for convenience also results in PIN reuse and reluctance to change PINs even after compromise, reducing security levels significantly. Many users justify their choice of simple, short PINs by the reduced risk of forgetting them, prioritizing ease of use and minimal entry time over security, which inadvertently lowers the barrier for potential attackers.

Smartphones, given their ubiquity and the sensitive data they hold, are particularly appealing targets for attackers. Users often prioritize convenience over security, making smartphones inadequately protected in many cases. Add a lack of awareness among some users about proper security measures, and the result is a high interest of malicious actors, leading to various forms of attacks.

1.2 Traditional Shoulder-Surfing Attacks

Next, we discuss the "usual", non-CV-assisted shoulder-surfing attacks. We explore their risks, consequences, and limitations.

Everyday Risks

Shoulder-surfing attacks, a subset of observational attacks, involve unauthorized viewing of content displayed on a device's screen, most commonly occurring in public spaces such as transit systems or cafés (Bošnjak and Brumen, 2020; Bâce et al., 2022). While not every instance of shoulder-surfing is malicious, the practice can significantly compromise privacy and security. Users' countermeasures, such as turning the screen away, are rarely observed even during sensitive inputs, highlighting a general lack of awareness.

Malicious actors exploit this lack of vigilance by observing users' interactions with their devices and noting down login credentials. The observer might be a stranger with a single opportunity or an insider with repeated access, enhancing the chances of successful credential acquisition. Attackers often use cameras or smartphones to zoom in and record the information, leveraging technology to their advantage.

Consequences of PIN Compromise

Shoulder-surfing, primarily aimed at spying on secrets like smartphone PINs, is often just the initial step in a broader malicious agenda. Subsequent attacks might include attempts at reusing the compromised credentials across various platforms in what is known as a Reused Credential Attack or Credential Stuffing. The simplicity and reuse of PINs can lead to unauthorized access to valuable assets, such as bank accounts or even personal properties.

If an attacker physically acquires the smartphone, the compromised PIN allows full access to the device's functionalities, stored data, and applications. Beyond the potential resale of the device, the attacker might exploit sensitive data for identity fraud (Credential Theft), sell the information (Credential Trading), or use it for extortion. The device could also be compromised further through malware installation for surveillance or inclusion in a botnet, broadening the scope of the threat.

A comprehensive overview of potential mobile security threats and attack vectors is available through resources such as MITRE ATT&CK: `https://attack.mitre.org/matrices/mobile/`, offering insights into the complexity of mobile security and the importance of robust protective measures

Limitations

Despite the potential severity of consequences of follow-up attacks after a successful shoulder-surfing attempt, these attacks inherently have significant limitations. The primary limitation is the physical proximity requirement. The attacker needs to be close enough to the victim, often within their personal space, to observe the PIN entry. Surveillance with the naked eye becomes ineffective over larger distances, and even smartphone cameras have limitations in terms of distance and quality necessary for creating usable recordings for the analysis by attacker. This proximity increases the risk of the attacker being caught.

Moreover, there is no guarantee of a reliable PIN capture. The victim might hold their phone in a way that obscures the view or makes the PIN entry hard to discern. Environmental factors, such as bystanders or movements in public transportation, can also affect the success of the attack, making it challenging to capture a clear recording. Another critical factor for many subsequent attacks is gaining physical access to the smartphone, which requires the device to be compromised or stolen. This adds another layer of risk for the attacker, who must invade the victim's personal space again, this time to steal one of their most personal possessions, ideally without being detected.

These challenges also affect the scalability of such attacks. An attacker can typically only focus on one target at a time in a given location. Attempting to observe multiple targets, possibly recording them and paying attention to PIN entries for quick analysis, requires much more effort and caution. The quality of PIN capture is likely to decrease.

1.3 Relevance

Our research addresses an opportunity in the security landscape by developing an attack strategy that leverages advancements in CV to expose vulnerabilities in PIN entry systems on smartphones. This work highlights the ease with which sophisticated shoulder-surfing attacks can be executed using available CV technologies. Such technologies are available to anyone. By employing modern segmentation and hand skeleton estimation methods, our approach can accurately track finger movements, even under partially obscured conditions. Thus, we are able to reveal the entered PIN with sufficient precision. Our method resulted in 44% success rate on a quite harsh dataset, which is comparable with other similar approaches (e.g., Cardaioli et al., 2022). The implications of this are far-reaching, suggesting that conventional PIN-based security measures may no longer suffice in protecting against modern threats. We reiterate that the means of reaching above goals are rooted deep in CV. This work is based on first author's thesis (Weich, 2023). The source code is available on GitHub under `https://github.com/PatP41/PIN-a-Boo`.

2 LEVERAGING THE COMPUTER VISION

To address the limitations of traditional shoulder-surfing attacks, integrating CV technology offers a significant enhancement. This approach eliminates the need for proximity to the target and improves the scalability of the attack.

PIN-based authentication and shoulder-surfing has been discussed by Lee (2014); Kwon and Hong (2015); Anthonio and Kam (2020); Kobayashi et al. (2020); Khan et al. (2018) among others. Surveys of PIN-entry (Binbeshr et al., 2021) and of graphical passwords (Por et al., 2024; Binbeshr et al., 2024) in the context of shoulder-surfing provide additional information.

Riyadh et al. (2024) are concerned with authentication in virtual reality. Singh and Koundal (2024) recognize "air-writing" (arguably, a more complicated category of gestures) with AI. Bhole et al. (2024) present a two-factor authentication method for visually impaired.

Security and Applications of CV

Exploring various attacks on knowledge-based authentication methods, particularly PIN entries, reveals the diverse application of CV and Machine Learning (ML) techniques in enhancing traditional methods. These attacks are broadly categorized below.

- *Attacks on visible hand movements:* These attacks do not require visibility of the keyboard or screen but rely on observing hand movements. For instance, Shukla and Phoha (2019) proposed an attack that deduces passwords from spatial and temporal hand movement information during input on a full-screen keyboard, extending beyond mere PIN recognition to include actions like pressing the shift key. This attack functions from a distance, reducing the need for the attacker's proximity to the victim. Another approach (Cardaioli et al., 2022) focuses on ATM (automated teller machine) PIN entries, achieving over 40% accuracy in recognizing PINs even when the user partially covers their hand, by analyzing movements of lower finger joints instead of just the fingertip.

- *Attacks on visible screens:* Yue et al. introduced two approaches where the first (Yue et al., 2014) involves identifying touchpoints on the screen using the shadow around the fingertip. The second (Yue et al., 2015) builds upon the first but employs language models to reconstruct English text inputs, aiming to disclose confidential emails. Chen et al. (2018) utilized foundational CV techniques to significantly enhance shoulder-surfing attack success rates.

- *Attacks utilizing display feedback features:* Maggi et al. (2011) proposed an attack exploiting the display feedback feature of Apple smartphones, which momentarily enlarges the pressed key, making it especially vulnerable to shoulder-surfing. This feature, typically enabled by default, allows for rapid processing and evaluation when automated.

- *Reflection Attacks* utilize reflections on objects like glasses or spoons to reconstruct on-screen information without needing to be close to the victim. This category includes techniques (Raguram et al., 2011; Xu et al., 2013) that reconstruct passwords or PINs from reflections and are less likely to attract attention, offering a stealthier approach to gathering sensitive information.

- *Smudge attacks:* Highlighting the traces left by fingers on touchscreen devices, these attacks derive graphical passwords from the smudges (Aviv et al., 2010). Information like movement direction and intensity can be gleaned from these residues, which remain largely unaffected by everyday actions like placing the phone in a pocket.

Further related work includes Corbett et al. (2024), where shoulder-surfing is detected using eye-tracking; Länge et al. (2024), where shoulder-surfing resistant authentication for virtual reality is presented; Imran et al. (2024) develop a shoulder-surfing resistant authentication using graphical passwords.

Other similar uses of the technology include sign language recognition (e.g., Al-Hammadi et al., 2020) and therapy applications (Rungruanganukul and Siriborvornratanakul, 2020).

Our attack aims to recognize smartphone PIN entries based on finger movements using existing CV techniques and limited resources. This work follows on the "visible hand movements" attack category, but aims to test the feasibility of an attack using available, off-the-shelf CV components. A possibility to do so makes the attack much easier for malicious actors. Further, we use only a monocular camera feed for a similar reasoning. (A fusion of multiple modalities is possible, see, e.g., Kalamkar and Amalanathan (2023), but it is not our focus.) Basically, if the above is feasible, then the attack no longer requires a researcher in CV and IT Security, but merely a skilled software developer.

3 DEVELOPMENT OF THE ATTACK

First, we discuss the assumptions and a possible high-level execution of the attack. The next sections will detail the utilized CV methodologies and technical requirements to execute our attack. Fig. 2 provides an overview of the CV pipeline.

3.1 Assumptions

In our scenario, it is assumed that the attacker's goal is to access sensitive data stored on a smartphone. The attacker is presumed to have a way to physically acquire the smartphone but needs the victim's PIN to unlock it. Key assumptions for the attack include:

- The attacker can record the victim's screen unlocking process. The video quality must be sufficient to apply CV techniques effectively. Ideally, the recording should be made from a distance no greater than 2.5 meters using a smartphone camera.

- The video should prominently feature the smartphone, with at least a portion of it visible within the frame. It is crucial that the finger movements, particularly those of the entering finger, are clearly observable. While the fingertip can be partially obscured, significant obfuscation of movements should be avoided.

- The PIN entry is assumed to be conducted using a single finger, typically the thumb, with the specific hand (right or left) being inconsequential.

- Environmental lighting conditions should neither be too bright nor too dark, ensuring clear visibility of finger movements and the smartphone screen. We include an evaluation with various scenes also without a complete visibility of either screen or of fingers or of both (see Sect. 4.2).

- The attacker must be familiar with the numeric layout typically used for PIN entries on smartphones.

3.2 The Attack

The essence of the attack is: The movement trajectory traced by the finger's motion, as observed from the camera's perspective, is analyzed to deduce potential keypresses. This process begins with capturing the movement trajectory and then transforming it to align with the user's perspective. Subsequently, the transformed trajectory is mapped onto a virtual keyboard to facilitate the identification of potential keypresses. The final step involves analyzing clusters of finger positions across consecutive frames to accurately deduce the corresponding PIN digits. The attack itself (in the broader scope) consists of:

1. *Initialization and preparation:* Select a public or semi-public location where individuals frequently use their smartphones, such as cafés, public transport, or waiting areas. The attacker prepares a high-resolution smartphone or a camera capable of recording detailed video. They also obtain or develop the software from this paper.

2. *Target selection:* Observe potential victims: The attacker monitors individuals using their smartphones, paying attention to those who frequently unlock their devices in view of others. They assess environmental conditions, pay attention to the visibility of the smartphone screen and of finger movements.

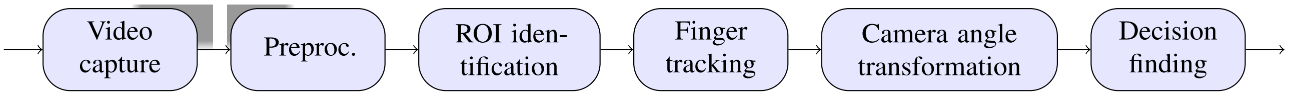


Figure 2: An overview of our method

3. *Recording:* The attacker positions themself discreetly within a suitable distance from the target (recommendation: up to 2.5 meters). Capture the video.

4. *Video pre-processing:* The recorded video is stabilized and enhanced in its clarity if necessary. We need to extract key frames for segmentation.

5. *Computer vision analysis:* As detailed in this paper, the attacker applies CV to track the movements of the finger and correlate them with the positions of the numbers on the keypad. In this manner they identify the PIN digits.

6. *Verification and refinement:* PIN sequence is verified by cross-checking the extracted PIN with common entry patterns; consistency is ensured in the detected movements. Optionally, the attacker simulates multiple PIN entries on a test device to verify the accuracy of the deduced PIN.

7. *Exploitation:* The attacker physically acquires the target's smartphone (if not already in possession). They unlock the device using the deduced PIN.

8. *Benefit realization:* Now it is possible to access and extract sensitive data stored on the device, such as personal information, financial data, or confidential communications. Optionally, the attacker can install malware or spyware on the device to maintain long-term access or monitor the victim's activities, if the smartphone is to return to the victim. The attacker can exploit the accessed data for financial gain, such as identity theft, unauthorized transactions, or selling the information on the black market.

In a nutshell, the attacker systematically leverages shoulder-surfing and computer vision techniques to compromise the security of smartphone PIN entry, ultimately gaining unauthorized access to sensitive data and resources. Next, we regard the CV-related parts of the attack, following Fig. 2.

3.3 Preprocessing

During preprocessing, video segmentation isolates key frames related to the phone unlocking process, identifying a tuple that marks the start and end of the PIN entry. To avoid errors, everything apart from the unlocking process should be excluded from the video.

Initially, velocity analysis was considered to identify the beginning and end of the PIN entry by detecting when the fingertip's velocity starts or drops to zero, respectively. However, this proved unreliable due to the constant motion of the fingertip and the difficulty in establishing a meaningful threshold.

The current method determines the start of the PIN entry when the finger enters the region of interest (ROI) and stops when the finger leaves the ROI. While this method works adequately in test environments, manual adjustments are still needed, particularly at the end of the PIN entry, as users in the wild typically continue using the phone after the unlocking process. This suggests the need for more refined and automated techniques in this stage.

3.4 Identification of the Region of Interest (ROI)

In the context of the proposed attack, the ROI is primarily focused on the fingertip, with at least the edges of the smartphone also necessary for later transformations related to the camera angle. Given the proximity of the fingertip to the smartphone during input, it is pragmatic to use object detection to locate the smartphone and designate this area as the ROI. We experimented with Canny edge detector and Haar feature cascade classifier, however, in the end a different approach was used.

We used a pre-trained object detection model. YouOnlyLookOnce (YOLO) is a popular one-stage object detection model (Redmon et al., 2016), notably for its high speed. Specifically, YOLOv8 by Ultralytics: `https://github.com/ultralytics/ultralytics` was used, which includes a pre-trained "Cell Phone" class in one of its models. This decision follows our general theme of using off-the-shelf components.

However, it is important to note that YOLO is trained for a wide range of classes and recognizes many classes. For optimal results, the pre-existing pre-trained model could be further trained to focus solely on the "Smartphone" class, which would significantly enhance efficiency. The current implementation could be optimized more, but this was not detrimental for the subsequent steps of the attack. The detected object was marked with a bounding box and classification label. The coordinates of the bounding box's corners were then used to define the region of the original image, thus establishing the desired ROI for the next phase of the attack.

3.5 Finger Tracking

For our attack, which aims to recognize PINs through finger movements, tracking these movements accurately is crucial. The precision of finger tracking directly impacts the reliability of the results.

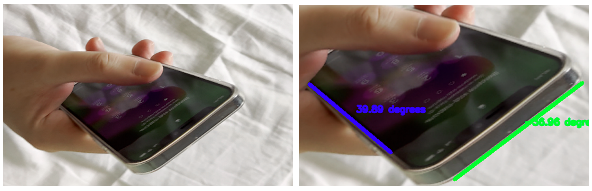
Initially, the approach of extracting finger contours, similar to Chen et al. (2018), was tested. This method leverages the fact that human skin's HSV values fall within a similar range despite variations in skin tone. An acceptable outcome was achieved using erosion, dilation, and a mask to isolate the finger. Contours were then defined to determine the fingertip's location. This technique works well when the fingertip is clearly visible and under adequate lighting conditions. However, due to the occasional absence of these conditions, alternative methods were explored.

MediaPipe (Lugaresi et al., 2019), a framework by Google for AI applications, offers various CV solutions. For this attack, the 'Hand landmark detection' feature was employed, which starts by identifying the palm before mapping 21 coordinates to the hand's joints, thereby calculating 3D coordinates for each hand segment. The decision to use MediaPipe was influenced by these 3D coordinates, anticipating that future transformations would be significantly simplified or potentially unnecessary.

The integration of MediaPipe into the existing concept was successful. With MediaPipe providing access to all 3D coordinates at any given time, it was possible to incorporate a check to determine the primary finger interacting with the smartphone. This finger is then monitored for PIN entry. For this implementation, the thumb was presumed to be the preferred finger for PIN entry, hence the thumb tip was used as the landmark for tracking the movement path. To illustrate the movement trajectory, the position of the thumb tip was plotted in each frame and connected to its position in the preceding frame (viz. Fig. 1). Note that the movement path was tracked from the camera's perspective. Before the trajectory can be analyzed, it needs to be transformed to account for this perspective.

3.6 Transformation of the Camera Angle

To derive the movement trajectory from the smartphone user's perspective, a transformation of the recorded path is required. The choice of MediaPipe for this task was partly due to its capability to output landmarks in 3D coordinates. This feature enables the normalization of the movement path using the edges of the smartphone. Initially, the Line Segment Detector (LSD) (von Gioi, 2014) was employed for line detection. However, for the final angle calculations, the LSD was replaced by the Hough Line Transform (Aggarwal and Karl, 2006), as it provided slightly more precise results for this specific purpose. However, we found out that z-coordinates provided by MediaPipe's 3D coordinates are not very precise. The estimation of the z-coordinates in space did not yield satisfactory results in the tests conducted. Attempts to filter out inaccurate movements using thresholds did not lead to improvements.



(a) Screenshot from the video. (b) Direct output of our angle estimation.

Figure 3: Estimating the camera angle.

Ultimately, rotation matrices were employed for transforming the movement trajectory. In this case, they were used to transform coordinates based on the camera angle. The initial step involves determining the angle relative to the camera. For instance, if the top end of the smartphone points directly at the camera, the angle would be 0°. With this baseline, the angle between the user and the camera can be easily calculated using one of the detected lines, as Fig. 3 shows. To give a further example, the angle in the scenario of Fig. 1 is approximately 70°. This angle is recalculated in each frame to account for any changes during the unlocking process. In this manner, we obtain the movement trajectory from the user's perspective.

3.7 Decision-Making

The decision-making phase is the final part of the workflow, where the PIN is reconstructed based on the information gathered in the previous steps. A significant challenge is determining whether the recorded path covers the entire numpad or only a small portion of it, which could be addressed by calculating the path's bounding box, normalizing the path points and by analyzing the width and height to determine the usage area.

The touch interface is divided into predefined regions, each representing a numeric key with specific normalized coordinate ranges. This depends on the smartphone's specific layout. Our approach involves evaluation of the normalized path by clustering proximate positions to identify stable fingertip locations within these numeric regions, and detecting transitions between clusters to infer the PIN. For each fingertip position, we compute the Euclidean distance to the last point in the current cluster; if this distance is below a defined threshold, the point is added to the cluster. Once a point falls outside this threshold, we evaluate the current cluster: if its size meets a minimum threshold, we calculate the centroid and map it to the corresponding numeric region, recording the detected number if it is not a duplicate of the last one detected. In cases where a PIN contains the same number consecutively, we handle this by checking if the cluster size is significantly larger than usual and then splitting the cluster into two separate clusters to detect the multiple presses. This process ensures that only stable fingertip positions within a numeric key's region are considered keypresses, allowing for accurate PIN detection from dynamic touch input paths. However, a significant challenge arises from the variability in user behavior during PIN entry, such as inconsistent breaks between keypresses and varying speeds of input.

3.8 Mathematical Description of the Attack

Our approach can be divided into the following abstract tasks. First, we apply preprocessing and temporal segmentation as described above. These steps are less foundational in nature. We then reconstruct the trajectory of the input finger cup G, represented as a sequence of 3D points over time. When multiple consecutive finger cup positions change only minimally, we consider the finger to be stationary. Next, we estimate the homography matrix H required to transform the camera's view into a normalized, frontal perspective of the smartphone screen. We use angle estimation and object detection methods as described above (see also Fig. 3), but other approaches could also be applied. In the final step, we compute the normalized trajectory $G' = H \cdot G$. By mapping the finger cup's points of contact onto the numpad in this normalized space, we can identify the digits pressed and thus reconstruct the entered PIN.

4 EVALUATION OF THE ATTACK

Next, we conduct an evaluation to assess the viability and potential effectiveness of the attack. This evaluation aims to determine the type of attacker for whom this method might be worthwhile and to propose possible defense mechanisms against such an attack. We also look into the success rate of our method in a controlled setting.

The complexity and technical requirements of this attack would suggest it is most suited for attackers with a sophisticated understanding of CV technologies. However, little development of own CV method is required, as a sufficiently successful attack can be composed from existing CV libraries. Several defense mechanisms can mitigate the risk of this type of attack:

1. *Behavioral dynamics:* Encouraging users to employ unpredictable finger movements or gestures that obscure the actual PIN entry can complicate the tracking process.

2. *Screen privacy filters:* Utilizing physical screen filters that narrow the viewing angle of a smartphone screen can prevent potential onlookers or cameras from capturing the PIN entry process. While our method does not need to see the screen, a decrease in visibility could undermine, e.g., the detection of non-standard PIN keyboard by an attacker.

3. *Biometric authentication:* Shifting from PIN-based authentication to biometric methods (fingerprint, facial recognition) can eliminate the vulnerability altogether, though it is crucial to consider the security and privacy implications of biometric data.

4. *Randomized keyboard layouts:* Implementing a dynamic keyboard layout that changes the position of digits each time a PIN is entered can render the tracking of finger movements ineffective.

4.1 Ethics

An automated shoulder-surfing attack is clearly unethical. An interception of credentials can violate privacy laws. Just as peering over someone's shoulder to read their private information is unacceptable, using CV methods for the same purpose is equally problematic. Moreover, CV-based methods can be replicated as software systems on an economy-of-scale basis—organized crime could invest in developing such a system once and then deploy it widely. The crucial point of this work is: a sufficiently effective attack no longer requires specialized expertise in IT Security or CV; it can be implemented using standard off-the-shelf components.

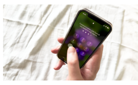
Precisely for these reasons, research on such attacks must continue. Developing better countermeasures requires a baseline attack for comparison, and raising awareness among general users—while teaching proper defense mechanisms—is equally important. To protect the privacy of potential victims and to establish a viable baseline, we generated our own dataset for this study.

All the PINs we used were created specially for the purpose of this study. Most of the PINs were random. Both phones (the one used for PIN entry and the one capturing the video feed) belonged to us, so no personal information or PINs were compromised. No ethics committee statement was required for the above reason. No leverage of CV to infer additional personal data was possible in our test scenario.

4.2 Evaluation on a Synthetic Dataset

For this study, a small dataset of PIN entries was specifically prepared and evaluated. The dataset consists of PIN entries on an iPhone 12 Pro, distinct from the device used before. It includes random numerical sequences and a subset simulating typical PIN patterns, such as calendar dates and plausible birth years. All PINs had four digits. Many of the hand and smartphone poses were not flattering our method, with screen not visible or fingers occluded. All videos were recorded on an iPhone 15 Pro in 4K at 60 fps under natural indoor lighting with a bright, unobtrusive background. Detailed information

Table 1: Evaluation dataset and results of our method. "R?" stand for "is random?", "F?" means "is the finger is fully visible?", "S?" means is the screen is fully visible?, "Eval" stands for evaluation result.

No.	Sample	R?	F?	S?	Eval
1		yes	no	no	success
2		yes	yes	yes	success
3		yes	yes	badly	failure
4		yes	no	no	failure
5		yes	no	no	mixed
6		yes	yes	no	failure
7		no	yes	yes	success
8		no	badly	no	failure

on the videos and the results of our method applied to this dataset are presented in Table 1. The videos were not altered of pre-processed.

In our dataset 75% of the PIN entry videos had a truly random PIN; in about 56% the screen was visible to the attacker, in about 31% was the fingertip visible. Application of our method resulted in about 44% overall success rate. This result is comparable with Cardaioli et al. (2022). MediaPipe is able to track the fingertip in some quite hard cases, such as #1 and, partially, #5. Bad viewing angles (present abundantly in our dataset, e.g., #4, #8) can pose a problem for our instance of YOLO. Fine-tuning of the model might improve on this issue. Notice, however, that all the "typical" scenes for shoulder-surfing were handled correctly, and also some that would be hard to impossible for a human.

4.3 Attack Potential

The development of the attack concept was largely successful, with its potential deemed promising despite the lack of a polished smartphone app, primarily due to time constraints rather than complexity. From an attacker's perspective, the execution mirrors that of a traditional shoulder-surfing attack, but with the significant advantage that the evaluation process is automated through CV, eliminating the need for manual analysis by the attacker. This automation allows even those without specialized knowledge to deploy the attack once it is developed into a smartphone app; they only need to provide suitable video material.

Reliable recognition enables the attacker to record video from a distance, reducing risks such as third-party intervention or detection by the target. The quality of the camera and its zoom capabilities extend the potential distance from which the attacker can operate while maintaining necessary video quality. Alternatively, hidden or small cameras placed near the target could minimize exposure or allow the attacker to inconspicuously capture better footage on-site. Since the attack requires only the visibility of finger movements and not the smartphone screen content, the attacker has greater flexibility in positioning relative to the target.

An implementation of the attack directly on the attacker's smartphone could allow for live evaluations, with all essential libraries already available in an Android environment. This would enable the attacker to adapt, such as changing positions or making multiple recordings in succession, and make immediate decisions regarding potential smartphone theft. Summarizing, this attack represents a significant enhancement over traditional shoulder-surfing techniques, offering increased flexibility and reduced risk for the attacker.

4.4 Discussion

Shoulder-surfing attacks often occur in public transport, where camera shake during video recording can be an issue. Although the procedures were tested in a controlled and stable environment, and a method to handle camera shake was introduced, such countermeasures are only effective up to a certain extent; excessive shaking severely impacts the evaluation. An application of existing video stabilization methods (Wang et al., 2023) might be of a benefit.

Lighting conditions significantly affect the evaluation's quality. Detection issues of both the smartphone and fingers arose in overly bright or dark environments, greatly increasing the error rate in subsequent computations. The impact of lighting is highly dependent on the camera used. Tests with a smartphone camera (OnePlus Nord2) and a mirrorless system camera (Sony Alpha 5000) unsurprisingly showed that the smartphone produced much poorer quality in low light. The camera quality also influences the maximum feasible distance from the target, with the Sony camera's zoom proving more reliable. However, using a camera for recording is more conspicuous than using a mobile phone in real-world scenarios.

With CV handling the evaluation, the risk of human error is replaced by the potential for AI errors, making the reliability of the results dependent on the AI techniques' implementation. While MediaPipe is generally reliable for finger tracking, it is not infallible, with occasional detection errors and frame jumps observed. To maintain a clear view of the input finger (see Table 1) is of importance; while MediaPipe can estimate the position of the fingertip using the finger joints, the data quickly becomes inaccurate if too much of the input finger is obscured.

Users have varying preferences for PIN entry, and while the attack can evaluate both left and right-handed inputs and is indifferent to the choice of input finger, it does not yet cover PIN entries made with two fingers.

The most volatile part of our pipeline is the search for the correct angle to transform the trajectory to the user perspective. It should be improved in the future, with more advanced methods. All other parts of the pipeline are quite stable, as we found out in our experiments.

The scalability limitation is significantly mitigated with CV evaluation, allowing faster and parallel processing of video material. The attack also improves the scalability of video material acquisition since capturing extensive views of the screen and fingers is unnecessary, offering the attacker more flexibility in recording or the option to use hidden cameras, given the broader tolerance for the recording angle. However, to access the smartphone for potential subsequent attacks, physical possession is still required. The application of CV merely replaces the video material evaluation aspect, necessitating the attacker's proximity to the target.

While the proposed attack demonstrates an application of CV technology to bypass smartphone security, its practical implementation faces significant hurdles, including the requirement for high-quality video capture and the potential for user behavior to thwart the tracking process. The development of robust defense mechanisms further challenges the attack's feasibility, suggesting that its application may be limited to highly targeted scenarios where alternative methods of attack are not viable.

It appears that alternative authentication methods—such as passwords ("alphanumeric PINs"), randomized PIN keyboards, graphical PINs, or biometric authentication (e.g., FaceID)—offer much stronger security against large-scale, CV-based shoulder-surfing attacks. These methods should be recommended to the general public.

4.5 IT Security Evaluation

The attack described is feasible under somewhat favorable conditions, but does not currently threaten the average smartphone user on a large scale. Rather, it poses a realistic risk for high-value targets, such as individuals of interest to state intelligence agencies or organized crime, and may be employed by adversaries able to automate and scale video processing.

A key concern is the low entry barrier: building such a system no longer requires specialized computer vision research but can be achieved by a moderately skilled programmer using existing libraries. The subtlety of recording finger movements from a distance, without needing direct shoulder-surfing proximity, makes the method especially appealing to potential attackers.

While it is unlikely that smartphone manufacturers will implement technical countermeasures anytime soon, users can protect themselves through the same practices advised for traditional shoulder-surfing attacks—namely, covering inputs or otherwise preventing direct observation when entering sensitive information. User vigilance thus remains the primary defense.

5 CONCLUSION, OUTLOOK, AND REFLECTION

Smartphones, with their widespread use and storage of sensitive data, are attractive targets for malicious actors. User behavior often compromises authentication security due to a preference for convenience and a lack of security awareness. Traditional shoulder-surfing attacks, despite their potential for significant harm, are less viable due to the need for proximity and clear visibility of the device, which presents practical challenges for attackers.

Our current threat model involves using a commodity smartphone with a monocular video feed, as previously discussed. In this scenario, a network of opportunistic criminals could capture video feeds of potential targets, deduce the PIN as described here, and then decide on further criminal actions—such as stealing the phone once the PIN is known. Another concern is state-level surveillance that could involve large-scale PIN inference from video feeds. However, since surveillance state actors typically have more advanced methods at their disposal, we consider this issue to be of lower priority.

CV offers a solution to the limitations of traditional shoulder-surfing, allowing the proposed attack to use basic CV techniques widely available in current applications. The close relationship between AI and cybersecurity today makes AI a dual-use technology: it enables new or enhanced attacks for malicious actors and serves as a crucial tool for defenders to address emerging threats.

Our attack was conceptualized to be a modular approach, with each step reflecting the attack process, similar in coarse steps to existing approaches (Yue et al., 2014). With basic CV principles and publicly available libraries promising solutions were developed and tested for each step. This success indicates not only the attack's feasibility but also its ease of execution.

The use of off-the-shelf components makes our method accessible to criminal actors. Our method enables discreet and flexible attacks on high-interest individuals from a distance, rather than it targeting the general population. The main threat to average citizens comes from petty criminals.

Besides technical defenses, enhancing user security awareness is the key. This attack remains relevant until users become proactive in their defense; security measures are ineffective if unused. Users must be aware of potential surveillance and understand the consequences of device compromise to develop cautious routines for sensitive inputs.

5.1 Future Work

The present work focuses on monocular visible-light video feeds, as can be obtained from any smartphone or surveillance camera. Stereo camera setups or other modalities, such as thermal or infrared imaging, might help achieve higher detection rates. The drawback, of course, is the limited availability of specialized hardware.

The next step involves conducting a more comprehensive evaluation on a larger real-life dataset and developing a streamlined solution, potentially culminating in a finalized mobile app. Future developments could include tailoring techniques to the application, such as training models, e.g., YOLO, with custom data, which was not feasible due to time constraints. The compensation of the viewing angle was the most fragile part of our pipeline. Additionally, generalizing the attack to cover other knowledge-based authentication methods beyond PINs and accommodating two-finger password inputs could enhance its applicability. Such generalization goals appear feasible, motivating further exploration in CV and IT Security. Combining the presented methods with predictive models to infer missing finger trajectory points may lead to higher detection rates.

As some smartphones include LiDAR sensors, augmenting the camera feed with further modalities is a viable future approach. Stereo cameras or, generally, time-of-flight depth sensors would also improve the accuracy, especially since we currently rely on angle estimation. Also, more advanced ML-based estimations of depth would also work similarly, but would require no further hardware. This paper focuses on data that can be collected using commodity smartphones.

In the presented dataset, the environmental conditions for PIN entry videos are kept fairly constant. Extending these videos to public spaces, outdoor settings, and transportation systems is a feasible direction. It would be of interest to experiment with different lighting conditions, camera shake, additional obstructions, and varied video resolutions and frame rates. Crowdsourcing videos is also a possibility. However, licensing issues and the sensitive nature of PIN entry may hinder the creation of third-party video samples. A further concern is the use of additional, erratic finger movements, randomized keyboard layouts, or employing another hand or an obstacle to cover the input area—common practices recommended for PIN entry at ATMs. These factors were not addressed here. However, countermeasures such as live obstacle detection and adjustments for them may also be of interest. We leave these concerns to future work.

ACKNOWLEDGMENTS

During the preparation of this work the authors used ChatGPT-4o and o1 in order to rephrase some sentences for more clarity. After using this service, the authors carefully reviewed and edited the content as needed and take full responsibility for the content of the publication.

REFERENCES

N. Aggarwal and W. C. Karl. Line detection in images through regularized Hough transform. 15 (3):582–591, Mar. 2006. ISSN 1941-0042. 10.1109/TIP.2005.863021.

M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, T. S. Alrayes, H. Mathkour, and M. A. Mekhtiche. Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation. *IEEE Access*, 8:192527–192542, 2020. ISSN 2169-3536. 10.1109/ACCESS.2020.3032140.

H. Anthonio and Y. H.-S. Kam. A shoulder-surfing resistant colour image-based authentication method using human vision perception with spatial frequency. In *Int. Conf. Internet Technology and Secured Transactions*, ICITST '20, pages 1–5, 2020. 10.23919/ICITST51030.2020.9351349.

A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In *USENIX Conf. Offensive Tech.*, WOOT '10, pages 1–7, Aug. 2010.

P. V. Bhole, Z. Li, S. Bokolia, T. Oh, G. W. Tigwell, and R. L. Peiris. Haptic2FA: Haptics-based accessible two-factor authentication for blind and low vision people. *Proc. ACM Hum.-Comput. Interact.*, 8(MHCI), Sept. 2024. 10.1145/3676509.

F. Binbeshr, M. L. Mat Kiah, L. Y. Por, and A. A. Zaidan. A systematic review of PIN-entry methods resistant to shoulder-surfing attacks. *Computers & Security*, 101:102116, 2021. ISSN 0167-4048. https://doi.org/10.1016/j.cose.2020.102116.

F. Binbeshr, K. C. Siong, L. Y. Por, M. Imam, A. A. Al-Saggaf, and A. A. Abudaqa. A systematic review of graphical password methods resistant to shoulder-surfing attacks. *Int. J. Inf. Secur.*, 24(1):46, Dec. 2024. ISSN 1615-5270. 10.1007/s10207-024-00956-3.

L. Bošnjak and B. Brumen. Shoulder surfing experiments: A systematic literature review. 99:102023, Dec. 2020. ISSN 0167-4048. 10.1016/j.cose.2020.102023.

M. Bâce, A. Saad, M. Khamis, S. Schneegass, and A. Bulling. PrivacyScout: Assessing vulnerability to shoulder surfing on mobile devices. 2022(3):650–669, July 2022. ISSN 2299-0984. 10.56553/popets-2022-0090.

M. Cardaioli, S. Cecconello, M. Conti, S. Milani, S. Picek, and E. Saraci. Hand me your PIN! inferring ATM PINs of users typing with a covered hand. In *USENIX Secur. Symp.*, pages 1687–1704, 2022. ISBN 978-1-939133-31-1.

T. Chen, M. Farcasin, and E. Chan-Tin. Smartphone passcode prediction. 12(5):431–437, 2018. ISSN 1751-8717. 10.1049/iet-ifs.2017.0606.

M. Corbett, B. David-John, J. Shang, and B. Ji. ShouldAR: Detecting shoulder surfing attacks using multimodal eye tracking and augmented reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 8 (3), Sept. 2024. 10.1145/3678573.

M. Harbach, E. v. Zezschwitz, A. Fichtner, A. De Luca, and M. Smith. It's a hard lock life: A field study of smartphone (un)locking behavior and risk perception. In *Symp. Usable Priv. Secur.*, SOUPS '14, pages 213–230, 2014. ISBN 978-1-931971-13-3.

F. E. Imran, V. T. Goh, S.-C. Yip, T. T. V. Yap, and Y. H.-S. Kam. A shoulder surfing resistant authentication scheme using polymorphic cipher. In *Multimedia University Engineering Conference*, MECON '24, pages 1–5, July 2024. 10.1109/MECON62796.2024.10776103.

S. Kalamkar and G. M. Amalanathan. Multimodal image fusion: A systematic review. *Decis. Analyt. J.*, 9:100327, 2023. ISSN 2772-6622. 10.1016/j.dajour.2023.100327.

H. Khan, U. Hengartner, and D. Vogel. Evaluating attack and defense strategies for smartphone pin shoulder surfing. In *Conf. Human Factors in Computing Systems*, CHI '18, page 1–10. ACM, 2018. ISBN 9781450356206. 10.1145/3173574.3173738.

K. Kobayashi, T. Oguni, and M. Nakagawa. A series of pin/password input methods resilient to shoulder hacking based on cognitive difficulty of tracing multiple key movements. *IEICE Transactions on Information and Systems*, E103.D(7):1623–1632, 2020. 10.1587/transinf.2019EDP7181.

T. Kwon and J. Hong. Analysis and improvement of a PIN-entry method resilient to shoulder-surfing and recording attacks. *IEEE T. Inf. Foren. Sec.*, 10(2): 278–292, 2015. 10.1109/TIFS.2014.2374352.

M.-K. Lee. Security notions and advanced method for human shoulder-surfing resistant PIN-entry. *IEEE T. Inf. Foren. Sec.*, 9(4):695–708, 2014. 10.1109/TIFS.2014.2307671.

C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann. MediaPipe: A framework for building perception pipelines, June 2019. URL `http://arxiv.org/abs/1906.08172`.

T. Länge, P. Matheis, R. Düzgün, M. Volkamer, and P. Mayer. Vision: Towards fully shoulder-surfing resistant and usable authentication for virtual reality. In *Usable Security and Privacy*, 2024. ISBN 979-8-9894372-5-2. 10.14722/usec.2024.23092.

F. Maggi, A. Volpatto, S. Gasparini, G. Boracchi, and S. Zanero. Poster: fast, automatic iPhone shoulder surfing. In *ACM Conf. Comp. Comm. Secur.*, CCS '11, pages 805–808, Oct. 2011. ISBN 978-1-4503-0948-6. 10.1145/2046707.2093498.

P. Markert, D. V. Bailey, M. Golla, M. Durmuth, and A. J. Aviv. This PIN can be easily guessed: Analyzing the security of smartphone unlock PINs. In *IEEE Symp. Secur. Priv.*, SP '20, pages 286–303, May 2020. ISBN 978-1-72813-497-0. 10.1109/SP40000.2020.00100.

L. Y. Por, I. O. Ng, Y.-L. Chen, J. Yang, and C. S. Ku. A systematic literature review on the security attacks and countermeasures used in graphical passwords. *IEEE Access*, 12:53408–53423, 2024. 10.1109/ACCESS.2024.3373662.

R. Raguram, A. M. White, D. Goswami, F. Monrose, and J.-M. Frahm. iSpy: automatic reconstruction of typed input from compromising reflections. In *ACM Conf. Comp. Comm. Secur.*, CCS '11, pages 527–536, Oct. 2011. ISBN 978-1-4503-0948-6. 10.1145/2046707.2046769.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conf. Comp. Vis. Pat. Rec.*, CVPR '16, pages 779–788, 2016. 10.1109/CVPR.2016.91.

H. T. M. A. Riyadh, D. Bhardwaj, A. Dabrowski, and K. Krombholz. Usable authentication in virtual reality: Exploring the usability of pins and gestures. In C. Pöpper and L. Batina, editors, *Applied Cryptography and Network Security*, pages 412–431. Springer, 2024. ISBN 978-3-031-54776-8.

M. Rungruanganukul and T. Siriborvornratanakul. Deep learning based gesture classification for hand physical therapy interactive program. In V. G. Duffy, editor, *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Posture, Motion and Health*, pages 349–358. Springer, 2020. ISBN 978-3-030-49904-4. 10.1007/978-3-030-49904-4_26.

B. Schneier. *The Coming AI Hackers*. Harvard Kennedy School, Apr. 2021.

D. Shukla and V. V. Phoha. Stealing passwords by observing hands movement. 14(12):3086–3101, Dec. 2019. ISSN 1556-6013, 1556-6021. 10.1109/TIFS.2019.2911171.

A. K. Singh and D. Koundal. A temporal convolutional network for modeling raw 3D sequences and air-writing recognition. *Decis. Analyt. J.*, 10:100373, 2024. ISSN 2772-6622. 10.1016/j.dajour.2023.100373.

R. G. von Gioi. *A contrario line segment detection*. SpringerBriefs in Computer Science. Springer, Mar. 2014. ISBN 978-1-4939-0574-4. 10.1007/978-1-4939-0575-1.

C. Wang, Y. Wang, Y. Chen, H. Liu, and J. Liu. User authentication on mobile devices: Approaches, threats and trends. 170:107118, Apr. 2020. ISSN 1389-1286. 10.1016/j.comnet.2020.107118.

D. Wang, Q. Gu, X. Huang, and P. Wang. Understanding human-chosen PINs: Characteristics, distribution and security. In *ACM Asia Conf. Comp. Comm. Secur.*, ASIA CCS '17, pages 372–385. ACM, Apr. 2017. ISBN 978-1-4503-4944-4. 10.1145/3052973.3053031.

Y. Wang, Q. Huang, C. Jiang, J. Liu, M. Shang, and Z. Miao. Video stabilization: A comprehensive survey. 516:205–230, Jan. 2023. ISSN 0925-2312. 10.1016/j.neucom.2022.10.008.

P. Weich. Erkennung von passworteingaben am smartphone anhand von fingerbewegungen, 2023.

J. Willomitzer, A. Heinemann, and M. Margraf. Zur Benutzbarkeit der AusweisApp2. In *Mensch und Computer*, MuC '16. GI, Aachen, Germany, 2016. 10.18420/muc2016-ws03-0002.

Y. Xu, J. Heinly, A. M. White, F. Monrose, and J.-M. Frahm. Seeing double: reconstructing obscured typed input from repeated compromising reflections. In *ACM SIGSAC Conf. Comp. Comm. Secur.*, CCS '13, pages 1063–1074. ACM, Nov. 2013. ISBN 978-1-4503-2477-9. 10.1145/2508859.2516709.

Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao. Blind recognition of touched keys on mobile devices. In *ACM Conf. Comp. Comm. Secur.*, CCS '14, pages 1403–1414, Nov. 2014. ISBN 978-1-4503-2957-6. 10.1145/2660267.2660288.

Q. Yue, Z. Ling, W. Yu, B. Liu, and X. Fu. Blind recognition of text input on mobile devices via natural language processing. In *Priv.-Aware Mob. Comp.*, MobiHoc '15, pages 19–24, June 2015. ISBN 978-1-4503-3523-2. 10.1145/2757302.2757304.