EYE-SHIELD: Real-Time Protection of Mobile Device Screen Information from Shoulder Surfing

Brian Jay Tang
University of Michigan
bjaytang@umich.edu

Kang G. Shin
University of Michigan
kgshin@umich.edu

Abstract

People use mobile devices ubiquitously for computing, communication, storage, web browsing, and more. As a result, the information accessed and stored within mobile devices, such as financial and health information, text messages, and emails, can often be sensitive. Despite this, people frequently use their mobile devices in public areas, becoming susceptible to a simple yet effective attack – *shoulder surfing*. Shoulder surfing occurs when a person near a mobile user peeks at the user's mobile device, potentially acquiring passcodes, PINs, browsing behavior, or other personal information. We propose, `Eye-Shield`, a solution to prevent shoulder surfers from accessing/stealing sensitive on-screen information. `Eye-Shield` is designed to protect all types of on-screen information *in real time*, without any serious impediment to users' interactions with their mobile devices. `Eye-Shield` generates images that appear readable at close distances, but appear blurry or pixelated at farther distances and wider angles. It is capable of protecting on-screen information from shoulder surfers, operating in real time, and being minimally intrusive to the intended users. `Eye-Shield` protects images and text from shoulder surfers by reducing recognition rates to 24.24% and 15.91%. Our implementations of `Eye-Shield` achieved high frame rates for 1440 × 3088 screen resolutions (24 FPS for Android and 43 FPS for iOS). `Eye-Shield` also incurs acceptable memory usage, CPU utilization, and energy overhead. Finally, our MTurk and in-person user studies indicate that `Eye-Shield` protects on-screen information without a large usability cost for privacy-conscious users.

1 Introduction

Mobile devices, such as smartphones, laptops, and tablets, have become ubiquitous throughout society [1]. People use them anytime and anywhere to communicate, store data, browse content, and improve their lives. The information accessed and stored within mobile devices, such as financial and health information, text messages, photos, and emails, is often sensitive and private.

Despite the private nature of information stored in mobile devices, people often choose to use them in public areas. This leaves users susceptible to a simple yet effective attack – *shoulder surfing*. Shoulder surfing occurs when a person near a mobile device user peeks at the user's screen, potentially acquiring sensitive passcodes, PINs, browsing behavior, or other personal information. This form of visual hacking dates back to the 1980s when shoulder surfing occurred near public pay phones to steal calling card digits [2]. Shoulder surfing can be combined with other tools such as cameras or binoculars to increase the effectiveness of stealing information.

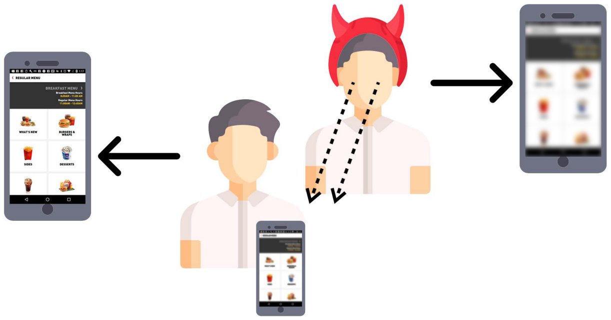


Figure 1: An example scenario of a shoulder-surfing privacy attack that could occur in any public setting. Using `Eye-Shield`, the shoulder surfer is prevented from seeing private content on the user's device screen.

Studies have shown that lack of screen protection in offices leaked information in 91% of shoulder surfing incidents [3]. Another study indicated that 85% of shoulder surfers acknowledged they observed sensitive information they were not authorized to see, such as login credentials, personal information, contact lists, and financial information [4]. Experiments indicate that you can hack into Snapchat or PayPal accounts by peeking at 2-factor authentication codes as they appear on a victim's mobile device screen [5, 6]. Shoulder surfing was also found to cause negative feelings and induce behavior changes [7]. Some shoulder surfing cases have sparked further discussion. For example, in 2017, someone had taken a picture and leaked Vitaliano Aguirre II's (Justice Secretary of Philippines) smartphone screen during a Senate hearing [8], proving he had been plotting against a senator. In 2018, Kanye West unlocked his smartphone in front of TV cameras in the White House revealing that his 6-digit PIN was 000000 [9].

Research has also demonstrated that shoulder surfers can obtain a 6-digit PIN 10.8% of the time with just one peek. While a person can limit his/her device's susceptibility to shoulder surfing by moving to a more private location, covering its screen, or turning its display away, these measures are not always feasible/effective (e.g., using a smartphone on a bus or airplane, using a laptop in an office or cafe). These privacy-preserving behaviors are typically employed as a response to protect against "detected" shoulder surfers, but studies have shown that mobile device users are aware of only 7% of shoulder surfing incidents [7]. The vast majority of shoulder surfing incidents and information leakage goes unnoticed, making it challenging for users to manually prevent information from being seen by shoulder surfers. Thus, effective defenses either automatically detect and notify users of unauthorized shoulder surfers, or continuously obfuscate information from potential shoulder surfers.

Users who seek protection against shoulder surfing may wish to hide sensitive information, keep others from stealing or peeking at login/PIN credentials, or desire peace of mind by having more control over private information. Many solutions have been proposed to thwart shoulder surfing, but each have their own drawbacks. One commonly used privacy-preserving mechanism is a privacy film that can be attached to a mobile device screen [10, 11]. These privacy films only allow light from the mobile device display to pass through the film within a narrow viewing angle [12]. Users can attach privacy films over their smartphone screen to prevent attackers outside of a certain viewing angle from seeing any content displayed on the smartphone's screen. However, screens covered with privacy films are still susceptible to shoulder surfers directly behind the user [13].

HCI and security researchers have explored various other defenses against shoulder surfing. They can be categorized into three main screen protection types: 1) shoulder surfer detection, 2) software solutions, and 3) authentication-specific approaches. Each of these solutions has its own advantages and drawbacks which we will discuss thoroughly in Section 2. No software-based defense has been developed for protecting the real-time usage of mobile devices, such as watching videos, playing games, and interacting with UI animations. Prior solutions are neither comprehensive nor capable of protecting all types of information from leaking to shoulder surfers.

Our goal is to prevent the leakage of *all* sensitive on-screen information to shoulder surfers without interrupting the intended user's device usage. To address this challenge, we develop Eye-Shield, a software solution that protects any on-screen information from shoulder surfers *in real time*. Eye-Shield can protect colored images, text, mobile app UIs, videos, and smartphone browsing from shoulder surfers. In this sense, it is a software version of privacy film protecting against shoulder surfers from any angle. While it protects information from shoulder surfers, it simultaneously allows the intended users to still view and comprehend the on-screen content. We envision that Eye-Shield can be deployed either as a feature of the device's operating system or an API for apps. Having a software solution to shoulder surfing implemented across mobile device platforms can increase awareness and protect sensitive information.

Throughout the development of Eye-Shield, we identify and address the following requirements/challenges:

1. The protection of *any* type of information displayed on-screen from shoulder surfers using purely software has not been developed before. Prior work has been tailored to protect certain types of information from shoulder surfers. In contrast, we explore and develop a universal shoulder surfing defense mechanism that functions with *any* type of content displayed on a device's screen. Section 3 details how the design of Eye-Shield achieves this. Eye-Shield protects images and text from shoulder surfers by reducing recognition rates to 24.24% and 15.91%. Sections 4 and 5 present an extensive evaluation of Eye-Shield's protection guarantees using several datasets of images, user interfaces (UIs), and videos.
2. Protecting on-screen information from shoulder surfers in *real time* without interruption to the intended device usage has not yet been addressed by purely software-based defense mechanisms. Previous systems rely on pre-rendering images and texts, or require the user to manually obfuscate portions of the screen. In contrast, Eye-Shield protects the entire screen with minimal to no disruptions to the intended device usage, while meeting real-time constraints. The implementation of Eye-Shield in mobile devices while achieving real-time performance, e.g., at a rate of 43 FPS on iOS devices for even the largest screen resolution sizes. Eye-Shield can achieve smoother performances of 60+ FPS by reducing the screen resolution. Section 5.3 provides performance details, such as latency, memory usage, CPU utilization, and energy overhead.
3. Eye-Shield must achieve the same level of protection that a privacy film provides, even if the shoulder surfer peeks at the screen from directly behind the intended user. Additionally, Eye-Shield should not cause noticeable usability disruptions, similar to a privacy film. Sections 4.4, 4.5, 5.4 and 5.5 detail the usability of Eye-Shield with two user studies. The first study (with $n = 22$ participants) demonstrates that users can still understand and use a mobile device employing Eye-Shield while shoulder surfers are unable to comprehend the screen content. The second study ($n = 99$) scales the findings of the first study. The participants found Eye-Shield to be easily usable, and privacy-conscious participants were satisfied with Eye-Shield's protection and its quality degradation.

2 Background

2.1 Threat Model

A typical shoulder surfing adversary is a curious or malicious person who seeks to observe or steal the information displayed on a victim's device. Most shoulder surfers do not wish to get caught, and hence we assume they would gather

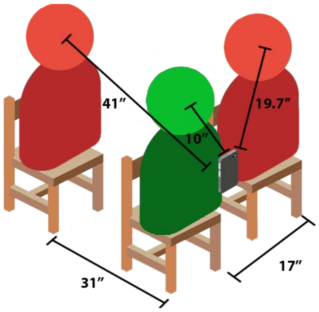


Figure 2: A diagram depicting the shoulder surfing threat model we consider. The intended user in green must be able to see and comprehend the information displayed on a screen.

information stealthily, i.e., the adversary peeks at the victim's screen either from an angle or a distance behind the victim. In realistic scenarios, an adversary who can view information on a victim's device is sitting either behind or next to the victim (e.g., public transportation seating, cafe, restaurant, auditorium, lecture, or office settings). In an airline setting, often the most cramped seating arrangement, the standard seat pitch is 31", and the standard seat width is 17" [14]. The majority of smartphone screen sizes are between 5" and 7" [15], with a resolution between 720×1280 and 1440×2960 [16], with most users operating them at a distance of 10" [17]. Our system should thus protect on-screen information from adversaries 41" directly behind the user's screen and 20" beside the user's screen. Figure 2 depicts this threat model. We use these measurements as the baseline setting and threat model for our experimental evaluation.

We assume that the adversary can either peek at the screen or use a camera to record the content on the victim's device. Since most shoulder surfing incidents are known to be out of curiosity rather than with malicious intent, `Eye-Shield` is not designed with the intent of protecting victims from adversaries using highly sophisticated tools or attacks. This, in turn, keeps the deployment of `Eye-Shield` easy and inexpensive. We also assume the adversary is interested in *any* type of content displayed on the victim's screen, not just passwords, text, or PIN entry.

2.2 Shoulder Surfing Defenses

Several existing defenses against shoulder surfing have been proposed. These defenses can be categorized into 4 main types: privacy films, authentication mechanisms, shoulder surfer detection, and software solutions.

Privacy Films: The most ubiquitous privacy defense against shoulder surfing is the privacy film. The film can be attached to a smartphone or laptop screen [10, 11], and only allows light from the mobile device display to pass through the film within a narrow viewing angle [12]. Users can attach privacy films on their smartphone screen to prevent attackers outside of a certain viewing angle from viewing any content presented on the smartphone's screen. However, screens covered with privacy films are still susceptible to shoulder surfers directly behind the user. Additionally, purchasing a privacy film incurs an added cost and must be physically attached to the mobile device screen – a price not all users are willing to pay.

Authentication Mechanisms: The problem of shoulder surfing has been most extensively explored in the context of authentication, particularly for PIN entry, password entry, game-based authentication, and drawing patterns for authentication. IllusionPIN uses hybrid images which encode the low spatial frequencies of one PIN digit with the high spatial frequencies of a different PIN digit to make the PIN entry appear different to a far-away shoulder surfer [18]. Zakaria *et al.*[19] implemented authentication by having users draw secret patterns. Kumar *et al.*[20] developed a system that allows users to enter passwords using just eye gaze on a keyboard. Another gaze-based authentication mechanism developed by Abdrabou *et al.* [21] used a combination of eye gaze and hand gestures as a potential mechanism for authentication.

Detection of Shoulder Surfers: This approach prevents shoulder surfers from unauthorized viewing of mobile device screens by detecting an additional set of eyes focusing on the screen and alerting the user to the potential shoulder surfing activity [22, 23]. Upon detecting a shoulder surfer, the system can either issue an alert and pause user activity, or the system could apply another software-based shoulder surfing protection mechanism.

Software Solutions: Software-based shoulder surfing defenses do not require any additional hardware beyond the typical functionalities available on a mobile device. The simplest of software implementations protect screen privacy by gray-scaling or darkening the screen. These approaches can also limit the visible regions of the screen via selective hiding and selective showing of screen content. There have been several approaches proposed by Zhou *et al.* [24], Khamis *et al.* [25], and BlackBerry [26]. Notably, the solution developed by Khamis uses eye-tracking to automatically selectively display and hide regions of the screen. Other software-based defenses aim to obfuscate specific types of information such as text. Eiband *et al.* [27] allow users to use their handwriting as the font, to increase the reading difficulty for shoulder surfers while maintaining font familiarity with the intended user. Von Zezschwitz *et al.* [28] seek to protect photo gallery browsing from shoulder surfers by pixelating or crystallizing the displayed images. This way, adversaries unfamiliar with the photos will be unable to extract meaningful information from peeking at a victim's device. Finally, `HideScreen` uses a grid-based approach to make the low spatial frequency components of an image or text appear like a completely gray background to shoulder surfers [29]. `HideScreen` encodes information as high spatial frequency components visible only to viewers close to the screen.

Other Related Approaches: Apple proposed using AR

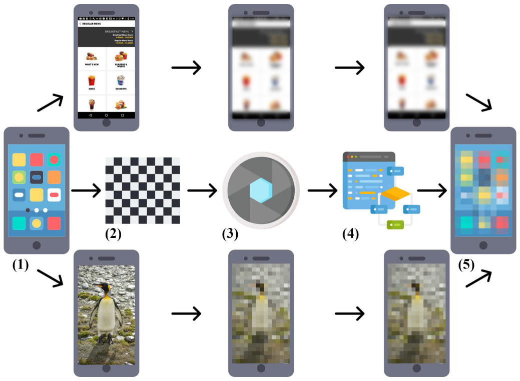


Figure 3: Eye-Shield works as follows: 1) an image or screen is input into the system, 2) a checkered grid mask is computed with the same dimensions as the input, 3) the input is blurred/pixelated, 4) Algorithm 1 uses the input, grid, and blurred input to compute, 5) the protected image.

glasses to un-blur mobile device screen content, such that it appears normal to the glasses wearer, by adjusting the prescription of the smart glasses for the wearer [30]. However, this approach has yet to be implemented, and it relies on additional expensive hardware not readily available to all users.

3 Design, Implementation, and Comparison

This section details Eye-Shield's design and implementation, and compares it with related work.

3.1 Design

Eye-Shield is designed to present the original screen content with only minor quality degradation to the intended user, but it can render shoulder surfers beyond a certain distance/angle away from the screen unable to discern the screen content. Eye-Shield achieves this by leveraging the fact that at a sufficient distance, it is impossible for an optical system to distinguish between two nearby light sources. By applying this theory of resolving power [31], we can construct checkered grids of pixels that can appear individually discernable at a close distance, but appear as a uniform average of the projected colors.

Fig. 3 provides an overview of the processing required to protect on-screen information. The main components required for the Eye-Shield algorithm are 1) the original screen/image, 2) a checkered grid mask in the dimensions of the original image, and 3) a blurred or pixelated version of the original image. Then, the protected output image will be computed with Algorithm 1.

Grid Generation: Eye-Shield's design is based on the observation that at angles smaller than $1.22\lambda/D$, where λ is the wavelength of light, and D is the lens aperture, it is no longer possible to distinguish two light sources from one another. So, using a checkered grid of pixels results in the pixels appearing as one uniform color to a user viewing from a far distance, whereas a user near the screen can distinguish between the individual pixels within the grid.

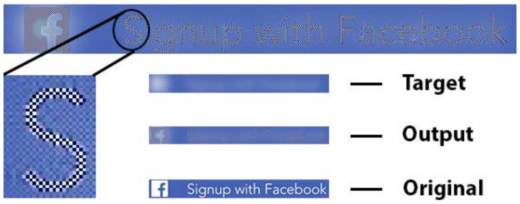


Figure 4: A close up look at a protected line of text.

Blurring/Pixelation: To enable Eye-Shield to function on colored content of any type, our system design makes the screen appear blurry (for text and mobile UIs) or pixelated (for images and videos) to users who perceive the pixels on the device screen from a small resolving power angular resolution (around 30–40" away from a smartphone or 20" with a 45°angle).

Computing Average Colors: We observe that two colors arranged in a checkered grid pattern and displayed on a screen appear as their average (additive color). While the best perceptual approximation of this averaged color can be achieved using a color approximation model such as CIECAM02 [32] or CIELAB [33], due to real-time computation constraints, we implement this color averaging as the root mean square in Eq. (1) to reduce the required computation time. Using the target (blurred/pixelated) image pixels as *rms* and the original image pixels as x, Eq. (1) computes the protected output image pixels as y.

Overall, after acquiring the required original image, target, and grid, to compute the protected image, we can represent Eq. (1) using the pseudocode in Algorithm 1:

An example of the resulting protected screen content can be seen in Fig. 4, where a full-size version of the protected content and downscaled versions of the target, protected output, and original text are presented. Note that the text output by Eye-Shield is much harder to discern compared to the original line of text. However, viewing the text in close proximity allows the user to still read the content.

Algorithm 1 Where img is the original $w \times h \times 3$ image
where grid is a $w \times h$ checkered grid of 1s and 0s
where targ is the $w \times h \times 3$ image, blurred or pixelated

```
procedure Eye-Shield ALGORITHM(img, grid, targ)
    $\text{complement} = (\text{targ}^2 \cdot 2) - \text{img}^2$
    $\text{delta} = (\text{complement} - \text{img}^2) \cdot \text{grid}$
    $\text{newimg} = \sqrt{\text{img}^2 + \text{delta}}$
    $\textit{clip}(\text{newimg}, 0, 255)$
```

3.2 Implementation of Eye-Shield

Eye-Shield utilizes the device's GPU on 4 platforms (Windows, Android, MacOS, iOS) to accelerate image processing and matrix operations. In the Windows implementation, we created both a CPU-only version and a GPU version for performance comparisons. The desktop implementation also supports video processing and writing using FFMPEG [34]. The full development stack of Eye-Shield for each platform can be found in Table 5 in the appendix. As it is unrealistic to expect app developers to implement a shoulder surfing solution on their platforms, Eye-Shield was devised as a proof-of-concept solution to be implemented on mobile device operating systems. As such, Eye-Shield acts more like a screen filter than an API for mobile app developers.

PC, CUDA: On several workstations and servers with access to an Nvidia GPU and CUDA drivers, Eye-Shield is able to run in real time using Python, OpenCV, and CUDA. Eye-Shield leverages CUDA to perform image blurring, grid generation, and the matrix operations used to compute the average colors.

Android, Vulkan: Eye-Shield runs in real time on Android mobile devices using C++, OpenCV, and Vulkan. Eye-Shield is capable of achieving real-time performance in image blurring without using OpenCL drivers. We note that Eye-Shield's implementation could improve with access to OpenCL drivers. Eye-Shield leverages Vulkan to perform the matrix operations used to compute the average colors. Here, the implementation of the grid generation is quick enough to be performed on the CPU in real time.

MacOS, iOS, Metal: Eye-Shield can run in real time on MacOS and iOS devices using Swift, CoreImage, and Metal. Eye-Shield leverages Metal to perform image blurring and the matrix operations used to compute the average colors. Grid generation is performed in C++ on the CPU.

3.3 Comparison with Related Work

Table 10 in the Appendix contains a comprehensive checklist of some notable shoulder surfing defense mechanisms and the types of content they can protect (not detection-based). Eye-Shield is capable of protecting all information categories except for PIN entry, since keypad reshuffling is required to make PIN entry fully secure.

Comparison to HideScreen [29]: The design of Eye-Shield is inspired by HideScreen [29], which leverages an observation that the human optical system is incapable of distinguishing between two adjacent light sources beyond a certain angle. This limit of the resolving power of an optical system can be leveraged to hide information from shoulder surfers on mobile devices by using a grid pattern of light and dark components so that it appears uniformly gray from far distances. HideScreen's design suffers from several key limitations: 1) selection of dark and light components limits the design to gray-scale images and text, 2) a significant amount of information is lost by only using 6 different grid patterns, and 3) the latency of hiding a 512×512 image is 1684ms, a run-time unsuitable for real-time screen usage.

Eye-Shield utilizes a similar grid-based design to shield on-screen content from shoulder surfers. However, the introduction of 3 channels for color poses a new challenge: the ability of Eye-Shield to preserve color information on screens. The approach of averaging everything into a monotone gray color fails to protect information from shoulder surfers when applied to full-color images. Eye-Shield does not attempt to perfectly hide all screen information from shoulder surfers, but it overlays screen information with a grid to make the screen appear blurry or pixelated to a far-away shoulder surfer. Besides enabling the protection of all types of screen content, Eye-Shield also achieves the performance required for real-time usage. These features allow for its integration with prevalent mobile device OSs and apps to protect a broad range of colored content, videos, and real-time browsing. It also removes many undesirable requirements in HideScreen such as needing app developers to adopt the HideScreen platform or display calibration, a time-consuming process. Finally, Eye-Shield causes no interruptions to usage and changes significantly less content than HideScreen. Overall, Eye-Shield's performance and usability is significantly improved over HideScreen.

Comparison to Privacy Films [11]: In many ways, Eye-Shield is most similar to privacy films in terms of functionality and interface. Eye-Shield is designed to protect and redraw the entire screen with minimal impacts on the screen refresh rate. The main differences are that privacy films are unable to protect against shoulder surfers directly behind the user, privacy films completely darken the screen from a shoulder surfer's perspective, and privacy films require users to be aware of shoulder surfing risks and be willing to pay and install films across all their devices. Eye-Shield protects against *any* shoulder surfers both beside and behind users, blurs and pixelates the screen rather than darkening it, and is free and implemented entirely in software.

Comparison to Zezschwitz *et al.* [28]: A photo browsing software capable of protecting information from shoulder surfers was developed by Zezschwitz *et al.* [28]. It provides a somewhat similar implementation to Eye-Shield in its usage of blurring, pixelation, and crystallization of images within the gallery. However, using the generated grid, Eye-Shield allows the intended user to still comprehend information and view low-level details within the pixelated images. The system in [28] completely blurs and pixelates content, even for the intended user.

Comparison to Eiband *et al.* [27]: This implementation allows users to view text messages in their personal handwriting. This personalization makes it more difficult for shoulder surfers to read information on the user's device. Eye-Shield

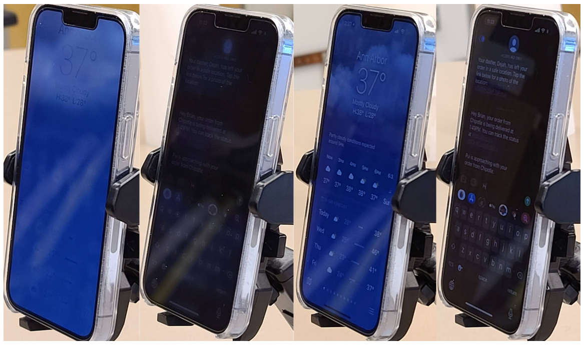


Figure 5: Photos of screens protected by Eye-Shield (left) and the unprotected original screens (right). Photos were taken at a distance of 19.7" and an angle of 45°, from the perspective of a shoulder surfer. The images were captured with a 108MP, f/1.8, 26mm smartphone camera at 3× zoom.

causes text to appear blurry to shoulder surfers without the need for customization or personalization. This allows users to more quickly adopt the privacy mechanism, further reducing barriers to usage.

Comparison to Blackberry Privacy Shade [26]: This software provides users with a tool for darkening all portions of the screen except for a small section. Users can then closely protect and hide the information from nearby shoulder surfers using their hands and body. This requires additional effort on the user's part and can interrupt a user's typical task flow. The implementation of Eye-Shield allows users to use their devices without actively worrying about their information by protecting the entire screen.

Comparison to IllusionPIN [18]: IllusionPIN uses hybrid images and keypad shuffling to protect PIN entry from the gaze of shoulder surfers. The combination of both tools makes the PIN appear to faraway shoulder surfers as a normal keypad, but the intended user can see the actual arrangement of the PIN numbers. While Eye-Shield does not explicitly support the protection of keypad entry, the combination of keypad shuffling and blurring can cause shoulder surfers to have difficulty in reading a user's PIN information.

4 Methodology for Evaluation of EYE-SHIELD

We empirically evaluate Eye-Shield's efficacy in protecting content, performance and resource consumption, and usability cost. This section describes the experimental setup and design details for each evaluation criterion as follows:

- **Protection:** Sections 4.1, 4.2, 4.4 and 4.5
- **Performance:** Section 4.3
- **Usability:** Sections 4.4 to 4.6

Our evaluation results will be provided in Section 5.

4.1 Perceptual Similarity

We generate protected images and videos using Eye-Shield on 3 datasets: 1) RICO, an image dataset of mobile app UIs [35], DIV2K, a diverse dataset of high resolution (2K) images [36], and DAVIS, a video dataset for object segmentation [37]. With the combined datasets, we evaluate 3,882 unique images. Using 4 different parameters for grid size and blurring/pixelation intensity, we generate a total of 124,224 protected images for use in our experimentation. To evaluate the information protection guaranteed by protecting images with Eye-Shield, we measure the SSIM index [38] (structural similarity) of the protected image and the blurred/pixelated target image. SSIM extracts and compares the luminance, contrast, and structure between two images. The formula is given in Eq. (2), where x, y are windows of size $N \times N$, μ is the average, σ is the variance/covariance, and c are normalization constants. In our experimentation, we use SSIM with $x, y = 7 \times 7$, and $c_1 = 0.01, c_2 = 0.03$. To simulate the distance from which a shoulder surfer views the protected image, we downscale the protected image and compare the SSIM with the target (blurred/pixelated) image. We also measure the SSIM of each full-scale protected image compared with the original image to evaluate the extent to which the protected image represents the original image.

4.2 Semantic Performance

To determine whether text and high-level details from images can be hidden from shoulder surfers by Eye-Shield at a large-scale, we leverage the Google Cloud Vision API to perform image recognition and optical character recognition (OCR). The "label detection" (image recognition) service provides labels and their associated confidence scores. Using OCR, we also detect the boundaries of texts and extract their content from images of mobile UIs. We evaluate the efficacy of Eye-Shield by performing label detection on downscaled protected images/videos from the DIV2K and DAVIS datasets and performing OCR on the mobile app UI screenshots from the RICO dataset. These results are compared with the API outputs of the unprotected downscaled images to provide a baseline for the percentage of labels and text protected.

4.3 Performance Evaluation

To determine whether Eye-Shield can run in real time, we benchmark the processing time and memory consumption on various devices, both with and without leveraging the devices' GPUs. We test a wide range of image resolutions — as small as 256×144 and as large as 3088×1440 (see Table 7). These encompass commonly-used video resolutions, mobile screen resolutions, and an image size for direct performance comparisons with HideScreen. The overall processing time of Eye-Shield is derived from a combination of the grid generation, blurring/pixelation of the original image, and the screen hiding algorithm that computes complementary colors. The performance data are gathered by logging processing times after running Eye-Shield for 100 image frames. We also evaluate the resource overhead of Eye-Shield by recording the maximum CPU utilization and the maximum memory usage after running Eye-Shield over the stream of 100 images. Finally, we measure energy consumption by using the Android Studio energy profiler [39] and the Xcode energy impact gauge [40]. These energy impacts are estimated based on GPU and CPU utilization, network and sensor usage, as well as other costs/overheads. The performance evaluations were run on 4 devices: a workstation with an AMD Ryzen 9 3900X CPU and an Nvidia GTX 2080 Super GPU, a 2021 MacBook Air with an M1 chip, a Samsung Galaxy S20 Ultra, and an iPhone 13 Pro. See Table 8 for more details.

4.4 MTurk Study

We conducted a user study through an Amazon Mechanical Turk (MTurk) survey to assess the protection strength of Eye-Shield on a diverse set of images and videos. 99 U.S. participants, aged 23–71 ($M = 43.19$, $SD = 10.37$; 55% men, 44% women), completed our survey. Our user study protocol was exempted (and approved) by our institution IRB, and participants who completed the study received $1.75 as compensation. We developed a series of questions where participants are presented with the original images/videos and the images/videos protected by Eye-Shield (in random order). To mitigate bias, participants were shown the protected images from the shoulder surfer's perspective, followed by the protected images from the intended user's perspective, and finishing with the unprotected images from the shoulder surfer's perspective. Participants are asked several text entry questions regarding the content within each image. To represent the distance at which a shoulder surfer sees the content, we also present a (4×) down-scaled version of both the original and protected content.

To derive the 4× downscaling, we calculate the angular diameter of an 5.78" iPhone 13 Pro (the device used in our in-person user study) at 2 distances, 10" and 41" (10" + 31", the average airplane seat pitch). We obtain an angular diameter of 8.064°and 32.239°, or roughly a 4× perceived size difference.

We collected responses from participants perceiving the protected and unprotected screens from the perspectives of the shoulder surfer and the intended user. We presented a total of 20 unique images, 6 unique videos, and 20 unique mobile app UIs to participants. These images and videos were randomly sampled from our evaluation datasets. Each participant answered questions regarding a random subset of 8 unique images/videos, portrayed as the downscaled protected screen, the full-size protected screen, and the downscaled original screen (24 images/videos per participant). Our survey averaged around 12.62 minutes for completion. With 99 participants, each question received an average of 19.8 responses, for a grand total of 3,180 responses. We measure a shoulder surfer or intended user's recognition rate (binary accuracy, R_{ss}, R_{iu}), or the percentage of text, images, and videos correctly labeled by the MTurk participants. In addition to evaluating the efficacy of Eye-Shield through this study, we also obtain user perceptions towards shoulder surfing and the users' inclination to use the protected screens. These responses were obtained using 5-point Likert survey responses ranging from "Strongly Disagree" to "Strongly Agree", normalized to values between 0–4. Finally, we recorded the response time for each question to better understand how Eye-Shield impacts the comprehension time of protected images. Sections 5.4 and 5.5 provide the MTurk study results.

4.5 User Study

We conducted an additional in-person user study to assess the usability and protection strength of Eye-Shield on a diverse set of images and videos. 22 U.S. participants, aged 22-63 ($M = 36.32$, $SD = 13.15$; 41% men, 59% women), completed our user study. We recruited participants with varying degrees of smartphone experience and visual health. Our protocol for this in-person user study was approved and exempted by our University IRB. The user study was conducted in a brightly lit lab with the device brightness at a moderate setting. We ensured the participants' vision was unobstructed by glare before continuing further in the study. Prior to conducting the user study, we placed the device (iPhone 13 Pro) displaying the images and videos protected with Eye-Shield on a smartphone mount on a table. The screen brightness of the device was set to 66%. Participants were instructed not to move or alter the device. This was done to ensure consistency between the evaluated study settings and to avoid the additional confounding factors that would arise if participants held the device. They were asked to evaluate the presented screen in 5 different settings: 1) a shoulder surfer 41" away from the screen protected with Eye-Shield, 2) an intended user 10" away from the screen, 3) a shoulder surfer 20" away from the protected screen at a 45° angle, 4) a shoulder surfer 41" away from the unprotected screen, and 5) a shoulder surfer 20" away from the unprotected screen at a 45° angle. This order was selected to mitigate bias from previous tasks. We also asked participants to evaluate a screen protected by a privacy film compared to a screen protected by both Eye-Shield and a privacy film by asking them to lean towards the device in setting 3 until they were able to read the content displayed on the device screen. We approximated the distance they had to lean. Setting 3, setting 5, and the evaluation with the physical privacy film were changes made to the original user study after suggestions from reviewers. This expanded user study along with the qualitative interview (Section 4.6) was conducted with 15 out of the 22 total participants. The total study took around 1 hour to complete with $20 for compensation. We developed a series of questions where participants are presented with the original image/video and the image/video protected by Eye-Shield (in random order). We presented a total of 6 images, 2 videos, 7 mobile app UIs, and 2 screen recordings to participants. Participants were asked several questions regarding the content within each image, involving reading text, describing images, and explaining videos. Some examples of these tasks were questions like "What is the current card balance?", "Can you read the first word in each sentence?", and "Can you describe the displayed image?". For texts, partial correctness was included in our accuracy metric. Although the correctness of the descriptions was subjective, most participants' answers were binary: accurate/specific (e.g., "ice rock climbing" and "person in red hiking mountain"), or no comprehension. We measured a shoulder surfer's or intended user's recognition rate (binary accuracy) as the percentage of text, images, and videos correctly labeled by the participants. We also asked participants to indicate the percentage of text they could read. The participants were asked to complete a system usability scale (SUS) survey [41] regarding the quality of images and text from the intended user's perspective.

4.6 User Study Interview

Finally, we discussed various topics related to Eye-Shield and received a variety of qualitative feedback. We obtained qualitative feedback (Section 5.6) after the user study experiments and the usability questionnaires were completed. We kept the interview open-ended and casual to learn about participants' initial perceptions of the system and to note ideas or opinions we would otherwise have missed. We transcribed and summarized the participants' discussion points and aggregated them by counting the number of participants who discussed similar topics. To help guide and continue the discussion, we asked participants about several topics. The questions we asked can be found in Table 11. [1]

5 Evaluation Results

We conducted the experiments and evaluations described in Section 4, answering the following key questions.

- **Experiment 1:** *How effective is our approach at protecting content such as colored images, videos, text, and mobile UIs?* In the experimentation with perceptual similarity metrics, Eye-Shield acts as expected, by very closely (*SSIM* > 0.9) resembling the blurred/pixelated content at smaller sizes and the original content at the original size. Cloud-based OCR and image recognition models are only able to recognize 8.45% of images and 3.16% of texts protected by Eye-Shield. Through user study tests, both in-person and crowdsourced, we find shoulder surfers are only able to recognize 25.00% of images and 18.78% of text. The crowdsourced study indicates that shoulder surfers can only recognize 32.24% – 35.50% of the content within images, videos, and texts. Our experiments indicate also Eye-Shield provides additional protection beyond using solely a privacy film. (Sections 5.1 to 5.2, 5.4 and 5.7)

- **Experiment 2:** *Does* Eye-Shield *meet real-time constraints for its screen hiding algorithms?* Eye-Shield is capable of achieving smooth performance (43 FPS) at even the highest screen resolutions (3088×1440). At lower resolutions, Eye-Shield can achieve 60+ FPS, the optimal performance for screens with a 60Hz refresh rate. While running, Eye-Shield consumes an acceptable amount of memory and CPU while having a moderate impact on power consumption. (Section 5.3)

- **Experiment 3:** *Is the usability cost of applying* Eye-Shield *to users' smartphones acceptable?* Users who are bothered and uncomfortable with shoulder surfing are more likely to use Eye-Shield, with users on average reacting positively towards the quality of text and images. The usability of the system was deemed above average, with a SUS score of 68.86. Our interviews with 15 participants provide additional insights into the usability of Eye-Shield, discussing concerns related to eye-strain, activation, and use cases. (Sections 5.4 to 5.6)

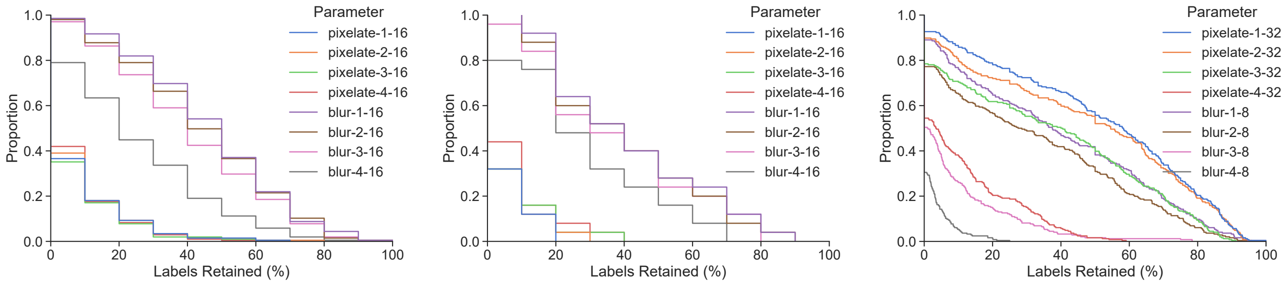
5.1 Perceptual Similarity

Fig. 7 shows the measured SSIM scores, where scores near 1.0 indicate high structural similarity, and scores near 0.0 indicate low similarity. We test for several parameters, such as grid size, downscaling size, and the window size for blurring. These 4× area downscaled images are consistent with the images presented to our MTurk participants in Section 5.4. Our algorithm was found to closely mimic the blurred images with blurring intensity of up to $\sigma = 24$, with the performance starting to degrade at $\sigma = 32$ (Fig. 7c). SSIM scores are > 0.9, demonstrating the high efficacy of Eye-Shield. Likewise, the structural similarity using pixelation degrades when the number of pixelated blocks is < 16. The results in Fig. 7b also suggest that Eye-Shield can provide some protection even at distances of only 20", though distances of 30" and beyond afford the best protection guarantees. Finally, while the smallest grid size achieves the best performance, larger grid sizes are more effective at hiding larger text fonts.

5.2 Semantic Performance

Eye-Shield is capable of reducing the detection rate of both image recognition and OCR systems. From an evaluation using the Google Cloud Vision API, Fig. 6 shows how a majority of the protected images retain fewer than 50% of the detected content compared to the original images. With certain parameters, around 20% of the evaluated images retain 0% of the

[1] The exact phrasing when asking about these questions differed from participant to participant, but the overall ideas and topics remained consistent.

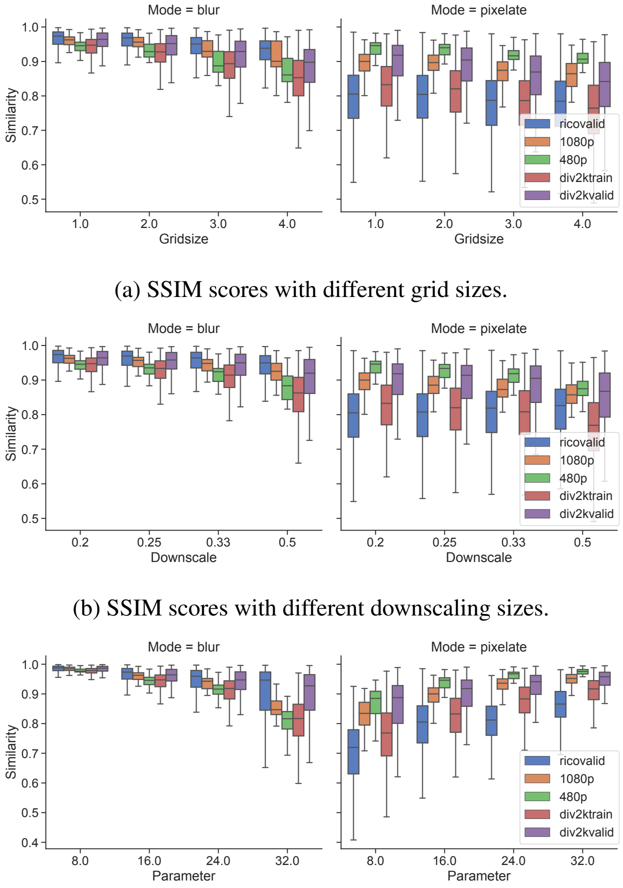


(a) DIV2K Train Dataset

(b) DIV2K Valid Dataset

(c) RICO Dataset

Figure 6: Results of using Google Cloud Vision API on protected images from 3 datasets. Table 9 provides total image counts.



(a) SSIM scores with different grid sizes.

(b) SSIM scores with different downscaling sizes.

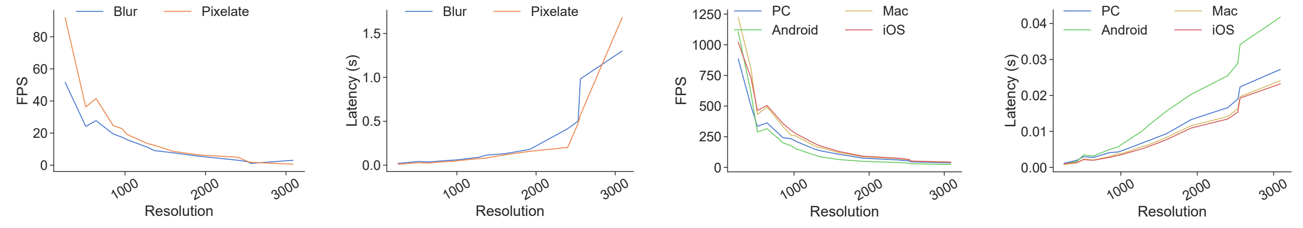
(c) SSIM scores with different blurring intensities.

Figure 7: SSIM scores for each evaluated dataset while varying grid size, downscaling size, and blurring/pixelation intensity. SSIM scores are computed by comparing the downscaled protected image and blurred/pixelated image.

originally detected labels. This observation holds for each evaluated dataset. In total, the DIV2K Valid, DIV2K Train, and RICO datasets protected by Eye-Shield retain 22.93%, 24.60%, and 47.71% of the original labels and text across all parameters. The high-resolution images protected using pixelation preserve much more information than the images protected using blurring. For images of mobile app UIs, blurring degrades the performance of the OCR system more than pixelation. When using blurring for text and pixelation for images, Eye-Shield-protected images retain 8.45%, 9.48%, and 42.38% for the DIV2K Valid, DIV2K Train, and RICO datasets. At larger grid sizes, the original text retained by the protected images decreases to as small as 3.16%. These results indicate that Eye-Shield may function best when switching between blurring and pixelation, depending on the type of interface presented on the device's screen. Selecting between blurring and pixelation can be made possible by detecting the amount of text on-screen using features such as OCR or using a phone's accessibility suite to determine the quantity of on-screen text. These results suggest that both shoulder surfers and vision systems may struggle with recognizing the information on a screen protected with Eye-Shield.

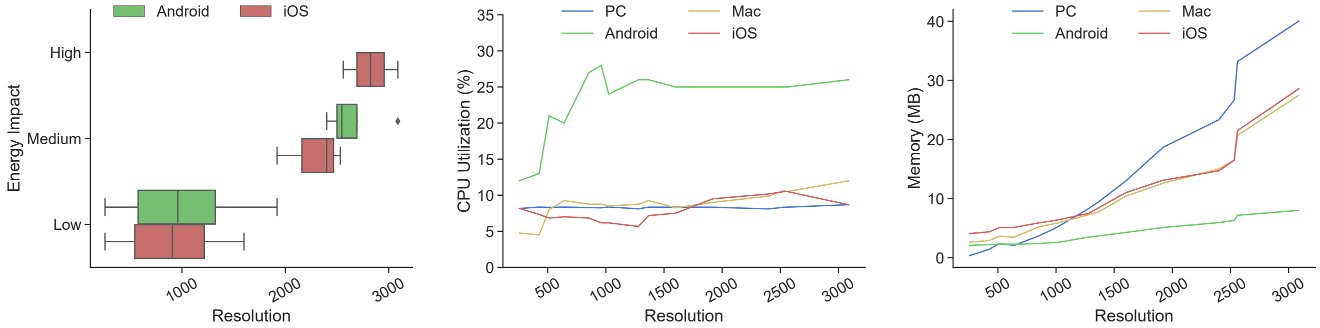
5.3 Performance Evaluation

As Eye-Shield was designed to protect mobile device screens with real-time constraints in mind, we conducted a set of performance evaluations aimed at measuring Eye-Shield's performance on a variety of screen sizes, video resolutions, and image sizes. The processing time for Eye-Shield with GPU support achieves 36.72 FPS, 75.27 FPS, and 241.72 FPS for resolution sizes of 3088 × 1440, 1920 × 1080, and 854 × 480 (Fig. 8). At an average latency of 3.457ms on 512 × 512 *full-color* images, the Android implementation of Eye-Shield provides a 487× speed-up compared to the 1684ms achieved on HideScreen with 512 × 512 *grayscale* images. The iOS, MacOS, and PC implementations of Eye-Shield achieve 2.153ms, 2.317ms, and 2.986ms, respectively. The average latency of running Eye-Shield on an image or frame of 3088 × 1440 is 27.23ms (PC), 41.75ms (Android), 24.10ms (MacOS), and 23.23ms (iOS). As shown in Fig. 8, Eye-Shield can achieve an average of 43 FPS on 3088 × 1440 screen resolutions using the iOS implementa-



(a) CPU-Only FPS (b) CPU-Only Latency (c) With-GPU FPS (d) With-GPU Latency

Figure 8: Measured FPS and latency using CPU-only and with-GPU implementations.



(a) Energy Impact (b) CPU Utilization (c) With-GPU Memory

Figure 9: Resource consumption such as energy usage, CPU utilization, and memory usage are measured and reported.

tion. This performance would allow the screen protected by Eye-Shield to update according to the user's inputs at an effective refresh rate of 43Hz. Considering most smartphones run at a 60Hz refresh rate, our implementation provides an acceptable trade-off for increased privacy. Alternatively, users who prefer responsiveness to high resolution may display the protected screen at a lower resolution to attain > 60 FPS. Our evaluation shows Eye-Shield achieves a > 60 FPS at screen resolutions of 1170×2532 and smaller on iOS (Table 6). The trade-offs between latency and screen resolution are similarly present in energy consumption and memory usage. In the case where most users prefer stable performance over higher screen resolution, Eye-Shield will prioritize selecting a screen resolution that can match the device's refresh rate.

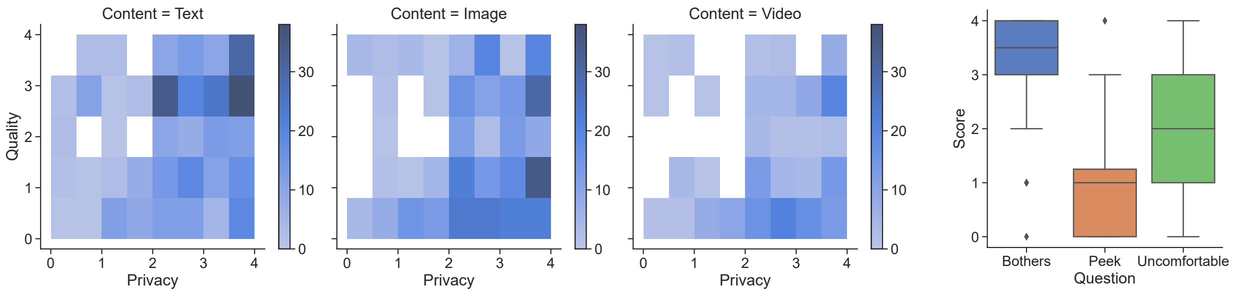
Since Eye-Shield must be lightweight enough to run on mobile devices, we ensure that the memory usage is small enough to not cause significant memory errors or stuttering. Figure 9c shows the memory usage of with-GPU and non-GPU implementations of Eye-Shield across each platform. The memory usage scales with the resolution size, reaching up to 40.04MB per frame in the GPU implementation of Eye-Shield. Using only the CPU consumes more memory with each frame (up to 299.95MB).

On mobile devices, Eye-Shield is observed to achieve low energy consumption for resolution sizes of up to 1920×1080. For larger resolution screen sizes, the measured energy impact is medium to high (see Fig. 9a). The maximum measured CPU utilization of Eye-Shield for each of the four evaluated devices is shown in Fig. 9b. Utilization is a percentage of all cores on the device (Table 8). For the most part, utilization of all the CPU cores on the device remains below 25%. Overall, especially at lower resolutions, the measured resource usage and estimated energy impacts indicate that Eye-Shield is lightweight enough to be run on mobile devices without overtaxing battery life or causing overheating/thermal issues.

5.4 MTurk Study

The main purpose of our MTurk study is a large-scale evaluation of Eye-Shield's efficacy in defending against shoulder surfing. Our study demonstrates that shoulder surfers can only recognize as low as 32.24% of the text on our protected images from an effective distance of 41" (Table 1). As a comparison, shoulder surfers can recognize up to 83.84% of the unprotected text from the same distance. This degrades the shoulder surfers' recognition rate by 51.60 percentage points. Our results for images and videos achieve similar protection improvements, with decreases in recognition rate of 60.75 and 61.61 percentage points, respectively. These reductions in recognition rate demonstrate Eye-Shield's potential for reducing the amount of information a shoulder surfer can glean from an unwitting user.

We also assessed our participants' perceptions of shoulder



(a) Likelihood of using Eye-Shield conditioned on responses from Fig. 10b.

(b) Privacy Attitudes

Figure 10: (a) Likelihood of using Eye-Shield conditioned on responses from (b). (b) Distribution of MTurk privacy responses.

Table 1: MTurk and in-person content recognition rates.

Platform	Setting	Text(%)	Image(%)	Video(%)
MTurk	Intended User (Protected)	97.22	81.75	79.80
	Shoulder Surfer (Original)	83.84	96.25	96.46
	Shoulder Surfer (Protected)	32.24	35.50	34.85
In-Person	Intended User (Protected)	98.05	82.58	86.64
	Shoulder Surfer (Original)	77.27	100.0	100.0
	Shoulder Surfer (Protected)	15.91	24.24	47.04
	Shoulder Surfer (Original, 45°)	100.0	100.0	100.0
	Shoulder Surfer (Protected, 45°)	21.90	22.22	51.67

Table 2: Overall in-person recognition rates

Setting	Recognition Rate (%)	Text Visible (%)
Intended User (Protected)	89.57	94.05
Shoulder Surfer (Original)	90.37	62.96
Shoulder Surfer (Protected)	26.94	5.876
Shoulder Surfer (Original, 45°)	100.0	96.82
Shoulder Surfer (Protected, 45°)	27.84	14.67

surfing. The mean 5-point Likert scores for 1) how bothered users were by others peeking at their phones, 2) how often users peeked at others' phone screens in public, and 3) how uncomfortable users were with looking at their phones in public areas were 3.28, 1.02, and 1.83, respectively. Fig. 10b shows that the participants were generally averse to being involved in shoulder surfing both as the victim and adversary.

We gathered participants' feedback on their contentedness of the quality of images using Eye-Shield for different content types conditioned on their attitudes towards shoulder surfing. For those who were both bothered by shoulder surfing and uncomfortable with using their smartphones in public settings, the mean Likert score for the likelihood of using the screen protection in public settings was 2.61, 2.21, and 2.00 for mobile UIs, images, and videos, respectively. These indicate that, on average, privacy-conscious participants were mostly happy with using Eye-Shield in public settings to protect their privacy. Fig. 10a depicts how, especially for images of mobile app UIs protected using blurring, participants were happy with the quality of images. The density plots indicate a correlation between concern with shoulder surfing and the likelihood of using Eye-Shield. Participants were less satisfied with the image quality for the high-resolution images and videos protected using pixelation.

5.5 User Study

We assessed the recognition rate of screens protected by Eye-Shield for an in-person setting and observed an overall recognition rate of 26.94% for shoulder surfers. For users close to the screen as the intended user, the recognition rate is 89.57%. For text visibility, we observed stronger protection with a shoulder surfer recognition rate of around 5.88%. Close to the screen, almost 100% of the text is visible to the intended user. Table 1 shows how in-person participants were only able to recognize 15.91% of the protected texts and 24.24% of the protected images. The video domain represented a much more challenging problem, as participants were still able to recognize the scenes 47.04% of the time As a comparison, without the protection provided by Eye-Shield, participants could clearly see and recognize almost 80% of the texts and 100% of the images and videos.

After answering questions about the content displayed on Eye-Shield, our participants responded to a SUS questionnaire. The average SUS score of Eye-Shield was 68.86, where a SUS score above 68 is deemed above average [41]. The distribution of responses is presented in Fig. 11. These SUS scores include the original 7 participants along with the 15 new participants after the study design changed. Participants responded negatively mainly to the consistency and cumbersomeness of Eye-Shield, due to certain types of content being more easily recognizable than others.

While we did not explicitly measure the time taken for participants to comprehend the information displayed, we observed that all participants took longer to respond when viewing the protected screen than the unprotected screen from the same distance of 41" and 20" with a 45 ° angle.

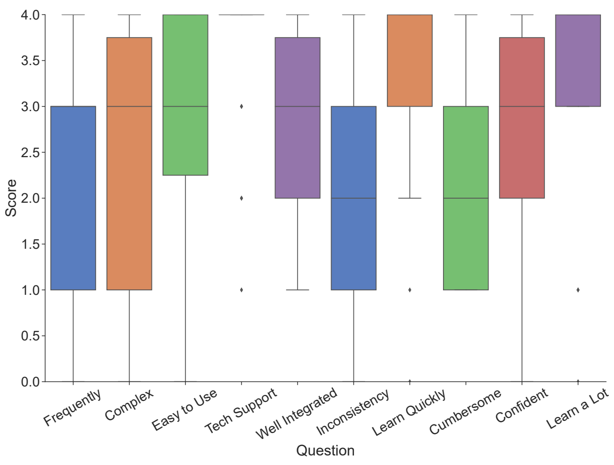


Figure 11: Normalized SUS scores for each question.

5.6 Qualitative Feedback

Overall, out of 15 participants, 8 participants indicated they would be uncomfortable with shoulder surfers peeking at their devices (Q7), 7 participants indicated they preferred using Eye-Shield over using a privacy film (Q2), 7 participants found an option to set the blurring intensity to be useful (Q11), and 6 participants stated that the ability to toggle Eye-Shield would be helpful. (Q11) 7 participants stated they would use Eye-Shield for protecting financial data and PIN entry (Q3), and 3 participants said they would use it to protect personal texts and photos (Q4). 7 participants found the blurring that Eye-Shield introduces to the user to be slightly annoying (Q5), and only 3 participants stated they had eyestrain as the intended user (Q6). Overall, these results support our claim that Eye-Shield would be helpful for protecting privacy-conscious users who are concerned about shoulder surfing (more in Table 11).

Generally, participants wanted Eye-Shield for PIN entry, and some participants wanted Eye-Shield to activate automatically for certain apps with more sensitive information (Q11). Participants also indicated that zooming, leaning in, and increasing brightness improved the usability of Eye-Shield as the intended user (Q13). We also observed generational differences in our responses, for example, older participants generally used their phone less than younger participants in public and found less of a need for Eye-Shield. One participant suggested that Eye-Shield may perform differently for other languages (Q14).

Some participants were excited and wanted to see Eye-Shield implemented on their smartphones, while others did not see themselves ever personally using Eye-Shield. Overall, participants reacted positively towards Eye-Shield, noting that it was very difficult to see the protected on-screen information from the perspective of a shoulder surfer.

5.7 Comparison with Privacy Films

We purchased several privacy films for use in our evaluations and user studies. During our in-person user study, we also conducted several experiments comparing the angle of readable text between the privacy film and Eye-Shield. Participants could not read any text on the screen protected by the privacy film from the 45° angle, although they could easily see everything whenever the phone was tilted sideways. When we asked participants to lean over towards the screen, they only needed to lean over an average of 4.55". After activating Eye-Shield with the privacy film still on, we asked participants to lean over, and they needed to lean an average of 10.55". Thus, combining Eye-Shield and privacy films provides the best protection. We also observed that the efficacy of privacy films was dependent on the environmental brightness relative to the screen brightness. For example, they are not effective at darkening the screen in dark settings, nor at 100% screen brightness levels, especially on laptops. Additionally, privacy films generally cost $10–$30 and are designed for specific device screen sizes/types. Several participants indicated annoyance at the inability of disabling the physical privacy film for smartphones. For example, one participant noted that the screen would appear dark and blurry if the phone was turned slightly, although they would consider using the film for large stationary devices like laptops. We performed an additional evaluation with the Google OCR service and observed that privacy films provide no protection at angles of $\leq n30^\circ$, whereas Eye-Shield still provides some protection. We note that privacy films provide protection against optical zoom whereas Eye-Shield is not. Table 3 depicts these results, demonstrating that in some settings, Eye-Shield protects more information than a privacy film, and using both Eye-Shield and a privacy film offers complete protection in the evaluated settings.

6 Discussion

6.1 Implications of Findings

Our experimental evaluation indicates the feasibility of implementing a software-based privacy film for mobile device screens. Having a widely accessible low-latency screen protection mechanism could increase users' awareness of shoulder surfing attacks and preserve their privacy without significantly disrupting their device usage. Without the need for additional physical components, Eye-Shield could be implemented agnostic of both apps and devices. Privacy-conscious users would no longer need to purchase and install new films whenever they change mobile devices, averaging once every 22.7 months for American adults (and more frequently for younger users) [42]. Highly cautious users will find increased privacy guarantees by applying both Eye-Shield and a privacy film to protect their on-screen information.

6.2 Prototype User Interface

As shown in Fig. 13, we have developed a low-fidelity prototype for toggling `Eye-Shield` on both the iOS and Android operating systems. `Eye-Shield` would be most naturally implemented as a toggle-able widget, with more advanced users being able to adjust individual parameters and features in the device's system settings and preferences. Users can manually toggle `Eye-Shield` upon entering public settings. We expect most users to activate `Eye-Shield` before viewing private or sensitive content. For most users, the default parameters can be set to *gridsize* = 1 and blurring with $\sigma = 8$, which achieves the best overall performance in the evaluation performed in Section 5.2. The default screen resolution size can be set to the maximum size in which the system achieves 60 FPS. Some users may opt to adjust these parameters to attain higher screen resolution or increased protection guarantees. We consider several pre-designed parameter settings:

1. High protection [low-brightness, low-contrast, high-blur]
2. Med protection [med-brightness, med-contrast, med-blur]
3. low protection [high-brightness, high-contrast, low-blur]

6.3 Limitations

The most notable limitation of `Eye-Shield` is the loss of some information and image quality. For example, users responded more negatively to pixelation than blurring, despite pixelation providing stronger privacy guarantees. Likewise, the blurring intensity and pixelation block size can affect interpretability and readability. Nonetheless, `Eye-Shield` still allows users to continue with their intended applications with minimal interruptions to their workflow. This trade-off between usability and privacy guarantees is inevitable when simultaneously providing intended users with their information and protecting information from shoulder surfers in the same screen. Additionally, shoulder surfers will still be able to infer certain high-level details. Several participants in our in-person user study were able to guess the type of mobile app being displayed and the general category of images. However, they were unable to interpret or read any specific details.

People of older age or those with visual impairments, vision degradation, or color-blindness may also find it slightly more challenging to use `Eye-Shield`. Our user study included 6 participants with minor visual impairments who were able to read and recognize almost all displayed content as the intended user. For users with severe visual impairments, mobile OSs provide accessibility APIs for toggling color filters and increasing contrast. Through preliminary tests, it is possible to use `Eye-Shield` with color-blindness filters with little to no impact on the comprehension of content. However, increasing contrast will negatively impact the protection guarantees. Regarding the risk of eyestrain, the length of the in-person study totaled about 60 minutes, with time spent staring at the screen averaging around 30 minutes. We offered the participants at several instances a chance to take a break, but none of the participants required a break. Only 3 participants noted experiencing minor eyestrain as the intended user. However, it is unclear whether using `Eye-Shield` for prolonged periods of time (>1 hour) will cause eyestrain for users due to looking at blurry/pixelated content.

6.4 Future Work

We have presented and demonstrated a shoulder surfing protection mechanism, and left integration of `Eye-Shield` into Android OS and iOS as ongoing/future work. Creating an adaptive solution to adjust parameters such as blurring and pixelation intensity and grid size is another technical challenge which requires further investigation.

Other directions for future work include assessing the efficacy of `Eye-Shield` using various device screen sizes and other blurring or pixelation methods. Finding the optimal trade-off between usability and privacy in each of `Eye-Shield`'s parameters is another useful topic to explore.

7 Conclusion

We have presented and evaluated `Eye-Shield` to prevent the shoulder surfing of information displayed on mobile devices. It is designed to protect all types of on-screen information — text, colored images, mobile app UIs, videos, and smartphone browsing — *in real time*, without significantly hampering the user's interactions with the mobile device. `Eye-Shield` can be regarded as a software version of a privacy film. By blurring/pixelating a screen, generating a checkered grid, and computing complementary colors, `Eye-Shield` can generate images that appear readable and interpretable at close distances, but appear blurry and pixelated at distances of 30" and angles of 45" and beyond. Having a software-based defense against shoulder surfing built into devices can increase user awareness of shoulder surfing and prevent adversaries from accessing/stealing sensitive information. `Eye-Shield` was designed and implemented as a low-cost and easily-adoptable defense mechanism against shoulder surfing.

Acknowledgments

The work reported in this paper was supported in part by the Army Research Office (ARO) under Grant No. W911NF-21-1-0057. Special thanks to the shepherd, the RTCL members, and all study participants for their incredibly helpful feedback!

Availability

The code, datasets, and user study samples are provided at https://github.com/byron123t/eye-shield, and https://www.bjaytang.com/projects/post_008/.

References

[1] P. R. Center, “Mobile fact sheet,” *Pew Research Center*, 2019.

[2] “Social Engineering: Manipulating the Source | SANS Institute,” Oct 2008, [Online; accessed 2. Mar. 2022]. [Online]. Available: https://www.sans.org/white-papers/32914

[3] P. I. LLC, “Global visual hacking experimental study: Analysis,” *Ponemon Institute Research Report*, 2016. [Online]. Available: https://multimedia.3m.com/mws/media/1254232O/global-visual-hacking-experiment-study-summary.pdf

[4] B. Honan, “Visual data security white paper,” *Secure*, 2012. [Online]. Available: https://multimedia.3m.com/mws/media/950026O/secure-white-paper.pdf

[5] “SnapHack: Watch out for those who can hack into anyone’s Snapchat! | WeLiveSecurity,” Dec 2021, [Online; accessed 2. Mar. 2022]. [Online]. Available: https://www.welivesecurity.com/2021/12/09/snaphack-watch-out-those-who-can-hack-anyones-snapchat

[6] “How I hacked my friend’s PayPal account | WeLiveSecurity,” Feb 2022, [Online; accessed 2. Mar. 2022]. [Online]. Available: https://www.welivesecurity.com/2022/01/24/how-i-hacked-my-friends-paypal-account

[7] M. Eiband, M. Khamis, E. Von Zezschwitz, H. Hussmann, and F. Alt, “Understanding shoulder surfing in the wild: Stories from users and observers,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 4254–4265.

[8] “Aguirre furious at photo leak of private text message |,” Sep 2017, [Online; accessed 2. Mar. 2022]. [Online]. Available: https://politics.com.ph/aguirre-furious-photo-leak-private-text-message

[9] C. Gartenberg, “Kanye West’s iPhone passcode is 000000,” *Verge*, Oct. 2018. [Online]. Available: https://www.theverge.com/tldr/2018/10/11/17964848/kanye-west-iphone-passcode-trump-iplane-apple-meeting

[10] GearBrain Editorial Team, “5 top privacy screen protectors for iPhone and Android phones,” *Gearbrain*, Aug 2019. [Online]. Available: https://www.gearbrain.com/privacy-screen-protector-iphone-android-2639955681.html

[11] “Privacy & Screen Protectors | Privacy & Protection | 3M US,” Jun. 2022, [Online; accessed 6. Jun. 2022]. [Online]. Available: https://www.3m.com/3M/en_US/privacy-screen-protectors-us

[12] “What you need to know about privacy screen protectors,” Jun 2021, [Online; accessed 2. Mar. 2022]. [Online]. Available: https://www.androidauthority.com/privacy-screen-protectors-explained-970541

[13] Linshang, “Angle of Mobile Phone Privacy Film and VLT Test,” *Linshang*, Jul. 2021. [Online]. Available: https://www.linshangtech.com/tech/privacy-film-angle-and-transmittance-tech1437.html

[14] “Airline Seat Pitch Guide | SKYTRAX,” Apr 2018, [Online; accessed 2. Mar. 2022]. [Online]. Available: https://www.airlinequality.com/info/seat-pitch-guide

[15] “Global smartphone shipments by screen size 2018-2022 | Statista,” Mar 2022, [Online; accessed 2. Mar. 2022]. [Online]. Available: https://www.statista.com/statistics/684294/global-smartphone-shipments-by-screen-size

[16] “Most used smartphone screen resolutions in 2019,” Oct 2019, [Online; accessed 2. Mar. 2022]. [Online]. Available: https://deviceatlas.com/blog/most-used-smartphone-screen-resolutions#us

[17] M. Yoshimura, M. Kitazawa, Y. Maeda, M. Mimura, K. Tsubota, and T. Kishimoto, “Smartphone viewing distance and sleep: an experimental study utilizing motion capture technology,” *Nature and science of sleep*, vol. 9, p. 59, 2017.

[18] A. Papadopoulos, T. Nguyen, E. Durmus, and N. Memon, “Illusionpin: Shoulder-surfing resistant authentication using hybrid images,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2875–2889, 2017.

[19] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan, “Shoulder surfing defence for recall-based graphical passwords,” in *Proceedings of the seventh symposium on usable privacy and security*, 2011, pp. 1–12.

[20] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, “Reducing shoulder-surfing by using gaze-based password entry,” in *Proceedings of the 3rd symposium on Usable privacy and security*, 2007, pp. 13–19.

[21] Y. Abdrabou, M. Khamis, R. M. Eisa, S. Ismail, and A. Elmougy, “Just gaze and wave: Exploring the use of gaze and gestures for shoulder-surfing resilient authentication,” in *Proceedings of the 11th acm symposium on eye tracking research & applications*, 2019, pp. 1–10.

[22] K. Bradshaw, “Future Chromebooks may alert you when someone is snooping over your shoulder,” Nov 2021. [Online]. Available: https://9to5google.com/2021/11/24/future-chromebooks-snooping-protection

[23] S. Lian, W. Hu, X. Song, and Z. Liu, “Smart privacy-preserving screen based on multiple sensor fusion,” *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 136–143, 2013.

[24] H. Zhou, V. Ferreira, T. Alves, K. Hawkey, and D. Reilly, “Somebody is peeking! a proximity and privacy aware tablet interface,” in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, 2015, pp. 1971–1976.

[25] M. Khamis, M. Eiband, M. Zürn, and H. Hussmann, “Eyespot: Leveraging gaze to protect private text content on mobile devices from shoulder surfing,” *Multimodal Technologies and Interaction*, vol. 2, no. 3, p. 45, 2018.

[26] “BlackBerry Privacy Shade - Apps on Google Play,” Apr 2022, [Online; accessed 11. Apr. 2022]. [Online]. Available: https://play.google.com/store/apps/details?id=com.blackberry.privacyfilter&hl=en_US%E2%89%B7=US

[27] M. Eiband, E. von Zezschwitz, D. Buschek, and H. Hußmann, “My scrawl hides it all: protecting text messages against shoulder surfing with handwritten fonts,” in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2016, pp. 2041–2048.

[28] E. von Zezschwitz, S. Ebbinghaus, H. Hussmann, and A. De Luca, “You can’t watch this! privacy-respectful photo browsing on smartphones,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 4320–4324.

[29] C.-Y. Chen, B.-Y. Lin, J. Wang, and K. G. Shin, “Keep others from peeking at your mobile device screen!” in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[30] S. C. Lee, “Systems and methods for switching vision correction graphical outputs on a display of an electronic device,” Nov 2021, united States Patent Application: 0210350769. [Online]. Available: https://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.html&r=1&f=G&l=50&s1=%2220210350769%22.PGNR.&OS=DN/20210350769&RS=DN/20210350769

[31] D. Singh, *Fundamentals of optics*. PHI Learning Pvt. Ltd., 2015.

[32] N. Moroney, M. D. Fairchild, R. W. Hunt, C. Li, M. R. Luo, and T. Newman, “The ciecam02 color appearance model,” in *Color and Imaging Conference*, vol. 2002, no. 1. Society for Imaging Science and Technology, 2002, pp. 23–27.

[33] X. Zhang, B. A. Wandell *et al.*, “A spatial extension of cielab for digital color image reproduction,” in *SID international symposium digest of technical papers*, vol. 27. Citeseer, 1996, pp. 731–734.

[34] S. Tomar, “Converting video formats with ffmpeg,” *Linux journal*, vol. 2006, no. 146, p. 10, 2006.

[35] B. Deka, Z. Huang, C. Franzen, J. Hibschman, D. Afergan, Y. Li, J. Nichols, and R. Kumar, “Rico: A mobile app dataset for building data-driven design applications,” in *Proceedings of the 30th Annual Symposium on User Interface Software and Technology*, ser. UIST ’17, 2017.

[36] E. Agustsson and R. Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[37] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. Van Gool, “The 2019 davis challenge on vos: Unsupervised multi-object segmentation,” *arXiv preprint arXiv:1905.00737*, 2019.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[39] “Inspect energy use with Energy Profiler | Android Developers,” Aug. 2020, [Online; accessed 24. May 2022]. [Online]. Available: https://developer.android.com/studio/profile/energy-profiler

[40] “Energy Efficiency Guide for iOS Apps: Measure Energy Impact with Xcode,” Sep. 2016, [Online; accessed 24. May 2022]. [Online]. Available: https://developer.apple.com/library/archive/documentation/Performance/Conceptual/EnergyGuide-iOS/MonitorEnergyWithXcode.html

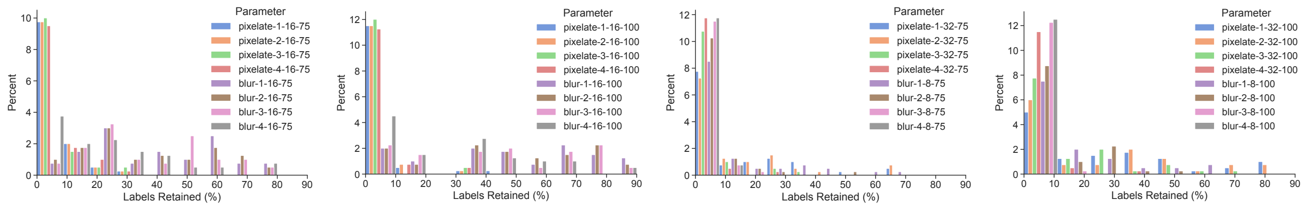
[41] J. Brooke, “Sus: a “quick and dirty’usability,” *Usability evaluation in industry*, vol. 189, no. 3, 1996.

[42] “What’s next for the smartphone industry - Global site - Kantar Worldpanel,” Aug. 2022, [Online; accessed 29. Aug. 2022]. [Online]. Available: https://www.kantarworldpanel.com/global/News/2017-smartphone-industry-insight-report

Appendix

A Contrast, Brightness, and Environment

We conducted an additional set of experiments to evaluate the efficacy of protecting on-screen information with



(a) DIV2K, Low Contrast

(b) DIV2K, Moderate Contrast

(c) RICO, Low Contrast

(d) RICO, Moderate Contrast

Figure 12: Results of using Google Cloud Vision API on protected images from 2 datasets with low contrast and moderate contrast settings. The unique images evaluated on each dataset total 50 images.

high-contrast colors and different device and environmental brightness settings. We evaluated images protected using Eye-Shield and displayed on a smartphone with 3 different device brightness settings (33%, 66%, and 100%) in a lab with the lights on and off. Figure 15 shows the experiment setup and the example photos used for the experiment. Note that at a close distance, there is little degradation in readability. Photos were taken of the device screen in all the described settings, and the cropped photos were evaluated on Google's OCR system to determine the impact of brightness on the efficacy of Eye-Shield. The word detection rate for images taken from the intended user's perspective is only decreased by 14 words and 9.5 words in the dark and bright environments, respectively. From the shoulder surfer's perspective, the average word detection rate decreases by as much as 21 words from the side angle, and 8.25 words from the direct angle. While the intended user can still read most words at any brightness setting, in the darkest brightness settings, a shoulder surfer may be unable to read *any* words on the protected screen (Table 4). Finally, we added a feature to decrease the contrast of the content displayed using Eye-Shield and evaluated the impact of decreasing contrast for both OCR and image recognition. The impact of the new feature on performance was negligible. Upon generating protected images with low and moderate-low contrast, we observed the impact of the decreased contrast on the Google Cloud APIs. For texts, the percentage of protected content is greatly increased for all parameters, while images receive little benefit (Fig. 12). This suggests that decreasing brightness and contrast is an effective method to further improve the protection rate for large fonts and high-contrast UIs.

Table 3: Results of using Google Cloud OCR on photos taken of a text message UI with several different protection mechanisms at 50% screen brightness with 5× optical zoom.

Environment	Angle	Protection	Word Recognition Rate
Dark	30 °	Eye-Shield	87
Dark	30 °	Film	83
Dark	30 °	Both	0
Dark	30 °	None	90
Dark	45 °	Eye-Shield	54
Dark	45 °	Film	0
Dark	45 °	Both	0
Dark	45 °	None	80
Light	30 °	Eye-Shield	60
Light	30 °	Film	74
Light	30 °	Both	0
Light	30 °	None	88
Light	45 °	Eye-Shield	48
Light	45 °	Film	0
Light	45 °	Both	0
Light	45 °	None	70

B Supplemental Experiment Setup Information

In this section, we provide several additional experiment details. We describe the development stack of Eye-Shield for each platform-specific implementation in Table 5. Our stack consists of image processing and general purpose GPU processing libraries.

In Table 9, we provide the distributions each dataset used in our evaluations in Sections 5.1 and 5.2.

Finally, we provide a comprehensive comparison of many other closely related works that seek to provide defenses against shoulder surfing. We do not include shoulder surfer detection systems, since they are dissimilar from Eye-Shield. Table 10 demonstrates how Eye-Shield is capable of protecting all of the information types except for PIN. Unlike privacy films, it is also able to protect against shoulder surfers directly behind the user.

C Supplemental Images, Examples, Prototypes

In this section, we provide several supplemental figures portraying Eye-Shield protected screens and images. Note that due to artifacts from downsampling and camera capture, the supplementary images and examples provimded in the paper are not as clear as they would appear to an intended user

Table 4: Results of using Google Cloud OCR on photos taken of a text message UI with several different brightness settings.

Environment	Angle	Screen Brightness	Word Recognition Rate
Dark	Close	Darkest	42.50
Dark	Close	Moderate	56.50
Dark	Far	Darkest	0.500
Dark	Far	Moderate	15.25
Dark	Side	Darkest	0.000
Dark	Side	Moderate	5.250
Light	Close	Darkest	62.00
Light	Close	Moderate	62.50
Light	Close	Brightest	71.50
Light	Far	Darkest	20.00
Light	Far	Moderate	19.00
Light	Far	Brightest	28.25
Light	Side	Darkest	0.000
Light	Side	Moderate	2.750
Light	Side	Brightest	21.00

Table 5: The development stack of for each platforms.

Platform	Language	Image Processing	GPGPU
Windows PC	Python	OpenCV	CUDA 11.6 (CuPy)
MacOS Laptop	Swift	CoreImage	Metal
Android Smartphone	C++	OpenCV	Vulkan (Kompute)
iOS Smartphone	Swift	CoreImage	Metal

viewing the original in-person.

Figure 14 shows that without a camera with optical zoom, it is very difficult to discern any of the text on the protected screen compared to the unprotected screen. Information such as names, phone numbers, and account details are redacted.

D Equations

$$\text{rms} = \sqrt{\frac{x^2+y^2}{2}} \quad \text{and} \quad y = \sqrt{\left(\text{rms}^2 \cdot 2\right) - x^2}. \tag{1}$$

$$SSIM\,(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{2}$$

$$\delta = 2\arctan\left(\frac{d}{2D}\right). \tag{3}$$

The grid square size g has a direct effect on the range at which the user can effectively operate their device, and it also has an effect on the range that a shoulder surfer can obtain information from viewing the user's device screen information. The grid square size should ideally be based on 3 factors: (1) the distance from the user to the device screen, (2) the device's pixel density (pixels per square inch, ppi), and (3) the font size. We found it optimal to set $g = 1 \times 1$ in most of our experiments performed on the iPhone 13 Pro which has a 2532 × 1170 resolution with 460 ppi display, the user viewing the device from 10″, and the default application font size. Eye-Shield can adapt this to other configurations with

Table 6: Mobile performance of `Eye-Shield` on large screens.

Resolution	Android (FPS)	iOS (FPS)
1920 × 1080	49.25	91.39
1080 × 2400	39.20	74.29
1170 × 2532	34.52	64.95
2560 × 1440	29.27	51.95
1440 × 3088	23.95	43.05

Table 7: A list of the image resolutions used in our evaluation.

Resolution	Aspect Ratio	Purpose
256 × 144	16:9	Video Resolution
426 × 240	16:9	Video Resolution
640 × 360	16:9	Video Resolution
854 × 480	16:9	Video Resolution
960 × 540	16:9	Video Resolution
1024 × 576	16:9	Video Resolution
1280 × 720	16:9	Video Resolution
1366 × 768	16:9	Video Resolution
1600 × 900	16:9	Video Resolution
1920 × 1080	16:9	Video Resolution
2560 × 1440	16:9	Video Resolution
512 × 512	1:1	`HideScreen` Comparison
1080 × 2400	9:20	Mobile Screen Resolution
1170 × 2532	90:195	Mobile Screen Resolution
1440 × 3088	90:193	Mobile Screen Resolution

Table 8: Each device used in our performance evaluations.

Device	CPU Cores	GPU	Resolution
PC Workstation	12 Cores	RTX 2080 Super (432 Cores)	1920 × 1080
2021 Macbook Air	8 Cores	Apple M1 (8 Cores)	2560 × 1600
Samsung Galaxy S20 Ultra	8 Cores	Mali G77 (11 Cores)	3200 × 1440
iPhone 13 Pro	6 Cores	A15 (5 Cores)	2532 × 1170

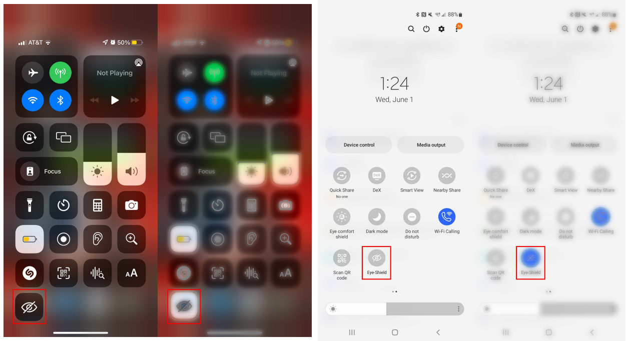
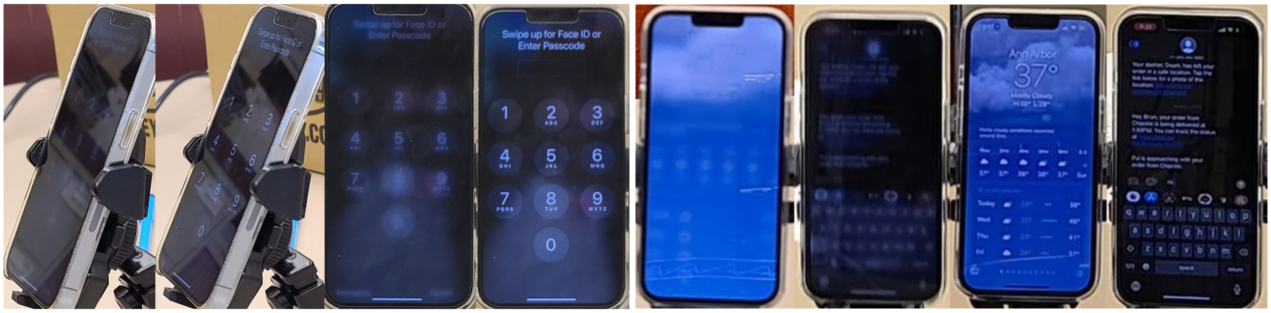


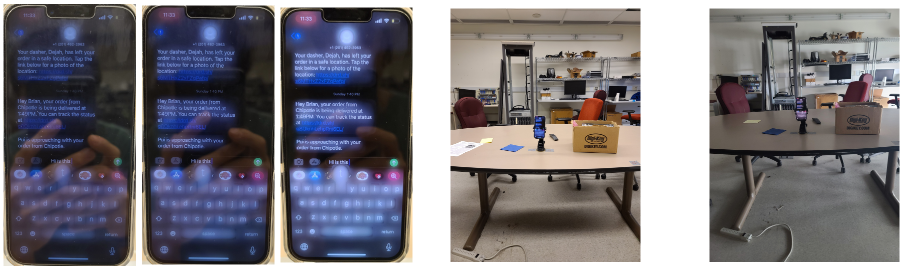
Figure 13: iOS (left) and Android (right) UI prototypes for toggling `Eye-Shield`. The red box highlights the toggle.



(a) Distance of (left) *19.7" and 45°angle* and (right) *41"*.

(b) Distance of *41"*.

Figure 14: Photos of protected and unprotected screens captured with (a) *48MP, f/1.8, 103mm smartphone camera at* $5\times$ *zoom.* and (b) *108MP, f/1.8, 26mm smartphone camera at* $1\times$ *zoom.*



(a) 33%

(b) 66%

(c) 100%

(d) Bright Env

(e) Dark Env

Figure 15: Device screen brightness from the intended user's perspective (a,b,c). Environment lighting experiment setup (d,e).

Table 9: Total number of images/frames for each dataset.

DIV2K Train	DIV2K Valid	RICO Valid	DAVIS 480p	DAVIS 1080p
800	100	1460	761	761

different pixel densities or different text sizes by using this default $g = 1 \times 1$ for 460 ppi and medium font size. We can calculate the optimal g by scaling this proportionally to the device's ppi and font size, and rounding to the nearest pixel. For example, the original HideScreen paper used this equation $(d + 12'')/3333$ (where d =distance from device to intended user) to calculate the optimal g in inches, which for 460 ppi and $d = 10''$ comes out to $g = 3 \times 3$ pixel squares. This was optimal for HideScreen which used larger font sizes.

Where g =grid square size in pixels (variable to solve for)

f =fontsize in inches (verified with 0.25)

d =distance to screen in inches (10")

p =pixel density in pixels per inch (460 PPI iPhone 13 Pro)

α =human eye resolving power in degrees (0.0167 degrees)

Using the equation of angular size: $\alpha = 2\arctan\frac{g/p}{2d}$

We get: $g = p \cdot 2d \cdot \tan(\frac{\alpha}{2})$

Scaling for font size: $g = p \cdot 2d \cdot \tan(\frac{\alpha}{2}) \cdot 4f$

Example: $g = 460 \cdot 20'' \cdot \tan(\frac{0.0167}{2}) \cdot 1.0$

Example: $g = 1.34$ pixels

Based on how much privacy protection the user wants, they can choose several protection strength settings. The examples we define in our prototype include: "full", "strong", "moderate", and "weak" protection. The contrast is adjusted using the OpenCV addWeighted() function on both the original image and the blurred target image. The privacy protection settings alter the blurring (σ) and contrast parameters as seen below:

- Full: $\sigma = 24$, contrast = 80/127
- Strong: $\sigma = 20$, contrast = 100/127
- Moderate: $\sigma = 16$, contrast = 115/127
- Weak: $\sigma = 8$, contrast = 127/127

σ refers to the standard deviation of the Gaussian blur filter. The Gaussian filter convolves over an image with a specific window size with a Gaussian distribution. This means that a larger σ value blurs/blends together a larger radius of pixels compared to a smaller σ value. This contrast parameter decreases the contrast of the output by computing:

$$\alpha = \mathit{float}(131 \cdot (\mathit{contrast} + 127))/(127 \cdot (131 - \mathit{contrast}))$$
$$\gamma = 127 \cdot (1 - \alpha)$$
$$\mathit{output\_img} = (\mathit{img} \cdot \alpha) + \gamma$$

Table 10: A comparison of Eye-Shield with other non-detection based shoulder surfing defense mechanisms.

	Eye-Shield	HideScreen [29]	**Scrawl** [27]	**Gallery** [28]	**IllusionPIN** [18]	**Privacy Shade** [26]	**Privacy Film** [11]
PIN	X	✓	X	X	✓	X	X
Text	✓	✓	✓	X	X	✓	✓
Grayscale Photos	✓	✓	X	✓	X	✓	✓
Color Photos	✓	X	X	✓	X	✓	✓
Videos	✓	X	X	X	X	X	✓
Mobile UIs	✓	X	X	X	X	✓	✓
Entire Screen	✓	X	X	X	X	X	✓
No Interruption	✓	✓	✓	✓	✓	X	✓
Protects Behind	✓	✓	✓	✓	✓	✓	X

Table 11: Main discussion questions for interview. Exact phrasing differed between participants, but overall topics were consistent.

Code	**Question**	Count/Responses
Q1	Does previous familiarity with a particular app UI help with your use of Eye-Shield?	3
Q2	Would you prefer using Eye-Shield over using a physical privacy film?	7
Q3	Would you use Eye-Shield to protect financial app information or PIN entry?	7
Q4	Would you use Eye-Shield to protect your text messages or photos?	3
Q5	Did you find the blurring that Eye-Shield causes annoying as the intended user?	7
Q6	Did you experience any eye strain when using Eye-Shield as the intended user?	3
Q7	Would you feel uncomfortable if you saw someone shoulder-surfing your device?	8
Q8	Have you experienced shoulder surfing before as the shoulder surfer or the victim?	13
Q9	Do you prefer blurring over pixelation as the intended user?	3
Q10	In what use cases could you see yourself using Eye-Shield?	Time-sensitive tasks, public transport, family snooping, lecture, church
Q11	What method would you prefer Eye-Shield to be activated with?	Widget in control center, blurring/brightness meter, automatic activation
Q12	What protection methods have you used to protect yourself from shoulder surfing?	Cover phone, lower brightness, stop usage, privacy film, Face ID
Q13	What methods did you find improved the usability as the intended user?	Zooming in, increasing brightness, leaning in
Q14	Various useful comments the participants brought up.	Difficulty with trifocals, accessibility for languages, pixelation eyestrain, PIN-entry default, older people less usage, annoyance for reading quickly