

University of Arkansas, Fayetteville

ScholarWorks@UARK

Graduate Theses and Dissertations

7-2021

Automated Privacy Protection for Mobile Device Users and Bystanders in Public Spaces

David Darling

University of Arkansas, Fayetteville

Follow this and additional works at: <https://scholarworks.uark.edu/etd>



Part of the [Graphics and Human Computer Interfaces Commons](#), [Information Security Commons](#), and the [Software Engineering Commons](#)

Citation

Darling, D. (2021). Automated Privacy Protection for Mobile Device Users and Bystanders in Public Spaces. *Graduate Theses and Dissertations* Retrieved from <https://scholarworks.uark.edu/etd/4218>

This Thesis is brought to you for free and open access by ScholarWorks@UARK. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of ScholarWorks@UARK. For more information, please contact scholar@uark.edu.

Automated Privacy Protection for Mobile Device Users and Bystanders in Public Spaces

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science

by

David Darling
University of Arkansas
Bachelor of Science in Computer Science, 2018

July 2021
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

Qinghua Li, PhD
Committee Chair

Brajendra Panda, PhD
Committee Member

Dale Thompson, PhD
Committee Member

Abstract

As smartphones have gained popularity over recent years, they have provided users convenient access to services and integrated sensors that were previously only available through larger, stationary computing devices. This trend of ubiquitous, mobile devices provides unparalleled convenience and productivity for users who wish to perform everyday actions such as taking photos, participating in social media, reading emails, or checking online banking transactions. However, the increasing use of mobile devices in public spaces by users has negative implications for their own privacy and, in some cases, that of bystanders around them.

Specifically, digital photography trends in public have negative implications for bystanders who can be captured inadvertently in users' photos. Those who are captured often have no knowledge of being photographed and have no control over how photos of them are distributed. To address this growing issue, a novel system is proposed for protecting the privacy of bystanders captured in public photos. A fully automated approach to accurately distinguish the intended subjects from strangers is explored. A feature-based classification scheme utilizing entire photos is presented. Additionally, the privacy-minded case of only utilizing local face images with no contextual information from the original image is explored with a convolutional neural network-based classifier. Three methods of face anonymization are implemented and compared: black boxing, Gaussian blurring, and pose-tolerant face swapping. To validate these methods, a comprehensive user survey is conducted to understand the difference in viability between them.

Beyond photographing, the privacy of mobile device users can sometimes be impacted in public spaces, as visual eavesdropping or "shoulder surfing" attacks on device screens become feasible. Malicious individuals can easily glean personal data from smartphone and mobile

device screens while they are accessed visually. In order to protect displayed user content, a novel, sensor-based visual eavesdropping detection scheme using integrated device cameras is proposed. In order to selectively obfuscate private content while an attacker is nearby, a dynamic scheme for detecting and hiding private content is also developed utilizing User-Interface-as-an-Image (UIaI). A deep, convolutional object detection network is trained and utilized to identify sensitive content under this scheme. To allow users to customize the types of content to hide, dynamic training sample generation is introduced to retrain the content detection network with very few original UI samples. Web applications are also considered with a Chrome browser extension which automates the detection and obfuscation of sensitive web page fields through HTML parsing and CSS injection.

Acknowledgments

I thank my dissertation advisor, Dr. Qinghua Li, for all of his helpful advice and aid throughout my studies in both undergrad and graduate school. The guidance throughout my studies has proven immensely valuable, as I navigate the peaks of higher education.

I extend my appreciation to my committee members, Dr. Thompson and Dr. Panda, for their valuable advice and cooperation in completing this thesis.

I thank Yaling Liu for her helpful contributions in implementing the browser-based content detection and obfuscation extension.

I would also like to thank my parents Web and Terri for all their support during the completion of my research topics. This work wouldn't have been possible without their help.

Table of Contents

| | |
|---|----|
| 1 – Introduction..... | 1 |
| References..... | 4 |
| 2 – Automated Bystander Detection and Anonymization in Mobile Photography | 5 |
| 2.1 – Introduction..... | 5 |
| 2.2 – Related Works..... | 6 |
| 2.3 – System Overview | 9 |
| 2.4 – Feature-Based Bystander Classifier | 10 |
| 2.4.1 – Feature Identification | 10 |
| 2.4.2 – Feature Extraction and Computation | 11 |
| 2.4.3 – Supervised Learning Model Consideration | 18 |
| 2.5 – CNN-Based Bystander Classifier | 19 |
| 2.5.1 – Network Architecture..... | 19 |
| 2.6 – Model Evaluation..... | 19 |
| 2.6.1 – Dataset..... | 19 |
| 2.6.2 – Feature-Based Bystander Classification | 20 |
| 2.6.3 – CNN-Based Bystander Classification..... | 24 |
| 2.7 – Anonymizing Bystander Faces | 26 |
| 2.7.1 – Implementation of Obfuscation Methods | 27 |
| 2.7.2 – Survey of Users on Face Anonymization | 28 |
| 2.8 – Conclusion | 32 |

| | |
|---|----|
| References | 34 |
| Appendix | 37 |
| A – IRB Approval | 37 |
| 3 – Sensor-Based Detection and Dynamic Mitigation of Visual Eavesdropping on Mobile Devices and Web Browsers | 38 |
| 3.1 – Introduction | 38 |
| 3.2 – Related Work | 39 |
| 3.3 – Mobile Attack Scenario | 41 |
| 3.4 – Sensor-Based Eavesdropping Detection Scheme | 42 |
| 3.4.1 – Overview | 42 |
| 3.4.2 – Magnetometer Normalizing and Thresholding | 43 |
| 3.4.3 – Facial Detection Network | 44 |
| 3.5 – Automated Private Content Detection and Hiding | 46 |
| 3.5.1 – User Interface as an Image (UIaaI) | 46 |
| 3.5.2 – Private Content Detection Network | 49 |
| 3.5.3 – User-Defined Content Retraining | 52 |
| 3.5.4 – Evaluation of Attacker Detection and Content Hiding Scheme | 55 |
| 3.6 – Browser Extension-Based Private Content Hiding | 56 |
| 3.6.1 – Chrome Extension Overview | 56 |
| 3.6.2 – Identifying Sensitive Content | 56 |
| 3.6.3 – Performing Dynamic Content Detection | 57 |
| 3.6.4 – Applying CSS Styles | 58 |

| | |
|-------------------------------------|----|
| 3.6.5 – Evaluation | 59 |
| 3.7 – Conclusion | 60 |
| References..... | 62 |
| 4 – Conclusion and Future Work..... | 64 |

List of Figures

| | |
|--|----|
| Figure 1: Example image with target clearly featured in the foreground and bystanders included to the left and right in the background. Image source http://mensstreetfashion.weebly.com/home/kanye-west-style | 9 |
| Figure 2: Architecture of the system..... | 10 |
| Figure 3: Standard template of facial landmarks fitted by common face detection networks..... | 12 |
| Figure 4: Gaze vectors from the CLNF model in green with face landmarks from the CE-CLM model on an example image with no bystanders. Original image by Steve Granitz, WireImage. | 18 |
| Figure 5: Overall architecture of the convolutional neural network..... | 19 |
| Figure 6: Progressive loss of the network for each mini-batch. | 25 |
| Figure 7: Progressive accuracy over training mini-batches for the CNN classifier. | 26 |
| Figure 8: From left to right: original image, image with black boxing, image with blurring, image with face swapping. Image source: https://cdn.ebaumsworld.com/2008/08/861915/phelps01.jpg | 27 |
| Figure 9: Survey responses to opinion questions 1-7. | 32 |
| Figure 10: Survey responses to rating the impact of anonymization methods on photos. | 32 |
| Figure 11: Survey responses to rating how willing users would be to use each anonymization method..... | 32 |
| Figure 12: Visualization of a potential attack scenario. The vulnerable “danger areas” relative to the user are shown in red with attacker gazes shown as black dashed lines..... | 41 |
| Figure 13: Architecture of feature extraction portion of facial detection network | 45 |
| Figure 14: Example detected face with bounding box in green and facial anchor points plotted in red. | 46 |
| Figure 15: Architectural overview of the private content detection network. | 50 |
| Figure 16: Bounding box mean average precision, precision, and recall metrics over progressive training epochs. Note that charts are smoothed with original data shown as a shadow behind. The model is found to converge under all metrics after 450 epochs. | 51 |
| Figure 17: Sample text detection output with confidence metric. The network is capable of accurately detecting text messages of varying sizes and positions. | 52 |

| | |
|---|----|
| Figure 18: Visualization of random transformations and augmentations applied over a subset of the full text message dataset. | 54 |
| Figure 19: CSS blurring styles applied to the automatically detected Twitter sign up page. | 59 |

List of Tables

| | |
|---|----|
| Table 1: Accuracy, precision, and recall metrics for all classifier models. (T) and (B) indicate that the metric was computed with target or bystander respectively as the positive class. | 23 |
| Table 2: Validation accuracy for models trained using feature-subsets. | 23 |
| Table 3: Average single prediction forward-pass runtime (Intel i9-10900k) | 23 |
| Table 4: Trial outcomes for each simulated visual eavesdrop attack. Attacker position is relative to the user. Refer to Fig. 12 for positioning information. | 55 |
| Table 5: Private content detection results for examined web pages based on login, sign up, and misc. categories. | 60 |

List of Published Papers

Chapter 2: Automated Bystander Detection and Anonymization in Public Photography

This work has been published as a preliminary workshop version and as a full conference version.

Workshop version:

D. Darling, A. Li and Q. Li, "Identification of Subjects and Bystanders in Photos with Feature-Based Machine Learning," *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 1-6.

Conference version:

Darling D., Li A., Li Q. (2020) Automated Bystander Detection and Anonymization in Mobile Photography. In: Park N., Sun K., Foresti S., Butler K., Saxena N. (eds) Security and Privacy in Communication Networks. SecureComm 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 335. Springer, Cham. https://doi.org/10.1007/978-3-030-63086-7_22

1 – Introduction

Due to the rapid increase in popularity of smartphones and mobile devices such as tablets and smart watches, their use has consequently risen among consumers in public spaces. Although this trend offers unparalleled levels of convenience for users, there are negative consequences for the privacy of individuals in public. Two main privacy concerns are identified for mobile device users and bystanders alike: the increase of digital photography in public causing bystanders to be inadvertently captured and the potential for visual information leakage by eavesdroppers maliciously looking at device screens while they are in use by unaware users. In order to provide solutions for these growing issues, this thesis presents approaches which attempt to solve the separate privacy concerns.

Digital photography is an enormously growing trend among the general public spurred primarily by the prevalence of smartphones with cameras in the daily lives of users. Attempts at estimating the number of digital photos captured annually have shown significant increases year over year with no sign of slowing down. In 2016, 660 billion photos were estimated to have been captured which increased to 1.2 trillion by 2017. Additionally, smartphones were estimated to be primarily responsible for these photos, as they captured roughly 85% of the photos taken. Digital cameras, the next largest device group, made up only 10.3% by comparison [1]. This significant increase in digital photography demonstrates that people are more likely to be taking photos across many scenarios, including in public locations where oftentimes other people are nearby. Due to the crowded nature of public locations, strangers are frequently included in personal photos by circumstance. Fig. 1 provides an example of such an image taken in public including both a target person and strangers. These strangers, or bystanders, often are completely unaware of photos being taken of them. Even realizing that a photo was taken, individuals usually have

very little recourse to ask to be removed from the photo or have the photo deleted in a public setting.

To address these issues caused by smartphone photography, Chapter 2 details and evaluates a solution for the privacy impacts caused by digital photography in public places [cite your workshop paper and conference paper here]. An automated solution for distinguishing bystanders from subjects in photos is explored. Additionally, several different methods for automatically obfuscating bystander faces caught in photos are explored. Specifically, black boxing, blurring, and dynamic face swapping are implemented and evaluated in terms of potential privacy protection and visual impact on photo quality. A user study is carried out with over 80 respondents to further evaluate the approach and ensure that the privacy concerns generated from public photography are a valid concern among users.

Similar to increasing digital photography trends, smartphones and mobile devices such as tablets, smart watches, etc. have rapidly grown in popularity in recent years. These types of devices offer unparalleled convenience and ease of access for end users who increasingly need to be able to access applications and services on the go. Estimates of smartphone sales trends have shown that the market has rapidly grown since the early 2000s. In 2019 alone, smartphone manufacturers sold an estimated 1.5 billion devices [2]. This is an enormous increase over estimates in 2007 which place sales at only 122 million units [3]. With this saturation of mobile devices among consumers, usage has likewise increased rapidly in public spaces.

Due to the convenience of having access to emails, text messages, and other applications, individuals frequently utilize devices in scenarios such as eating, riding a subway, or while walking where others behind or to the side of a user can easily see or visually eavesdrop the content on their device screens. This clearly constitutes a security risk, as a majority of users are

likely to access sensitive apps in public [4] such as text or productivity. Users accessing their smartphones generally have no way of easily knowing if an attacker is spying on their screen.

Chapter 3 proposes a sensor-based scheme to provide smartphone users with a means to quickly and easily detect attackers attempting to glance at their phone screen. A magnetometer binary thresholding method is utilized to filter out user faces while utilizing integrated front facing cameras to discretely scan around and behind the user to find potential attackers. To allow users to still utilize apps while protecting sensitive content, user interface as an image (UIaaI) is proposed to pre-render UI views. These pre-rendered views can be used to automatically detect private content to avoid the necessity of large code changes for complex applications. Beyond mobile applications, a web browser extension is developed to obfuscate sensitive content in web applications using HTML parsing and CSS injection techniques.

References

- [1] Richter, F.: Infographic: Smartphones cause photography boom (Aug 2017), <https://www.statista.com/chart/10913/number-of-photos-taken-worldwide/>
- [2] L. Goasduff, "Gartner says global smartphone sales fell slightly in the fourth quarter of 2019," Mar 2020. [Online]. Available: <https://www.gartner.com/en/newsroom/press-releases/2020-03-03-gartner-says-global-smartphone-sales-fell-slightly-in>
- [3] A. Liang, "Market share: Smartphones, worldwide, 4q07 and 2007, "2008. [Online]. Available: <https://www.gartner.com/en/documents/619509/market-share-smartphones-worldwide-4q07-and-2007>
- [4] L. Rainie and K. Zickuhr, "How americans use their cell phones in public," May 2020. [Online]. Available: <https://www.pewresearch.org/internet/2015/08/26/chapter-2-phone-use-in-public-areas/>

2 – Automated Bystander Detection and Anonymization in Mobile Photography

2.1 – Introduction

With the rising popularity of mobile photography by consumers causing an unprecedented increase in photos taken in public spaces, bystanders are frequently caught unaware by user cameras. To help mitigate the privacy concerns caused by this trend, this work explores solutions for bystander detection and studies effective ways to obfuscate bystander faces. To this end, both a feature-based classification approach utilizing entire photos and a privacy-minded, convolutional neural network (CNN)-based approach utilizing only local face images with no contextual information from the original photo are proposed to investigate how different models learn the intrinsic visual differences between targets and bystanders. In order to train and evaluate the models, a real-world dataset consisting of over 200 photos and over 500 faces has been created to provide a generalized representation of the types of images that commonly can be found uploaded to social media. Generally, the photos provide a mixture of both celebrities appearing in public with bystanders behind them and typical people in front of landmarks or other locations of interest with strangers also inadvertently captured in the photo.

In addition to the classifiers, methods for effective facial obfuscation also are implemented and examined through a user study (with IRB approval) as a part of this work. The methods for anonymizing faces not only include standard methods such as Gaussian blurring and black boxing but also a novel approach of face swapping using a state-of-the-art position map regression network [2]. To gain a fuller understanding of how these obfuscation methods impact and are perceived by end users, an in-depth survey is carried out and evaluated to determine the preferred methods of actual users. General opinions on privacy as it relates to digital photos capturing strangers are also collected to validate the assumptions behind this work.

This work's main contributions are summarized as follows:

- An automated system for protecting bystander's privacy in mobile photography with a unique feature that it can work as a standalone tool on a user's smartphone, without relying on inter-user interaction or any online platforms which are commonly needed by previous solutions.
- A novel, feature-based classification approach utilizing entire photos and a privacy-minded CNN-based approach utilizing local face images for automatically distinguishing targets from bystanders in mobile photography. The two approaches are evaluated and compared to explore the tradeoff between the distinguishing accuracy and privacy.
- The implementation of three face obfuscation methods for complete face anonymization including black boxing, blurring, and face swapping. A user study is carried out with 89 respondents to evaluate opinions on bystander privacy and on the acceptability of the three face anonymization methods.

The remainder of this chapter is organized as follows. Chapter 2.2 provides an overview of related works. Chapter 2.3 presents an overview of the design for the system. Chapter 2.4 describes the feature-based approach for target/bystander classification. Chapter 2.5 describes the convolutional neural network approach. Chapter 2.6 provides evaluations of both models. Chapter 2.7 provides a technical description of proposed facial obfuscation methods and presents the results from the user survey. Chapter 2.8 presents conclusions.

2.2 – Related Works

Works related to improving facial privacy in photography have previously followed trends such as utilizing photographer-bystander cooperation. Li et al. [3,4] design systems for

cooperation between smartphone users to blur requesting users' faces. Jung and Philipose [5] utilize a method of gesture recognition to detect a person who wants to be excluded from a photo. MarkIt [6] can perform automated covering of user-defined objects in photos, but users must manually predefine objects to be hidden. This work, by comparison, explores a fully automated approach towards the identification of bystanders to be obfuscated and requires no manual interaction from photographers or those captured in photos.

Other approaches in automated anonymization require users interested in having their privacy protected to wear specialized markers. Schiff et al. [7] propose a scheme whereby specialized markers that users wear can be recognized by an automated system which will then blur their faces. Bo et al. [8] similarly utilize worn QR-codes on clothing to automatically determine individual privacy preferences. Our approach does not require any worn markings; only visual attributes of persons within individual images are used to anonymize those who are inadvertently captured.

Some works explore photo privacy protection for individuals in online social network settings [9, 10, 11] or in individual phones [12]. Li et al. proposed HideMe [10], a system for defining scenarios for photo access control and distance-based face blurring. Xu et al. [11] developed a facial identification system to incorporate captured persons into the decision process of sharing photos. Ilia et al. [9] proposed fine grained access control based on face detection to prevent individual faces from being viewed by other users. Each of these works depend upon either accurate facial recognition to identify bystanders for anonymizing or specific scenario definitions from the photographer. This work circumvents the need for both face recognition and input from the photographer by automating the detection of bystanders and providing low-impact solutions for obfuscating their faces.

One recent work in parallel to this one also explores detection of bystanders utilizing features computed over individuals [13]. However, the features used in this work are different from theirs. Also, that work focuses entirely on feature engineering and effectiveness of predictive models, but this work presents a full system for both detecting and anonymizing bystanders and explores the trade-off between bystander protection and photo quality/usability. Additionally, privacy-aware machine learning is considered with the development of a CNN which does not require access to an entire user photo and can predict solely based on facial images. While Hasan et al. [13] explore both transfer learning on deep neural networks, this approach utilizes a simplified feature-set which allows for simpler and smaller classifiers.

Our preliminary work [14] introduced the computation of facial features including face size, face deviation, and gaze direction for distinguishing targets and bystanders. However, that work only introduced basic methods of gathering these features and did not propose a full feature set. The features proposed in this work expand upon the face size, deviation, and gaze concepts with a more accurate CE-CLM model and more explicitly define feature computations. Additionally, this work utilizes a full feature set to create a complete scheme for bystander protection with automated obfuscation methods.



Figure 1: Example image with target clearly featured in the foreground and bystanders included to the left and right in the background. Image source <http://mensstreetfashion.weebly.com/home/kanye-west-style>

2.3 – System Overview

The proposed system works on the photographing phone and consists of four main processes that can automatically run over taken photos. The first is an initial pass of face detection which identifies face regions and positions. This process of facial detection is fully automated and requires no manual user input specifying specific photos or regions to be focused by the algorithm. Facial region data from this pass can then be forwarded either to the feature-based or CNN classifiers which automate the detection of targets and bystanders. The classifications along with face landmarks are then used to perform obfuscation processing with the black boxing, blurring, and face swapping methods which can be selected by the user as a system configuration parameter. The resulting anonymized photo is the final output. Fig. 2 provides a visualization of this system. Both classifiers are robust against nonstandard scenarios

where a photo might not include any bystanders or any human targets (pictures of scenery are an example of this.)

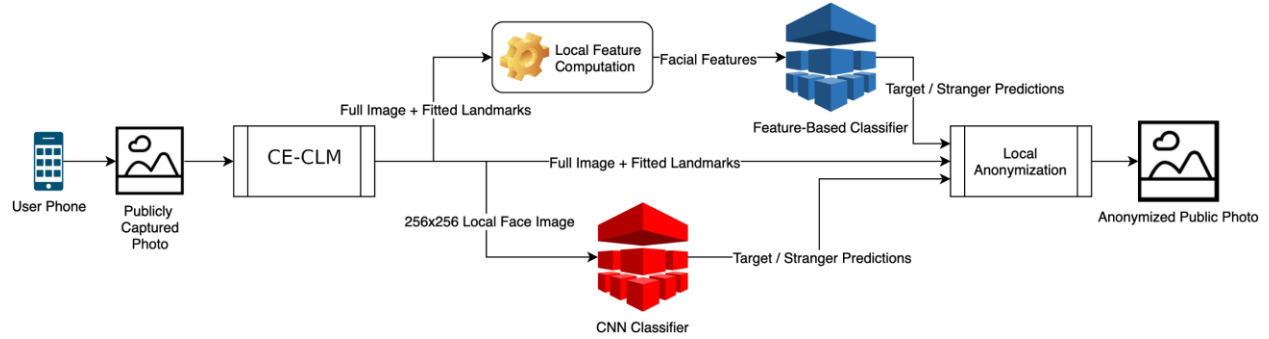


Figure 2: Architecture of the system.

2.4 – Feature-Based Bystander Classifier

2.4.1 – Feature Identification

In order to begin designing a model for classification of an abstract concept such as who is the desired target of a photo, it is essential to correctly break the problem down into quantitative measurements which provide a suitable amount of information to distinguish classes. Relative face size (the size of a given face relative to the maximum face size in the image) and face deviation from center are identified to model the fact that bystanders are often included in the background or periphery of photos. Local blurriness of a face is also found to regularly indicate a person was not the intended focus of a picture. Beyond these metrics of relative visual differences, additional features are identified to model the fact that bystanders are usually unaware of having their photo captured. Additionally, non-relative features are needed to account for scenarios where photos might exclude any human targets (for example bystanders are captured in an image of scenery or some other non-human target). Head pose angles and gaze deviation from the camera are decided as good indicators of this, as bystanders will commonly not be looking at the camera capturing them if they are unaware of it. Considering all of these

metrics in a single model, relative face size, face deviation from photo center, local blurring, head pose, and gaze deviation are identified as a sufficient number of features to capture the complexity of the target/bystander classification problem.

2.4.2 – Feature Extraction and Computation

Relative Face Size

In order to accurately capture face size in an image, traditional bounding box methods used by most state-of-the-art object detection models such as YOLOv3 [15] are not sufficient. These bounding boxes normally have no guarantee of forming a tight bound on the object in question. Instead, we recognize that recent advances in constrained local neural fields (CLNFs) for facial landmark detection are more suited for the task of robust landmark placement for photos taken in the wild due to their resistance to factors such as pose differences, lighting changes, and local facial differences such as hair or accessories. To this end, we adopt a Convolutional Experts Constrained Local Model (CE-CLM) [16] [17] to perform accurate facial landmark placement. These types of models function by first capturing landmark shape variations with a point distribution model (PDM) and then modeling the local differences in the visual appearances of fitted landmarks with the use of local patch experts. This model, pretrained over LFPW [18] and Helen [19] training sets, is able to accurately determine landmark positions and provided higher detection rates on smaller faces as well as partially occluded faces as compared to more popular systems such as Dlib [20] during experimentation.

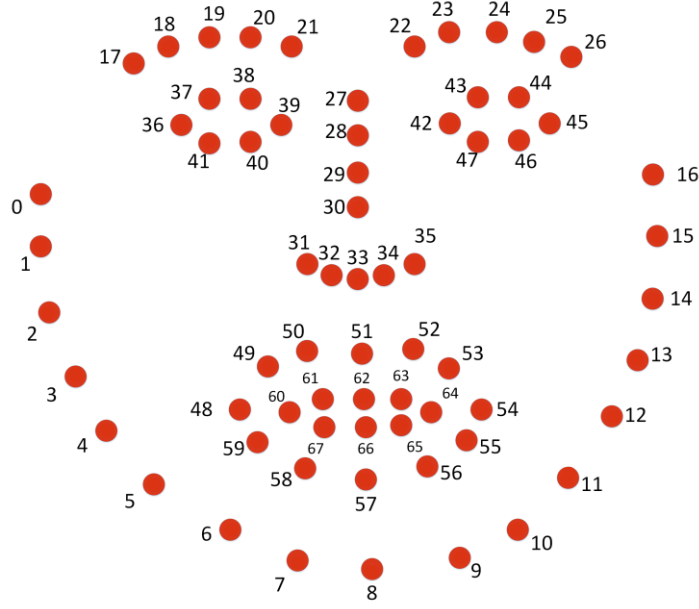


Figure 3: Standard template of facial landmarks fitted by common face detection networks.

Fig. 3 provides a visualization of the standard collection of facial landmarks which are fitted by networks such as Dlib and the CE-CLM model. Refer to this figure for locating any numbered facial landmarks mentioned subsequently. Once a tight bound has been formed around each face using these landmarks, the maximum size of all faces found in the photo is calculated as:

$$\max_{0 \leq i \leq n} S(i) = (\alpha_i - \beta_i)(\gamma_i - \delta_i) \quad (1)$$

where i refers to the index of the current face in the photo which can have n faces, α_i is the x-coordinate of facial landmark 16. β_i is the x-coordinate of landmark 0, γ_i is the y-coordinate of landmark 24, and δ_i is the y-coordinate of landmark 8. The subscript i for each of these variables indicates that these are specific to the face with index i . Similarly, the relative face size metric for each face is then calculated simply as:

$$R(x) = \frac{s_i}{\mu} \quad (2)$$

where s_i is the size of face i and μ is the maximum face size computed in equation 1. Thus, the relative size of any given face will always fall within the bounds of (0,1].

Deviation of Face from Center

Face position within an image can be extracted and computed in a method similar to the relative face size metric. Facial landmarks are fitted to each detected face in an image with the center of the face treated as landmark 33. By utilizing this landmark, it is possible to accurately extract what region of the image a given face is located assuming the dimensions of the image are known. To create a useful metric for supervised learning models, the face position is computed as the amount of deviation from center of the image. The intuition here is that the feature generally should (but not always) provide a positive correlation with the likelihood that a given face belongs to that of a stranger. To better capture the meaning behind a person's face deviation from the center of a photo, the metric should be normalized such that its range of values carry the same meaning across different photo sizes. To this end, the deviation of a detected face is calculated as:

$$D(i) = \frac{|\epsilon_i - \zeta|}{w} + \frac{|\eta_i - \theta|}{h} \quad (3)$$

Here, ϵ_i is the x-coordinate of landmark 30, ζ is the x-coordinate of the computed center of the image, η_i is the y-coordinate of landmark 30, θ is the y-coordinate of the center of the image, w is the width dimension of the image, and h is the height dimension of the image. The subscript i , as in previous equations, refers to the i -th detected face in the image. The deviation in both x and y coordinate axes is summed for this metric because their relative importance towards determining the likelihood of a person being a bystander in an image is unclear. For example, consider a case where a target is featured centrally in front of public stairs, and a bystander is captured farther up the stairs in the image. Although the bystander might be

centrally located in the x-axis, their y-axis deviation is more important in this case. Because of this ambiguity in importance across image cases, both axis deviations are weighted the same when computing the overall deviation. The bound of this metric then becomes $[0,2]$.

Local Blurring

Many methods exist to compute the amount of blurring that occurs over a localized region in an image. One of the most popular methods is to compute a Fast Fourier Transform over an image to break it down into its constituent frequencies and perform frequency domain analysis on the results. This method is not ideal for generating a general metric of blurriness across photos as it is difficult to identify the specific frequencies in the general case which mark a region as blurry vs. another. Instead, the convolution of the Laplacian method proposed by Pech-Pacheco et al. [21] is selected for its ability to provide average edge variance in an image as a single floating-point result. This is desirable from a feature engineering perspective because blurriness, as a measure of edge variance, effectively captures the desired information from an image region without requiring a secondary classifier to convert frequency component information into a blurriness boolean.

To perform this method, the Laplace operator which is defined in 2-dimensional Cartesian space canonically as:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (4)$$

However, in a discrete grid such as an image, the discrete Laplacian is used which essentially is a convolution of the following kernel:

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (5)$$

This operation is therefore used to generate a floating point “rating” of the blurriness over each local region of detected faces defined by their bounding boxes extracted by padding the space around the outer facial landmarks. Bounding boxes are used in this case rather than the tight bounds featured in previous features because the wider region around the face contains additional edges for the Laplacian convolution operation. This provides a better idea of how much variance is in the face's surrounding edges.

Head Pose Estimation

Head pose defines the way an individual's head is oriented in 3-dimensional space. Intuitively, being able to extract some measurements of head orientation should capture the trend of strangers having their heads turned away from a capturing camera. To define head orientation, 3 main parameters are needed:

- Pitch: This defines the angle a head is looking up or down. Essentially a measure of vertical tilt.
- Roll: The angle a head is tilted from side to side. Note that this is distinct from yaw in that roll can vary while the face remains looking forward.
- Yaw: The angle a head is turned from left to right. For example, when turning to look at something behind a person, the head yaw angle becomes more extreme.

Each of these angles are intrinsically within the CE-CLM models as they internally keep a 3D representation of the fitted face landmarks. These 3D representations can be utilized to estimate accurate head pose information by solving the n-point in perspective problem [16] [17] [22]. This procedure essentially allows for accurate estimation of head pose angles in general

images. These angles are represented as continuous floating-point values which can range both in positive and negative directions.

Gaze Deviation

This feature is intended to provide additional information about whether a given person has awareness of being included in a photograph. Although head pose angles can provide information whether the face in question is oriented toward the camera, these angles do not tell the whole story. A stranger in a photo could very well have their head oriented toward the camera if, for example, they are walking behind someone taking a selfie. To provide more information to learning models about these situations, gaze deviation from the center of the camera focus could be included.

In order to extract gaze angles, a slightly altered method of utilizing CLNFs proposed by [23] is adopted. In the original work, CLNFs were utilized to form a PDM of the eye landmarks for synthesized eyes. This estimation of the shape of the eye can then be used to further estimate what direction the eye is oriented in (and where the eye is looking). The accuracy of the gaze estimation was improved utilizing pre-determined intrinsic camera parameters. However, because photos captured by mobile devices in the wild provide no information about these parameters, we adopt the model to utilize some default values for these camera parameters. These include estimated distance of the face from the camera, focal length, and optical center. This sort of calibration cannot be done for images collected for the training dataset as there simply are too many camera-to-person-positionings to take into account. Therefore, the gaze vectors should be considered rough estimates of whether a given person is looking in the general direction of a camera. The rationale behind this is that a binary value of looking vs. not looking

at a camera is still a desirable trait in identifying bystanders in a mobile photo. Example gaze vectors plotted over an image can be seen in Fig. 4.

By utilizing extracted gaze vectors, it is possible to trace a ray from the center of a given face into the estimated camera location. Then, by organizing world coordinates with the camera at the origin (0,0,0), it is possible to estimate the position of a face from the camera using coordinates. The tracing process begins by setting up the following equation:

$$\kappa_z + u\vec{\lambda}_z = 0 \quad (6)$$

Here κ_z refers to the z-coordinate or estimated depth of facial landmark 30, u is a scaling factor which must satisfy the equation, and $\vec{\lambda}_z$ is the z-coordinate vector of the average gaze angle for a given face. Solving this equation for scaling factor u allows us to then scale the other components of the full gaze vector $\vec{\lambda}$:

$$\vec{\lambda}' = u\vec{\lambda} = u \begin{bmatrix} \vec{\lambda}_x \\ \vec{\lambda}_y \\ \vec{\lambda}_z \end{bmatrix} \quad (7)$$

To compute the point ρ where the ray traced from the face intercepts with the origin or camera z-plane, the scaled gaze vector must be added to κ (landmark 30):

$$\rho = \kappa + \vec{\lambda}' \quad (8)$$

The value of ρ enables a deviation estimate for anyone looking in the direction of the camera's z-plane. Using the x and y components of ρ , and assuming the camera is the origin of the world coordinates, we can use a simple 2-D distance formula calculation to find the final deviation:

$$D = \sqrt{\rho_x^2 + \rho_y^2} \quad (9)$$

The resulting value of D computed for any given detected face is the final metric for a supervised learning model.

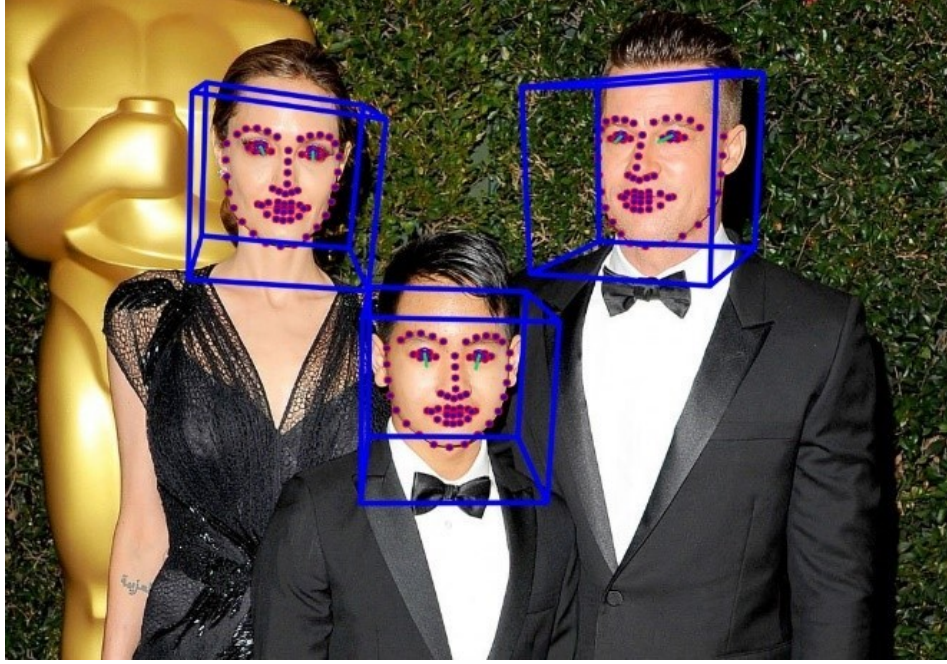


Figure 4: Gaze vectors from the CLNF model in green with face landmarks from the CE-CLM model on an example image with no bystanders. Original image by Steve Granitz, WireImage.

2.4.3 – Supervised Learning Model Consideration

Because all of the facial features are collectively designed to fully capture the complexity of the bystander classification problem, they should generalize very well to a wide range of supervised learning models. In this implementation, a collection of diverse classifiers are implemented and evaluated, including Gradient Boosted Decision Tree, Multilayer Perceptron, Random Forest, and Support Vector Machine. Detailed descriptions of these algorithms as well as discussions of their effectiveness in learning from the computed features are presented in the evaluation section.

2.5 – CNN-Based Bystander Classifier

2.5.1 – Network Architecture

Fig. 5 provides an overview of the complete network architecture. The network utilizes increasingly small convolutions separated by max pooling layers. The activation functions utilized by the network are rectified linear unit (Relu) for the two convolutional layers and sigmoid for the final activation layer. The dense layers in the latter portion of the model are intended to produce meaning from the large feature vector and condense them into more usable, countable features. The final dropout layer is included to reduce overfitting on the training set. It is set to drop inputs at a rate of 0.25 which experimentally achieved best results. The filter size of the convolutional layers is small, at a size of (2,2). This is to ensure that fine-detailed features such as eye direction might be captured, as eyes in the facial dataset can sometimes be very small (only being formed from a few 10s of pixels). Stride for each of these kernels is set to (1,1), such that a direct sweep of the kernel is performed over the image.

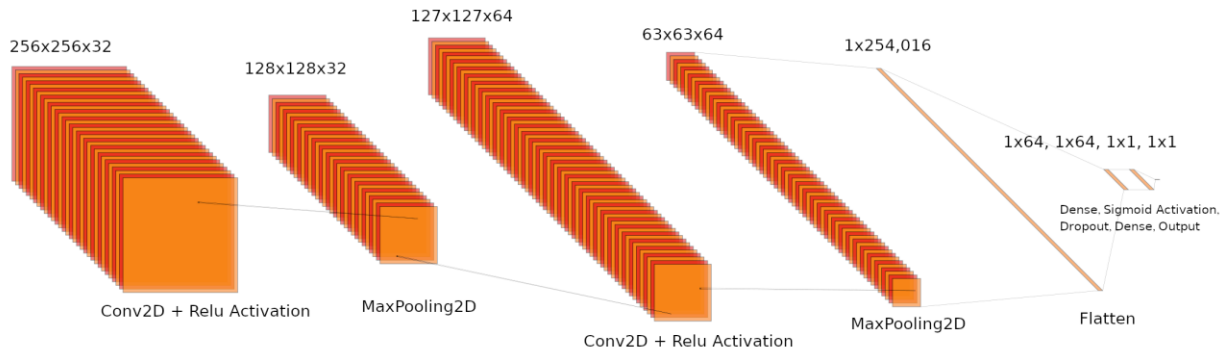


Figure 5: Overall architecture of the convolutional neural network.

2.6 – Model Evaluation

2.6.1 – Dataset

The dataset for this problem was created specifically for this project due to the fact that no existing datasets could be found at the time which provided the types of required images. The

images in the set were automatically collected from freely available online sources including social media platforms, public news sites, and image repository sites. Collected photos were manually reviewed to ensure the dataset would be able to provide a good generalization of the types of images that might be encountered in the wild. To be included, photos were required to have at least one human face. Photos could have all targets, all bystanders, or a mixture of both. The photos are all taken in various settings including indoor/outdoor locations, public venues with varying degrees of crowding, and daytime/nighttime lighting conditions. Photos excluding human targets such as scenery photos were also included to ensure the developed models could generalize to these challenging scenarios. In total, the dataset consists of 515 valid face images extracted from 222 photos. It is worth noting that our dataset is of comparable size to Hassan et al. [13] (515 facial images vs 600). In order to allow other researchers to utilize these images and contribute new images, we have made the dataset publicly available at [24].

2.6.2 – Feature-Based Bystander Classification

Model Selection and Implementation

The following supervised learning algorithms were selected and implemented to provide a good coverage of the varieties of popular classifiers and demonstrate that even with multiple different algorithms, our features generalize well.

Gradient Boosted Decision Tree (GBDT)

GBDTs are an enhancement over normal decision trees whereby an ensemble of weaker models are utilized to form a single classifier [25]. The number of estimators or trees used was 300. A maximum depth of 10 was also selected for the classifier. The learning rate was set to 0.03 after multiple training attempts.

Random Forest (RF)

Instead of the weak models favored by gradient boosting methods, the RF approach is to make use of deep, fully grown trees and average them together to reduce variance and overfitting [26]. The RF algorithm was selected to offer a good comparison with GBDTs. The number of estimators for this algorithm was selected as 50 with a max depth of 6.

Support Vector Machine (SVM)

Because SVMs are considered very suitable for binary classification [27], they were chosen for evaluation alongside other more advanced algorithms. Best results were achieved with a linear kernel, with hyperparameter C set to 10 and gamma set to 0.001.

Multilayer Perceptron (MLP)

Neural networks are a logical choice for a feature-based model such as this. The chosen architecture for the MLP is 3 hidden layers of size (7,5,3). The hyperbolic tangent function was selected for activation, and a learning rate of 0.03 is used with alpha parameter set to 0.0001. After multiple training attempts, this coupling of architecture and hyperparameter values provided best results.

Training and Evaluation

Each of the feature-based algorithms was trained over a random 80/20 train/test sample split of 515 feature sets (one set for each face image in the dataset). Validation metrics are shown in Table 1. Of the classifiers we trained, the GDBT and MLP neural network were able to achieve the best validation accuracy at 94.34%. The RF and SVM were still able to achieve acceptable accuracy over 90%. The reasoning for this is that the simpler SVM model had a higher tendency to overfit the training set and resulted in less generalizable models. The RF

model suffered from the lower depth of underlying decision trees relative to the GBDT but required less processing to perform prediction passes.

Different classifiers were able to predict targets and bystanders with varying degrees of effectiveness. This is shown by the precision, recall, and F-1 score of each algorithm. Precision is defined as the number of true positives out of the combined number of true positives and false positives. Recall is defined as the number of true positives out of the combined number of true positives and false negatives. F1-score is the harmonic mean of the two. The MLP actually had the overall highest F1 scores with 0.93 (target positive) and 0.94 (bystander positive). The GDBT suffered from significantly worse precision when predicting targets which indicates that it might have difficulty identifying relevant targets in the wild. It achieved the same target F1 score as the SVM model which had the lowest scores for both targets and bystander prediction.

In order to determine how effective each of the engineered features was individually to the classifiers, the algorithms were trained over different subsets of the complete feature-set. Each subset had one feature removed and the others included. Table 2 lists the validation accuracies for these models. Of the features tested for exclusion from the models, the gaze deviation metric, when removed, had no real impact on the performance of the GBDT and only a small impact to the other models. By contrast, both the face size and center deviation metrics had significant harmful impacts on all of the classifiers. This could be an indication that the inherent inaccuracy of the gaze metric itself was a problem for training. Additionally, the gaze metric could simply be redundant where head pose information might have been sufficient. For these reasons, the gaze deviation metric could be a candidate for removal in the interest of improving training and prediction speed. Table 3 shows the measured single forward-pass runtimes for each of the examined classifiers averaged over 1000 runs. All feature-based

classifiers have almost negligible runtime requirements for prediction operations on an Intel i9 platform. This indicates they would be excellent candidates to run directly on resource-constrained mobile devices. Because of the streamlined set of features used by the classifier, the computational complexity of all classifiers was kept low and should be suitable for direct implementation on resource constrained mobile devices.

Table 1: Accuracy, precision, and recall metrics for all classifier models. (T) and (B) indicate that the metric was computed with target or bystander respectively as the positive class.

| Learning Algorithm | Validation Accuracy | Precision (T) | Recall (T) | F1 Score (T) | Precision (B) | Recall (B) | F1 Score (B) |
|--------------------|---------------------|---------------|------------|--------------|---------------|------------|--------------|
| MLP | 94.34% | 0.98 | 0.88 | 0.93 | 0.90 | 0.98 | 0.94 |
| SVM | 90.57% | 0.94 | 0.87 | 0.90 | 0.87 | 0.94 | 0.90 |
| RF | 92.45% | 0.93 | 0.95 | 0.94 | 0.94 | 0.92 | 0.93 |
| GDBT | 94.34% | 0.83 | 1.00 | 0.90 | 1.00 | 0.88 | 0.94 |

Table 2: Validation accuracy for models trained using feature-subsets.

| Learning Algorithm | No Gaze Deviation | No Face Size | No Center Deviation | No Head Pose |
|--------------------|-------------------|--------------|---------------------|--------------|
| MLP | 90.57% | 77.36% | 81.13% | 85.71% |
| SVM | 92.45% | 81.13% | 84.91% | 86.68% |
| RF | 92.45% | 81.13% | 77.36% | 83.81% |
| GDBT | 94.34% | 84.91% | 83.02% | 84.76% |

Table 3: Average single prediction forward-pass runtime (Intel i9-10900k)

| Learning Algorithm | Average Runtime (ms) |
|--------------------|----------------------|
| MLP | 0.0743 |
| SVM | 0.0541 |
| RF | 0.0515 |
| GDBT | 0.2028 |
| CNN | 244.3 |

2.6.3 – CNN-Based Bystander Classification

In order to train the CNN, the same image set was used as for the feature-based model with a similar 80/20 train/test split. However, extracted local face images were used rather than entire photos. As mentioned previously, the goal of analyzing this network is to see if a privacy-concerned model could function well without needing to process entire images. Actual training took place utilizing mini-batch gradient descent with a batch size of 24 samples and 17 steps per epoch. Through experimentation, it was found that the CNN training accuracy generally converged after 15 training epochs. The testing accuracy was 81.55% with target precision and recall of 82.69% and 81.13% respectively which is significantly lower than the best feature-based models. Additionally, the predictive runtime of the model was found to be much higher than the feature-based models at around 244ms on an Intel i9 platform see (Table 3 for detailed runtime comparisons). However, these metrics are still impressive considering the loss of contextual information about a photo that the CNN experiences in comparison with the feature sets.

Originally, it appeared as though the networks would quickly converge during training due to the relatively stable loss value that was reported for the first 10 training epochs. However, by doubling the training epochs, it was found that true loss convergence did not generally happen until further training occurred. Through multiple experiments, the number of epochs that resulted in the lowest loss without overfitting and harming test accuracy occurred with 15 as mentioned previously. An example of a complete training sequence is provided in Fig. 6 with loss plotted. Additionally, Figure 7 shows the accuracy of the CNN as mini-batches progress during training. Taken in conjunction with Figure 6, the CNN appears to only converge after the 220th mini-batch where loss and accuracy variance is lowest.

It is interesting to note that just three epochs were enough to provide significant improvements in loss and training accuracy, but, in experiments, the model was not able to effectively generalize to the test set after such short training periods. It is also worth noting that with a training accuracy approaching 89% and test accuracy reaching 81.55%, overfitting still occurs in the model even with a reasonable dropout rate of 0.25.

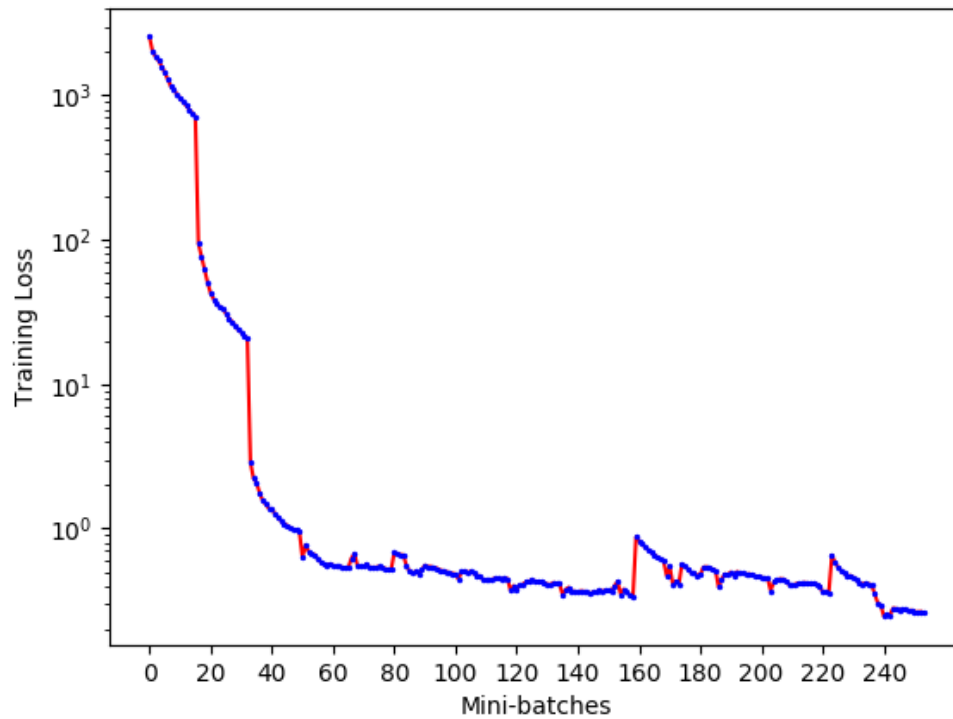


Figure 6: Progressive loss of the network for each mini-batch.

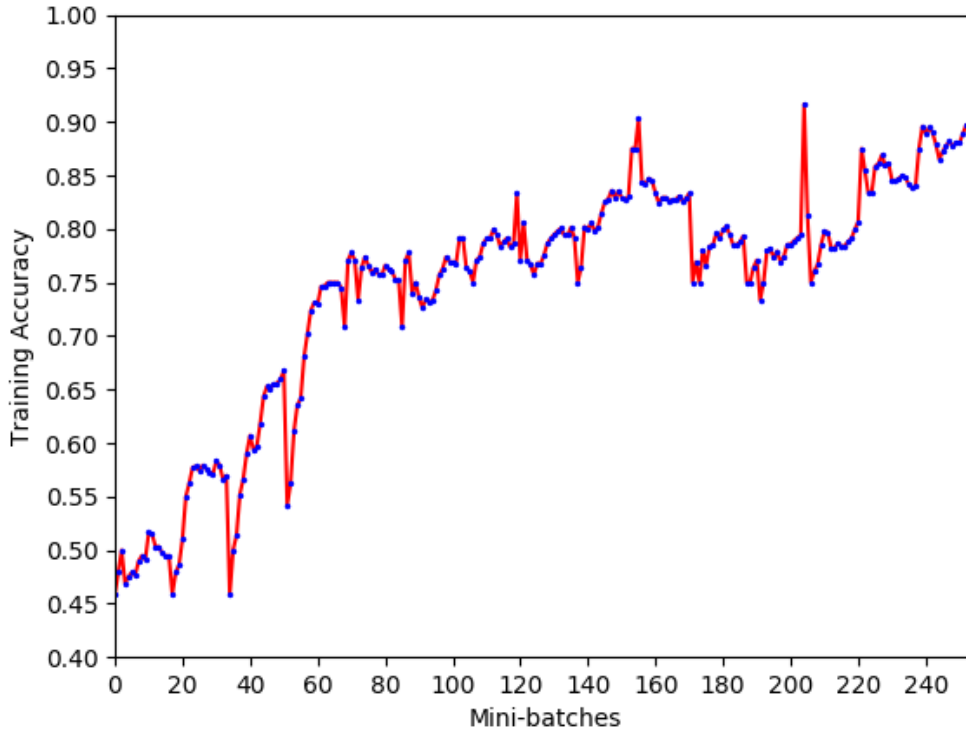


Figure 7: Progressive accuracy over training mini-batches for the CNN classifier.

The main contribution of this model is the fact that, unlike the feature-based approach which will require users to supply entire images to compute contextual features such as relative face size, this model is still able to effectively distinguish bystanders and targets with only face images, as it convolves over a “cropped” facial region. The hope is that users who might not want to supply entire photos to an automated system especially when the system is hosted in the cloud can still make use of a less invasive model which will not see any exposing information at the cost of a significant, flat decrease in performance in terms of accuracy, precision, and recall.

2.7 – Anonymizing Bystander Faces

For a complete system to ensure the privacy of strangers captured in images, it is necessary to automate the obfuscation of faces so that the ease of identifying someone is severely limited. Although there are some methods available to obfuscate faces, it is still unclear how

acceptable they are to users. In this section, we implement three different methods for facial anonymization and explore them with a user study. Fig. 8 provides a visualization of each method on an example photo from the training dataset.



Figure 8: From left to right: original image, image with black boxing, image with blurring, image with face swapping. Image source: <https://cdn.ebaumsworld.com/2008/08/861915/phelps01.jpg>

2.7.1 – Implementation of Obfuscation Methods

Black Boxing

Black boxing of the face is the simplest, and arguably most secure method for facial anonymization. The detected stranger face is completely removed from the image with every pixel RGB value being set to black. In this way, there is not any remaining information which can be gleaned from the face, such as race or face size which the other proposed methods can leave behind. Although this provides the strongest guarantee of privacy, the visual impact to a photo can be very harmful depending on the surrounding lighting conditions.

Gaussian Blurring

Blurring is considered an intermediary between the black boxing and face swapping methods in terms of intended impact to photo quality. Blurring on smaller faces can be relatively unnoticeable in images, especially on persons captured in the far background of an image.

Unlike black boxing, information such as race and even hair color can still be preserved depending on the resolution of the photo. However, facial features are always guaranteed to be completely anonymized. Gaussian blurring is used for this system due to its popularity and ease of computation. A kernel size of (70,70) is used and achieves acceptable blurring.

Face Swapping

Pose-tolerant face swapping traditionally required the use of deep CNNs such as the deepfake project which requires specific training for the two faces attempting to be swapped. However, recent advances in automated swapping, specifically the introduction of position map regression networks, have allowed for excellent generalized swapping of 3D face masks without any need for targeted training. Using an implementation of this method introduced by Feng et al. [2] allows for a novel technique of anonymizing bystander faces with any selection of “public” faces. These public faces could be commonly known celebrities or even artificially generated portraits. Assuming the face is realistic enough and lighting differences are not too extreme, the results can be very believable. For this project, faces taken from stock photos found online were used. To match skin tone for each detected stranger face, it is possible to compute an average pixel color value utilizing facial landmarks within the face region to gain a representation of their overall face color. This color is then compared with the precomputed averages of a collection of public faces. The public face which minimizes the difference is selected for swapping.

2.7.2 – Survey of Users on Face Anonymization

In order to validate our obfuscation methods and demonstrate that they are both effective at anonymizing faces and cause limited impact to user photos, we conduct a comprehensive user survey.

Questions

To gain a better understanding of how actual users regard the protection of privacy for strangers and themselves in photos, one portion of the survey asked participants for their opinions on a series of questions relating to digital photo privacy and stranger protection. These questions were presented as:

- Question 1: Ensuring the privacy of digital photos is important.
- Question 2: You would want your privacy protected if someone took a photo that captures you without you knowing.
- Question 3: It is a good idea to protect strangers' privacy in your photos.
- Question 4: It is sometimes hard to avoid including strangers' faces in photos taken in public places.
- Question 5: If there was an option to protect a stranger's face in your photo without affecting quality, you would use it.
- Question 6: If there was an option to protect a stranger's face in your photo while slightly degrading the quality, you would still use it.
- Question 7: If there was an option to protect a stranger's face in your photo while significantly degrading the quality, you would still use it.

Users were asked to rate their opinions to these questions on a Likert scale.

Another portion of the survey presented an unaltered photo compared with anonymized photos where strangers' faces had been obfuscated using each of the three proposed methods. Participants were asked to rate their opinion on how harmful each method was to the original photo on a sliding scale from 0 to 10, with 0 in this case being not harmful and 10 being

extremely harmful. In addition to rating the impact of each method on the photos, participants were also asked to rate their willingness to use each method on their own photos from 0 to 10. At the end of the section, participants could also optionally respond with testimonial as to which method they preferred and why.

In another portion of the survey, participants were presented with timed views of images. One image had strangers anonymized with face swapping and the other did not. The participants had to guess if any face swapping occurred or not in each of the two. This portion of the survey was intended to examine how noticeable face swapping could be in a fast browsing environment such as social media where average users generally only spend a few seconds looking at pictures before moving on.

Responses

In total, the survey received 89 anonymous responses over the course of 1 month. Participants were primarily recruited among university students mainly majoring in computer science and computer engineering although a minority of respondents were working adults. All respondents participated on a completely voluntary basis (no incentives were provided). Exact demographic information was not collected to preserve participant anonymity. Figure 9 shows detailed results of participant responses to survey questions 1-7.

In reviewing the responses to questions 1-7, it is clear that respondents had strong feelings in support of digital photo privacy. Most respondents likewise felt that both their privacy and stranger's privacy should ideally be protected in public photographs. Additionally, a large majority of respondents answered positively that they would make use of an anonymization system, assuming the impact to the photo was negligible while a majority responded negatively to any sort of significant impact to photo quality. Clearly, finding obfuscating methods with as

little impact as possible to photo quality is paramount in designing a system that would be well-regarded and actually used.

The results of the harmfulness ratings questions are shown in Fig. 10. The results of the willingness ratings questions are shown in Fig. 11. From these results, black boxing received the most negative feedback with a large majority of participants rating its impact the worst overall and usability the lowest. Interestingly, blurring seemed to score the highest among respondents for both usability and harmfulness. Face swapping had comparable harmfulness but was rated significantly lower on average for usability. To further examine these results, some participant responses provide helpful insight. Most participants who felt blurring was their preferred method seemed to find that face swapping was either unnatural looking or felt that they could not trust it to always create a believable swap. For example, one user responded, "The black box hurt the quality of the photo and the swapping was disturbing because you could tell it was the wrong face on the stranger.". Overall, it seemed that blurring would be the best approach from a usability and perceived impact perspective based on this participant feedback.

Analyzing the number of users who were able to detect face swapping in the final survey section demonstrates that face swapping certainly is still detectable among many users despite recent advances in realistic swapping technology. 53.9% of respondents were able to tell that face swapping was used in the photo they were presented compared with 8% detection for the control photo. Many users noticed that something was different about the photo compared with the control photo, although the detection responses were close to random guessing for the photo with swapping.

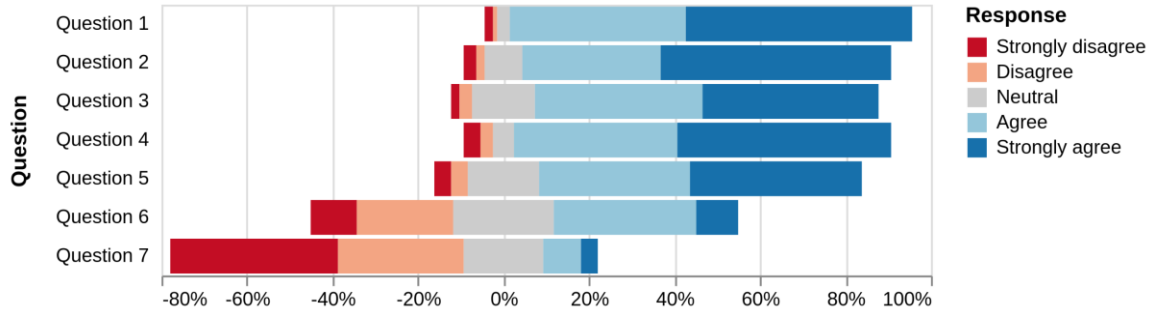


Figure 9: Survey responses to opinion questions 1-7.

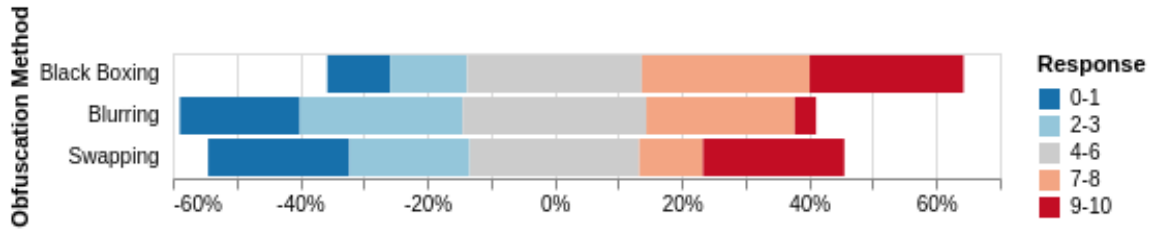


Figure 10: Survey responses to rating the impact of anonymization methods on photos.

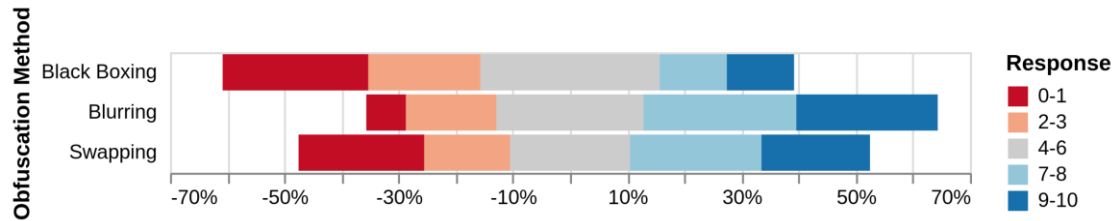


Figure 11: Survey responses to rating how willing users would be to use each anonymization method.

2.8 – Conclusion

In this work, we presented a novel approach for automating the detection and anonymization of bystanders in digital photos, applying both a feature-based model and a privacy-concerned convolutional neural network. Techniques for feature engineering were explored with methods for utilizing metrics such as relative face size and head-pose estimation. The MLP model achieved the highest validation accuracy at 94.34%, which demonstrates generalizability and promise for future use in an anonymization system for smartphone users. The convolutional neural network also demonstrated promising results with the highest achieved

accuracy of 81.55% and limited overfitting of the training set. This work is the first of its kind in being able to offer a fully privacy-concerned approach as all other works previously relied upon contextual information within a full image. Additionally, we eliminate the need for any sort of manual cooperation between photographer and bystanders as most other related works require. The hope is that being able to offer a system that automates the protection of individuals in mobile photos and preserves the privacy of those captured, the user trust and willingness to use such a method in a real-world system is greatly enhanced over any previous methods which all require some form of participation on the bystanders' parts.

Three fully automated approaches for face anonymization (black boxing, blurring, and face swapping) were presented for use with the classifying models. To better understand user opinions around public photo privacy and each of the presented methods, a comprehensive user study was carried out. Participant responses indicated that while privacy of photos and individuals in public settings was definitely a concern for most, developing anonymizing methods which do not harm photo quality is important for creating any sort of real-world system. Especially promising were the large number of positive responses on the blurring and swapping methods which indicate that the system has attraction to real-world users.

References

- [1] Darling, D., Li, A., Li, Q.: Feature-Based Model for Automated Identification of Subjects and Bystanders in Photos. In: IEEE International Workshop on the Security, Privacy, and Digital Forensics of Mobile Systems and Networks (MobiSec)(2019)
- [2] Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X.: Joint 3d face reconstruction and dense alignment with position map regression network. In: ECCV (2018)
- [3] Li, A., Du, W., Li, Q.: Politecamera: Respecting strangers' privacy in mobile photographing. In: 2018 International Conference on Security and Privacy in Communication Networks (SecureComm) (2018)
- [4] Li, A., Li, Q., Gao, W.: Privacycamera: Cooperative privacy-aware photographing with mobile phones. In: IEEE International Conference on Sensing, Communication, and Networking (SECON). pp. 1–9 (2016)
- [5] Jung, J., Philipose, M.: Courteous glass. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication. p. 1307–1312. UbiComp '14 Adjunct, Association for Computing Machinery, New York, NY, USA (2014).
<https://doi.org/10.1145/2638728.2641711>,<https://doi.org/10.1145/2638728.2641711>
- [6] Raval, N., Srivastava, A., Lebeck, K., Cox, L., Machanavajjhala, A.: Markit: Privacy markers for protecting visual secrets. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication. p. 1289–1295. UbiComp '14 Adjunct, Association for Computing Machinery, New York, NY, USA (2014).
<https://doi.org/10.1145/2638728.2641707>,<https://doi.org/10.1145/2638728.2641707>
- [7] Schiff, J., Meingast, M., Mulligan, D.K., Sastry, S., Goldberg, K.: Respectful cameras: detecting visual markers in real-time to address privacy concerns. In: 2007IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 971–978 (Oct 2007).
<https://doi.org/10.1109/IROS.2007.4399122>
- [8] Bo, C., Shen, G., Liu, J., Li, X.Y., Zhang, Y., Zhao, F.: Privacy.tag: Privacy concern expressed and respected. In: Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems. p. 163–176. SenSys '14, Association for Computing Machinery, New York, NY, USA (2014).
<https://doi.org/10.1145/2668332.2668339>,<https://doi.org/10.1145/2668332.2668339>
- [9] Ilia, P., Polakis, I., Athanasopoulos, E., Maggi, F., Ioannidis, S.: Face/off: Preventing privacy leakage from photos in social networks. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. p. 781–792. CCS '15, Association for Computing Machinery, New York, NY, USA (2015).
<https://doi.org/10.1145/2810103.2813603>,<https://doi.org/10.1145/2810103.2813603>

- [10] Li, F., Sun, Z., Li, A., Niu, B., Li, H., Cao, G.: Hideme: Privacy-preserving photo sharing on social networks. In: IEEE INFOCOM 2019 -IEEE Conference on Computer Communications. pp. 154–162 (April 2019). <https://doi.org/10.1109/INFOCOM.2019.8737466>
- [11] Xu, K., Guo, Y., Guo, L., Fang, Y., Li, X.: My privacy my decision: Control of photo sharing on online social networks. *IEEE Transactions on Dependable and Secure Computing* 14(2), 199–210 (March 2017). <https://doi.org/10.1109/TDSC.2015.2443795>
- [12] Li, A., Darling, D., Li, Q.: Photosafer: Content-based and context-aware private photo protection for smartphones. In: IEEE Symposium on Privacy-Aware Computing (PAC). pp. 10–18 (2018)
- [13] Hasan, R., Crandall, D., Fritz, M., Kapadia, A.: Automatically detecting by-standers in photos to reduce privacy risks. In: IEEE Symposium on Security and Privacy (S&P) (May 2020), <https://publications.cispa.saarland/3051>
- [14] Darling, D. (2018). Exploring Photo Privacy Protection on Smartphones. *Computer Science and Computer Engineering Undergraduate Honors Theses* Retrieved from <https://scholarworks.uark.edu/csceuht/62>
- [15] Redmon, J., Farhadi, A.: Yolov3: An incremental improvement (2018)
- [16] Zadeh, A., Baltrusaitis, T., Morency, L.: Convolutional experts constrained local model for facial landmark detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 2051–2059 (2017)
- [17] Baltrusaitis, T., Robinson, P., Morency, L.: Constrained local neural fields for robust facial landmark detection in the wild. In: 2013 IEEE International Conference on Computer Vision Workshops. pp. 354–361 (Dec 2013). <https://doi.org/10.1109/ICCVW.2013.54>
- [18] Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N.: Localizing parts of faces using a consensus of exemplars. *IEEE Transactions on Pattern Analysis Machine Intelligence* 35(12), 2930–2940 (Dec 2013). <https://doi.org/10.1109/TPAMI.2013.23>
- [19] Le, V., Brandt, J., Lin, Z., Bourdev, L., Huang, T.S.: Interactive facial feature localization. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C.(eds.) *Computer Vision – ECCV 2012*. pp. 679–692. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
- [20] King, D.E.: Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research* 10, 1755–1758 (2009)

- [21] Pech-Pacheco, J.L., Cristobal, G., Chamorro-Martinez, J., Fernandez-Valdivia, J.: Diatom autofocus in brightfield microscopy: a comparative study. In: Proceedings 15th International Conference on Pattern Recognition. ICPR-2000. vol. 3, pp.314–317 vol.3 (Sep 2000). <https://doi.org/10.1109/ICPR.2000.903548>
- [22] Baltrusaitis, T., Zadeh, A., Lim, Y.C., Morency, L.: Openface 2.0: Facial behavior analysis toolkit. In: 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018). pp. 59–66 (May 2018). <https://doi.org/10.1109/FG.2018.00019>
- [23] Wood, E., Baltruaitis, T., Zhang, X., Sugano, Y., Robinson, P., Bulling, A.: Rendering of eyes for eye-shape registration and gaze estimation. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 3756–3764 (Dec 2015). <https://doi.org/10.1109/ICCV.2015.428>
- [24] Darling, D.: Target bystander detection repository. <https://github.com/ddarling/target-bystander-detection> (2020)
- [25] Friedman, J.H.: Stochastic gradient boosting. Computational Statistics and Data Analysis 38(4),367–378(2002). [https://doi.org/https://doi.org/10.1016/S0167-9473\(01\)000652](https://doi.org/https://doi.org/10.1016/S0167-9473(01)000652), <http://www.sciencedirect.com/science/article/pii/S0167947301000652>, nonlinear Methods and Data Mining
- [26] Breiman, L.: Randomforests. Machine Learning 45(1), 532 (Oct 2001). <https://doi.org/10.1023/A:1010933404324>,<https://doi.org/10.1023/A:1010933404324>
- [27] Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their Applications 13(4), 18–28 (July 1998). <https://doi.org/10.1109/5254.708428>

Appendix

A – IRB Approval



To: David Webster Darling
From: Douglas James Adams, Chair
IRB Committee
Date: 09/09/2019
Action: **Exemption Granted**
Action Date: 09/09/2019
Protocol #: 1907204966
Study Title: Survey of User Opinions on Digital Photo Obfuscation Techniques to Protect Stranger Privacy

The above-referenced protocol has been determined to be exempt.

If you wish to make any modifications in the approved protocol that may affect the level of risk to your participants, you must seek approval prior to implementing those changes. All modifications must provide sufficient detail to assess the impact of the change.

If you have any questions or need any assistance from the IRB, please contact the IRB Coordinator at 109 MLKG Building, 5-2208, or irb@uark.edu.

cc: Qinghua Li, Investigator

3 – Sensor-Based Detection and Dynamic Mitigation of Visual Eavesdropping on Mobile Devices and Web Browsers

3.1 – Introduction

To address the growing problem of visual eavesdropping on smartphone screens in public, we propose the first, automated scheme for easily detecting visual eavesdroppers around a user's phone before a user accesses and potentially leaks sensitive app content. Specifically, this work explores a sensor-based approach utilizing the integrated, front-facing cameras which come standard on most modern smartphones. A state-of-the-art mobile face detection model is coupled with a magnetometer reading threshold to offer fast detection of eavesdroppers. Based on attempted eavesdropping trials, this solution is capable of accurately detecting surrounding eavesdroppers before a user checks their desired app.

Beyond detection, this work also explores the mitigation of visual eavesdropping to allow users to still access their desired apps or websites while protecting any sensitive content. We first propose modifying existing applications to utilize user-interface-as-an-image (UIaai) to enable various image processing and computer vision techniques to be applied to live app UIs. A YOLO style deep convolutional neural network (DCNN) is used to automatically detect sensitive content such as text messages in apps. Additionally, a method for dynamically retraining the model on user-specified content is presented to ensure real-world usability. Finally, for web browser environments, we propose a novel HTML/CSS injection scheme to automatically introduce blurring for sensitive content on web pages. This scheme is able to run over any web page without modification by the developers by running in a Google Chrome browser extension.

Users can utilize this complete scheme in public locations by immediately scanning their surroundings without leaving their desired app. In especially crowded scenarios, attackers

constantly walking behind or around the user that might avoid the scanning detection mode can be avoided with a manual activation of content hiding. However, the scanning mode should be sufficient in most public scenarios. Content hiding is then performed for the user while still allowing them to use their desired app. When a user feels they are in a safe location with no chance of shoulder surfing occurring, the content protection can be switched off manually at any time.

The remainder of this chapter is organized as follows. Chapter 3.2 provides an overview of related works. Chapter 3.3 describes the visual eavesdrop attack scenario in greater detail. Chapter 3.4 presents and details the sensor-based eavesdrop detection scheme. Chapter 3.5 describes the concept of UIaI and the methods for automated detection and hiding of sensitive content. Chapter 3.6 describes the design and effectiveness of the web-based content hiding system. Chapter 3.7 presents conclusions.

3.2 – Related Work

The majority of work relating to shoulder surfing attacks have traditionally focused on protecting specific attack vectors such as phone passcode input. Kumar et al. [1] present a gaze-based password entry method for preventing shoulder surfing attacks by utilizing a user's gaze rather than their hands to enter passcode digits. Chakraborty and Mondal [2] present a honeypot-based scheme whereby if a shoulder surfer attempts to enter a credential containing a tag digit they would be detected. Yu et al. [3] propose an evolvable password protection scheme using images as keys rather than digit-based authentication. Zhang et al. [4] presented a similar evolving password scheme using augmented reality displays to present an input field that is only visible to a specific user. Sun et al. [5] propose a graphical authentication scheme resistant to multiple camera screen spying attacks for passcode entry. This work attempts to prevent attacks

not only over specific scenarios but in the general case on any number of apps or services with a universal protection scheme.

Some works focus on detecting or preventing shoulder surfing attacks on specific non-mobile platforms. Watanabe et al. [6] developed a dummy cursor system for desktop or laptop platforms to hide the true cursor from attackers' views. Li et al. [7] present a shoulder surfing detection scheme for ATMs by tracking human bodies and faces. This work primarily focuses on mobile and web platforms due to their ubiquitous and uniquely susceptible nature.

Other works of interest focus specifically on the detection and alerting of shoulder surfing attacks for mobile devices. Saad et al. [8] study several different methods for actually alerting mobile users such as through vibration or visual indication. Brudy et al. [9] explore various methods users can employ to be more aware and protective of their mobile displays. Lian et al. [10] propose a system for tracking eyes of individuals looking at mobile screens and dimming the screen in response to eavesdrop detection. This work does not focus on methods for alerting users, rather content hiding is performed automatically. Additionally, our work does not rely on any specific facial attribute such as eyes to detect eavesdropping.

Beyond novel detection methods, our work is also the first to offer private content hiding with little modification for mobile apps in the general case. Existing approaches for content hiding require specific and time-consuming code changes for published applications. Our approach offers automated content detection and hiding with minimal code changes for existing apps by utilizing UIaAI. Detected private content can be automatically obfuscated with blurring or other hiding techniques whereas many existing methods require manual code changes specific to the type of private content such as password entry schemes.

3.3 – Mobile Attack Scenario

In this work, we consider an attack scenario as any occasion where a mobile device user is located in a public space with other individuals. In this environment, the attacking party could be any person near enough to the user to see the phone screen, but out of the user's field of view. This is because it is assumed that attackers within the vision of the user could be easily detected by a defensive user. Attackers then are ideally located to the sides or behind the user where the device screen is visible and not obstructed by the user's body. This work aims to detect these cases where the user is uniquely vulnerable to being spied on. Fig. 12 provides a visual representation of a possible attack scenario. The attackers are featured behind and to the sides of the user in the blind spots highlighted in red.

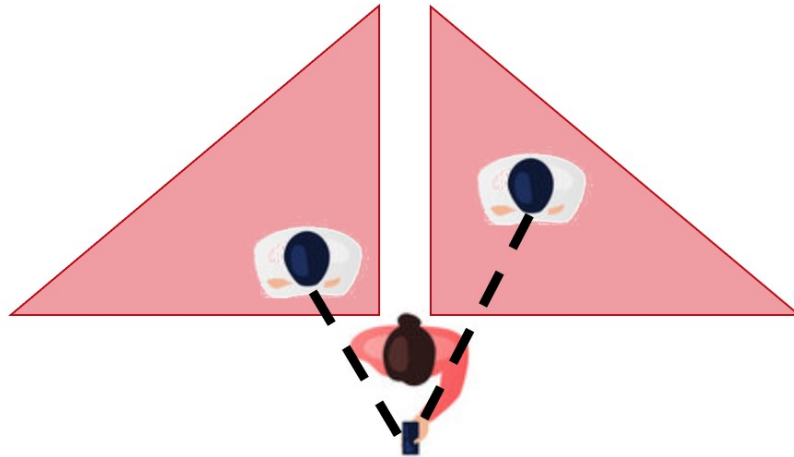


Figure 12: Visualization of a potential attack scenario. The vulnerable “danger areas” relative to the user are shown in red with attacker gazes shown as black dashed lines.

3.4 – Sensor-Based Eavesdropping Detection Scheme

3.4.1 – Overview

In order to be effective, an eavesdropping detection system must be able to locate attackers in the user's vulnerable blind spots where they cannot easily see. Additionally, the detection system should ideally be unobtrusive to users and convenient to use. To satisfy these requirements, we develop a scanning-based solution implemented as a mobile app which can be downloaded onto user phones. The scanning detection works by having a user quickly normalize the phone, and then, utilizing the front facing device camera, scanning the camera left and right. A magnetometer based binary thresholding method is used to eliminate the user's own appearance in any captured image frames. Captured images are passed through a BlazeFace [11] based facial detection network. If faces are captured after the thresholding process, there are potential attackers in the user's vulnerable fields who could view content on their phone screen.

The complete detection pipeline is outlined as follows:

1. The user phone is normalized by being oriented with the front camera facing the user face with the user holding it.
2. Attacker scanning is begun by the user holding a button in the mobile app. The user quickly scans the vulnerable areas around them with the front camera.
3. Magnetometer binary thresholding is used to eliminate images which contain the user's own face.
4. Remaining image frames are passed through the facial detection network.
5. Sensitive content on a user phone is obfuscated if potential attacker faces are identified.

3.4.2 – Magnetometer Normalizing and Thresholding

Implementation

Most smartphones are equipped with a magnetometer sensor for reading information about local magnetic fields as well as the Earth's magnetic fields for general positional information. We are able to utilize these types of sensors to compute the user's current orientation relative to magnetic north or, in other words, a compass heading for the user. The magnetic heading information can then be used to filter out images where the front facing camera has not been turned far enough to avoid capturing the user's own face.

Accessing the magnetic information on a smartphone differs across device platforms. iOS offers access to magnetometer sensor data through a *CLLocationManager* object. The object is able to deliver heading updates through the *locationManager(_:didUpdateHeading:)* delegate function. This allows real time updates for magnetic heading information. Android offers access to device magnetometers through use of a *SensorManager* object. The *getRotationMatrix(rotationMatrix, null, accelerometerReading, magnetometerReading)* function call in conjunction with *getOrientation(rotationMatrix, orientationAngles)* call can be used to compute accurate heading information independent of device rotation using both the device accelerometer and magnetometer. Either of these APIs enables accurate device orientation information.

Use in Attacker Detection

With magnetic orientation collected, it is possible to store device orientation, and monitor as the user rotates their device to scan around. During the attacker detection phase, the user is presented with a UI button element to first normalize their device orientation. The user must point their device with the front camera pointing directly at their face while holding the device in

front of them. After activating the button, the application stores the current heading information and begins scanning mode. During scanning, the phone's current heading value is first checked. If the x-axis heading degree is $\pm 15^\circ$, no image frame is captured. This threshold generally prevents any image of the user's face from being captured but can be tuned to allow for users who hold their phones closer or further from their face. Under this scheme, image frames which are captured and passed to the face detection network can be assumed to be free of the user's face.

3.4.3 – Facial Detection Network

In order to rapidly and accurately detect attackers around the user, a state-of-the-art facial detection network is employed. We utilize a convolutional neural network based around the BlazeFace architecture optimized specifically for mobile GPUs. The network architecture shown in Fig. 13 utilizes increasingly shrinking sizes of convolutional "blocks" which efficiently shrink the feature space to generate predictive bounding box information. The network uses a combination of single convolutional blocks with double convolutional blocks. The single blocks simply consist of a 5x5 kernel convolution, a 1x1 convolution, and a max pooling layer followed by an activation function. The double blocks consist of two of these single blocks, but with only one max pooling layer. In other words, max pooling is only performed once for each type of block.

This style of convolutional neural network is highly desirable for the domain of attacker detection due to its relatively fast inference time on mobile processors and GPUs. The model is capable of achieving 0.6 ms inference time on iPhone XS platforms which corresponds to faster than real time performance in forward pass detection. Tested against MobileNetV2 [12], the network offers precision within 1% of MobileNet, but with significantly faster performance (0.6

ms compared with 2.1 ms). Both bounding box information and facial anchor points are generated from the feature extraction network. Fig. 14 shows example output from the network visualized over a sample image.

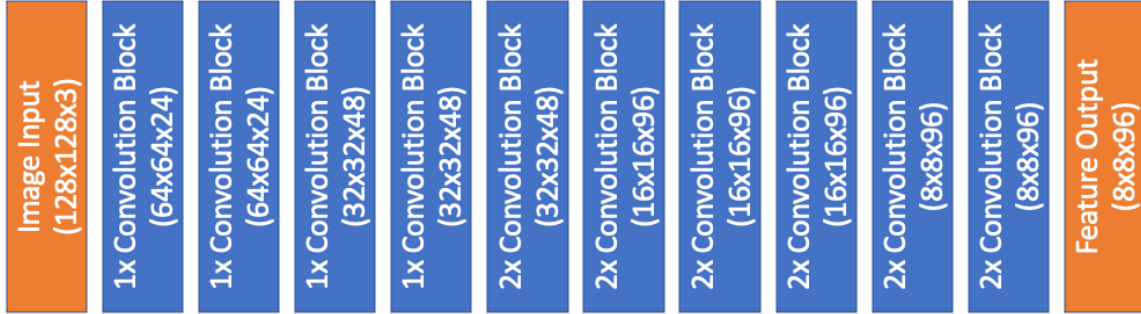


Figure 13: Architecture of feature extraction portion of facial detection network

This implemented network is capable of real time detection of attacker faces on most modern iOS and Android devices within front-facing camera image frames captured during the scanning mode of operation. The network is robust against pose deviation in faces as well as distance. Taken together with the magnetometer-based thresholding technique presented in the previous section, the complete detection scheme is fully capable of robust attacker detection. Due to the lack of forward-pass prediction delay, a user attempting to quickly scan their surroundings in a discrete manner can be alerted immediately if a potential attacker is located behind or around them in their blind spots.

Attacker detection results from the detection network are utilized to automatically censor content in apps to protect users until the scanning mode shows that there are no further attackers nearby. In the following section, we present a novel technique for allowing existing mobile applications to easily implement content-hiding and for automatic detection of sensitive content.

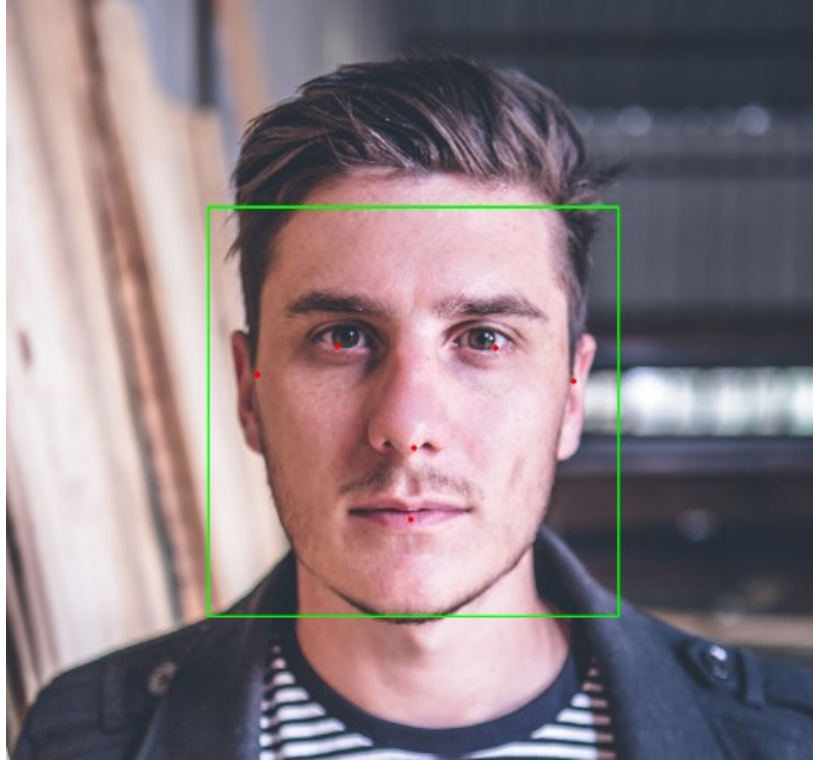


Figure 14: Example detected face with bounding box in green and facial anchor points plotted in red. Image source: <https://unsplash.com/photos/FGaknWvNbBk>

3.5 – Automated Private Content Detection and Hiding

3.5.1 – User Interface as an Image (UIaI)

Overview

With modern mobile applications, there are often many complex layers of UI code to dynamically generate and present content to users in an accessible manner. Because of this it may be difficult for many mobile app developers to add additional functionality for hiding private content dynamically. This could present an enormous commitment of time for large mobile apps with complex and nested user interface elements where it may be difficult to identify which content is currently visible and needs to be hidden from visual eavesdroppers.

To help mitigate this issue and provide a standardized methodology for dynamically hiding sensitive content on complex applications, we present UI as an Image. The basic idea of

this concept is that computer vision techniques such as object detection and image filtering are highly applicable to the domain of private content detection and could replace tedious and time-consuming code conversion of existing applications. Under UIaaI, dynamically generated UI views are presented to the user entirely as a pre-rendered, interactive image rather than as direct views of the app UI. This approach offers several benefits, namely the technique for converting applications to utilize UIaaI is standardized for all applications no matter the underlying UI complexity, computer vision techniques can be applied directly to existing views of UI without the need to directly program any content-hiding logic, and complex private content can be automatically identified. Although utilizing images as displayed UI elements eliminates some interactivity, UIaaI can be used temporarily for situations where users are in public spaces. Apps can easily be switched back to their default behavior once a user no longer has to worry about eavesdropping.

Conversion Pipeline

In order to make UIaaI as universally adoptable as possible, the pipeline for providing conversion in apps is simple. An application on iOS or Android that needs to implement dynamic content protection can follow these general guidelines:

1. Implement a switchable mode which a user can activate via a button or other UI element.
2. Once a user has switched to UIaaI mode, the application should pre-render its current UI view as a single image frame.
3. The pre-rendered UI view is displayed to the user on-screen.
4. A pass of content detection is performed over the image to identify any content that should be hidden.

5. A Gaussian blurring filter is utilized to dynamically hide any sensitive content.
6. Once a user has switched back to normal mode, the regular app display logic can be used again.

Utilizing this general guideline, even highly complex app UIs will not require complicated UI logic to enable dynamic hiding of sensitive content. The goal of this is to enable developers to easily make their applications shoulder surfing resistant and encourage adoption of content hiding to benefit end users. Additionally, being able to operate on pre-rendered images enables a variety of interesting computer vision techniques to be applied over the UI such as content swapping to hide content from view with dummy content. Most importantly, we propose that pre-rendered UI can be used to automatically detect private content with an object detection network.

The drawback to this design is a slight loss in user-interactivity. This is due to the fact that image-based views of complex UIs cannot directly emulate all of the visual actions that a traditional UI view is able to generate. For example, although a user's touch events and coordinates of the touch can be recorded and translated into the activation of a button or other element, the normal visualization of a button being depressed would require an additional series of pre-rendered views to be generated. Despite this, we maintain that for suitably large applications, the ease of enabling private content hiding through UIaaS outweighs this temporary lack of interactability.

3.5.2 – Private Content Detection Network

Network Design

In order to automate the detection of private content, we find that object detection networks such as YOLO [13] or EfficientDet [14], which have been used to great effect in fields of object tracking or generalized object detection, are uniquely useful in visually identifying user interface elements. This is due primarily to the fact that user interface elements generally share very similar visual attributes. For example, in an application which displays emails in an inbox, each email summary view will feature the same general attributes (a subject line, an icon representing whether the item has been read or not, and a brief excerpt from the email text). These similarities in visual appearance which are extremely common among UI elements should be learnable by an object detection network.

We implement an object detection network based off the YOLOv5 [15] architecture. This type of network features high inference speed and smaller model size relative to other state of the art networks. These features make it a prime candidate for inclusion onto mobile platforms. Fig. 15 provides an overview of the implemented network architecture. The network relies on a feature extraction backbone built around cross-stage-partial (CSP) network layers [16] along with a spatial pyramid pooling (SPP) layer [17] for getting feature tensors of fixed output size irrespective of input image size. The feature aggregation or neck portion of the network is built around the path aggregation network (PANet) architecture [18] which has been used to great success in competing object detection networks. The final output layers are output from different downsampled feature spaces in the network. This is done to allow the network to identify spatially larger or smaller objects in an input image at different levels of granularity.

Resulting output vectors contain bounding box anchors, box width, box height, class prediction, and confidence interval information.

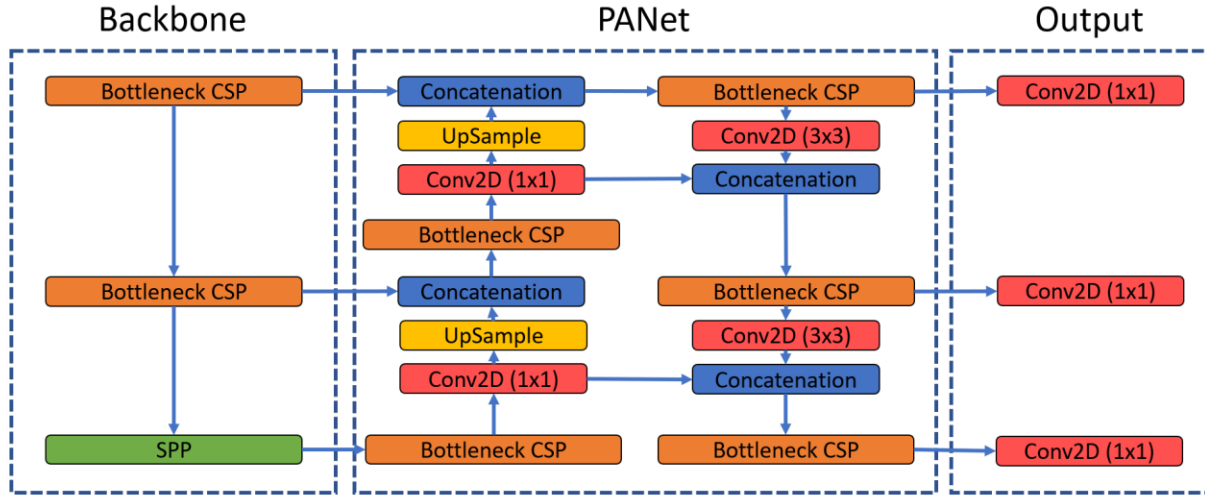


Figure 15: Architectural overview of the private content detection network.

Training for Sensitive Content Detection

In order to evaluate the content detection network's ability to detect and classify different types of common UI elements that would contain private content in the real world, a dataset for text messages in the iOS Messages app was created for training the network. Text messages were selected as the UI element of choice in this case because they feature many visual similarities to other common private elements such as email summaries or phone call notifications. Text messages also have the potential to be challenging for an object detection network due to the large variation in size across different messages.

The created dataset contains 188 rendered text message conversations. The network was trained over 1000 epochs to determine how quickly it was able to converge. The resulting model was able to achieve 98.17% mean average precision at 0.5 intersection over union (mAP@0.5). The model was able to achieve this performance after 450 epochs of training which corresponds to 14 minutes of training time on an Nvidia Tesla T4. The results for this training are visualized

in Fig. 16 along with precision and recall metrics. Sample test forward pass predictions are visualized in Fig. 17. These results generally show that the model is highly capable of learning to distinguish UI elements even in a complex layout such as text message stacks.

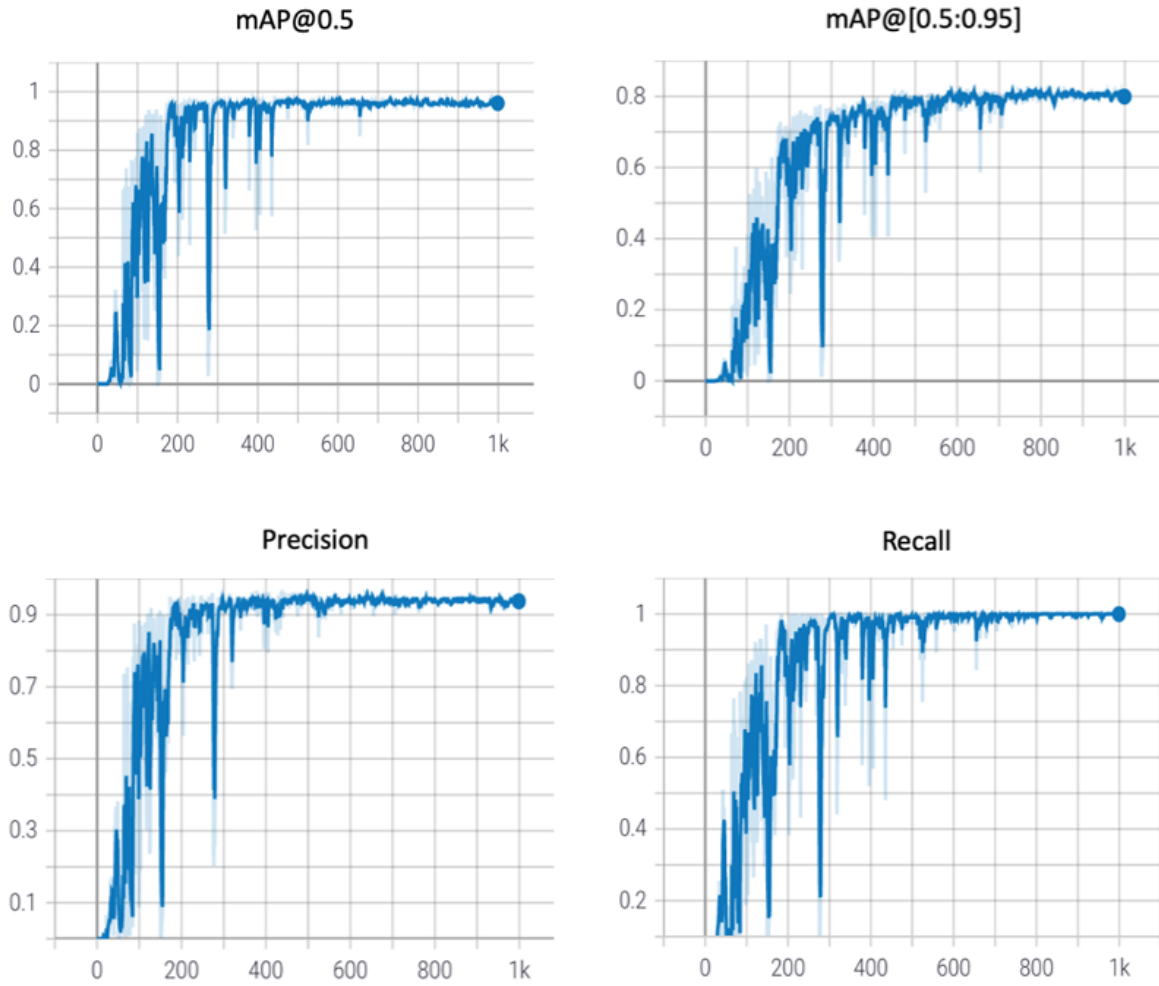


Figure 16: Bounding box mean average precision, precision, and recall metrics over progressive training epochs. Note that charts are smoothed with original data shown as a shadow behind. The model is found to converge under all metrics after 450 epochs.

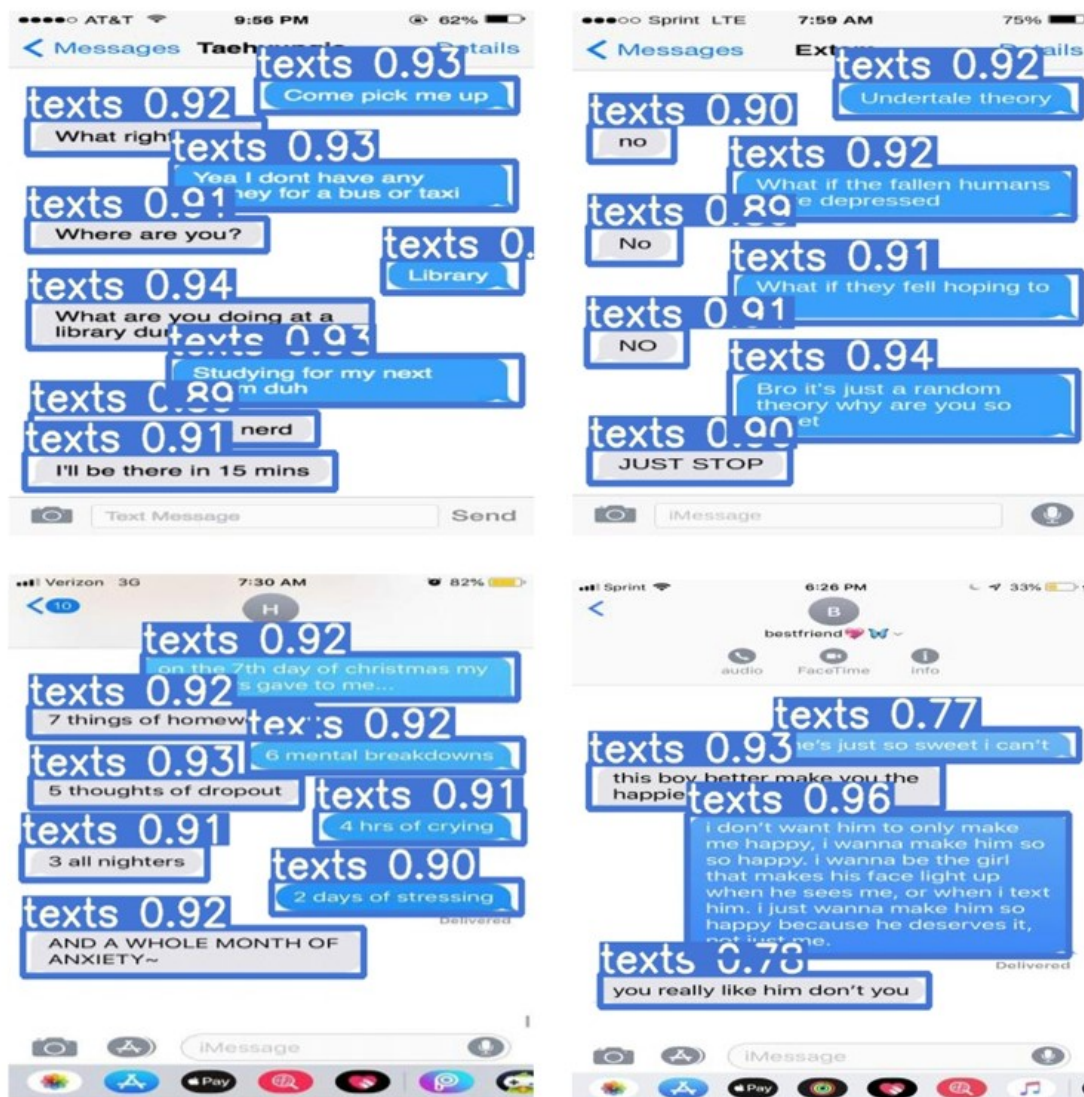


Figure 17: Sample text detection output with confidence metric. The network is capable of accurately detecting text messages of varying sizes and positions.

3.5.3 – User-Defined Content Retraining

We recognize that it is an impossible task to fully identify all forms of private content that a user might wish to be protected across many different mobile applications. Users may have a financial app which displays a summary of account balances which they would not want strangers in public to see for example. To this end, we propose a scheme for dynamic training sample generation. Under this scheme, users can manually identify a UI element they wish to automatically detect and hide within an app, and many training image samples can be generated

of the same element using visual transformations. Fig. 18 presents a visualization of all transformations/augmentations applied to sample text message images.

Flip Transformation

In order to simulate the different orientations that UI elements might take on a mobile phone, random horizontal and vertical flipping of pre-rendered views provides the model with variation in positioning that might be encountered in the wild as a user accesses their app.

Rotation Transformation

Similar to the flip transformation, the rotation transformation is proposed as a means to introduce more positional variation into a generated dataset. Rotations in 90° intervals are used, as mobile UIs almost never feature non-right-angle rotations in the wild. The rotations combined with flip transformations are found to offer a great deal of positional variety even for very small numbers of original training samples.

Random Cropping and Tiling

To simulate the fact that many complex mobile UIs stack or layer UI elements together in the same views, we utilize randomized image cropping and tiling. This method is used to combine together different views of the same elements to force the model to learn to identify elements at any region of a pre-rendered UI view. Complementary slices of training images are used to form a full-sized training sample.

Training Results

To validate the training data generation scheme, we utilize a subset of the original text message dataset with only 10 images. These images were used to generate a new dataset of 150 augmented images. The same model architecture as presented in section 3.5.2 was trained over these generated images. This model was able to nearly match the validation average precision of

the model trained on the full dataset, reaching a $mAP@0.5$ of 95.65%. The level of accuracy demonstrates that it is possible to train a robust model for UI recognition with fewer training samples than are traditionally required in object detection tasks. This is attributed primarily to the similarity in lighting conditions, visual properties, and shape of most UI views which are rarely so similar for more generalized recognition tasks.

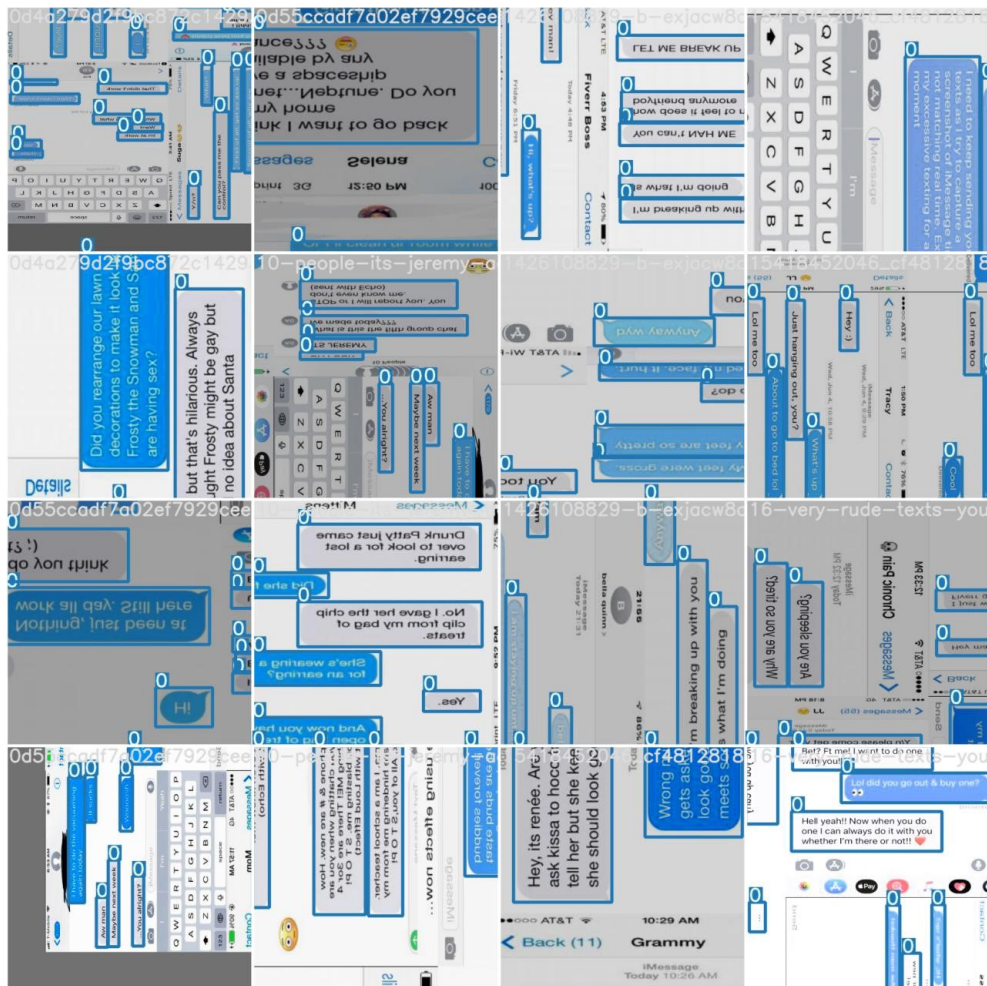


Figure 18: Visualization of random transformations and augmentations applied over a subset of the full text message dataset.

3.5.4 – Evaluation of Attacker Detection and Content Hiding Scheme

In order to evaluate the overall efficacy of our content protection scheme, we carry out a series of trials whereby an individual acting as an attacker attempts to glean some information from a user's screen. The trials are intended to mimic conditions that average users might encounter in public locations. The user is standing and holding their phone upright in front of them attempting to read a text conversation. The attacker could be located anywhere within the user blind spots identified in Fig. 12. Table 4 presents the results of these trials. The attacker was made to either stand stationary or move around behind the user. Even with both user and attacker moving, the detection network was able to identify an attacker in most scenarios. The failure cases in the trials were when the attacker was able to move very quickly behind the user to evade the camera. Because of the fast movement, the attacker appeared heavily blurred in captured images, and facial details were not defined enough for the detection network to identify a face. However, we note that because the attacker had to be moving quickly, they had difficulty actually reading information off of the user phone, so information leakage was minimal. The attacker was only ever able to glean 1-4 words from a text conversation.

Table 4: Trial outcomes for each simulated visual eavesdrop attack. Attacker position is relative to the user. Refer to Fig. 12 for positioning information.

| Trial # | Attacker Position | Attacker Movement | User Movement | Attacker Detected | Information Loss |
|---------|-------------------|-------------------|---------------|-------------------|------------------|
| 1-2 | Left | None | None | Yes | None |
| 3-4 | Left | Slow | None | Yes | None |
| 5-6 | Left | Slow | Slow | Yes | None |
| 7-8 | Left | Fast | Slow | No | Scattered words |
| 9-10 | Back left | None | None | Yes | None |
| 11-12 | Back left | Slow | None | Yes | None |
| 13-14 | Back right | None | None | Yes | None |
| 15-16 | Back right | Slow | None | Yes | None |
| 17-18 | Right | None | None | Yes | None |
| 19-20 | Right | Slow | None | Yes | None |
| 21-22 | Right | Slow | Slow | Yes | None |
| 23-24 | Right | Fast | Slow | No | Scattered words |

3.6 – Browser Extension-Based Private Content Hiding

3.6.1 – Chrome Extension Overview

Beyond mobile devices and applications, users frequently access private services via web application. Although these web services allow easy access to for users to perform actions like online banking or ordering goods from online stores, using these services in public on a laptop or other device can easily allow strangers to see personal information similar to mobile applications. To address this issue and provide content protection for as many devices as possible, we develop a browser extension for dynamic content hiding. Because Chrome is one of the most popular web browsers with large amounts of support for 3rd party extensions, we develop an extension targeting the Chrome platform to reach a large population of web users.

3.6.2 – Identifying Sensitive Content

Generalized private content detection is carried out primarily by searching for HTML patterns. Specifically, four HTML structures are used for detecting sensitive personal data.

- The HTML `<form>` elements that use POST as the Hypertext Transfer Protocol (HTTP) method. For any `<form>` that requires inputs of sensitive data such as passwords, the submitted data should not be exposed in the Uniform Resource Locator (URL). Thus, these forms must use POST as their HTTP methods.
- The HTML `<input>` elements with types of password, telephone, or email addresses. Some input fields that require sensitive personal information are not wrapped up in a form, so they need to be searched for individually.
- The HTML `<iframe>` elements. Some online payment pages enclose all input fields for private information (such as credit card number) in an `<iframe>`. Due to the same-origin policy, an iframe element cannot be accessed with different origins using JavaScript,

which makes it difficult to detect sensitive data among the internal HTML nodes.

Therefore, this extension checks whether the *allowpaymentrequest* attribute of a captured `<iframe>` is set to true. If so, the `<iframe>` must contain input fields that ask for input of credit-card-related information.

- Login buttons. Clicking these buttons changes the Document Object Model (DOM) tree. Sometimes, the change displays input fields for users to enter their usernames and passwords.

3.6.3 – Performing Dynamic Content Detection

There are two main cases where dynamic content detection is performed. The first and most common is when a page is first loaded. The content script of the Chrome extension sets the *run_at* field to *document_idle* indicating that either *window.onload* or *DOMContentLoaded* has been called. In either case, there might still be scripts running as the event is triggered. If a running script inserts any node into the DOM tree during this time, the new node will not be found when the content detection script is running and cannot be captured.

To avoid this problem, the extension can call *window.requestAnimationFrame* which takes as an argument a callback function the browser calls whenever a page repaint occurs. This implementation catches all freshly inserted nodes in the DOM tree, but page repainting occurs often when a page is loaded to set up styling. This can cause large performance impacts to a webpage.

A better approach is to invoke the *setTimeout* function. The content detection and CSS style injection logic can be directly passed as a parameter callback to this function. This process is less costly, in terms of performance, because the callback can be delayed a few milliseconds

before executing. This generally allows for enough time for all web page elements to load before attempting to detect sensitive content.

A final case where a content detection pass is performed is when any JavaScript event triggers a change in the DOM tree. This could occur when a user interacts with an element on a web page which activates a log in field. To get around these interactive elements, a list of keywords is used: "login", "log in", "signin", "sign in", "sign up", "signup", "register", "join", "create new account", and "try it free". Any interactive element containing these keywords is attached with a click event listener to the element's event list.

Finally, a manual toggle is provided to users to blur entire pages to deal with any other cases not covered by the extension.

3.6.4 – Applying CSS Styles

Once sensitive content has been detected on a web page, the extension applies Gaussian blurring effects to specific elements, so attackers will find it significantly more difficult to identify specific content like letters or numbers. The following algorithm applies CSS styles to sensitive content:

1. Apply styles to any captured HTML *<form>* elements that use POST methods. These could include forms such as payment information submissions.
2. If email input fields are the only captured elements, CSS styles are applied to these and the associated submit buttons. Buttons are identified as closest in the DOM tree rooted at the parent node of the email address input field. If no button is found, a recursive search of the parent tree is performed until one is found.
3. If the discovered content contains HTML elements other than email address input fields, the CSS styles are simply applied to the entire page. This is because the

DOM tree could contain iframes, password fields, or telephone fields. These categories of information are especially important to protect, and, in some cases, leaking the identity of the website could be risky.

Fig. 19 displays an example web page with obfuscating styling applied.

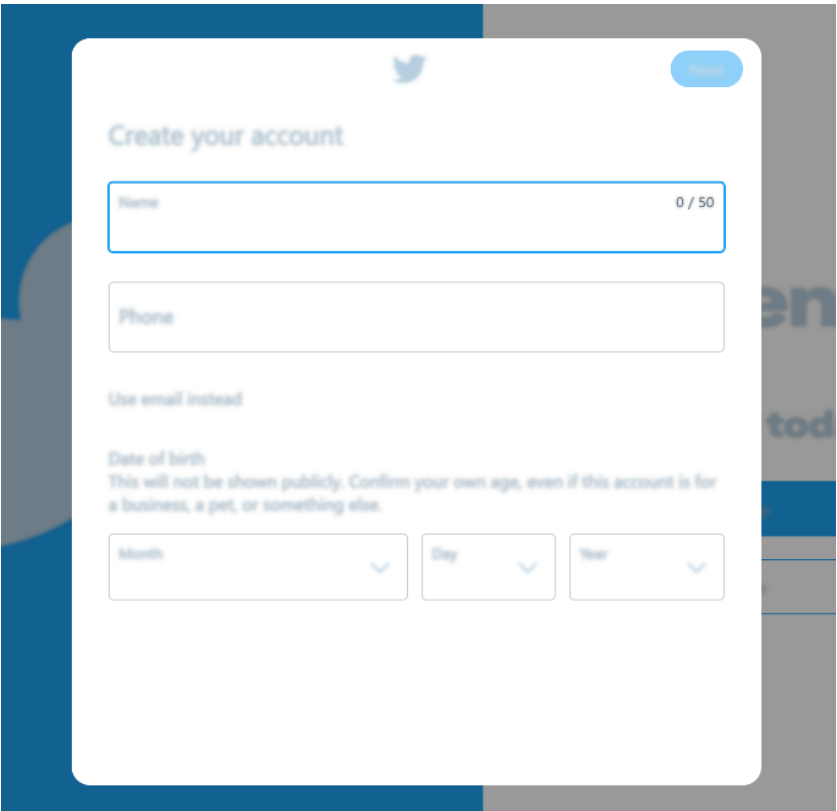


Figure 19: CSS blurring styles applied to the automatically detected Twitter sign up page.

3.6.5 – Evaluation

The content protection extension is tested over a series of highly trafficked websites. Primary testing was done on login and signup pages, as these are the most common forms that can contain private information. Table 5 contains results for testing on each web page. A pass indicates the extension correctly obfuscated private fields. The misc column indicates if any additional testing was done on the website with content specific to that page. Most websites

were successfully protected, with some failure cases. For att.com, the signup page routing is performed using JavaScript methods rather than loaded from a server. Thus, there was no detected page change by the extension. For pandora.com, the web pages fail to be detected due to a similar problem caused by JavaScript-based page loading. Pinterest and Quizlet obfuscations work correctly, but obfuscation styling is unable to be reverted due to difficulty in capturing login modal closing events. Correcting these specific issues are avenues for future work.

Table 5: Private content detection results for examined web pages based on login, sign up, and misc. categories.

| Website | Login Page | Sign Up | Misc |
|-------------------|-----------------------------|-----------------------------|----------------------------------|
| amazon.com | passed | passed | payment info: passed |
| att.com | passed | failed | payment info: passed |
| bbc.com | passed | passed | N/A |
| facebook.com | passed | passed | N/A |
| geeks4geeks.com | passed | passed | N/A |
| github.com | passed | passed | N/A |
| google.com | passed | passed | N/A |
| homedepot.com | passed | passed | email subscription field: passed |
| instagram.com | passed | passed, 3 second delay | N/A |
| linkedin.com | passed | passed | N/A |
| lowes.com | passed | passed | email subscription field: passed |
| npr.com | passed | passed | N/A |
| pandora.com | failed | failed | N/A |
| pinterest.com | passed, style cannot revert | passed, style cannot revert | N/A |
| quizlet.com | passed, style cannot revert | passed, style cannot revert | N/A |
| quora.com | passed | passed | N/A |
| reddit.com | passed | passed | N/A |
| stackoverflow.com | passed | passed | answer box: passed |
| twitter.com | passed | passed | N/A |
| udemy.com | passed | passed | N/A |
| w3schools.com | passed | passed | N/A |
| yahoo.com | passed | passed | email subscription field: passed |
| yamibuy.com | passed | passed | email subscription field: passed |
| youtube.com | passed | passed | N/A |

3.7 – Conclusion

In this work, we presented multiple solutions for protecting private content on mobile device screens from visual eavesdroppers in public spaces. A sensor-based eavesdrop detection

scheme was designed utilizing magnetometer and front-facing camera built in sensors for iOS and Android devices. The scheme was found to be effective in numerous trials with both a stationary and slow-moving attacker. In addition to attacker detection, we proposed an automated solution for detecting private UI elements through a YOLO style deep convolutional neural network which was able to achieve 98.17% mAP@0.5. To allow users to retrain the network to recognize custom identified content, we presented a series of training image transformations which allow for generation of new sample images from a small existing set. To protect content on web services, we proposed a Chrome browser extension which is capable of automatically detecting and applying blurring to web page fields that could contain private user information. We evaluated this extension on multiple high-traffic websites and found that the extension was able to perform well in most scenarios for both login, signup, and payment forms.

References

- [1] 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*, ser. SOUPS '07. New York, NY, USA: Association for Computing Machinery, 2007, p.13–19. [Online]. Available: <https://doi.org/10.1145/1280680.1280683>
- [2] N. Chakraborty and S. Mondal, “Tag digit based honeypot to detect shoulder surfing attack,” in *Security in Computing and Communications*, J. L. Mauri, S. M. Thampi, D. B. Rawat, and D. Jin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 101–110.
- [3] X. Yu, Z. Wang, Y. Li, L. Li, W. T. Zhu, and L. Song, “Evopass: Evolvable graphical password against shoulder surfing attacks,” *Computers and Security*, vol. 70, pp. 179–198, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S016740481730113X>
- [4] R. Zhang, N. Zhang, C. Du, W. Lou, Y. T. Hou, and Y. Kawamoto, “Augauth: Shoulder surfing resistant authentication for augmented reality,” in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
- [5] H. M. Sun, S. T. Chen, J. H. Yeh, and C. Y. Cheng, “A shoulder surfing resistant graphical authentication system,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 2, pp. 180–193, 2018.
- [6] K. Watanabe, F. Higuchi, M. Inami, and T. Igarashi, “Cursorcamouflage: Multiple dummy cursors as a defense against shoulder surfing,” in *SIGGRAPH Asia 2012 Emerging Technologies*, ser. SA '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 1–2. [Online]. Available: <https://doi.org/10.1145/2407707.2407713>
- [7] C. Li, M. Liang, K. Xiao, S. Fong, Q. Wang, and W. Song, “Human body and face detection based anti shoulder attack system on atm,” in *Proceedings of the International Conference on Big Data and Internet of Thing*, ser. BDIOT2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 145–148. [Online]. Available: <https://doi.org/10.1145/3175684.3175706>
- [8] A. Saad, M. Chukwu, and S. Schneegass, “Communicating shoulder surfing attacks to users,” in *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM 2018. New York, NY, USA: Association for Computing Machinery, 2018, p.147–152. [Online]. Available: <https://doi.org/10.1145/3282894.3282919>
- [9] F. Brudy, D. Ledo, S. Greenberg, and A. Butz, “Is anyone looking? mitigating shoulder surfing on public displays through awareness and protection,” in *Proceedings of The International Symposium on Pervasive Displays*, ser. PerDis '14. New York, NY, USA:

Association for Computing Machinery, 2014, p. 1–6. [Online]. Available:
<https://doi.org/10.1145/2611009.2611028>

- [10] 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.
- [11] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, “Blazeface: Sub millisecond neural face detection on mobile gpus,” 2019.
- [12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” 2019.
- [13] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” 2018.
- [14] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” 2020.
- [15] G. J. et al., “ultralytics/yolov5: v5.0 YOLOv5 P6 1280 models, AWS, Supervise.ly and YouTube integrations,” Apr. 2021. [Online]. Available:
<https://doi.org/10.5281/zenodo.4679653>
- [16] C.-Y. Wang, H. Y. M. Liao, I. H. Yeh, Y. H. Wu, P. Y. Chen, and J. W. Hsieh, “Cspnet: A new backbone that can enhance learning capability of cnn,” 2019.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *Lecture Notes in Computer Science*, p. 346–361, 2014. [Online]. Available: <http://dx.doi.org/10.1007/978-3-319-10578-923>
- [18] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” 2018.

4 – Conclusion and Future Work

This thesis addresses the growing privacy and security challenges posed by the growing proliferation of smartphones and mobile devices in public spaces. Two important privacy challenges were identified as posing a significant threat to users: the growth of digital photography in public spaces by smartphone users and the increasing potential for visual eavesdrop attacks on unknowing users who increasingly use mobile devices to access applications which can display sensitive data. Statistical studies and sources were presented to establish these growing problems. Each of these problems were addressed by this thesis.

Chapter 2 presented an automated solution for conveniently protecting the privacy of bystanders in digital photography. A feature-based approach was explored with feature engineering steps to produce local blurring, relative face size, gaze deviation, and deviation of face from center metrics were created to capture the ways humans identify bystanders in an image. To further preserve user privacy, a convolutional neural network model was introduced to utilize only local face images, so that users might not have to provide entire images for subject/bystander detection. Beyond identifying bystanders, several methods for protecting face images were explored including blurring, black boxing, and face swapping. Through a comprehensive user study, it was found that users generally found the privacy impacts of public photography concerning and favored blurring as a method for protecting faces.

Chapter 3 presented an attacker detection scheme for mobile device users with integrated cameras and magnetometer sensors to quickly detect individuals around them who might shoulder surf or eavesdrop their screen. A binary thresholding mechanism using magnetic heading coupled with a deep, convolutional facial detection network allows users to easily scan their blind spots and surroundings without seeming conspicuous. To automate detection of

private content displayed in applications, UIaAI is presented as a means to pre-render UI views and avoid substantial code-base changes for developers. A deep object detection network was implemented and evaluated for detecting challenging UI elements such as text messages. Finally, a web browser extension was developed to automatically detect and obscure private content on web applications to handle cases where users must use web services to access or enter important information.

Due to the lack of available image datasets for the target/bystander detection application, collecting usable images including a good diversity of photo types, locations, lighting, etc. was a time-consuming process that could hardly be fully automated as photos had to be manually reviewed for suitability. The induced small size of the dataset might limit the generality of the results. In the future, this dataset will be expanded on to greatly increase the number of images and faces included. The goal of this is to ensure that models trained over the expanded dataset will be able to better generalize to in-the-wild photos. Making the dataset publicly available will allow other researchers to contribute and refine the images as well. Additionally, a more significant user study with a larger population will be carried out in the future to examine the usability of our system in greater detail.

Because the eavesdropping attacker detection scheme was unable to identify attackers that were moving at a fast speed, we identify this as a possible future improvement that could be made to further protect users. Some methods for deblurring or sharpening facial features will be explored to enhance the face detection network's ability in the future. Additionally, the content hiding browser extension was unable to correctly detect private fields in all tested websites. One of the main causes for this was the use of JavaScript functions to load new pages rather than perform HTTP requests. This can make it difficult to always detect the HTML login elements,

as they are encapsulated in a span and link tag. In the future, we hope to improve the detection logic to handle cases such as these.