

A Computer Vision Approach to Prevent Screen Photographing and Ensure LGPD Compliance

Mateus Padilha Carestiato Daniel, Breno de Souza Moraes Rego, João Gabriel Rodriguez, Gabriel Franklin Monteiro, Rigel P. Fernandes^{id}, and Cassius M. do C. Figueiredo

Abstract—Traditional Data Loss Prevention (DLP) tools cannot prevent a smartphone from photographing sensitive information displayed on a monitor. This paper proposes a computer-vision-based system that detects smartphones positioned toward a screen and triggers immediate screen obfuscation. Using a standard webcam and a YOLOv8n pipeline implemented in Python on Google Colab, the final model achieved a precision of 100%, recall of 98.3%, mAP50 of 99.5%, and a total pipeline latency of 8.01 ± 0.57 ms, demonstrating strong feasibility as a low-cost LGPD compliance tool.

Keywords—Computer Vision, Data Privacy, LGPD, Screen Obfuscation, Visual Hacking, YOLOv8.

I. INTRODUCTION

Sensitive information displayed on a monitor can be captured by a smartphone, bypassing firewalls, encryption, and DLP tools entirely. This visual hacking threat is especially relevant under privacy regulations such as the Brazilian LGPD [1]. Existing object detection approaches [2], [3] demonstrate the feasibility of real-time smartphone recognition, and real-time surveillance systems based on YOLOv5 [4] further confirm that YOLO-family detectors are well suited for monitoring unauthorized device usage. Privacy-preserving video analytics frameworks such as *PrivacyEye* [5] show that efficient video processing can be performed without exposing personal data, while screen-photography tracing techniques based on optical patterns [6] reinforce the relevance of camera-screen interaction for data leakage scenarios.

Prior work also shows that computer vision outperforms physical sensors in flexible environments [7], and that pre-trained models require fine-tuning to handle occlusion [3]. Related systems have demonstrated computer-vision-based monitoring in remote work contexts, including liveness detection approaches that rely solely on webcam input [8]. Building on these findings, this work proposes a YOLOv8-based system [9] that monitors a workstation webcam and obfuscates the screen upon detecting a smartphone capture attempt, without requiring specialized hardware.

The rest of this work is organized as follows. Section II presents the problem and system overview. Section III details the proposed method. Section IV presents the results. Section V concludes the work.

The authors are affiliated with the Computer and Software Engineering programs at Centro Universitário Ibmec, Rio de Janeiro, Brazil. Emails: mateuspcdaniel@gmail.com, brenorego10@gmail.com, rodriguez.joao@outlook.com, gf.monteiro04@gmail.com, rigelfernandes@gmail.com.

II. PROBLEM STATEMENT AND SYSTEM OVERVIEW

Visual hacking occurs when sensitive screen content is captured by an external camera, bypassing all software-based controls. Despite robust cybersecurity infrastructure, this analog loophole poses significant risks to LGPD compliance [1]. The threat model considered here assumes a handheld smartphone directed toward the monitor and visible in the webcam’s field of view, including partially occluded scenarios.

The proposed system must satisfy three requirements: (1) low-latency detection to obfuscate the screen before a clear capture occurs; (2) minimal false positives to avoid disrupting the user; and (3) deployment without specialized hardware, relying only on a standard webcam. The system operates as a background software layer, continuously monitoring the workspace and triggering screen obfuscation upon detection.

III. PROPOSED METHOD

A. System Architecture

The pipeline consists of six stages: frame acquisition, pre-processing, YOLOv8 inference, confidence filtering, threat decision, and screen obfuscation. Fig. 1 summarizes this flow. When a smartphone is detected above threshold τ , the obfuscation routine is activated; otherwise, the screen remains visible and the next frame is processed.

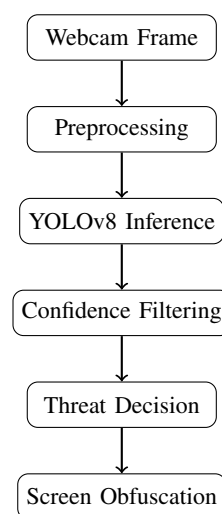


Fig. 1. Processing pipeline of the proposed system.

B. Model and Dataset

YOLOv8n was selected for its balance between accuracy and inference speed. Three training configurations were eval-

uated and compared, as detailed in Section IV. The dataset comprises 1,780 smartphone images from three sources: 100 images with Pascal VOC bounding-box annotations converted to YOLO format, and 1,680 classification images from public repositories for which synthetic full-frame bounding boxes were assigned. The final split is summarized in Table I.

TABLE I. Dataset organization.

Subset	Images
Training	1,424
Validation	356
Total	1,780

C. Detection and Obfuscation Logic

The obfuscation decision function $f(x)$ for a frame x is defined as:

$$f(x) = \begin{cases} 1, & \text{if } \exists_i (C_i \geq \tau \wedge ID_i = ID_{\text{phone}}), \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

When $f(x) = 1$, the screen is immediately obfuscated. The pseudocode below summarizes the routine:

Pseudocode: Detection and obfuscation routine

1. Initialize video stream and load YOLOv8n model.
2. Define confidence threshold τ .
3. While active:
 - 3.1. Capture and preprocess frame x .
 - 3.2. Run inference; for each detection i :
 - If $ID_i = ID_{\text{phone}}$ and $C_i \geq \tau$: obfuscate screen.
 - Otherwise: keep screen visible.

Fig. 2. Pseudocode of the smartphone detection routine.

IV. RESULTS

A. Quantitative Results

Three training configurations were systematically compared to identify the best-performing setup, as shown in Table II. The first two used only the 100-image annotated subset; the final configuration incorporated all 1,780 images with data augmentation (HSV jitter, horizontal flip, mosaic, mixup, and copy-paste). This dataset expansion was the key factor driving performance improvement, enabling the model to generalize substantially better and achieve near-perfect scores across all metrics.

TABLE II. Comparison of training configurations.

Configuration	Ep.	P (%)	R (%)	mAP50 (%)
YOLOv8n 100 imgs	30	46.7	41.2	51.4
YOLOv8n 100 imgs	60	48.9	52.9	56.5
YOLOv8n 1,780 + aug.	60	100.0	98.3	99.5

Table III details the final metrics for the selected model. The 100% precision confirms zero false positives on the validation set, meaning the system never incorrectly flagged a non-phone object. The 98.3% recall indicates that virtually all smartphone appearances were successfully detected, corresponding to an overall detection accuracy of 98.4% (370 out of 376 annotated instances correctly classified).

TABLE III. Final model performance (YOLOv8n, 1,780 images, 60 epochs).

Metric	Value
Precision	100.0%
Recall	98.3%
mAP50	99.5%
mAP50-95	97.4%
Accuracy	98.4%

B. Qualitative Results

Visual inspection confirms the model's ability to localize smartphones in practical scenarios, including partial occlusion. Fig. 3 shows a successful detection with a bounding box around a handheld device. Once the class condition is met, the screen-blocking routine is triggered, confirming the system's proactive capability.



Fig. 3. Successful detection under hand occlusion.

C. Latency

Table IV reports the mean and standard deviation of the processing time per pipeline stage, measured over 250 runs on an NVIDIA T4 GPU in Google Colab. The total latency of 8.01 ± 0.57 ms demonstrates strong real-time capability, well within the reaction window required to obfuscate the screen before a photograph can be taken.

 TABLE IV. Processing time per pipeline stage (NVIDIA T4, Google Colab, $n = 250$).

Stage	Mean $\pm$ Std (ms)
Preprocessing	3.37 ± 0.50
YOLOv8n inference	3.62 ± 0.06
Postprocessing	1.02 ± 0.35
Total	8.01 ± 0.57

V. CONCLUSION

This work demonstrated a real-time smartphone detection system using YOLOv8n and a standard webcam for LGPD-oriented visual data protection. By expanding the training dataset from 100 to 1,780 images combined with data augmentation, the model achieved 100% precision, 98.3% recall, 99.5% mAP50, and a total pipeline latency of 8.01 ± 0.57 ms — substantially outperforming the baseline configurations and confirming the system's feasibility for real-world deployment.

Remaining limitations include dependence on the webcam's field of view, sensitivity to concealed micro-cameras, and the use of synthetic full-frame bounding boxes for the 1,680 classification images, which may not precisely reflect real capturing postures. Future work should focus on annotating the full

dataset with precise bounding boxes, evaluating the system on continuous webcam streams under diverse lighting conditions, and integrating an operating-system-level obfuscation API for seamless deployment without GPU acceleration.

REFERENCES

- [1] Brasil, “Lei No 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD),” http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/113709.htm, 2018.
- [2] P. Rajput, S. Nag, and S. Mittal, “Detecting usage of mobile phones using deep learning technique,” in *Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good (GoodTechs '20)*. New York, NY, USA: ACM, 2020, pp. 96–101. [Online]. Available: <https://doi.org/10.1145/3411170.3411275>
- [3] P. Mustamo, “Object detection: TensorFlow object detection API case study,” Master’s thesis, University of Oulu, 2018.
- [4] R. H. P. Sejati, R. Mardhiyyah, Z. Zulkhairi, N. Istiqomah, and R. I. B. Prasetya, “Real-time smartphone usage surveillance system based on YOLOv5,” *IJID (International Journal on Informatics for Development)*, vol. 11, no. 2, pp. 242–251, 2023. [Online]. Available: <https://doi.org/10.14421/ijid.2022.3766>
- [5] W. Du, A. Li, P. Zhou, B. Niu, and D. Wu, “PrivacyEye: A privacy-preserving and computationally efficient deep learning-based mobile video analytics system,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3263–3279, 2022. [Online]. Available: <https://doi.org/10.1109/TMC.2021.3050458>
- [6] Y. Cheng, X. Ji, L. Wang, Q. Pang, Y.-C. Chen, and W. Xu, “mID: Tracing screen photos via Moiré patterns,” in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 2021, pp. 2969–2986. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity21/presentation/cheng-yushi>
- [7] H. G. V. Assumpção, C. de Souza de Medeiros, G. dos Santos Perrota Duarte, H. B. D. Rolan, and R. P. Fernandes, “Estacionamento inteligente: uma comparação entre sensores ultrassônicos e visão computacional,” in *XLII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - SBrT*. Belém, PA, Brazil: Sociedade Brasileira de Telecomunicações, 01–04 October 2024. [Online]. Available: <http://dx.doi.org/10.14209/sbrt.2024.1571036315>
- [8] F. O. de Saldanha da Gama, R. S. D. Jorge, C. R. Fernandes, M. M. do Nascimento Filho, R. P. Fernandes, and T. S. de Souza, “A computer vision-based system for monitoring liveness in remote work,” in *XLIII Brazilian Symposium on Telecommunications and Signal Processing - SBrT*. Natal, RN, Brazil: Sociedade Brasileira de Telecomunicações, 29 September – 02 October 2025. [Online]. Available: <http://doi.org/10.14209/sbrt.2025.1571143918>
- [9] M. P. C. Daniel, B. d. S. M. Rego, J. G. Rodriguez, and G. F. Monteiro, “Detecção de celulares com YOLOv8,” <https://github.com/mateuscaretiato/Deteccao-Celulares-YOLOv8>, 2026.