engineering proceedings

MDPI

Proceeding Paper

From Camera to Algorithm: OpenCV and AI Workshop for the Cybersecurity of the Future †

Pablo Natera-Muñoz [1], Fernando Broncano-Morgado [2] and Pablo Garcia-Rodriguez [1,*]

[1] Escuela Politécnica, University of Extremadura, 10003 Cáceres, Spain; natera@unex.es
[2] Escuela de Ingenierías Industriales, University of Extremadura, 06006 Badajoz, Spain; fbroncano@unex.es
* Correspondence: pablogr@unex.es; Tel.: +34-656375058
† Presented at the First Summer School on Artificial Intelligence in Cybersecurity, Cancun, Mexico, 3–7 November 2025.

Abstract

Artificial vision and artificial intelligence (AI) are increasingly interconnected in cybersecurity. This work presents an overview of OpenCV-based visual computing as a core tool for intelligent security systems that analyze real-time visual data. It includes practical exercises on face, edge, motion, and color detection, forming the basis for advanced object recognition using YOLOv10. Real applications, such as document processing and camera-based anomaly detection, are implemented in a microservice architecture with OpenCV, and deep learning frameworks. Integrating computer vision and AI is shown to be essential for developing resilient and autonomous cybersecurity infrastructures.

Keywords: computer vision; OpenCV; cybersecurity; artificial intelligence; CNN; deep learning

1. Introduction

The rapid evolution of cybersecurity increasingly relies on the integration of artificial intelligence (AI) and computer vision to detect, interpret, and respond to visual information in real time [1,2]. Modern infrastructures demand systems capable of not only monitoring but also understanding complex visual contexts, from surveillance cameras to identity verification systems.

OpenCV, an open-source library for image and video processing, provides a flexible and accessible foundation for developing such intelligent visual systems. It allows the implementation of real-time algorithms for face, motion, edge, and color detection, serving as the first layer of perception for higher-level AI models [3]. When combined with deep learning frameworks and object detection architectures such as YOLOv10 [2], OpenCV becomes a powerful tool for automated scene interpretation [4,5].

This paper explores the use of OpenCV in the context of cybersecurity applications, including automated surveillance, deepfake verification, and document analysis. The proposed approach employs a microservice-based architecture using Flask and distributed processing to ensure scalability and modularity. The aim is to demonstrate how combining computer vision and AI contributes to building resilient, autonomous, and intelligent cybersecurity systems capable of responding to emerging digital and physical threats [6].

2. Why Cybersecurity Needs "Eyes"

As cyber threats evolve, digital protection can no longer rely solely on data encryption or network monitoring. Modern cybersecurity increasingly depends on the system's ability

Check for updates

Academic Editors: Luis Javier García Villalba, Héctor Manuel Pérez Meana, Gabriel Sánchez Pérez and Ana Lucila Sandoval Orozco

Published: 30 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

to perceive and interpret the physical world. Vision-based AI systems provide these "eyes", enabling real-time detection of anomalies, behaviors, and manipulations that traditional algorithms cannot perceive.

Emerging challenges such as adversarial attacks, where attackers manipulate visual inputs to deceive AI models, demonstrate the urgency of integrating robust computer vision into cybersecurity frameworks. Similarly, the rise in deepfakes and synthetic media has blurred the boundaries between genuine and manipulated content, making visual verification essential for digital trust.

Moreover, automated surveillance systems equipped with AI can monitor hundreds of cameras simultaneously, identifying suspicious activities and abnormal patterns without human fatigue or bias. By combining OpenCV for low-level visual processing and AI-based interpretation for contextual understanding, cybersecurity systems gain the capability to "see", "analyze", and ultimately "decide" with enhanced autonomy and precision.

In short, integrating vision into cybersecurity represents a paradigm shift, from defending data to protecting digital and physical environments through perception and intelligence.

3. Basic OpenCV Exercises for Video-Based Cybersecurity

Practical experimentation with OpenCV offers a direct way to understand how visual perception can enhance cybersecurity systems. Working with real-time video streams allows developers to design algorithms that detect, track, and analyze activity within digital and physical environments, forming the basis for intelligent monitoring and automated threat response.

A first essential step is face detection, implemented through Haar Cascade classifiers or deep-learning–based models, which enables the identification and tracking of individuals in live video. This capability supports security tasks such as access control, identity verification, and behavioral monitoring in sensitive or restricted areas.

Another fundamental operation is edge detection, typically achieved with the Canny algorithm, which isolates structural features within each frame. Detecting edges allows systems to recognize object boundaries and identify tampering or manipulation in surveillance footage, contributing to visual integrity verification.

Motion detection further extends this capacity by comparing consecutive frames to measure variations in pixel intensity, enabling the identification of unexpected movements or intrusions. This technique provides the foundation for automated alarms and perimeter monitoring systems.

Finally, color-based tracking in the HSV color space facilitates the detection of objects or regions of interest based on color characteristics. It can be applied to identify specific objects, detect hazardous materials, or even recognize adversarial patterns designed to mislead AI-based recognition systems.

Through these exercises, OpenCV demonstrates how fundamental video-processing techniques can evolve into intelligent components of cybersecurity infrastructures, where perception, analysis, and automated decision-making converge to strengthen system resilience.

4. Use Case 1: Automated Spanish ID (DNI) Detection and OCR

We developed a pipeline to detect and read Spanish ID cards (DNI) in images using a combination of object detection, classical feature-based alignment, and zonal OCR. The dataset was generated by augmenting only 100 original DNI photographs to obtain 1600 training images; to simulate real-world noise and labeling errors, 5% of the dataset was intentionally replaced with driving licenses, producing controlled mismatches in the

https://doi.org/10.3390/engproc2026123004

database. Initial localization of the DNI in each frame is performed with an object detector (YOLO), which produces a bounding box that is used to crop the card region with OpenCV. This crop reduces background clutter and standardizes input for the next stages.

Within the cropped region, ORB keypoint detection and matching is applied against a canonical DNI template to establish correspondences. Robust matching yields a homography matrix that aligns the detected card to the template coordinate frame. The computed transformation is applied to generate a rectified, template-registered image, allowing consistent extraction of template zones (zonal OCR). Using OpenCV crops defined by the template, each zone (name, surname, document number, issue date, etc.) is isolated and preprocessed: illumination normalization, morphological filtering, and adaptive thresholding methods such as Otsu's algorithm are applied to enhance text contrast and reduce background noise prior to OCR.

Finally, text recognition is carried out with EasyOCR on the preprocessed zone crops. Postprocessing routines validate and normalize outputs (for example, enforcing numeric formats for DNI numbers, checking checksum rules, and cross-validating dates). The inclusion of 5% driving licences in training allows the system to learn robustness to impostor samples and to trigger verification flags when OCR content or layout deviates from expected templates. The overall pipeline is organized as modular stages (detection $\rightarrow$ alignment $\rightarrow$ zonal crop $\rightarrow$ preprocessing $\rightarrow$ OCR $\rightarrow$ validation), which facilitates deployment in a Flask microservice for real-time operation and logging.

Evaluation focuses on localization accuracy (IoU of predicted bounding boxes), homography quality (reprojection error), OCR accuracy per zone (character and field-level correctness), and the false-accept/false-reject balance introduced by impostor cards. Privacy-preserving measures and secure handling of personally identifiable information are implemented at the ingestion and storage layers to comply with regulatory constraints. This combined approach, fast, template-aligned crops plus classical feature alignment and robust zonal OCR, delivers a practical and scalable solution for automated DNI extraction in surveillance and identity-verification contexts.

5. Use Case 2: Video Surveillance with OpenCV and YOLO (People and Vehicles)

We implemented a real-time video-surveillance pipeline to detect and analyze people and vehicles using OpenCV for low-level video processing and YOLO for object detection. Video streams are ingested from IP cameras and normalized (resolution, frame rate, color space), with optional stabilization to mitigate vibrations. A motion prior is computed via background subtraction (MOG2) to focus inference on active regions. YOLO provides frame-level detections for the classes person and car (optionally truck, motorbike), followed by non-maximum suppression and multi-object tracking (e.g., ByteTrack/DeepSORT) to assign stable IDs across frames. Tracking enables higher-level analytics such as dwell time, line-crossing, and occupancy statistics.

To translate pixel coordinates into scene semantics, we calibrate each camera with a planar homography to a floor map or reference plane. This mapping supports zone-of-interest definitions and geofencing, enabling rules like "alert if a person enters a restricted area" or "count vehicles crossing a virtual gate." For traffic-like scenes, the calibrated geometry allows speed approximation and flow estimation. Color-based analysis in HSV complements detection for tasks such as selective highlighting (e.g., high-visibility garments) or basic vehicle color classification. For security, tamper detection monitors abrupt loss of scene content (occlusion, defocus, lens covering) and triggers maintenance alarms.

All events are published by a Flask microservice that aggregates per-camera summaries (counts, heatmaps, and time-of-day profiles) and writes structured logs to a message

queue for downstream analytics. A zonal cropping step produces privacy-preserving views: faces and license plates can be blurred in defined polygons to comply with data-protection policies while retaining aggregate statistics. Session summaries—counts per class, zone violations, and temporal histograms—are exported to spreadsheets to mirror the workshop's "session recaps." Model updates and thresholds are hot-reloadable to adapt sensitivity to each site.

Evaluation considers detection accuracy (mAP for person and vehicle), tracking consistency (IDF1, ID switches), and application reliability (false-alarm and miss rates in restricted zones). During operation, the system optimizes performance by adjusting the processing rate under load and skipping static frames based on motion priors. This integration of OpenCV preprocessing, YOLO detection, and privacy-aware postprocessing provides a robust and scalable framework for real-time video-based cybersecurity.

6. Conclusions

The integration of computer vision and artificial intelligence represents a crucial advancement in the design of modern cybersecurity systems. Through the use of OpenCV and YOLO-based architectures, this work has shown how real-time video analysis and document recognition can be effectively combined to enhance both digital and physical security. From identity verification using OCR-based DNI detection to multi-object video surveillance, the presented approaches demonstrate that perception-driven cybersecurity is achievable with open-source tools and modular architectures. Future work will focus on expanding dataset diversity, improving adversarial robustness, and integrating federated learning for secure, distributed model updates.

Author Contributions: Conceptualization, P.N.-M. and P.G.-R.; methodology, P.N.-M. and F.B.-M.; software, P.N.-M.; validation, P.N.-M., F.B.-M. and P.G.-R.; formal analysis, F.B.-M.; investigation, P.N.-M.; resources, P.G.-R.; data curation, P.N.-M.; writing—original draft preparation, P.N.-M.; writing—review and editing, F.B.-M. and P.G.-R.; visualization, P.N.-M.; supervision, P.G.-R.; project administration, P.G.-R.; funding acquisition, P.G.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found on YouTube by searching for public security camera footage. The specific list of video identifiers and links used as the dataset for this research is available on request from the corresponding author.

Acknowledgments: This initiative is carried out within the framework of the funds of the Recovery, Transformation and Resilience Plan, financed by the European Union (Next Generation)—National Cybersecurity Institute (INCIBE) in the project C107/23 "Artificial Intelligence Applied to Cybersecurity in Critical Water and Sanitation Infrastructures".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mittal, P. A comprehensive survey of deep learning-based lightweight object detection models for edge devices. *Artif. Intell. Rev.* **2024**, *57*, 242. [CrossRef]
2. Cong, X.; Li, S.; Chen, F.; Liu, C.; Meng, Y. A Review of YOLO Object Detection Algorithms based on Deep Learning. *Front. Comput. Intell. Syst.* **2023**, *4*, 17–20. [CrossRef]
3. Kultan, J.; Meruyert, S.; Danara, T.; Nassipzhan, D. Application of Computer Vision Methods for Information Security. *R&E-SOURCE* **2025**, *12*, 161–177. [CrossRef]

4. Villegas-Ch, W.; García-Ortiz, J. Authentication, access, and monitoring system for critical areas with the use of artificial intelligence integrated into perimeter security in a data center. *Front. Big Data* **2023**, *6*, 1200390. [CrossRef] [PubMed]
5. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458. [CrossRef]
6. Sánchez, M.; Cuartero, A.; Barrena, M.; Plaza, A. A New Method for Positional Accuracy Analysis in Georeferenced Satellite Images without Independent Ground Control Points. *Remote Sens.* **2020**, *12*, 4132. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.