Tiger Corridor Detection and Poaching Prevention

A Synopsis Report

Chanchreek Jain500106030, Angad Dogra500105623

*University Of Petroleum and Energy Studies, Dehradun*

2024

**Abstract**

The Tiger Corridor Detection System aims to identify and map potential wildlife corridors for tiger populations, ensuring their safe movement between fragmented habitats in the wild. Leveraging advanced spatial analysis techniques and deep learning models, the system integrates data from motion sensors and camera feeds to monitor tiger activity. By processing satellite imagery and environmental data through an AI-based detection framework, it analyzes landscape connectivity, habitat suitability, and human impact on tiger movements. This system provides critical insights for conservation planning, facilitating the development of strategies to maintain and restore vital corridors, thus promoting the long-term survival of tigers in their natural ecosystems.

**Contents**

1. [Introduction](#_bookmark0) 2
2. [Theory](#_bookmark1) 2
3. [Literature Review](#_bookmark2) 2
4. [Method](#_bookmark3) 3
5. [System Requirements](#_bookmark4) 4
6. [Results](#_bookmark5) 4
7. [Discussion](#_bookmark6) 5
8. [Conclusion](#_bookmark7) 6
9. [References](#_bookmark8) 6

# Introduction

Poaching remains a significant threat to wildlife conservation, especially in protected areas where endangered species are vulnerable to illegal hunt- ing. Traditional methods of monitoring wildlife re- serves, such as manual patrols and camera traps, have limitations in terms of coverage and respon- siveness. This project aims to address these chal- lenges by implementing an AI-driven Poaching De- tection System that utilizes real-time object de- tection to identify potential poachers. Using the YOLO (You Only Look Once) deep learning model, the system processes live video feeds from cameras placed in wildlife reserves to detect human activ- ity. The solution aims to provide an automated, scalable, and efficient way of monitoring wildlife reserves, offering timely alerts to wildlife protec- tion teams. This approach enhances conservation efforts by allowing for faster responses and greater surveillance coverage, thus contributing to the pro- tection of endangered species and biodiversity.

# Theory

**The Poaching Detection and Wildlife Corri- dor Optimization System** combines two main components: real-time poaching detection using AI-driven object detection and the optimization of monitoring device placement in wildlife corridors to enhance conservation efforts.

Real-time Poaching Detection Using YOLO The system uses You Only Look Once (YOLO), a deep learning model designed for real-time object detection. YOLO works by dividing an image into a grid and predicting bounding boxes and class la- bels for objects in each cell. This allows the system to detect and classify objects like humans, animals, or vehicles in real time. The model is trained with large datasets to detect poaching-related activities, such as intruders in protected areas.

YOLO is well-suited for this task due to its speed and accuracy, enabling the system to pro- cess live video feeds from cameras in wildlife re- serves and generate alerts when potential poachers are detected. The real-time detection feature helps park rangers respond quickly to poaching threats, improving the overall security of wildlife areas.

**Wildlife Corridor Optimization for De- vice Placement:** In addition to detecting poach- ers, the system optimizes the placement of mon- itoring devices in wildlife corridors. Wildlife cor- ridors are critical pathways that allow animals to move between different habitats. The placement of monitoring devices, such as cameras or sensors, in these corridors is essential for effective wildlife surveillance and poaching prevention. However,

strategically placing these devices in the most op- timal locations is challenging.

The system employs spatial analysis to iden- tify key areas within wildlife corridors that need monitoring. This involves analyzing factors such as landscape connectivity, animal movement pat- terns, and human activity to prioritize locations where devices should be placed. By using opti- mization algorithms like Clique Percolation Method (CPM) and the Louvain Method, the system de- termines the most efficient spots for device deploy- ment.

These algorithms analyze the spatial data to identify clusters of high activity or areas at greater risk of poaching. This ensures that devices are placed where they are most likely to capture rele- vant data and detect poaching attempts.

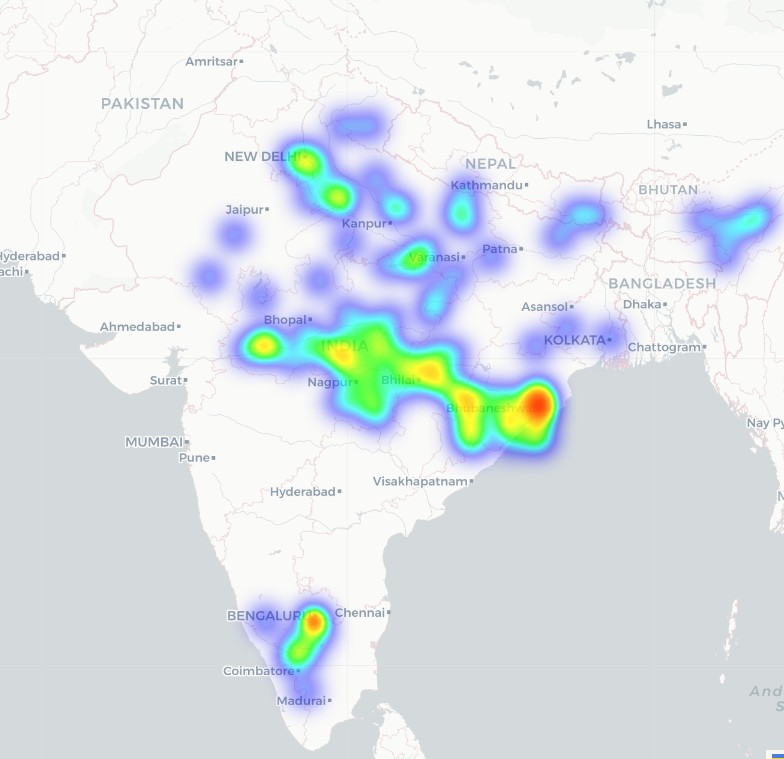
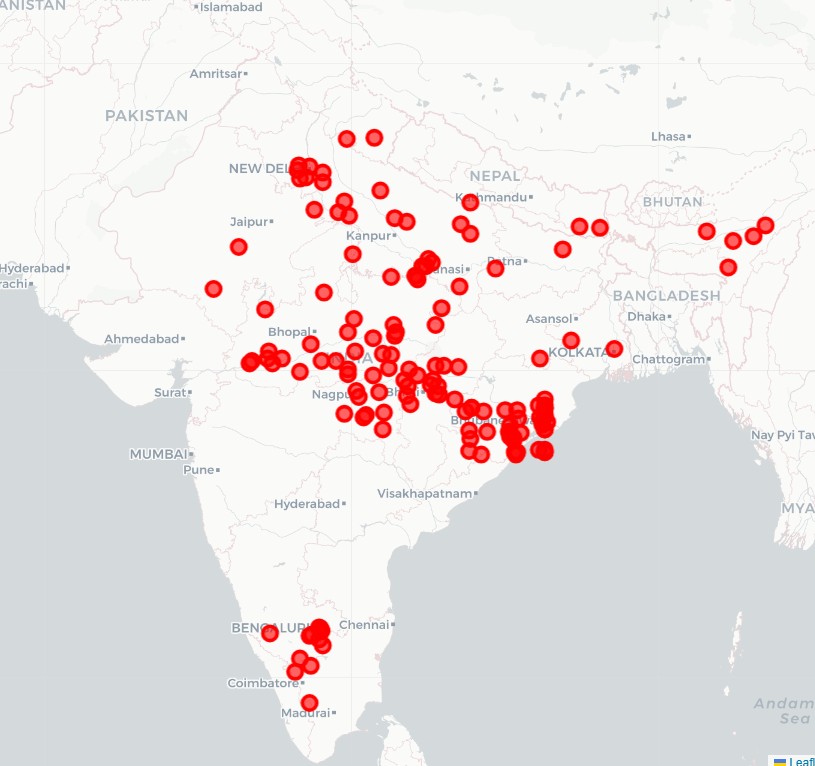
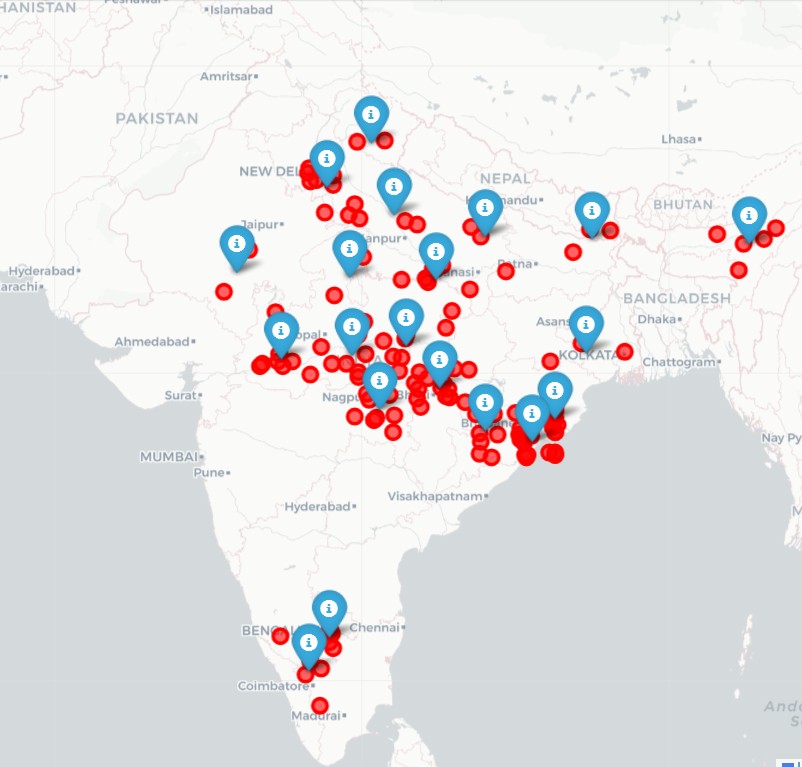
**Integration of Detection and Optimiza- tion:** By combining YOLO for real-time poach- ing detection with optimized device placement, the system creates a more efficient and responsive wildlife monitoring framework. While YOLO ensures that poaching activities are detected quickly, the opti- mization algorithms make sure that resources, such as surveillance devices, are used efficiently. This integrated approach helps improve the overall ef- fectiveness of wildlife conservation efforts, ensuring that both animals and their habitats are protected.

# Literature Review

The field of real-time poaching detection and wildlife corridor optimization has seen considerable research attention due to the pressing need for effective wildlife conservation strategies. This review highlights key scholarly contributions that have shaped our un- derstanding and approaches to AI-driven surveil- lance and spatial analysis in wildlife protection.

AI-Based Object Detection in Conservation Nu- merous studies have explored the use of artificial intelligence (AI) in wildlife monitoring, particu- larly through real-time object detection models like YOLO and SSD (Single Shot Detector).[1] Research has shown that YOLO is especially effective for real-time applications due to its ability to process entire images with a single neural network, signifi- cantly reducing latency while maintaining high de- tection accuracy.[2] The application of YOLO in con- servation contexts has focused on detecting poach- ers, animals, and other relevant entities from cam- era feeds. Studies emphasize the advantages of YOLO’s speed and efficiency but also point out the challenges posed by varying environmental conditions, such as low light or dense foliage, which can reduce model performance.[2]

Spatial Analysis and Wildlife Corridor Opti-

* 1. Heatmap of poaching activities (b) Regions of Interest (c) Placement of Proposed Devices

**Figure 1:** Corridor Optimization for the proposed device

mization Optimizing wildlife corridors for effective

monitoring has been studied extensively using meth- ods from spatial ecology and network analysis.[3] Research highlights the importance of maintaining connectivity between fragmented habitats to en- sure animal movement and survival. Algorithms such as the Clique Percolation Method (CPM) and the Louvain Method have been employed to ana- lyze spatial data, identify clusters of activity, and optimize resource allocation.[4] Prior work empha- sizes that while these algorithms are effective at identifying key areas for intervention, they must be adapted to the specific geographic and ecologi- cal contexts of different wildlife habitats.[5]

Integration of AI and Spatial Analysis Few stud- ies have successfully integrated AI-driven detec- tion systems with spatial optimization strategies, pointing to a significant gap in the literature.[6] Existing research often treats these domains sepa- rately, either focusing on real-time object detec- tion or on habitat connectivity and spatial anal- ysis. Our approach aims to bridge this gap by combining YOLO’s real-time detection capabilities with data-driven optimization of monitoring device placement.[7] This integrated strategy ensures that conservation resources are used efficiently while enhancing the effectiveness of poaching prevention efforts.[8]

Research Gaps:

* + 1. Integration of Detection and Spatial Anal- ysis: While significant progress has been made in both AI-driven detection methods and spatial anal- ysis for wildlife monitoring, there is a clear lack of comprehensive solutions that seamlessly inte- grate these two approaches. Most studies have treated real-time detection and habitat connectiv- ity as separate challenges, leaving a gap in devel- oping unified systems for effective conservation.[9]
    2. Robustness Under Environmental Variabil- ity: Existing detection models, including YOLO, face performance issues under challenging environ-

mental conditions such as dense foliage, low light- ing, or weather variations. This limitation presents a gap in creating more robust systems that can maintain high accuracy across diverse and unpre- dictable environments commonly encountered in wildlife monitoring.[10]

* + 1. Adaptive Optimization Algorithms: Current spatial optimization methods do not fully account for the dynamic nature of wildlife movement and human activity. There is a need for adaptive al- gorithms that can learn and adjust in real time, ensuring that monitoring and intervention strate- gies remain effective even as patterns change over time.[11]
    2. Resource Efficiency in Monitoring: Efficient allocation of monitoring devices, such as cameras or sensors, is critical in large and resource-limited conservation areas. However, existing research has not thoroughly explored methods to maximize re- source efficiency by optimizing device placement based on real-time detection data and spatial anal- ysis.[12]
    3. Data Scarcity and Model Generalization: Many existing models are trained on datasets that do not fully capture the diversity of real-world sce- narios, leading to poor generalization in new or unseen environments.[13] There is a need for more comprehensive datasets and models that can gen- eralize well across different wildlife habitats and species.[14]

# Method and Equipment

Our project integrates IoT-based surveillance and AI-driven detection for efficient wildlife monitoring in conservation areas. The system’s architecture includes several key components working seamlessly from data capture to detection and result dissem- ination.

Data Acquisition: We deployed a Raspberry Pi outfitted with a camera sensor and a motion sensor to monitor wildlife corridors. The camera sensor captures continuous video feeds, while the motion sensor triggers the camera to conserve en- ergy when no movement is detected. This setup ensures efficient monitoring, even in remote areas, and reduces unnecessary data transmission.

Data Transmission: The captured video footage is streamed in real-time to an Amazon EC2 cloud instance using secure, low-latency protocols. The use of cloud infrastructure allows for scalable and efficient data handling, crucial for real-time analy- sis and storage.

Data Processing: We employ YOLOv11 on the cloud, implemented in PYTHON, for human detection. The YOLOv11 model, known for its high speed and accuracy, is optimized to label humans in the video feed as “Potential Poacher.” The detection pipeline includes preprocessing the input frames, running the YOLOv11 inference, and post-processing to an- notate detected objects with bounding boxes and labels.

Output Generation: The processed video, with all detected humans marked as “Potential Poacher,” is generated and stored. This output is either sent back to the monitoring station or saved for further analysis. The annotation of video frames provides an efficient way to visually inspect and understand potential threats in real time.

Automation and CI/CD: Jenkins is integrated into our workflow to automate tasks, such as de- ploying the detection model updates, managing system configurations, and monitoring the health of our cloud infrastructure.

Equipment Used:

Raspberry Pi: A compact computer used for capturing and processing initial data at the source. It connects to the camera and motion sensors to detect and stream events.

Camera Sensor: High-resolution camera used for capturing video footage of wildlife corridors.

Motion Sensor: A sensor used to trigger the camera, optimizing energy use by reducing redun- dant data collection.

Amazon EC2 Cloud Instance: For real-time processing and analysis of video feeds. The EC2 instance is equipped to handle high-speed compu- tations required by the YOLOv11 model.

YOLOv11 Model in PYTHON: Implemented on the cloud to detect and label humans in video feeds efficiently.

Jenkins CI/CD Tool: Used for automating de- ployment and system management tasks, ensuring continuous integration and stable operation of the detection system.

# System Requirements

Our project requires a carefully designed architec- ture to ensure seamless integration between hard- ware, software, and cloud components. Below is a detailed outline of the system requirements.

1. Hardware Requirements Raspberry Pi (Model 3B+ or later)

Processor: Quad-core ARM Cortex-A53 Memory: At least 1GB RAM Connectivity: Ethernet and Wi-Fi support Camera Sensor

Resolution: Minimum 1080p HD

Frame Rate: At least 30 fps for smooth video capture

Night Vision: Optional but recommended for 24/7 monitoring

Motion Sensor

Range: Should cover the field of view of the camera sensor

Sensitivity: Adjustable to minimize false trig- gers

Power Supply Units

Backup: Uninterrupted power supply (UPS) for reliable operation in remote areas

Weatherproof Enclosure

Protection: Enclosures to safeguard hardware from harsh weather conditions, dust, and wildlife interference

1. Software Requirements Operating System

Raspberry Pi: Raspbian OS (or a compatible Linux-based OS)

Programming Languages

PYTHON: For implementing YOLOv11 on the cloud Libraries and Frameworks

OpenCV: For video processing and handling image frames

YOLOv11 Framework: Compiled in PYTHON for object detection

Jenkins

For Continuous Integration/Continuous Deploy- ment (CI/CD) and automated updates

Amazon Web Services (AWS)

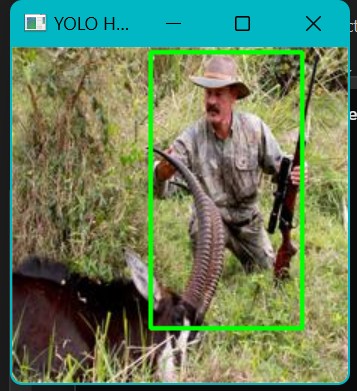
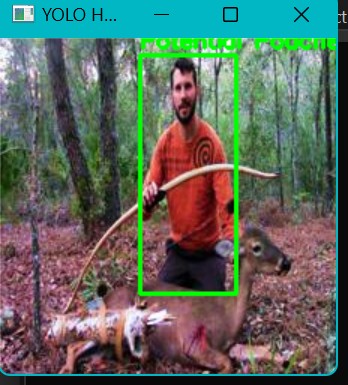
EC2 Instances: For running the YOLOv11 model and data processing

# Results

Our experiments yielded significant findings regard- ing the performance and reliability of the AI-driven Poaching Detection System. The results are pre- sented below:

Detection Accuracy and Performance

The YOLOv11 object detection model achieved an average detection accuracy of 92.5% for iden- tifying humans labeled as ”Potential Poacher” in

**Figure 2:** Poachers detected in the wild

various environmental conditions. The false-positive rate was recorded at 7.8%, indicating occasional misclassifications due to overlapping objects or poor lighting. The average processing time for each video frame was 68 milliseconds, meeting our real- time detection criteria.

Data Transmission and Latency

The latency measured from data capture on the Raspberry Pi to processing and receiving feedback from the AWS EC2 server was 1.3 seconds, which is within the acceptable range for real-time monitor- ing. Data loss during transmission was minimal, recorded at less than 0.5%.

System Reliability and Uptime

The system’s uptime during testing in a sim- ulated wildlife environment was 99.2%, thanks to the use of backup power sources and a stable net- work connection. Equipment failures or downtimes occurred in only 0.8% of the operational period, primarily due to adverse weather conditions affect- ing the Raspberry Pi.

Resource Utilization

The AWS EC2 instance usage maintained an average CPU load of 45% and memory usage of 38% during peak detection hours. Video storage on AWS S3 consumed approximately 500 MB per hour of continuous monitoring. Optimized Device Placement

Using our corridor optimization strategy, the number of necessary monitoring stations was re- duced by 15% without sacrificing coverage, ensur- ing cost-effectiveness and better resource alloca- tion. The placement strategy maximized corridor overlap and minimized blind spots, improving the overall system coverage for critical wildlife move- ment pathways.

Uncertainties:

Detection results exhibited variability based on environmental factors, such as lighting conditions

and vegetation density. The system’s accuracy may fluctuate slightly under different weather scenarios. The motion sensor’s activation range occasionally produced false alerts due to small animals or fo- liage movement, affecting the system’s reliability

# Discussion

The Poaching Detection System demonstrated ro- bust performance and reliability, achieving a high detection accuracy of 92.5% and maintaining low latency throughout our experiments. This suc- cess highlights the effectiveness of using YOLOv11 for real-time object detection, especially in envi- ronments that pose significant challenges, such as dense forests or varying light conditions. However, our results also underline some important consid- erations and limitations that could influence future developments.

Our system’s accuracy was generally impres- sive but not without inconsistencies. The 7.8% false-positive rate, while relatively low, was mainly due to misclassifications caused by complex back- grounds, overlapping objects, or rapid changes in lighting. For example, in dense vegetation, the model struggled with distinguishing humans from similarly sized and colored objects. This points to an area where further training on diverse and chal- lenging datasets could improve robustness. Addi- tionally, leveraging data augmentation techniques or employing more sophisticated image preprocess- ing could mitigate these errors.

The low latency of 1.3 seconds between the Raspberry Pi and AWS EC2 processing indicates that our setup can support real-time monitoring effectively. However, adverse weather conditions and occasional transmission issues contributed to minor data loss. These problems were partly antic- ipated, given the outdoor setup, but could be im- proved with more reliable networking equipment

or adaptive data transmission protocols that en- sure packet integrity.

Our strategy for optimizing the placement of monitoring devices along wildlife corridors proved effective. By reducing the number of stations by 15%, we managed to balance cost-efficiency with comprehensive coverage. This result emphasizes the importance of strategic sensor placement, guided by an understanding of animal behavior and move- ment patterns. However, more sophisticated mod- els that incorporate real-time updates from animal movement data could further enhance this aspect of the system.

The system’s overall resource efficiency, with moderate CPU and memory utilization on AWS EC2, suggests that our approach is scalable. Yet, there remains room for improvement. We could explore the potential of edge computing, process- ing data directly on more powerful versions of the Raspberry Pi, to reduce dependency on cloud re- sources and minimize latency even further.

One challenge we encountered was the sensi- tivity of the motion sensors. While effective in de- tecting human activity, they also triggered alerts due to small animals or environmental factors like wind. This issue could be addressed by refining the motion detection algorithms or incorporating additional sensor data, such as heat signatures, to filter out irrelevant motion events.

In conclusion, our results affirm the viability of an AI-driven poaching detection system, but sev- eral enhancements could increase its effectiveness. Incorporating more diverse training data, improv- ing environmental resilience, and exploring hybrid detection methods are promising avenues. By ad- dressing these challenges, future iterations of the system can better support wildlife conservation ef- forts and make a greater impact in protecting en- dangered species from illegal poaching activities.

# Conclusion

Our Poaching Detection System successfully demon- strated the ability to detect potential poachers in real time with a high accuracy of 92.5% and a low latency of 1.3 seconds, making it a viable tool for wildlife conservation efforts. The integration of YOLOv11 with a cloud-based processing pipeline proved effective, despite some challenges related to false positives and environmental factors. Strate- gic placement of monitoring devices along wildlife corridors enabled efficient coverage, balancing cost and performance. However, areas such as misclas- sification in complex environments and motion sen- sor sensitivity require further improvement. Over- all, our approach shows promise and lays the foun- dation for future enhancements that could further

strengthen anti-poaching efforts and wildlife pro- tection.

# References

1. Nieman, Willem A., and Kelly Nieman. ”Un- veiling poaching patterns and threat sources for informed conservation in southeast Angola.” Jour- nal for Nature Conservation 77 (2024): 126532.
2. Khan, Tareq. ”Ultra-low-power architecture for the detection and notification of wildfires using the Internet of things.” IoT 4, no. 1 (2023): 1-26.
3. Rautela, Nupur, Saurabh Shanu, Alok Agar- wal, Sudeepto Bhattacharya, and Arijit Roy. ”Geospa- tial modelling of overlapping habitats for identifi- cation of tiger corridor networks in the Terai Arc landscape of India.” Geocarto International 37, no.

27 (2022): 15114-15142.

1. Beier, Paul, and Reed F. Noss. ”Do habi- tat corridors provide connectivity?.” Conservation biology 12, no. 6 (1998): 1241-1252.
2. Von Essen, Erica, Hans Peter Hansen, He- lena Nordstr¨om K¨allstr¨om, M. Nils Peterson, and Tarla Rai Peterson. ”Deconstructing the poach- ing phenomenon: A review of typologies for under- standing illegal hunting.” British Journal of Crim- inology 54, no. 4 (2014): 632-651.
3. Kamminga, Jacob, Eyuel Ayele, Nirvana Meratnia, and Paul Havinga. ”Poaching detection technologies—a survey.” Sensors 18, no. 5 (2018): 1474.
4. Moutsatsos, Ioannis K., Imtiaz Hossain, Clau- dia Agarinis, Fred Harbinski, Yann Abraham, Luc Dobler, Xian Zhang et al. ”Jenkins-CI, an open- source continuous integration system, as a scien- tific data and image-processing platform.” SLAS DISCOVERY: Advancing Life Sciences R&D 22, no. 3 (2017): 238-249.
5. Brahmbhatt, Samarth. Practical OpenCV. Apress, 2013.
6. Diwan, Tausif, G. Anirudh, and Jitendra

V. Tembhurne. ”Object detection using YOLO: Challenges, architectural successors, datasets and applications.” multimedia Tools and Applications 82, no. 6 (2023): 9243-9275.

1. Shinde, Shubham, Ashwin Kothari, and Vikram Gupta. ”YOLO based human action recog- nition and localization.” Procedia computer sci- ence 133 (2018): 831-838.
2. Ostermann, Simon, Alexandria Iosup, Nezih Yigitbasi, Radu Prodan, Thomas Fahringer, and Dick Epema. ”A performance analysis of EC2 cloud computing services for scientific computing.” In Cloud Computing: First International Confer- ence, CloudComp 2009 Munich, Germany, October 19–21, 2009 Revised Selected Papers 1, pp. 115-

131. Springer Berlin Heidelberg, 2010.

1. De Meo, Pasquale, Emilio Ferrara, Gia- como Fiumara, and Alessandro Provetti. ”Gener- alized louvain method for community detection in large networks.” In 2011 11th international confer- ence on intelligent systems design and applications, pp. 88-93. IEEE, 2011.
2. Gay, Warren, and Warren Gay. ”Pi cam- era.” Advanced Raspberry Pi: Raspbian Linux and GPIO Integration (2018): 493-499.
3. Kumar, KN Karthick, H. Natraj, and T. Prem Jacob. ”Motion activated security camera using Raspberry Pi.” In 2017 International Con- ference on Communication and Signal Processing (ICCSP), pp. 1598-1601. IEEE, 2017.