International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Anti-Drone Protection and Surveillance for Vips in Public Spaces

Dr. S. Brindha[1], Ms. T. P. Kamatchi[2], Mr. G. Parthive[3], Mr. Tamizh Eniyan Sarvanan[4], Mr. Niranjan Kumar Manjunathan[5], Mr. R. Namith Kumar[6], Mr. K. S. Harish Siva[7]

[1]HoD, [2]HoD, [3, 4, 5, 6, 7]Students
[1]HoD, Department of Computer Networking, PSG Polytechnic College
[2]HoDDepartment of Computer Engineering, PSG Polytechnic College

[3, 4, 5, 6, 7]Students, Department of Computer Networking, PSG Polytechnic College

Abstract:
This project focuses on developing a real-time weapon and intrusion detection system using drones and Convolutional Neural Networks (CNN). A drone equipped with a Wi-Fi camera captures live video, from which frames are extracted and processed by the Raspberry Pi Pico. The CNN algorithm accurately detects weapons and intrusions, triggering instant alerts via a GSM modem to a connected PC. A GPS module provides precise location tracking for enhanced monitoring.
The system is divided into three modules: hardware integration, CNN-based detection, and drone deployment for real-time surveillance. This cost-effective solution improves security in sensitive areas, ensuring quick threat detection and response while addressing challenges like processing speed, accuracy, and false positives.

Keywords: Weapon detection, Convolutional Neural Networks (CNNs), GSM, Alert Mechanism.

INTRODUCTION:

Security and surveillance are essential for preventing unauthorized access and ensuring public safety. Traditional security systems often rely on fixed cameras and manual monitoring, which can limit coverage and delay response time. To overcome these limitations, this project employs drones equipped with Convolutional Neural Networks (CNNs) for real-time weapon and intrusion detection. By integrating drone mobility with intelligent detection algorithms, the system aims to enhance security measures with rapid threat identification and alert mechanisms.
CNNs are highly effective in detecting complex visual patterns, making them well-suited for identifying weapons and intruders in diverse environments. The drone captures live footage, which is processed directly on the Raspberry Pi Pico using CNNs[29] to detect potential threats instantly. Once a weapon or unauthorized individual is identified, the system triggers a GSM alert, providing immediate notification to the relevant authorities.
Weapon and intrusion detection systems play a crucial role in public safety, especially in high-risk areas. Several approaches have been explored to improve detection accuracy, categorized as follows:

1. Weapon Detection Algorithms: CNN-based models to identify suspicious objects such as weapons.
2. Intrusion Detection Mechanisms: Real-time identification of unauthorized access through motion tracking and behavioral analysis.
3. Alert Systems: Instant communication via GSM for quick response and intervention.
4. Environmental Adaptation Techniques: Enhancing detection accuracy across various lighting and environmental conditions.

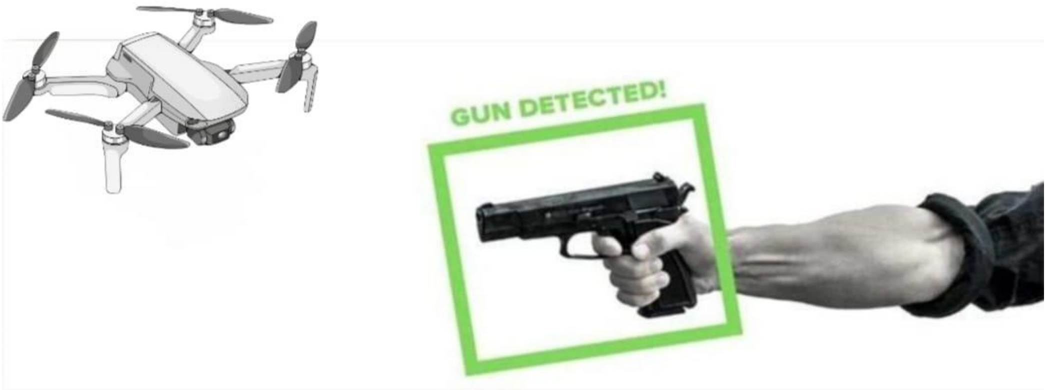

Fig. 1. Demonstration of automated weapon detection using Drone camera.

1. BACKGROUND:

Traditional surveillance systems often fall short in providing real-time detection and response. To address these challenges, integrating drones with Convolutional Neural Networks (CNNs) offers a promising solution for weapon and intrusion detection. CNNs excel at identifying patterns in visual data, making them highly effective for detecting objects like weapons and unauthorized individuals. The drone's camera captures live footage, which is processed by the Raspberry Pi Pico using a CNN model to detect potential threats instantly.

This system combines drone mobility with CNN's accuracy to enhance coverage and responsiveness. Upon detection, alerts are sent via a GSM modem, ensuring timely intervention. The use of CNNs enables precise object recognition, even in complex environments, making this project a reliable, real-time security solution for safeguarding sensitive areas.

1.1 CNN (Convolutional Neural Network)

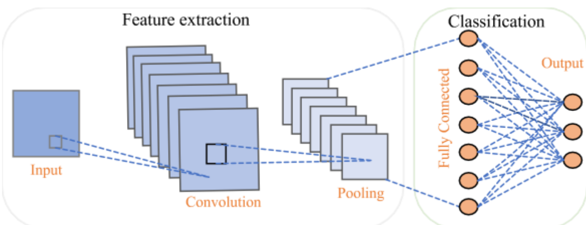

Fig. 2. Convolutional Neural Network

The CNN or ConvNet is a special kind of deep-learning architecture that has gained much attention in computer vision and robotics. The initial idea of CNN, called *neocognitron*, was presented in 1979 by Kunihiko Fukushima, which later became known as the predecessor of CNN. Furthermore, the CNN architecture has been explained by Le-Cun *et al.* later, an improved version was explained in.

A developed CNN network called LeNet-5 was found to be able to classify handwritten digits. Popular architectures from 2012 to 2015 are examined in, along with their basic components. The basic structure of the CNN model comprises three types of layers: convolutional, pooling, and fully connected. The purpose of the convolution layer is to perform feature extraction. In the convolutional operation, an array of numbers (kernel) is applied across inputs (tensor) to construct the feature map. The procedure of constructing a feature map is an element wise product between each element of the kernel and the input tensor, and the outputs are summed to obtain the element of the kernel.

The kernel convolves across all the elements on the input tensor to construct the elements of the feature map for that kernel. An arbitrary number of feature maps can be obtained by implementing the convolution operation with different kernels. While training, the convolution operation is called forward propagation; during back propagation, the gradient descent optimization technique updates the learnable parameters (kernels and weights) according to the loss value.

A pooling layer provides a typical down sampling operation to reduce the dimensionality of the feature maps to introduce translation invariance to small shifts and distortions and thereby decrease the number of subsequent learnable parameters. The pooling function is *pool*(·); for each feature *j* is a local neighborhood around location (*i*, *j*). The fully connected layers are the final outputs of the CNN, such as the probabilities for each class in classification tasks. The number of output nodes in the final fully connected layer is usually equal to the number of classes. A nonlinear function, such as ReLU, follows each fully connected layer. Finally, a loss function is calculated to assess the compatibility of the CNN's forward propagation output predictions with the provided ground truth labels.

The loss function for CNN optimization is given by**:**

$$L = \frac{1}{N}\sum_{n=1}^{N}(y_n \log(y_n) + (1 - y_n)\log(1 - y_n))$$

where N is the number of samples, y_n is the actual label, and $\hat{y}_n$ is the predicted probability

Training a CNN determines the global minima, which identify the best-fitting set of parameters by minimizing the loss function. Currently, many CNN models exist, such as AlexNet, ZFNet, VGGNet, GoogLeNet/Inception and ResNet.

2. PROPOSED SYSTEM

2.1 Proposed System Overview

The proposed system utilizes a Convolutional Neural Network (CNN) to detect weapons and intruders in real-time through drone-captured footage. CNNs are highly effective in processing visual data, making them ideal for object detection tasks. The live video feed is processed frame-by-frame on the Raspberry Pi Pico, which runs a lightweight CNN model optimized for speed and accuracy.

Convolutional Layer: The CNN applies convolution operations to extract features from each frame. The convolution operation is represented as:

Z=(X∗W) + B

Where:

- X = Input image/frame
- W = Filter (kernel)
- B = Bias term
- Z = Feature map output

Activation Function: After convolution, an activation function like ReLU (Rectified Linear Unit) is applied to introduce non-linearity:

f(x)=max (0, x)

Pooling Layer: To reduce the spatial dimensions while retaining key features, max pooling is used:

P=max(Z)

Alert Mechanism: If the CNN [1] detects a weapon or intruder with high confidence, the system triggers an alert through the GSM modem and sends the location coordinates via the GPS module. Additionally, the LCD screen displays the real-time status of the surveillance area.

Fig. 3. Samples from the detection

2.2 WORKING of Proposed System:

2.2.1 Drone Surveillance and Video Capture:

The drone continuously patrols the surveillance area, capturing live video footage through its onboard Wi-Fi camera. The live feed is transmitted wirelessly to the Raspberry Pi Pico for real-time processing, ensuring dynamic coverage across large areas.

2.2.2 Frame Extraction and Preprocessing:

The incoming video stream is broken down into individual frames for analysis. Each frame is resized to fit the CNN input dimensions and normalized to improve model performance. Preprocessing techniques, such as noise reduction and contrast enhancement, are applied to handle varying lighting conditions and reduce detection errors.

2.2.3 Feature Extraction via Convolutional Layers:

The CNN analyzes each frame, identifying potential weapons by extracting spatial features like shapes, edges, and textures. Multiple convolutional layers progressively refine these features, producing high-confidence detections while minimizing false positives. The CNN outputs confidence scores for each detected object, and a threshold is applied to classify objects as weapons only if the confidence score exceeds a predefined value.

2.2.4 Weapon Detection Using CNN:

The CNN analyzes each frame, identifying potential weapons by extracting spatial features like shapes, edges, and textures. Multiple convolutional layers progressively refine these features, producing high-confidence detections while minimizing false positives. The CNN outputs confidence scores for each detected object, and a threshold is applied to classify objects as weapons only if the confidence score exceeds a predefined value.

2.2.5 Threat Verification and Decision-Making:

To improve accuracy, the system employs a multi-frame verification process, ensuring the detection is consistent across consecutive frames before raising an alert. This reduces the risk of false positives caused by sudden movements or environmental factors.

2.2.6 Real-Time Alert System:

The Raspberry Pi Pico triggers the GSM module to send an immediate alert to a connected PC or directly to security personnel. The alert message includes critical information, such as the type of detected weapon and GPS coordinates, enabling a rapid response.

2.3 Weapon Detection Using CNNs:

Weapon detection using CNNs involves training a deep learning model to identify weapons in images or video frames captured by a drone. The CNN extracts features from the input data, such as edges and shapes, to classify objects as weapons or non-weapons.

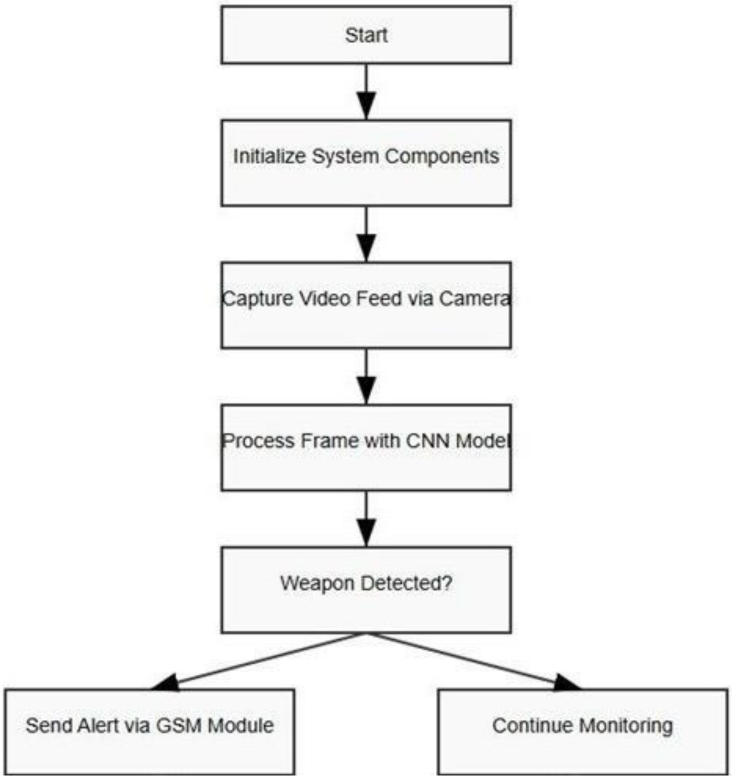

Fig. 4. Flow chart of weapon detection

2.3.1 Drone Surveillance and Video Capture:

The drone continuously patrols the surveillance area, capturing live video footage through its onboard Wi-Fi camera. The live feed is transmitted wirelessly to the Raspberry Pi Pico for real-time processing, ensuring dynamic coverage across large areas.

2.3.2 Frame Extraction and Preprocessing:

The incoming video stream is broken down into individual frames for analysis. Each frame is resized to fit the CNN input dimensions and normalized to improve model performance. Preprocessing techniques, such as noise reduction and contrast enhancement, are applied to handle varying lighting conditions and reduce detection errors.

2.3.3 Feature Extraction via Convolutional Layers:

The CNN analyzes each frame, identifying potential weapons by extracting spatial features like shapes, edges, and textures. Multiple convolutional layers progressively refine these features, producing high-confidence detections while minimizing false positives. The CNN outputs confidence scores for each detected object, and a threshold is applied to classify objects as weapons only if the confidence score exceeds a predefined value.

2.3.4 Weapon Detection Using CNN:

The CNN analyzes each frame, identifying potential weapons by extracting spatial features like shapes, edges, and textures. Multiple convolutional layers progressively refine these features, producing high-confidence detections while minimizing false positives. The CNN outputs confidence scores for each detected object, and a threshold is applied to classify objects as weapons only if the confidence score exceeds a predefined value.

2.3.5 Threat Verification and Decision-Making:

To improve accuracy, the system employs a multi-frame verification process, ensuring the detection is consistent across consecutive frames before raising an alert. This reduces the risk of false positives caused by sudden movements or environmental factors.

2.3.6 Real-Time Alert System:

The Raspberry Pi Pico triggers the GSM module to send an immediate alert to a connected PC or directly to security personnel. The alert message includes critical information, such as the type of detected weapon and GPS coordinates, enabling a rapid response.

2.4 Weapon Detection Using CNNs:

Deepfake detection becomes more complex because it has to consider temporal dependencies between frames. For this, CNNs are used to extract spatial features from each frame, analyze the sequence of frames over time to capture the motion and temporal patterns.

Deepfake Detection Using CNNs

2.4.1 Drone Surveillance System:

Drones[13] have revolutionized security and surveillance by offering mobility, wide-area coverage, and real-time monitoring. Equipped with a Wi-Fi camera, the drone captures live footage of its surroundings, enabling constant surveillance of high-risk areas. This aerial vantage point enhances the system's ability to detect potential threats over large environments.

2.4.2 Convolutional Neural Networks (CNN):

At the heart of the detection system lies the Convolutional Neural Network (CNN), a deep learning architecture specially designed for image processing tasks. CNNs excel in extracting spatial features like edges, shapes, and textures, making them ideal for detecting weapons. Layers of convolutions, pooling, and fully connected nodes enable the model to distinguish weapons from other objects with high accuracy.

2.4.3 Real-Time Video Processing:

The video feed from the drone is processed frame by frame in real time[29]. Each frame is resized and preprocessed to match the input dimensions required by the CNN. Techniques such as noise reduction and contrast adjustment are applied to ensure consistency, enhancing detection performance in dynamic environments.

2.4.4 Threat Classification and Confidence Thresholds:

The CNN analyzes each frame to classify objects, assigning a confidence score that reflects the certainty of a weapon's presence. A predefined confidence threshold is used to minimize false positives — only objects surpassing this threshold are classified as weapons, ensuring reliable detection.

2.4.5 Alert and Notification System:

Upon detecting a weapon, the system triggers the GSM module to send immediate alerts to security personnel or a central monitoring station. These alerts contain crucial information about the location and nature of the threat, ensuring a rapid response to potentially dangerous situations.

2.4.6 GPS Tracking and Location Identification:

A GPS module integrated into the system provides real-time location data of the drone and the detection site. When a weapon is detected, the system records the coordinates, helping responders precisely locate the area of concern and track the drone's movements.

2.4.7 LCD Display for Visual Feedback:

The Raspberry Pi Pico is connected to an LCD that continuously displays system status. When no weapon is detected, the screen shows a "No Weapon Detected" message. Upon detecting a weapon, the display updates instantly, providing visual feedback for on-site personnel.

2.4.8 Performance Evaluation and Optimization:

System performance is evaluated through metrics like accuracy, precision, recall, and F1-score. Optimization techniques, such as reducing the CNN's complexity and adjusting frame rates, are employed to ensure real-time processing on the Raspberry Pi Pico without compromising accuracy.

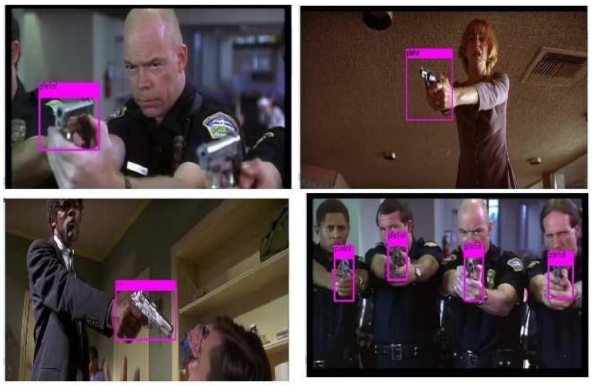

Fig. 5. Weapon detection and localization results

2.5 CNN Use Cases:

1. Weapon Detection in Surveillance: CNNs can identify weapons like guns[32] or knives[21] in real-time security footage by analyzing spatial features in each frame. This helps in quickly recognizing potential threats.
2. Integration of CNN and RNN**:** The CNN extracts spatial features from each frame (e.g., shape and texture of objects), and these feature vectors are passed into the CNN. The CNN captures temporal patterns, such as detecting if someone is holding a weapon consistently across multiple frames, reducing false positives caused by brief occlusions or reflections.

3. Final Classification: The combined CNN output is fed into a fully connected layer that classifies whether the detected object is indeed a weapon. This improves accuracy by considering both spatial and temporal features, ensuring more reliable threat detection.

2.6 CNN for Weapon Detection:

Input Stage: In the input stage, video frames from the drone camera are captured and preprocessed. This includes resizing, normalization, and noise reduction to ensure consistent input for the CNN. These frames serve as the raw data for weapon detection.

CNN Stage: The CNN stage processes the input frames through multiple layers. Convolutional layers extract features like edges and shapes, while activation functions (e.g., ReLU) introduce non-linearity. Pooling layers reduce dimensions, preserving essential features. Finally, fully connected layers classify the detected object, triggering alerts if a weapon is identified.

3. CHALLENGES FOR CREATION DEEPFAKE and DETECTION OF DEEPFAKE:

3.1 CHALLENERS FOR CREATION:

Developing a robust weapon detection system using drones and CNNs presents several challenges. These include model optimization, dataset diversity, environmental factors, and real-time processing constraints.

Model Optimization: Designing a CNN model that balances accuracy and efficiency is crucial, especially when deploying on resource-constrained hardware like the Raspberry Pi Pico. Reducing model complexity while ensuring effective feature extraction is a challenging task.

Dataset Diversity: Training a CNN requires a comprehensive dataset that includes various weapon types, angles, and lighting conditions. Collecting diverse datasets to prevent model bias and ensure generalization across different scenarios is time-consuming and complex.

Environmental Factors: Real-world conditions such as varying lighting, weather changes, and camera angles can significantly impact detection performance. Addressing these factors during training and optimizing the model for diverse conditions remains a key challenge.

Real-time Processing: Ensuring the CNN[29] processes frames in real-time without lag is crucial for timely alerts. Optimizing Python code for frame extraction, processing, and communication with GSM and GPS modules requires careful handling of computational overhead.

3.2 CHALLENGES FOR DETECTION:

Once the system is deployed, several challenges arise in detecting weapons accurately in real-world scenarios. These challenges include false positives, occlusion, and system integration.

False Positives: Distinguishing between real weapons and objects with similar shapes or features can lead to false positives. Implementing filtering techniques such as confidence thresholds and multiple-frame verification is necessary to reduce errors.

Occlusion and Motion Blur: Fast-moving drones and objects can introduce motion blur, reducing detection accuracy. Additionally, partial occlusion of weapons due to environmental obstacles or human positioning adds complexity to accurate detection.

System Integration: Integrating CNN-based detection with drone footage, Raspberry Pi Pico processing, and GSM alert transmission requires seamless coordination. Ensuring smooth data flow from the drone's camera to the detection system and real-time alert generation is a technical challenge.

Future Work:

- Improving CNN architecture for better accuracy while maintaining low latency on resource-constrainned devices.

- Enhancing the dataset with diverse weapon types and real-world scenarios to increase model robustness.
- Implementing advanced filtering techniques to reduce false positives and enhance detection reliability.
- Optimizing GSM alert systems for quicker and more reliable communication with authorities.
- Exploring additional features such as automatic drone path adjustment based on detection outcomes to improve surveillance coverage.

CONCLUSION:

The weapon and intrusion detection system using drones and Convolutional Neural Networks (CNNs) presents a significant advancement in real-time surveillance technology. By leveraging a drone-mounted camera, the system captures live video, extracts frames, and processes them on the Raspberry Pi Pico, ensuring lightweight, efficient object detection without relying on external computational power. The integration of CNNs enhances detection accuracy by identifying weapons and intruders with high precision, while minimizing false positives through techniques like confidence thresholds and multi-frame verification.

The system's ability to send immediate alerts via the GSM module and pinpoint locations using GPS ensures a rapid response to potential threats. This makes it highly adaptable for diverse security applications, including border patrol, event monitoring, and public safety enforcement. The modular architecture allows seamless integration with existing security infrastructures and ensures portability across different environments.

Despite challenges such as hardware limitations and real-time processing constraints, the project demonstrates the feasibility of deploying CNN-based detection models on low-power embedded devices.

REFERENCES:

1. Mehta, P., Kumar, A. and Bhattacharjee, S., "Fire and gun violence based anomaly detection system using deep neural networks" in International Conference on Electronics and Sustainable Communication Systems (ICESC): IEEE, pp. 199-204, July 2020.
2. Tiwari, R.K. and Verma, G.K., "A computer vision based framework for visual gun detection using harris interest point detector" Procedia Computer Science, vol. 54, pp.703-712, 2015.
3. Krizhevsky, A., Sutskever, I. and Hinton, G.E., "Imagenet classification with deep convolutional neural networks" Advances in neural information processing systems, vol. 25, pp.1097-1105, 2012.
4. Zhang, Q., Yang, L.T., Chen, Z. and Li, P., "A survey on deep learning for big data" Information Fusion, vol. 42, pp.146-157, 2018.
5. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., "Imagenet large scale visual recognition challenge" International journal of computer vision, vol. 115(3), pp.211-252, 2015
6. Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., "Object detection with deep learning: A review" IEEE transactions on neural networks and learning systems, vol. 30(11), pp.3212-3232, 2019.
7. Olmos, R., Tabik, S. and Herrera, F., "Automatic handgun detection alarm in videos using deep learning" Neurocomputing, vol. 275, pp.66-72, 2018.
8. Verma, G.K. and Dhillon, A., "A handheld gun detection using faster r- cnn deep learning" in Proceedings of the 7th International Conference on Computer and Communication Technology, pp.

84-88, November 2017.

9. Grega, M., Matiolański, A., Guzik, P. and Leszczuk, M., "Automated detection of firearms and knives in a CCTV image" Sensors, vol. 16(1), p.47, 2016.
10. Redmon, J. and Farhadi, A., "Yolov3: An incremental improvement"
11. arXiv preprint arXiv:1804.02767, 2018.
12. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., "You only look once: Unified, real-time object detection" in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788, 2016.
13. Hashmi, T.S.S., Haq, N.U., Fraz, M.M. and Shahzad, M., "Application of Deep Learning for Weapons Detection in Surveillance Videos" in International Conference on Digital Futures and Transformative Technologies (ICoDT2): IEEE, pp. 1-6, May 2021.
14. Luvizon, D.C., Picard, D. and Tabia, H., "2d/3d pose estimation and action recognition using multitask deep learning" in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5137- 5146, 2018.
15. Elmir, Y., Laouar, S.A. and Hamdaoui, L., "Deep Learning for Automatic Detection of Handguns in Video Sequences" in JERI, April 2019.
16. Warsi, A., Abdullah, M., Husen, M.N., Yahya, M., Khan, S. and Jawaid, N., "Gun detection system using YOLOv3" in IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA): IEEE, pp. 1-4, August 2019.
17. Tiwari, Rohit Kumar, and Gyanendra K. Verma. "A computer vision- based framework for visual gun detection using harris interest point detector." Procedia Computer Science, vol. 54, pp. 703-712, 2015.
18. González, J.L.S., Zaccaro, C., Álvarez-García, J.A., Morillo, L.M.S. and Caparrini, F.S., "Real-time gun detection in CCTV: An open problem" Neural networks, vol. 132, pp.297-308, 2020.
19. Akcay, S., Kundegorski, M.E., Willcocks, C.G. and Breckon, T.P., "Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery" IEEE transactions on information forensics and security, vol. 13(9), pp.2203- 2215, 2018.
20. Egiazarov, A., Mavroeidis, V., Zennaro, F.M. and Vishi, K., "Firearm detection and segmentation using an ensemble of semantic neural networks" in European Intelligence and Security Informatics Conference (EISIC): IEEE, pp. 70-77, November 2019.
21. Buckchash, H. and Raman, B., "A robust object detector: application to detection of visual knives" in IEEE International Conference on Multimedia & Expo Workshops (ICMEW): IEEE, pp. 633-638, July 2017.
22. Castillo, A., Tabik, S., Pérez, F., Olmos, R. and Herrera, F., "Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning" Neurocomputing, vol. 330, pp.151-161, 2019.
23. Tiwari, R.K. and Verma, G.K., "A computer vision based framework for visual gun detection using SURF" in International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO): IEEE, pp. 1-5, January 2015.
24. Singleton, M., Taylor, B., Taylor, J. and Liu, Q., "Gun identification using TensorFlow" in International Conference on Machine Learning and Intelligent Communications, Cham: Springer, pp. 3-12, July 2018.
25. Lai, J. and Maples, S., "Developing a real-time gun detection classifier"
26. Course: CS231n, Stanford University, July 2018.

27. R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, 2014.
28. R. Girshick, “Fast r-CNN,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448, 2015.
29. S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real- time object detection with region proposal networks,” in Advances in Neural Information Processing Systems, pp. 91–99, 2015.
30. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788, 2016.
31. J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271, 2017.
32. Li, P. and Zhao, W., “Image fire detection algorithms based on convolutional neural networks” Case Studies in Thermal Engineering, vol. 19, p.1006, 2020.