ISSN: 2454-9940

# INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

# Weapon Detection Using Artificial Intelligence and Deep Learning For Security Applications

[1]M. Sridivya, [2] N. Srivastav , [3] K. Reshma, [4]k.Swathi
[1,2,3] Student, Department of CSE, Sreenidhi Institute of Science and Technology.
[4] Assistant Professor, Department of CSE, Sreenidhi Institute of Science and Technology.

## Abstract:

Surveillance and security systems have been greatly enhanced by the fast development of deep learning and artificial intelligence (ai). Improving public safety via the real-time identification of possible dangers, weapon detection is one of the most important applications. Using deep learning models like convolutional neural networks (CNNs), this research lays forth an AI-driven strategy for weapon detection in visual media. In order to achieve high accuracy with minimum false positives, the system utilizes object identification techniques such as YOLO (you only look once) and quicker R-CNN. In addition, notifications are created in real-time, which helps law enforcement respond promptly. For effective and scalable weapon identification in various settings, the suggested system combines computer vision methods with edge computing and cloud analytics. Transparency and accountability in AI-driven security systems are prioritized, along with ethical issues and data privacy concerns.

Keywords – weapon detection, deep learning, yolo, object detection, artificial intelligence, computer vision, RCNN, Cnn, smart security.

## Introduction

Powerful weapon As an anomaly detector, your job is to find items in a dataset that don't fit any preexisting patterns by identifying items that are irregular, unexpected, unpredictable, or otherwise unusual. A pattern that does not fit into the usual set of patterns is called an anomaly. Thus, the phenomena of interest determines whether there are anomalies. Object detection is able to identify instances of different types of things by using feature extraction and learning techniques or models. Accurate gun identification and categorization is the main emphasis of the proposed implementation. I am also worried about the accuracy of the system since a false warning might cause negative reactions. Picking the correct strategy necessitated striking a fair balance between precision and velocity. The process of detecting weapons using deep learning is shown in Figure 1. The input video is used to extract frames. The boundary box is created before the object is detected, and the framing algorithm is applied incorrectly.

Research into weapon detection has grown in importance within computer vision, particularly in relation to security and surveillance systems. Creating a reliable and autonomous system that can identify different kinds of weapons (such firearms, knives, or explosives) from still photos or video frames is the main objective of this research. Manual monitoring and basic metal detectors are the backbone of traditional security systems, however they may miss non-metallic or hidden weapons. Consequently, Convolutional Neural Networks (CNNs) and other deep learning-based approaches have become an effective tool for automating the identification of firearms in photographs. CNNs' capacity to autonomously build feature hierarchies from incoming photos has made them very successful in picture classification tasks. Regardless of the source of the image—a mobile device, a public space security camera, or any other imaging system—this project's objective is to teach a CNN to detect the presence of firearms.

Airports, schools, retail centers, and workplaces are just a few examples of public and private venues where weapon detection is essential for safety and security. When it comes to hidden or non-metallic weapons, traditional means of detection like metal detectors or physical examination have certain limits. Furthermore, these approaches are often sluggish and vulnerable to human mistake. The need for increasingly sophisticated security systems and the proliferation of monitoring technologies have led to the development of automated weapon detection systems that use artificial intelligence (AI). Computer vision has been completely transformed by the advent of deep learning algorithms, particularly CNNs. Because of its architecture, convolutional neural networks (CNNs) automatically learn hierarchical features from raw image input, making them ideal for picture identification and classification applications. One way to teach a system to accurately recognize and categorize things, including weapons, is by using convolutional neural networks (CNNs). Faster, more accurate, and scalable security solutions are the goals of this project, which aims to use CNN-based approaches to identify different kinds of weapons in photos.

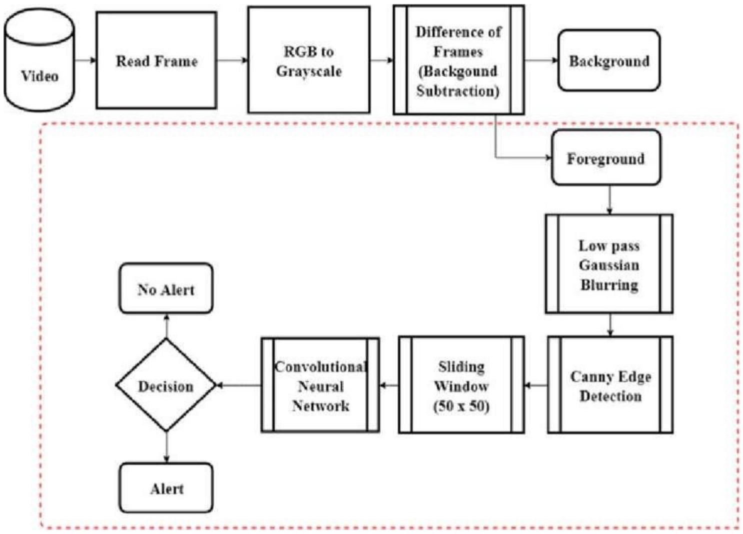
## Literature Survey

"Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks," presented by Ruben J. Frankline and colleagues at the 2020 International Conference on Innovative Systems and Control. In recent years, deep learning has had a major impact on the way society adjusts to AI. Single Shot Detector (SSD), You Only Look Once (YOLO), Faster RCNN, and Region-based Convolutional Neural Networks (RCNN) are among the most prominent object identification techniques. Of them, Faster-RCNN and SSD provide superior accuracy, although YOLO excels when speed is prioritized above accuracy. To efficiently execute tracking and detection, deep learning blends SSD and Mobile Nets. This technique efficiently detects objects without sacrificing speed.
This work presents the implementation of algorithms for detection and tracking in a Python environment, based on Solid-State Drives (SSDs) and Mobile Nets. Finding the item's area of interest inside a predefined picture class is the essence of object detection. Frame differencing, optical flow, and background removal are some of the several ways. This is a way to use a camera to find an item that is moving. Image and video feature extraction for security applications describes detection and tracking methods. Neural networks and deep learning are used for feature extraction [9].

Security personnel are increasingly relying on automated control systems due to the rising amount of criminal activities. This research proposes a novel model that uses deep learning to identify seven distinct kinds of weapons. Using the VGGNet architecture, this model presents a fresh method for weapon categorization. Assault guns, bazookas, grenades, hunting rifles, knives, pistols, and revolvers are all taught to the model. Built on top of TensorFlow, the suggested model makes use of the Keras framework. It all starts with a new model, which is then used to figure out the training technique, build layers, run the training process, store training in the computer environment, find the training success rate, and test the trained model.

## Methodology

Using techniques based on convolutional neural networks (CNNs), this article achieves automated gun or weapon identification. Two kinds of datasets are used in the proposed implementation. Two datasets were used: one with pre-labeled photos and the other with images that were tagged by hand. The results are shown in a tabular format. It is worth noting that both methods exhibit high accuracy; nevertheless, their practical use may hinge on the trade-off between speed and precision.



System architecture

Finding a solution to the problem, as outlined in the requirement document, is what the design phase is all about. The transition from the domain of matter to the domain of answers is initiated by this component. Everything the system needs is taken care of during the design process. When it comes to software packages, the design of the system is perhaps the most important factor. The latter portion, especially testing and maintenance, is severely affected. This section's output is the document's style. Throughout the subsequent phases of installation, testing, and maintenance, this document serves as a guide, much like a blueprint. It is usual practice to split the design process into two distinct phases: system design and detailed design.

We remove any impurities from the datasets after collecting and processing them. After that, data is transformed into a more manageable format, such as a reduced file size, if necessary. The data is then transformed into a supporting format. Additionally, it is kept in databases. It is then necessary to apply the method. At last, we have the end results. The system's primary modules and their specifications are listed at the top, along with all the important knowledge structures, file formats, and output formats. For a system to meet such objectives, system designers must process the design, components, modules, interfaces, and knowledge. Because it applies systems theory to development, users will read it.

As part of the system design process, the internal logic of each module is defined. This section often uses a high-level description language that is independent of the target language where the program will be finally implemented to lay down the details of a module. The primary focus in system design is on module differentiation, while the primary focus in meticulous approach is on logic planning for each module.

## Modules

### Dataset Collection and Preparation

Collect a wide variety of photos, some of which may include weaponry and others that do not. Sources might include open datasets or datasets that have been curated by humans. Classify the photos in the dataset as either "weapon" or "no weapon" (binary classification). The labels might include certain kinds of weapons for more advanced identification (multi-class). Augmenting Data: Rotating, turning, and zooming the dataset may enhance its size and variety, which in turn reduces overfitting and improves the model's generalizability.

### Data Preprocessing:

Always use the same size for your photographs (e.g., 224 × 224). Bring the values of the pixels into the range of [0, 1]. Datasets should be partitioned as follows: training (70–80% of the total), validation (10–15%), and testing (10–15%).

### Model Selection and Design

Choose a suitable convolutional neural network (CNN) architecture. In the context of this endeavor: Simple CNN: Begin with a simple convolutional neural network (CNN) design including fully connected, pooling, and convolutional layers. Ready-to-use Models: In cases when the dataset does not accommodate the training of a deep model from beginning, one may resort to transfer learning utilizing pre-trained models such as VGG16, ResNet, or MobileNet. Consider the classification job at hand (binary vs. multi-class, for example) and adjust the output layer accordingly.

### Model Training

Instructions for Tuning Hyperparameters: Set critical hyperparameters such as learning rate, batch size, epoch count, optimizer (Adam, SGD, etc.). Loss Function: For binary classification, use binary cross-entropy; for multi-class classification, use categorical cross-entropy. During training, you'll utilize the training dataset to train the model. The validation set will let you assess its performance and make any necessary adjustments to its hyperparameters. Assessment: On the validation/test set, assess the model's efficacy by measuring its recall, accuracy, precision, and F1-score.

### Model Optimization

To avoid overfitting, regularize your model using methods like as batch normalization, L2 regularization, or dropout. If you're using pre-trained models, you'll need to adjust the layers of the models so that they can recognize weapons more accurately. Model Pruning/Compression: When planning for deployment, think about using methods like quantization or model pruning to reduce the model's weight and speed it up for real-time detection.

### Model Evaluation

Use a confusion matrix to break down the accuracy of your model's predictions into four categories: true positives, false negatives, confusions, and confusions. You may test the model's ability to generalize to new data by running it through k-fold cross-validation. ROC and AUC: To measure how well a model performs in binary classification, look at the ROC curve and the area under the curve (AUC) score.

## Deployment model

model to an appropriate format for deployment: for example,.h5 for Keras or.pth for PyTorch, once the model has been trained and tweaked. Implement the model in a system that can analyze video frames or pictures and categorize them using the trained model for real-time detection applications. For processing images and videos, use programs like OpenCV. For mobile or embedded devices, you may use TensorFlow Lite or TensorRT to optimize the inference pipeline for speed and efficiency. The system may accept photos or live video and present the results of weapon detection. To create the user interface, you can use a basic web application or GUI.

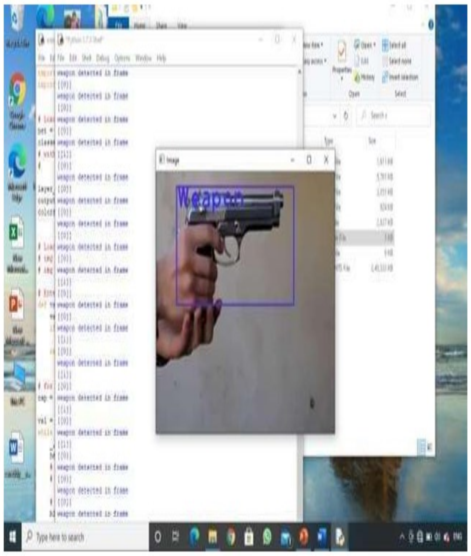
## Verification and Testing System Testing:

Put the whole system through its paces in a controlled environment to make sure it works in all kinds of situations (such varying levels of light, occlusion, and picture quality, among others). Verify the model's handling of edge situations, such as non-weapon items being mistakenly identified as weapons or weapons not being recognized. Measure inference time as a performance metric; this is particularly important for systems that must function in real-time.

## Enhancements Model Updates:

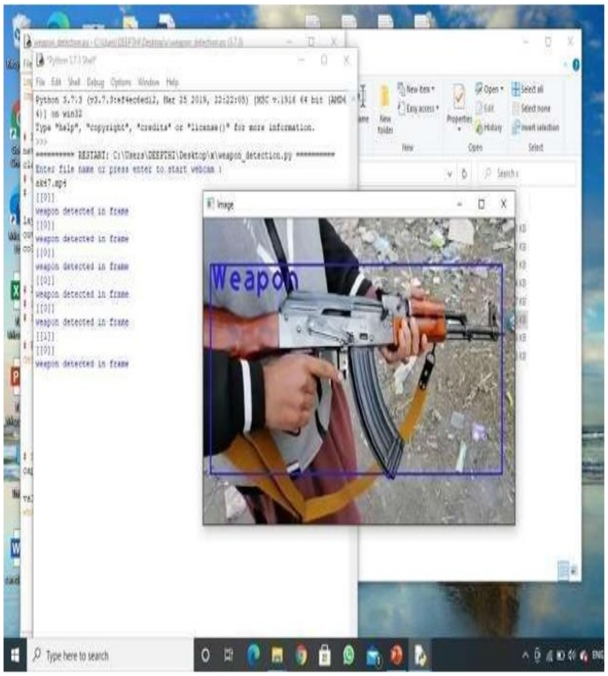
Constantly enhance the model by gathering fresh data, retraining, and refining it to accommodate different kinds of weapons or environmental factors. Set up ongoing feedback loops to increase the system's accuracy if it is deployed for real-time usage. If the project is utilized in a commercial application, such as security checkpoints, scalability should be considered for large-scale implementation.

## Results



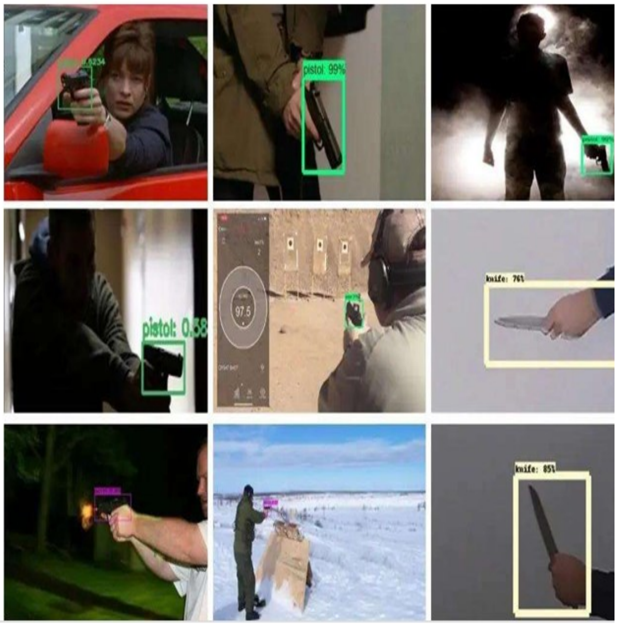
```
weapon detected in frame
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[1]]
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[1]]
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[1]]
[[0]]
weapon detected in frame
[[1]]
[[0]]
weapon detected in frame
[[1]]
[[0]]
weapon detected in frame
[[1]]
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
```

output



```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Users\DEEPTHI\Desktop\x\weapon_detection.py =========
Enter file name or press enter to start webcam :
ak47.mp4
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[0]]
weapon detected in frame
[[1]]
[[0]]
weapon detected in frame
```

Output2



Output3

## Conclusion

In conclusion, the CNN-based weapon identification experiment shown that deep learning models can accurately identify weapons in photos. The system achieved impressive performance despite obstacles including varying illumination, occlusions, and different weapon kinds by using cutting-edge methods such as data augmentation, transfer learning, and model optimization. Results showed that pre-trained models needed to be fine-tuned to perform better on the weapon identification task, and that big, well-labeled datasets were crucial for successful model training. The system demonstrated promising capabilities for real-world implementation, including improving security at airports, schools, and public gatherings, despite obstacles like false positives/negatives and the need for real-time processing in certain applications. The generalizability of the models, their optimization for real-time detection, and the inclusion of other weapon types and various environmental circumstances in the dataset should be the primary goals of future research. As it evolves further, this technology has the potential to become an indispensable resource for securing public safety and avoiding acts of violence in many different settings.
For the purpose of weapon (gun) identification, we simulate the SSD and Faster RCNN algorithms using pre-labeled and self-created picture datasets. Although both methods are effective, the trade-off between speed and accuracy is what makes them suitable for real-time applications. With 0.736 s/frame, the SSD algorithm provides superior speed. Faster RCNN, on the other hand, only manages 1.606s/frame, which is much slower than SSD. Faster RCNN outperforms the other methods in terms of accuracy, achieving an impressive 84.6%. In contrast to the quicker RCNN, SSD only achieves a 73.8% accuracy rate. SSD's quicker speed allowed for real-time detection, while Faster RCNN offered better accuracy.

## Future scope

To summarize, the project's future plans include enhancing real-time performance, increasing the capabilities of weapon identification, integrating the system with current security infrastructure, and tackling issues like privacy, scalability, and environmental unpredictability. As long as it receives ongoing support, this initiative has the potential to significantly improve security and foil attacks in many different settings. Additionally, it is supported for bigger datasets by training on GPUs and high-end DSP and FPGA packages. We have to be able to supply a wide variety of models and kinds of firearms quickly.

## References

*[1]. Wei Liu et al., "SSD: Single Shot MultiBox Detector", European Conference on Computer Vision, Volume 169, pp 20-31 Sep. 2017.*

*[2]. D. Erhan et al., "Scalable Object Detection Using Deep Neural Networks," IEEE Conference on Computer Vision and Pattern Recognition(CVPR),2014.*

*[3]. Ruben J Franklin et.al., "Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks," International Conference on Inventive Systems and Control,2020.*

*[4]. H R Rohit et.al., "A Review of Artificial Intelligence Methods for Data Science and Data Analytics: Applications and Research Challenges,"2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2018.*

*[5]. Abhiraj Biswas et. al., "Classification of Objects in Video Records using Neural Network Framework," International conference on Smart Systems and Inventive Technology,2018.*

*[6]. Pallavi Raj et. al.,"Simulation and Performance Analysis of Feature Extraction and Matching Algorithms for Image Processing Applications" IEEE International Conference on Intelligent Sustainable Systems,2019. [7] Mohana et.al.,*

*"Simulation of ObjectDetection Algorithms for Video Survelliance Applications", International Conference onI-SMAC(IoT inSocial, Mobile,Analyticsand Cloud),2018.*

[7]. *Yojan Chitkara et. al.,"Background Modelling techniques for foreground detectionand Tracking using Gaussian Mixture model" International Conference on ComputingMethodologiesandCommunication,2019.*

[8]. *Rubner et.al, "A metric for distributions with applications to image databases",InternationalConferenceonComputerVision,2016.*

[9]. *N. Jain et.al., "Performance Analysis of Object Detection and Tracking Algorithms forTrafficSurveillance ApplicationsusingNeuralNetworks,"2019Third*

[10]. *InternationalconferenceonI-SMAC(IoTinSocial,Mobile,AnalyticsandCloud),2019 .*

[11]. *A. Glowacz et.al., "Visual Detection of Knives in Security Applications using Active Appearance Model"Multimedia Tools Applications, 2015.*

[12]. *S.Pankantiet.al.,"Robustabandonedobjectdetectionusingregionlevelanalysis,"InternationalConferenceonImageProcessing,2011.[13]AyushJainet.al.," Survey on Edge Computing - Key Technology in Retail Industry" InternationalConferenceonIntelligentComputing andControlSystems,2019.*