Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 7001388, 11 pages
https://doi.org/10.1155/2022/7001388

Research Article

Towards Proactive Surveillance through CCTV Cameras under Edge-Computing and Deep Learning

Abdul Jaleel,[1] **Syed Khaldoon Khurshid**,[2] **Rehman Mustafa**,[2] **Khalid Mehmood Aamir**,[3] **Madeeha Tahir**,[4] **and Ahmad Ziar**[5]

[1]*Department of Computer Science (RCET, GRW), University of Engineering and Technology, Lahore, Pakistan*
[2]*Department of Computer Science, University of Engineering and Technology, Lahore, Pakistan*
[3]*Department of CS & IT, University of Sargodha, Sargodha, Pakistan*
[4]*Department of Mathematics, Government College Women University, Faisalabad, Pakistan*
[5]*Department of Computer Science, Kardan University, Kabul 1007, Afghanistan*

Correspondence should be addressed to Ahmad Ziar; r.ziar@kardan.edu.af

Received 11 March 2022; Accepted 9 June 2022; Published 13 July 2022

Academic Editor: Musavarah Sarwar

Weapons, usually a handgun, a revolver, or a pistol, are used in the majority of criminal acts. The traditional closed-circuit television (CCTV) surveillance and control system requires human intervention to detect such crime incidents. The purpose of this research is to develop a real-time automatic weapon carrier detection system that may be used with CCTV cameras and surveillance systems. The goal is to alarm and alert the security officials to take proactive action to prevent violent activities. In deep learning literature, region-based classifiers (R-FCN and Faster R-CNN) and regression-based detectors (Yolo invariant) are being used as promising object detection methods. Although region-based classifiers are accurate, they lack the speed of detection required for real-time detection, whereas regression-based detectors (for example, YoloV4 invariant) are fast enough for real-time detection, but lack accuracy. The method applied in this study relies on Yolov4 to quickly detect anomalies, followed by R-FCN to boost detection accuracy by filtering out any false positives. A weapon dataset comprising 4430 locally and internationally available weapon photos with a 70–30 split ratio is used to train and test the system, which is subsequently evaluated using a live surveillance camera system. This hybrid system achieved a 90% accuracy with a low false positive rate, as well as 94% precision, 86% recall, and 89% *F*1 score. Our results prove that the proposed hybrid system is useful for proactive real-time surveillance to alarm the existence of a suspicious weapon carrier in a surveillance area.

1. Introduction

With the increase in world population and unemployment ratio, criminal activities are increasing with each day. It is imperative to improve the conventional surveillance and security methods. The reactive approach of the conventional policing system begins investigations following the occurrence of robbery, snatching, and assault incidents [1]. Reactive efforts, on the other hand, are insufficient to prevent violent events [2]. Technology has evolved into a critical component of public and national security in the modern era [3]. Closed-circuit television (CCTV) cameras-based surveillance and control system are used to monitor such incidents all over the world [4–6]; however, identifying the occurrences involves human personnel. This human-based continuous monitoring in surveillance camera systems is error-prone because it is not humanly possible to monitor the surveillance area throughout the day and night minutely [7]. Although human intervention helps detect anomalous activities, they can make errors, while monitoring for a long duration. Without automatic surveillance, there is a high probability that the system can make errors in detection. To minimize the errors, the surveillance system should be automated.

A large number of weapon detection algorithms can be found in the literature, which claim to detect weapons in

real-time surveillance environments; however, most of these systems fail to achieve desirable precision and accuracy [8]. In the past, most of the studies addressed the problem of weapon detection with machine learning classical methods applied over RGB images [4, 5]. Nowadays, region proposal network (RPN) based on deep learning models are widely regarded as the most practical detection models [9], as they improve accuracy, while reducing the computational cost. Faster R-CNN (region-based convolutional neural networks) [10] and R-FCN (region-based fully convolutional network) [11] are the most prominent CNN (convolutional neural network) models, which have outclassed the traditional machine learning-based detection methods in terms of accuracy as well as speed [12, 13]. The researchers [14–17] have worked on real-time weapon detection to reduce unlawful activities by applying Faster R-CNN on videos. However, existing researches are mostly conducted on data sets containing handheld weapon images downloaded from online databases, and video frames are processed with CNN models (Faster R-CNN and R-FCN) for detecting weapons [12, 14, 18, 19]. Moreover, region-based detection models are slow, whereas regression-based models have low accuracy. There is a need to develop such a system, which is faster in anomaly detection as well as have good accuracy and gives less false positives.

We designed a hybrid system for weapon carrier detection from live scenes for the types of weapons being used for crimes in Pakistan and worldwide, including pistols, revolvers, shotguns, and submachine guns. A robust weapon detection system is presented that quickly identifies the weapon carriers from video streams of surveillance cameras, which is an aid to proactive security measures. The hybrid method is applied to identify the weapons in real-time video frames, which includes a regression-based model containing YOLO V4, Yolo V4 tiny, or Yolo V4 CSP approach in step one, and it implements the region-based classification model such as faster-RCNN or R-FCN in step two. We have obtained improved results for weapon detection in CCTV cameras. The following is a list of this work's key contributions.

(i) A novel weapon detection framework based on Yolo v4 and R-FCN as a hybrid model is implemented.

(ii) A labeled dataset is generated for our problem in the context of available weapons. The rules are defined for labeling the problem-specific dataset, and suitable parameters are defined for generating the problem-specific dataset.

(iii) A comparative study is performed by training the data set with Yolo v4 invariants, R-FCN, Faster R-CNN, and a hybrid model. The findings for YoloV4 cum RFCN based hybrid system were 90 percent accuracy, 94 percent precision, 86 percent recall, and an 89 percent $F1$ score, demonstrating that it outperformed solitary models.

The rest of the paper is organized as follows. Section 2 describes the background, where different region-based classification and regression-based detection models are described, and related works are summarized. Section 3 describes materials and methods. Evaluations and results are given in Section 4 where we analyzed and discussed the performance of our proposed system. Finally, in Section 5, a conclusion has been stated.

2. Literature Review

This section summarizes the related region-based classification models, R-CNN, Fast R-CNN, Faster R-CNN, and R-FCN, and the related regression-based object detection models. Later, we discuss the related works.

2.1. Region-Based Classification Approaches. The first CNN-based detection model is the R-CNN [20]. In R-CNN, firstly an exterior boundary box generator generates 2,000 region ideas. After that, each of the region proposals is subjected to a VGG (visual geometry group in the University of Oxford)-based feature extractor [21]. CNN's final product is then input into two forecasters: a support vector machine (SVM) is a classifier program that predicts the class, while a linear regressor is a method to estimate the future and regresses the box. R-CNN performs reasonably well; however, due to a high number of computations of CNN, it is slow to perform real-time detection [22].

Fast-RCNN [23] is the successor of R-CNN. It demonstrates an improved performance by extracting features from the complete input image prior to producing region proposals. [22]. It changes the first predictor in R-CNN i.e., SVM, with the soft-max-based RoI (region of interest) pooling layer, where the features of a qualified region are converted into a compact feature map by RoI [24]. However, it uses the selective search for generating region proposals, which is the bottleneck and slow down the algorithm performance significantly. Other pitfalls of Fast R-CNN include multistage expensive training, which slows down the testing time [25].

The subsequent improvisation in this category is Faster R-CNN [10], where the R-CNN algorithm uses a faster neural network to replace the selective search algorithm. It trains a single CNN, which is then used for region proposals and classification [26]. Faster R-CNN brings in a region proposal network (RPN), which is efficient enough to generate RoIs in real-time [24, 27]. RPN is the key distinction between Fast R-CNN and Faster-RCNN [26]. In Faster R-CNN, RPN generates region proposals much faster than a selective search algorithm, and it shares the maximum computation with the detection network [12, 14]. However, the object-like regions and backgrounds get produced instead of object instances. Also, the algorithm is weak in dealing with objects having extreme scales or shapes.

Next improvement is R-FCN [11]. R-FCN creates nine number of region-based position-sensitive feature maps (top-left, top-middle, top-right, center-left,. . ., and bottom-right) for the input image. The pool of RoI with position sensitivity was then applied. It gives results comparable to the RoI pool used in the Fast R-CNN [28]. It determines position as well as objects' class by integrating voting

outcomes from various feature maps [29]. In R-FCN, the detection speed was improved over Faster R-CNN by single CNN sharing with maximizing the shared computation. The object classification and detection in an image are done with position-sensitive score maps [28]. However, it runs into specific problem sets due to its convolution property in model design and the relative position of an object class being represented by a position-sensitive score map [9, 13, 18].

2.2. Regression-Based Detection Approaches. Redmon et al. [30] created the first "You Only Look Once" (YOLO) model as a customized Darknet framework [31]. The Darknet, a comprehensive research platform, was developed in low-level languages. It has yielded some of the most effective real-time object detectors, such as YOLOv1, YOLOv2, YOLOv3, and YOLOv4, and the latest one is YOLOv5. Yolo models isolate a specific image into regions and visualize each region's confined-edge box and probability, as they are entirely dependent on CNN. At the same time, they anticipate various confined-edge boxes and their possibilities. The basic YOLO model has drawbacks, such as difficulty identifying small objects and objects with odd aspect ratios. Compared to the region-based competitor Fast R-CNN, it committed more localization errors. In 2017, YOLO v2 was introduced, which used anchor boxes to forecast the placement of objects in a picture. In convolutional layers, it performs normalization of the batch, and is a high-resolution classifier. After a year, numerous enhancements to YOLOv3 were made by adding Darknet-53 as the backbone network in place of Darknet-19 that was being used in YOLOv2.

YOLOv4 is the most recent stable version, which greatly outperforms prior approaches in terms of performance detection and speed. It is described by the working group [32] as a speedily operating detector that is trained and employed for fast object detection. YOLOv4 has a backbone of CSPDarknet53, a neck of Spatial Pyramid Pooling, a neck of PANet path-aggregation, and a head of YOLOv3. After it was built, Wang et al. [33] changed the structure of YOLOv4 to allow scaling for a range of applications. YOLOv4-tiny was created to maximize the speed and to minimize the computational costs. Then, YOLOv4-CSP and other larger variants of YOLOv4 were developed to enhance accuracy with changing computational needs.

2.3. Related Works. Deep learning has presented the latest techniques for fast detecting objects from live scenes. The models developed recently are providing promising results. Wu et al. [13] has given a comprehensive overview of recent improvements in object detection using deep learning. They carried out a thorough review of deep detection models and classified them into detection categories, learning methodologies and applications, and benchmarking-based evaluations. The element that influences the performance of a detection model is also highlighted.

CCTV cameras serve an important function in surveillance. Alexandrie [3] finds that CCTV cameras can minimize the crime rate in several aspects by capturing random events using CCTV cameras. Ashby [4] elaborated on the importance of CCTV cameras as a primary tool of investigation in the prevention of crime. Authors have analyzed 251195 records of crime cases registered by the British Transport Police, which happened from 2011 to 2015 on the British railway system. For the 45% cases, video evidence was available to the investigators, out of which 29% was successfully judged.

Olmos et al. [14] evaluated Faster-RCNN with feature extractor VGG16 for the detection of the gun in videos by using deep learning. They compared the selective window approach and region proposal network-based approaches and priorities of RPN-based detection. They also trained the model and configured it with the alarm system named alarm activation per interval where the alarm is activated in five successful true positives in between 0.2 seconds among 30 scenes. The results were promising, with zero false positives, 100% recall, 84.21% precision but with many true negatives.

Ren et al. [18] has given a brief comparative study of region proposal network (RPN) based models and expressed CNN architectures. It highlights the importance of data set design and deep convolutional networks, e.g., feature classifiers, and gives a novel approach to weapon detection using Faster RCNN using different feature extractors. Castillo et al. [34] introduced a guided brightness-based preprocessing for recognition of cold metal weapons in surveillance camera footage, employing a deep convolutional neural network. The goal was to use an automatic alert system to detect cold metal (steel) weapons in diverse lighting circumstances. Authors have analyzed different combinations of region selection techniques and CNN classifiers for their work. They employed R-FCN with ResNet 101, which gave 93% $F1$ measures.

Olmos et al. [14] developed a system "binocular image fusion" to outline a method for reducing false positives. It used a fusion method to target a classifier on the area of the scene, where suspicious activity is happening. The authors provide a dual-camera system for computing the disparity chart and content harvesting at a low cost to enhance the choice of eligible input frame regions. It was concluded that the presented approach has lessened the number of false-positive with improved performance for object detection.

Luo et al. [35] have taken the existing approach to backbone networks, which relied on pretrained models, and retrained them on fresh data in order to achieve an improved result for the new objective, mostly leading to weakening generalization resulting in overfitting. They presented a framework that included a more robust backbone network. They employed a twin backbone network structure encoder for better and more diverse feature extraction and evaluated their approach on six public datasets. The method used a backbone augmented network to evaluate relevant object detection.

Hashmi et al. [36] presented a comparative analysis of YOLOV3 and YOLOV4, the two versions of object detection algorithms, for the weapons detection task. The performance

of the presented work was estimated using precision, recall, quality, $F1$ Score, and mAP. It was demonstrated that the performance of YOLOV4 is superior to YOLOV3 for the weapons detection task in terms of all performance parameters.

3. Materials and Methods

The proposed system is a weapon detection model for national and public security, which helps in proactive security measures. This section provides detail about our method for weapon carrier detection through surveillance cameras. As given in Figure 1, the system works in four major steps, (a) data set generation for training, (b) training of the object detection models, (c) testing of the proactive surveillance CCTV system, and (d) deploying the model for real-time surveillance. The process starts with gathering a weapons dataset, which is then used for training of the deep learning models. Testing of the trained models is performed for hybrid combination, and the model is then used in real-time surveillance.

3.1. Data Set Generation and Annotations. We generated a dataset containing 1,444 images of locally available firearms. The captured images are of the weapons used by security guards, rescue officers, patrolling guards, policemen, gunmen, and different forces. We also visited multiple weapon showrooms in various cities of Pakistan to take snapshots of locally available weapons. Weapon images are also taken from the internet for model training. A weapon dataset containing 2986 images is also taken from Kaggle [37]. The total dataset of 4430 images is divided into a 70 : 30 ratio. The 70% of the images are annotated with labels and boundary boxes for the training of the RPN models. The under-given criteria are followed during data gathering.

(i) Multiple data sources are used for unbiasedness.

(ii) Same weapon images with different lighting effects and color saturation are captured.

(iii) Variety of weapons in one image is captured.

(iv) Snapshots are taken from different angles of handling a weapon. Different gripping styles are captured for the weapons.

(v) There is no redundancy of weapon images with angels and distance to avoid over-fitting.

The dataset is made up of many different weapon imageries collected from various sources and has several color schemes (like colored, grey-scale, and black and white) and pixel density. The snapshots contain different types of weapons, numbers, and different angles, and each image has at least one gun and may have a maximum number of weapons. In a preprocessing step, resolution of 1280×720 pixels is used to resize the images, and then, annotation is performed for weapons inside the photographs. The PASCAL VOC [38] standard is used to annotate the dataset as per the above-given norms. Sample images of locally available weapons are shown in Figure 2. Following are the norms we used for the annotation of the training data.

(i) Correct labeling of images is ensured.

(ii) Each weapon present in an image is separately labeled to avoid drawing a label box on multiple weapons.

(iii) Complete labeling of each weapon is ensured. The drawing of a label box on some parts of the weapon is prevented.

(iv) Multiple labels for one weapon in separate areas are avoided.

(v) Label within a label is also avoided.

3.2. Training of Deep Learning Models. The training data contain images of various weapons (most of them are locally available in Pakistan) that are labeled for supervised learning. A 70 percent of the data set was used to train the model, and it is generated by following a strict set of rules. We reformulated the object detection problem by training the best region proposal-based object detector Yolo4 and its variants. The performance is optimized by reducing the false positives through training and using the region-based classification models R-FCN. The hybrid combination of deep learning models including regression-based object detection models Yolo v4, Yolo v4 CSP, and Yolo v4 tiny, and region-based classification models Faster-RCNN and R-FCN are trained to identify the existence of a weapon carrying person at the place under surveillance.

3.2.1. Regression-Based Object Detection Steps

(1) The YoloV4 method divides the image into regions initially. The image is segmented into several grids, also known as residual blocks. Each grid is $S \times S$ in size, and the objects that exist within each grid cell will be detected separately.

(2) To forecast the objects' class, width, height, and center, the YOLO model uses regression with a single bounding box. Thus, it imagines a confined-edge box with probability for each region. Simultaneously, it calculates the probabilities of various confined-edge boxes and their classes.

(3) In object detection, the intersection over union or IoU is a notion that depicts how the boxes are overlapping. In YOLO, the IOU builds an output box to surround the items accurately. The bounding boxes and their confidence scores are estimated in each grid cell. If the expected and actual bounding boxes are similar, the IOU is assigned value of 1. This approach removes the bounding boxes that aren't the same in size as the real box. The final detection is made up of unique bounding boxes customized to the objects in concern.

Yolov4's specific constructs are listed as follows:

(i) The backbone network of YOLO-v4 for feature formation is typically pretrained on ImageNet classification and the weights are then adjusted in the new object detection task.

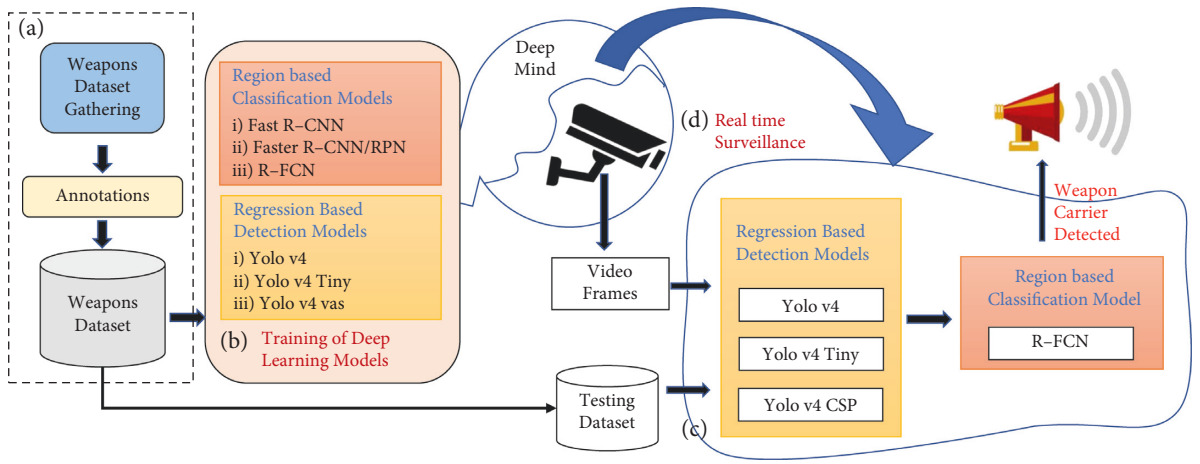

Figure 1: Proposed methodology for real-time weapon detection in surveillance CCTV system using a hybrid approach.

Figure 2: Sample images of our dataset with a variety of locally available weapons.

(ii) For feature aggregation, the features created in the ConvNet backbone are mixed and combined in the YOLOv4 Neck.

(iii) The YOLO-V4 head uses the same Yolov3 head to execute the anchor-based detection stages and three levels of granularity detection.

(iv) The majority of the freebies in Yolov4's "Bag of Freebies" are related to data augmentation, which increases network performance without increasing inference time in production.

(v) Yolov4 employs "Bag of Specials" tactics that add little inference time but considerably improve performance.

3.2.2. Region-Based Classification Steps.

(1) Feature Extraction: features extractor is CNN architecture that performs a set of experiments on the input dataset images for feature extraction [14]. We used VGG16 for feature extraction, trained on ImageNet containing a 1.28 million pictures dataset. In VGG16 "Conv5" layers [14, 15] are used with millions of parameters and RELU [39] is implemented to all convolutional layers. These are the convolution layers that are used for the prediction of region proposals as demonstrated in Figure 3.

We trained the VGG for minimization of loss prediction, as given in (1), defined in [11].

$$X(w) = \sum_{i=1}^{N} L\left(f\left(w; a^i\right)b^i\right) + \lambda R(w). \quad (1)$$

Here, a and b are the training input images used to iteratively reduce the average loss. "N" represents data instances number in iterations, "L" represents loss function, "X" represents predicted output, "w" represents current weights, "R" represents weight decay, and "λ" represents lag-range multiplier.

The feature extraction process results in a feature score map "M" provided as input to the next layer, the region proposal network.

(2) Region Proposal Network: we have trained RPN on our dataset. The RPN uses a feature map produced in the previous stage and generates proposals on objects regardless of whether it is a weapon or background. The process of RPN uses anchor boxes embedded on images during annotation and uses a CNN to construct the image's regions of interest.

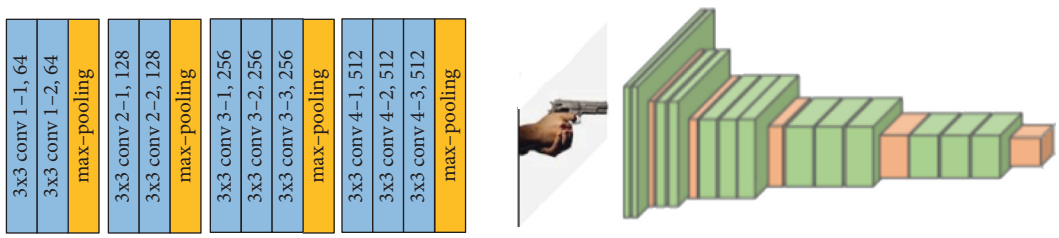

Figure 3: VGG-16 convolution layers.

(3) Position-Sensitive Score Map: from the feature map "M" containing the weapon, the position-sensitive score map is generated by dividing the region of interest into 3×3, 9 regions. It generates nine feature maps based on regions, each of which detects an object's top-right, top-middle, top-left, bottom-right, middle-middle, and so on. Then, it generalizes the regions into $k \times k$ RoIs, for which a score map of $k^2(C+1)$ is used. It receives feature maps to apply convolution and builds position-sensitive score maps with a depth of $k^2(C+1)$. For each ROI, the position-sensitive ROI creates a $K * K$ vote array. It uses soft-max to classify the objects after averaging the array.

(4) Pooling: pooling comes after the generation of the position-sensitive score map. It helps in reducing overfitting. In R-FCN average pooling is used instead of max pooling, whereas Faster-RCNN use max pooling. We employed an approach that pools the maximum value from the preceding layer's output, which is in the form of a matrix as given in the following equation:

$$\text{Max\_pooling}\left(i_{xy}\right) = \max\left(i_{11}, i_{12}, i_{13}, \ldots, i_{nm}\right). \tag{2}$$

(5) Classification and boundary box regression: after pooling is done, we applied soft-max layer [40], which gives the probability of the class with a class and boundary box regression. It is done to precise the boundary box drawn on the object. $k^2(C+1)$ feature maps are created for classification [9], and the same procedure is done for boundary box regression. $4K^2$ maps are used from the same score maps and applied position-based ROI pooling to compute the k^2 array of elements containing the boundary box, and the final prediction is made by taking a maximum of all these elements.

(6) The effectiveness of a classification method/model with an output probability value ranging from 0 to 1 is measured using cross-entropy loss, often referred to as log loss. [41]. A model with a log loss of 0 is ideal. Cross-entropy calculated our binary classification by the following equation:

$$-(B \log(P) + (1-B)\log(1-P)). \tag{3}$$

Here, B is the binary indicator, P represents predicted probability, and log is the natural log.

(7) For model optimization utilizing a minibatch, a stochastic gradient descent optimizer [42] is used. The SGD optimizer adjusts the parameters based on the data $(a)^i$ and label $(b)^i$ as given in the following equation:

$$\theta = \theta - \eta.\nabla\theta J\left(\theta; (a)^i; (b)^i\right). \tag{4}$$

As given above, instead of using a single sample, minibatches are employed to improve the parameters, so the parameters are optimized using the following equation:

$$\theta = \theta - \eta.\nabla\theta J\left(\theta; (a)^{i:\, i+n}; (b)^{i:\, i+n}\right). \tag{5}$$

The SGD optimizer traverses to the overall minimum loss by progressing to where the loss lessens. However, because there is no next point when the loss reaches a local depth, the SGD gets trapped in the local minima. The solution to this problem is momentum, which causes the SGD to accelerate in the appropriate direction, as shown in the following equation:

$$\theta = \gamma u_{t-1} + \eta.\nabla\theta J\theta, \tag{6}$$

where θ represents weights, η is learning rate, $\nabla\theta J$ is gradient, u_t is the updated weights, and γ represents momentum and in our case it is 0.9.

Following the loss computation, the parameters are optimized through the optimization function. Our training data was incredibly vast, and loading it all at once required a lot of memory. The data are split into minibatches to optimize the model's attributes as a solution to this problem.

4. Deep Learning Models Testing and Real-Time Surveillance

This research employs a strategy that uses Yolo-V4 models to detect threats in real-time from video frames. The anomalies are then reported to the region-based classifiers for confirmation of the threat. The step-wise methodological flow of the proposed threat detection model is described below.

(i) The video frames module extracts frames from video received through a CCTV camera used as input to the frame extractor.

(ii) Extracted video frames are passed to the regression-based detection models. The system is configured for YoloV4, YoloV4Tiny, or YoloV4CSP, as per the

processing device's deep learning accelerator availability or speed.

(iii) A frame detected for a "weapon carrier" anomaly is forwarded to the R-FCN running at the edge computer.

(iv) R-FCN uses Feature Extractor to generate feature map, applies region proposal network to produce RoI(s), and a position-sensitive score map is generated through convolution. Then pooling is to produce prominent features. The classification and boundary box regression predict classes of weapons or not weapons.

(v) In the last step, an alarm is generated if a weapon carrier is detected by YoloV4 first and then confirmed by R-FCN.

4.1. Training Results for Yolo V4. There are multiple variations available for yolov4 in the open-source market. We implemented popular variations including "yolov4 original," "yolov4 CSP," and "yolov4 tiny" to determine which one performs better for proactive surveillance under CCTV cameras. The dataset images are set with bounding boxes as text files. The three variations of YoloV4 models are trained on the same weapon dataset of 4430 images split under 70%–30%. We have used google provided free notebooks for the training purpose of our algorithms named "Google Collab." YOLOv4 is completely dependent on the "graphical processing unit" (GPU). GPU is mainly used for gaming purposes, but due to the high rate of number-crunching, these can be used for the training of deep learning algorithms. Thanks to "Nvidia" for providing CUDA cores in their GPUs that can be used to run algorithms faster. We used Google collab daily resource usage limit on a free subscription with one of the top-end GPU "tesla v4 (32 GB)," 12 GB of RAM, and 70 GB of cloud storage. We used YOLO's "darknet" framework. For vanilla yolov4 loss dropped to 0.6 with a training precision of 90% after 3000 iterations, whereas, loss in yolov4-CSP dropped to 6.24 with a training precision of 76% after 3000 iterations, and finally yolov4-tiny dropped the loss to 13.6 with 79% training precision after 3000 iteration. Other results for Recall, $F1$-score, Mean Avg precision (@50%), and Avg. Intersection over Union(@50%) are given in Table 1.

Based on the above results, this can be inferred that "yolov4-tiny" is good for a quick training process with a slight loss in performance for prediction, but is a quite light model for starter projects. In comparison, "yolov4-CSP" is better than yolov4-tiny, but takes more time for training. The reason for Yolov4-CSP to take more time is, that it has a more complex structure than yolov4-tiny. Base yolov4 took the longest time for training due to the complex structure but with the best performance.

4.2. Training Results for R-FCN. For region-based classification evaluations, we trained faster R-CNN and R-FCN models over the 70% of the weapon dataset, following the steps involved in a model's architecture. We took an image as input and passed it to the ConvNet, which returned the feature map for that picture, resulting in a faster RCNN. The feature maps were subjected to region proposal networks. The object proposals are returned, together with their objective score. To bring all of the recommendations down to the same size, the RoI pooling layer was applied to the region proposals. Finally, to categorize and produce the bounding boxes for objects, the proposals were given to a fully connected layer containing a softmax layer and a linear regression layer at the top. For R-FCN model training, an image's feature map produced through CNN was used by RPN to generate the position-sensitive score map.

We determined the accuracy, loss, class loss, and box loss for the two models used in our experiments to show the improvement during training. The training accuracy and losses are tested for 5000 iteration data values divided into five sets of 1000 chunks. It is accessed that the accuracy during the training process is improved from Faster-RCNN to RFCN. And the loss, class loss, and box loss during the training process are lessened from Faster-RCNN to RFCN. The testing outcome of our R-FCN based system was 91%, while the confusion matrix shows the results as follows:

(i) Accuracy = 91.17%,

(ii) Precision = 93.63%,

(iii) Recall = 88.03%,

(iv) $F1$ score = 91.35%.

4.3. Evaluation of YOLO V4 cum R-FCN-Based Hybrid Setup. We presented the plots for accuracy and loss of the proposed hybrid model in Figure 4 to show the convergence in 5000 epochs. The graph in 4(a) presents the accuracy achieved during the training and testing comparatively, the training accuracy is represented by the blue line, while the test accuracy is represented by the orange line. Accuracy convergence is shown up to 5000 epochs. The graph in Figure 4(b) depicts the comparative loss calculated during the training and testing. The training loss is given by the blue line, whereas, the test loss is shown by the orange line. It depicts the convergence of loss to a minimum of upto 5000 epochs. To forecast about the test data instances of the dataset, an assessment is performed about either the image frame is containing a weapon (positive) or not containing a weapon (negative). The following are the four basic constructs determined for this prediction:

(i) True positives representing the correct positive predictions about weapon carriers,

(ii) False positives representing the incorrect positive predictions, guessing something else as a weapon,

(iii) True negatives are correct negative predictions, classifying a nonweapon carrier to the negative class,

(iv) False negatives are Incorrect negative predictions that miss-classifies a weapon-carrier as a noncarrier.

Table 1: Yolo results table.

	Epochs	Precision	Recall	$F1$ score	Mean avg precision (@50%)	Avg. intersection over union (@50%)
Yolo-v4	1000	75	81	80	85.3	56.82
(Loss 0.6)	2000	85	82	86	88.13	67.78
	3000	90	84	87	90.22	71.18
Yolo-v4 (csp)	1000	74	75	74	73.85	54.26
Loss (6.24)	2000	79	86	82	86.64	62.75
	3000	76	88	81	87.69	61.22
Yolo-v4 (tiny)	1000	56	54	55	54.56	37.59
(Loss 13.6)	2000	82	75	79	82.14	62.02
	3000	79	81	80	85.01	60.1

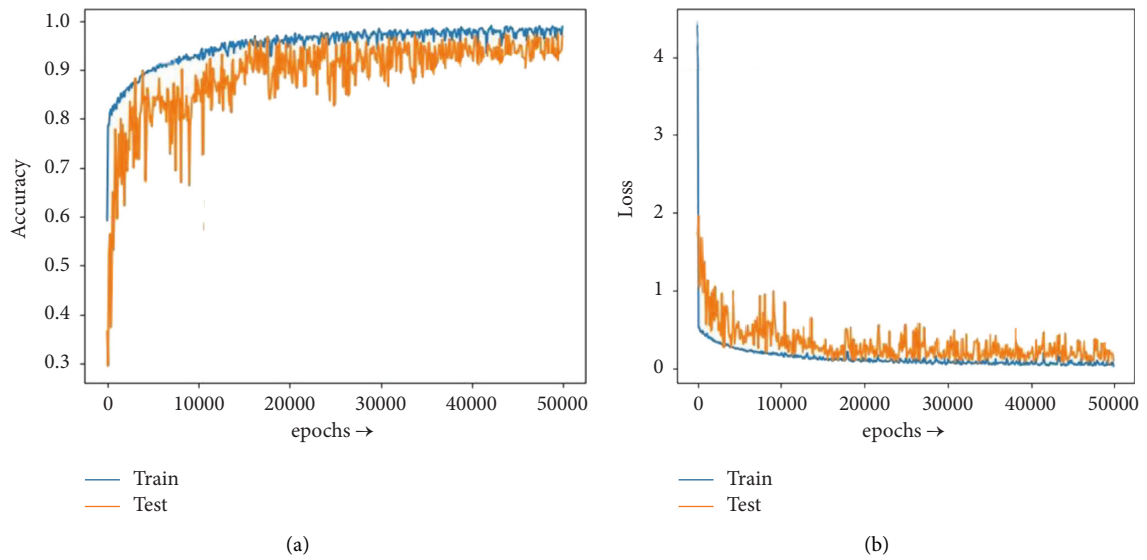

Figure 4: Hybrid model training. (a) Hybrid model accuracy graph. (b) Hybrid model loss graph.

For classification of "weapon carrier" or "not weapon carrier" classes, Table 2 presents the results of numerous evaluation matrices for binary classifiers, including Faster-RCNN, R-FCN, YoloV4, and Hybrid Method. These models are compared for accuracy, recall, precision, and $F1$ score. We have tested the models on our dataset to make an unbiased comparison, presented in Table 2. In hybrid mode, results given under case, (i) are for Yolo v4 whereas case and (ii) denotes the activation of RFCN after Yolo V4 detected a weapon to recheck for any false positives.

The testing results given in Tables 1 and 2 showed that the Yolo-based regression models are faster in speed than the region-based classification models. However, the accuracy of weapon detection is better for the region-based classification models. In regression-based models, Yolo-v4-Tiny is the faster in detection speed, Yolo v4 CSP is at the second number in speed, and then comes the number of original Yolo v4. The training time for each of these models is proportional to their speed rankings. Nevertheless, the accuracy of weapon detection improves in reverse order, i.e., Yolo tiny is at the lowest accuracy, Yolo CSP is at second, and the original Yolo v4 is the best out of these three in terms of accuracy. The speed and accuracy of classification-based models improve as we advance from RCNN to fast-RCNN to faster-RCNN to RFCN. Comparing the speed and accuracy of regression-based models with classification-based models shows that the former is better in speed, whereas, the latter is better in accuracy. This work then applied a hybrid method that uses Yolo-V4 models for live detection of threats from video frames. If an anomaly is observed, it is sent to the region-based classification to confirm it as a true or false positives threat. The proposed method achieves better accuracy and speed than the individual categories of deep learning models.

4.4. Proactive Surveillance through Live Detection and Alarm Generation. To test proactive surveillance, as depicted in Figure 5, surveillance camera systems were installed in public places, and threat situations were explicitly simulated, where one or more men with a weapon in hand were introduced in the vicinity of the camera. In our experimental setup, surveillance cameras were set up with a Raspbian Jessie OS installed Raspberry Pi 3B+, and it was booted from a MicroSD card 64G. It worked as a module of intelligence attached to the camera. The models were trained on the TensorFlow framework of deep learning with the resources listed in Table 3. It takes almost 4–5 hours for each model to

Table 2: Results comparison of all trained models.

Evaluation matrices	Faster RCNN	R-FCN	YoloV4	Hybrid method
True positive	551	581	576	586
False negative	99	77	105	98
False positive	89	33	65	35
True negative	590	638	593	610
Accuracy	0.85	0.91	0.87	0.90
Precision	0.86	0.93	0.90	0.94
Recall	0.84	0.88	0.84	0.86
*F*1 score	0.85	0.91	0.87	0.89
Avg frames/sec	5–17	7	55	(i) 55 -> (ii) 31

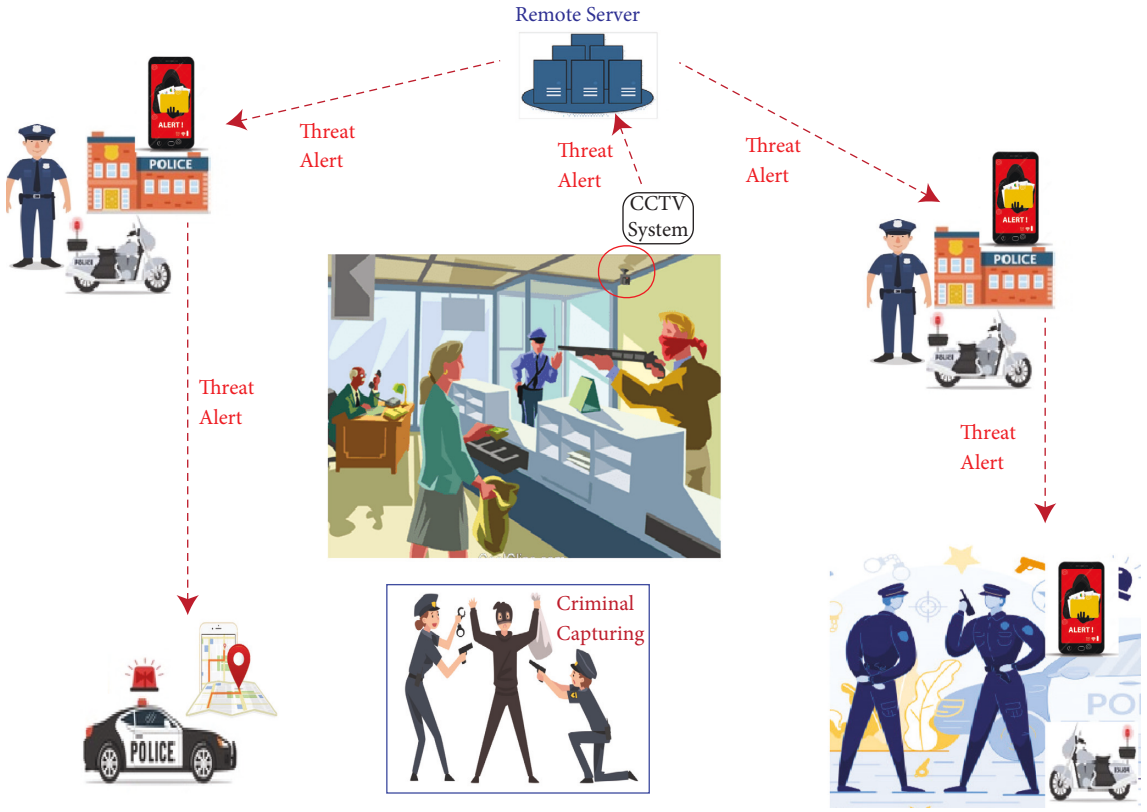

Figure 5: Surveillance system working procedure.

fine-tune. The training is performed till the convergence of the learning rate to $1 \times e^{-3}$, where the loss is measured using SGD optimization with momentum, and cross-entropy is applied to calculate the loss. In a threat situation, the intruder showed off his weapon. The video was captured at 25 frames per second. The video clips were classified as frames. The "Video Frames" module extracted frames from video coming from CCTV cameras to produce Test Data, which were used to evaluate our proposed hybrid model. A threat was detected from the live scenes in real-time, and the surveillance system-generated alarm at the surveillance computer at the security office, as depicted in Figure 5. The detected threats were also stored in a common database, later used for comparative analysis of results. In our system, the person's normal behavior was labeled as "no threat" while the abnormal behavior was labeled as "weapon threat." The proposed hybrid method tries to increase accuracy, reduce execution time and computational cost. In weapon carrier detection, false positive minimization is also an important factor. The proposed method achieved an accuracy of up to 90% with low false positives for images containing weapons. This is achieved with Yolo v4 cum R-FCN models, which has given promising results.

Table 3: Resources used for region based classification models training.

Resource	Detail
Operating system	UBANTU 16.04
Central processing unit	Intel Xeon E5640
Graphical processing unit	Nvidia 1070 Ti GPU 8 GB memory
RAM	16 GB

Example outputs for real-time weapon detection in a surveillance camera are shown in Figure 6. A boxed area

Figure 6: Hybrid setup based proactive surveillance outcomes.

with a label shows the percentage of confidence about whether a detected object is a weapon or not a weapon. When a weapon carrying individual appeared in front of a CCTV camera, the system immediately recognized the threat and raised an alarm, allowing preemptive action to be taken.

5. Conclusion

This research trained the deep learning model for proactive surveillance under CCTV systems. We prepared a weapon data set from multiple resources and annotated the images for bounding boxes on weapons in the view. The dataset is split into a 70–30% ratio for training and testing purposes. The deep learning models of regression and classification categories are trained using the collected weapons dataset for proactive surveillance against handheld weapons. We trained Yolo v4, Yolo v4 tiny, and Yolo v4 CSP models from the regression category, whereas, from the classification category, the deep learning models trained are RCNN, Fast-RCNN, Faster- RCNN, and RFCN.Yolo v4 and R-FCN. A comparative study has been done for accuracy and speed of regression-based object detection and region-based classification models, and a CCTV cameras based weapon detection system has been developed under the hybrid approach. The models were trained in a deep learning accelerator, and a comparative study has performed under edge computing. In comparing YOLOv4, Yolov4-CSP, and YOLOv4-tiny, the latter is better for real-time object detection having a faster inference time; however, Yolov4 is better in precision and accuracy for real-time object detection scenarios. To improve the accuracy of the presented system, RFCN based trained model is added before the alarm generating stage to decrease the false positive rate. Experiments have shown that the proposed hybrid method obtained relatively good results for proactive surveillance.

Data Availability

Data are available on request from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] P. P. Vuma, "Measuring the ability of the police to prevent crime: could this assist in stressing the importance of crime prevention?" *Acta Criminologica: Southern African Journal of Criminology*, vol. 29, no. 1, pp. 98–112, 2016.

[2] F. D. Azumah, J. O. Nachinaab, C. D. Sintim, and S. Krampah, "Crime analysis and conventional policing strategies: evidence from a community in the western region, Ghana," *International Journal of Social Science Studies*, vol. 7, no. 4, p. 1, 2019.

[3] G. Alexandrie, "Surveillance cameras and crime: a review of randomized and natural experiments," *Journal of Scandinavian Studies in Criminology and Crime Prevention*, vol. 18, no. 2, pp. 210–222, 2017.

[4] M. P. J. Ashby, "The value of cctv surveillance cameras as an investigative tool: an empirical analysis," *European Journal on Criminal Policy and Research*, vol. 23, no. 3, pp. 441–459, 2017.

[5] S. Cosar, G. Donatiello, V. Bogorny, C. Garate, L. O. Alvares, and F. Bremond, "Toward abnormal trajectory and event detection in video surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 683–695, 2017.

[6] J. Park, D. H. Kim, Y. S. Shin, and S.-h. Lee, "A comparison of convolutional object detectors for real-time drone tracking using a ptz camera," in *Proceedings of the 2017 17th International Conference on Control, Automation and Systems (ICCAS)*, pp. 696–699, IEEE, Jeju, Korea (South), October 2017.

[7] K. Mehtre and B. M. Mehtre, "Automated camera sabotage detection for enhancing video surveillance systems," *Multimedia Tools and Applications*, vol. 78, no. 5, pp. 5819–5841, 2019.

[8] G. Mathur and M. Bundele, "Research on intelligent video surveillance techniques for suspicious activity detection critical review," in *Proceedings of the 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1–8, IEEE, Jaipur, India, December 2016.

[9] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," pp. 91–99, 2015, https://arxiv.org/abs/1506.01497.

[11] J. Dai, Y. Li, K. He, J. Sun, and R-fcn, "Object detection via region-based fully convolutional networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 379–387, Barcelona, Spain, December 2016.

[12] H. Buckchash and B. Raman, "A robust object detector: application to detection of visual knives," in *Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 633–638, IEEE, Hong Kong, China, July 2017.

[13] X. Wu, D. Sahoo, and S. C. Hoi, "Recent Advances in Deep Learning for Object Detection," *Neurocomputing*, vol. 396, 2020.

[14] R. Olmos, S. Tabik, and F. Herrera, "Automatic handgun detection alarm in videos using deep learning," *Neurocomputing*, vol. 275, pp. 66–72, 2018.

[15] R. K. Verma and G. 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.

[16] G. K. Verma and A. Dhillon, "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, Allahabad, India, November 2017.

[17] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, and T. Huang, "Revisiting rcnn: on awakening the classification power of faster rcnn," in *Proceedings of the European Conference on Computer Vision*, pp. 453–468, ECCV), Munich, Germany, September 2018.

[18] Y. Ren, C. Zhu, and S. Xiao, "Object detection based on fast/faster rcnn employing fully convolutional architectures," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–7, Article ID 3598316, 2018.

[19] L. Liu, W. Ouyang, X. Wang et al., "Deep learning for generic object detection: a survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.

[20] 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, Columbus, OH, USA, June 2014.

[21] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014, https://arxiv.org/abs/1409.1556.

[22] R. Gandhi, R. Cnn, and Fast R. Cnn, "Faster R-Cnn, yolo — Object Detection Algorithms," July 2018, https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.

[23] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, Washington, DC, USA, 2015.

[24] J. Hui, "What do we learn from region based object detectors (faster r-cnn, r-fcn, fpn)? https://medium.com/@jonathan_hui/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9.

[25] S. Ananth, "Fast R-Cnn for Object Detection. A Technical Summary," https://towardsdatascience.com/fast-r-cnn-for-object-detection-a-technical-summary-a0ff94faa022.

[26] M.-C. Roh and J.-y. Lee, "Refining faster-rcnn for accurate object detection," in *Proceedings of the 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pp. 514–517, IEEE, Nagoya, Japan, May 2017.

[27] S. M. Abbas and D. S. N. Singh, "Region-based object detection and classification using faster r-CNN," in *Proceedings of the 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT)*, February 2018.

[28] A. F. Agarap, "Deep Learning Using Rectified Linear Units (Relu)," 2018, https://arxiv.org/abs/1803.08375.

[29] J. Hui, "Understanding Region-Based Fully Convolutional Networks (R-fcn) for Object Detection," 2016, https://arxiv.org/abs/1605.06409.

[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, Las Vegas, NV, USA, June 2016.

[31] J. Redmon, "Darknet: Open Source Neural Networks in C," 2016, https://pjreddie.com/darknet/.

[32] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal Speed and Accuracy of Object Detection," 2020, https://arxiv.org/abs/2004.10934.

[33] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: scaling cross stage partial network," in *Proceedings of the IEEE/cvf Conference on Computer Vision and Pattern Recognition*, pp. 13 029–113 038, Nashville, TN, USA, June 2021.

[34] A. Castillo, S. Tabik, F. Pérez, R. Olmos, and F. Herrera, "Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning," *Neurocomputing*, vol. 330, pp. 151–161, 2019.

[35] R. Luo, H. Huang, and W. Wu, "Salient object detection based on backbone enhanced network," *Image and Vision Computing*, vol. 95, Article ID 103876, 2020.

[36] T. S. S. Hashmi, N. U. Haq, M. M. Fraz, and M. Shahzad, "Application of deep learning for weapons detection in surveillance videos," in *Proceedings of the 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, pp. 1–6, IEEE, Islamabad, Pakistan, May 2021.

[37] U. of Grenada, "Handgun Detection Kaggle," https://www.kaggle.com/andrewmvd/handgun-detection.

[38] H. Kim, H. Kim, Y. W. Hong, and H. Byun, "Detecting construction equipment using a region-based fully convolutional network and transfer learning," *Journal of Computing in Civil Engineering*, vol. 32, no. 2, Article ID 04017082, 2018.

[39] E. Granger, M. Kiran, L.-A. Blais-Morin, and L. T. Nguyen Meidine, "A comparison of cnn-based face and head detectors for real-time video surveillance applications," in *Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–7, IEEE, Montreal, Canada, November 2017.

[40] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," *ICML*, vol. 2, no. 3, p. 7, 2016.

[41] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 8778–8788, Montréal, Canada, December 2018.

[42] J. Duda, "Sgd Momentum Optimizer with Step Estimation by Online Parabola Model," 2019, https://arxiv.org/abs/1907.07063.