

# Image and video-based crime prediction using object detection and deep learning

Mohammed Boukabous, Mostafa Azizi

Mathematics, Signal and Image Processing, and Computing Research Laboratory (MATSI), Superior School of Technology (ESTO), Mohammed First University, Oujda, Morocco

## Article Info

### Article history:

Received Nov 6, 2022

Revised Dec 16, 2022

Accepted Jan 2, 2023

### Keywords:

Crime

Deep learning

Object detection

Security intelligence

YOLO

## ABSTRACT

In recent years, the use of artificial intelligence (AI) for image and video-based crime detection has gained significant attention from law enforcement agencies and security experts. Indeed, deep learning (DL) models can learn complex patterns from data and help law enforcement agencies save time and resources by automatically identifying and tracking potential criminals. This contributes to make deep investigations and better steer their targets' searches. Among others, handheld firearms and bladed weapons are the most frequent objects encountered at crime scenes. In this paper, we propose a DL-based surveillance system that can detect the presence of tracked objects, such as handheld firearms and bladed weapons, as well as may proceed to alert authorities regarding eventual threats before an incident occurs. After making a comparison of different DL-based object detection techniques, such as you only look once (YOLO), single shot multibox detector (SSD), or faster region-based convolutional neural networks (R-CNN), YOLO achieves the optimal balance of mean average precision (mAP) and inference speed for real-time prediction. Thus, we retain YOLOv5 for the implementation of our solution.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Mohammed Boukabous

Mathematics, Signal and Image Processing, and Computing Research Laboratory (MATSI)

Higher School of Technology (ESTO), Mohammed First University

Oujda 60000, Morocco

Email: m.boukabous@ump.ac.ma

## 1. INTRODUCTION

Social media has profoundly transformed the way information is currently exchanged and used up, and has become an integral operations' element of government agencies and businesses. Social media has enabled users to exchange materials and thoughts without relying on conventional and centralized news channels. This perhaps results in a more democratic distribution of viewpoints by allowing individuals to reach a broad segment of the population [1], [2]. It is true to assume that the coronavirus pandemic had a significant impact on the way people utilize social media, leading to 3.78 billion social media users worldwide in 2021, and more than half of the world's population is expected to be on social media over the next five years, according to the most recent social media data [3].

Traditionally, monitoring social media exchanged messages requires security officers to visually detect the presence of security threats (such as the presence of weapons) by observing exchanged images and video messages on social media websites and making quick judgments based on them. In this study, we overcome the limitations of manual analysis by processing the exchanged images or video streams using deep learning (DL) object detection algorithms to automate threat detection. Naturally, videos are sequences of images.

The majority of state-of-the-art image classification systems employ different types of DL approaches, convolutional neural network (CNN) based object detection systems have been applied to a variety of image processing applications recently [4], [5]. That is why there is a high demand for building efficient CNN architectures capable of predicting crimes [6]. There are several architectures that are both computationally efficient and accurate: the region-based convolutional neural networks (R-CNN) variants [7]–[9], single shot multibox detector (SSD) [10], and you only look once (YOLO) [11].

The rest of this paper is structured as follows. Section 1 as introduction, section 2, we recall some backgrounds of our issue and discuss relevant related works. Our proposal method is presented in section 3. Experiments and obtained results are described in section 4. Finally, section 5 concludes this work and gives some perspectives.

## 2. BACKGROUND

To develop an effective crime prediction model, various data inputs must be considered, analyzed, and categorized. Meanwhile, analyzing a big volume of data is a difficult task, and extracting knowledge from them is another challenging job. Even if it appears impossible to accurately predict every type of crime, we can at least try to do so using available datasets [12].

Object detection is a computer vision technology that identifies and locates objects in a digital image or video [13]. Algorithms such as R-CNN, YOLO, SSD, and others have been designed to discover these instances quickly. Face recognition and pedestrian detection tasks are two well-studied object detection issues. Object detection has a wide range of applications in computer vision, including image retrieval and video surveillance.

### 2.1. Convolutional neural network architectures

In the past few years, CNNs have shown impressive results in the fields of image processing and object detection. A CNN is a type of neural network specifically designed for image recognition [14], [15]. It is composed of several layers, each of which performs a specific task. The first layer (the input layer) takes in an image as input. The next layer (the convolution layer) convolves the input image with a set of filters, which are small images themselves, this layer extracts features from the input image. The next layer (the pooling layer) downsizes the output of the previous layer by taking a fixed-size square from the input image and pooling together all the pixels in the square. This layer reduces the number of parameters and makes the network more tolerant of errors. The next layer (the fully connected layer) takes the output of the previous layer and feeds it into a number of neurons, one for each object that the network is trying to detect. The final layer (the output layer) outputs the class of the object that was detected, along with the coordinates of the object's center [16], [17].

A CNN can be trained to detect a wide variety of objects, including people, cars, animals, and even individual letters of the alphabet. The network is first trained on a large dataset of images that contain the objects to detect [18]. The network learns to identify the features associated with each object and to associate a specific class with each of these features [19]. After the network has been trained, it can be applied to new images to detect the presence of the desired objects. CNNs have shown impressive results in the field of object detection. It can detect a wide variety of objects, are tolerant to errors, and can learn the features associated with each object. This makes it a very promising technology for use in a wide range of applications, such as self-driving cars [20], [21], facial recognition [22], crime detection [23], internet of things-based photovoltaics monitoring [24]–[26], and even the detection of COVID-19 [27].

Backbones are the weights or parameters that are used to generate the feature map. In the context of object detection, the backbone is the component of the feature extractor that is responsible for generating the features that will be used to detect objects. Several backbones can be used for object detection, each with its own set of advantages and disadvantages as shown in Figure 1. The most known DL CNNs backbones used in object detection algorithms are visual geometry group (VGG), residual neural network (ResNet), inception, and DarkNet. Each of these architectures has different trade-offs in terms of speed, efficiency, and accuracy.

- VGG is a CNN model originally developed by the VGG at the University of Oxford and was released as part of the Caffe DL framework [28]. The VGG16 object detection model is composed of 16 layers, while the VGG19 model has 19 layers. VGG can be used as the feature extractor backbone in algorithms such as fast R-CNN, faster R-CNN, and SSD.
- ResNet is an object detection network that is composed of a large number of layers. The network is designed to enable the accurate detection of objects in a wide variety of scenes. The network can detect objects even when they are partially hidden or when they are in challenging environments, such as outdoors or in low light conditions [29]. ResNet can be used as the feature extractor backbone in both faster R-CNN, and SSD algorithms.

- Inception is a CNN model that was developed by Google and it is considered to be one of the most accurate architectures for object detection. It is composed of a deep network of 22 layers (27 with the pooling layers) [30]. Inception can be used as the feature extractor backbone in the faster R-CNN algorithm.
- DarkNet is a highly effective open-source object detection tool that has been fully implemented in a variety of practical applications, including autonomous driving, medical image analysis, and video surveillance. It is an extremely effective method for detecting small objects in images, and it can also track objects as they move through a scene, making it a valuable tool for automated surveillance [31]. DarkNet is used in YOLO.

Many popular object detection algorithms, such as the R-CNN variants, SSD, and YOLO, rely on these algorithms as the feature extractors' backbone. There are generally two types of object detection algorithms: one-stage and two-stage. One-stage algorithms try to find objects in an image in one go, while two-stage algorithms divide the task of object detection into two parts. In the first part, a classifier is used to find possible locations of objects in an image. In the second part, a region proposal algorithm is used to identify the most likely locations of objects in the image as shown in Figure 1. One-stage object detection algorithms are faster but less accurate, while two-stage object detection algorithms are slower but more accurate. In general, one-stage object detection algorithms are used for real-time applications where accuracy is not as important as inference speed, while two-stage object detection algorithms are used for more critical applications where accuracy is more important.

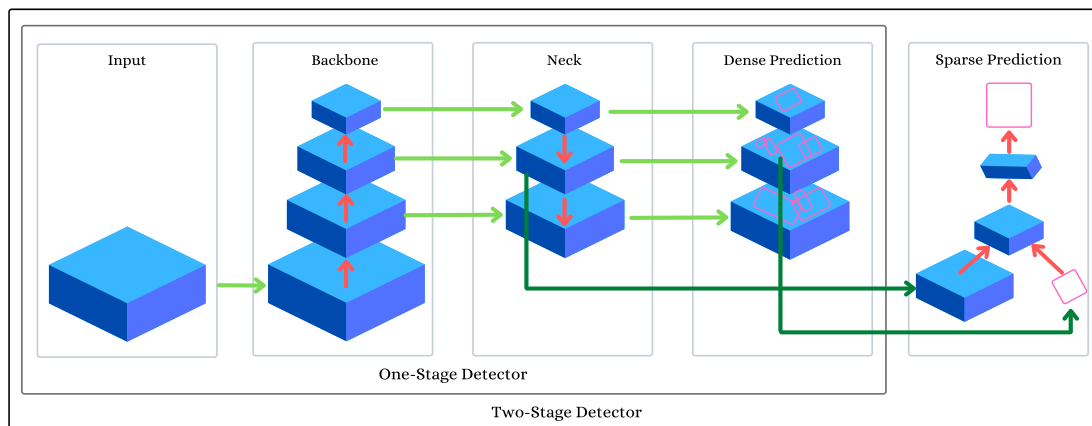


Figure 1. Object detection models architectures

Object detection architectures typically consist of five major components: the input, the backbone, the neck, the dense, and the sparse. The input refers to the image that is being fed into the network. The backbone is the main part of the network that is responsible for extracting features from the input image. The neck is a smaller part of the network that aggregates the extracted features from the backbone. The dense is a fully connected layer that takes the features from the neck and outputs the final object detection results. The sparse is a sparsely connected layer that takes the features from the neck and outputs the final object detection results.

## 2.2. Single shot multibox detector

When it comes to object detection, the SSD algorithm developed by Liu *et al.* [10] has set new benchmarks for speed and accuracy, achieving more than 74% mean average precision (mAP) at 59 frames per second (FPS) on datasets like Pascal visual object classes (VOC) and Microsoft common objects in context (COCO). The backbone model and the head are the two parts of the SSD. SSD head adds a new set of convolutional layers on top of this one, with the results interpreted as the bounding boxes and classes of objects located where the SSD head is activated. SSD divides the image into grids instead of utilizing a typical sliding window technique, and each grid cell is in charge of recognizing objects in that section of the image. The output will be null (0) if no object was found.

## 2.3. You only look once

This is an algorithm for identifying and classifying distinct objects in an image (in real-time). YOLO performs object detection as a regression problem and returns the class probabilities for the detected objects.

This method detects objects by performing a single forward propagation across a neural network. This implies that the complete image is predicted in a single algorithm run. The CNN is used here to simultaneously predict multiple class probabilities and bounding boxes [11].

YOLO enhances the detection speed by predicting objects in real-time, providing accurate results with minimal background errors. However, the YOLO model predicts only two boxes per grid, making it more difficult to detect small objects that appear in groups [32]. It also struggles to generalize to objects with novel or unusual dimensions.

#### 2.4. Region-based convolutional neural networks variants

Research by Girshick *et al.* [7] developed an approach called R-CNN where they utilize selective search to pick just 2,000 regions from the image and named them region proposal. Therefore, rather than attempting to categorize a large number of regions, we focus only on these 2,000 regions. The author of the earlier paper (R-CNN) fixed some of R-CNN shortcomings to design a quicker object detection system called fast R-CNN [8]. It is similar to the R-CNN model, except that instead of feeding the CNN by regions' suggestions, the input image is used to build a convolutional features' map.

Both R-CNN and fast R-CNN employ a selective search to get the region's proposals. As a result, the network's performance suffers while performing a search using the selective search algorithm. Therefore, research by Ren *et al.* [9] designed an object detection method called faster R-CNN that does away with the selective search algorithm and instead allows the network to learn the region proposals.

#### 2.5. Related works

Research by Grega *et al.* [33] proposed a system for detecting knives and firearms in closed circuit television systems (CCTV) footage using MPEG-7 (multimedia content description standard) and principal component analysis (PCA) with a sliding window technique, they achieved 96.69% of specificity and 35.98% of sensitivity in firearm detection, and on other side, 94.93% of specificity and 81.18% of sensitivity in knife detection. Research by Olmos *et al.* [34] employed a sliding window and region proposal approach to detect firearms in real-time. The region suggestion technique yielded the best results. The sliding window approach took 14 s/image, but the region proposal method took 140 ms/image with 7 FPS. Research by Iqbal *et al.* [35] suggested a method for object detection that is aware of its orientation. This technique is more suited for long and thin objects such as rifles and other such items.

According to Verma and Dhillon [36] employed faster R-CNN to detect firearms. The work was conducted using a dataset called internet movie firearms database (IMFDB). They achieved an accuracy of 93.1% for firearm detection. Research by Mehta *et al.* [37] used the same dataset (IMFDB) to develop a DL model based on the YOLOv3 algorithm in which they process videos frame-by-frame in real-time to detect anomalies such as gun violence, mass shootings, home fires, industrial explosions, and wildfires. They achieve a detection rate of 45 FPS, and their final model had a validation loss of 0.2864.

### 3. METHOD

Nowadays, it is a hard task to detect crime objects in images and videos. With the rapid development of society and technology, more and more images and videos are exchanged every day. Human beings cannot check all of them. So, we need machines to do so. Our proposed method is to develop an algorithm that can be used to detect objects related to crime, such as guns, knives, and other weapons. By using this algorithm, law enforcement agencies can quickly and accurately identify objects that may be related to criminal activities. This may help to prevent crime and keep public safety. One way is that it can be used by police officers to help them find evidence at crime scenes. This can be particularly helpful in cases where there are multiple weapons present. It can also be used by security guards to help them identify potential threats in real-time or even prevent crimes before their occurrence. Our proposed method is divided into five steps as follows (as shown in Figure 2):

- First step: data collection, in this stage, data for our analysis is collected from the open images dataset V6 which is a large-scale dataset that contains over nine million images [38]. This dataset is popular among researchers who are looking to train their machine learning (ML) models on a large dataset. This dataset provides a variety of images containing all types of firearms and bladed weapons, which will be helpful in training object detection algorithms to more accurately identify crime in images. We extract six categories from this dataset: Handgun (607 images), Rifle (2072 images), Shotgun (476 images), Knife (785 images), Dagger (349 images), and Sword (492 images). Then, we grouped them into two categories: firearms (that contains handgun, rifle, and shotgun) and bladed weapons (that contains knife, dagger, and sword).
- Second step: data preprocessing, it refers to how an image is prepared for analysis. This can involve everything from resizing the picture to changing the way data is represented. One of the most important

things to do in data preprocessing is to resize the picture to  $320 \times 320$ . This ensures that all the data is of the same size, and thus can be more easily compared. It also makes it easier to work with smaller images, which can be useful when training a computer to recognize objects. Another important step in data preprocessing is to augment the data. This can be done in several ways, but the most common is to flip, rotate, or scale the image. This is done to increase the amount of data available for training and to make it more likely that the computer will be able to learn to recognize objects from a variety of angles. Data preprocessing is a critical step in object detection and can make the difference between a successful and unsuccessful object detection system. By resizing and augmenting the data, we can give ourselves a better chance to succeed.

- Third step: backbone model choice, when it comes to choosing a backbone model for object detection models, there are a few options to consider. The four most popular models are the VGG 16-19, ResNet, and inception v3. All four models have their pros and cons, so it is important to benchmark each one, to see which model works best for our specific data. We used transfer learning to fine-tune those models on our dataset. Lastly, for YOLO, it is without a doubt DarkNet that has been used as the backbone model.
- Fourth step: object detection models training, we train our object detection models on the train (70%) and validation sets (20%) using the best backbone model. This allows us to create a highly accurate and reliable model that can be used to detect a variety of objects. The backbone model provides us with a robust foundation upon which we can build our model. By using the best backbone model available, we can ensure that our model is as accurate and reliable as possible. Object detection models used are faster R-CNN, SSD, and YOLO v5. We used faster-RCNN because it is a more accurate and efficient object detection algorithm than R-CNN and Fast R-CNN. It can handle more complex images and learn high-level features, making it an ideal choice [39]. The newest version of YOLO, YOLOv5, is the best version yet. It is faster and more accurate than the previous versions, and it is also easier to use. From the perspective of both functionality and user-friendliness, v5 is simply the best choice [40]. Most object detection algorithms require some form of labeled data to train the model. The most common way to label data for object detection is to use bounding boxes. In a bounding box, the algorithm draws a box around the object in the image, and the label includes information about what class of object is contained in the box. Faster R-CNN and SSD use TensorFlow record (TFRecord) files as input data. YOLOv5 uses TXT annotations and YAML config files.
- Fifth step: models' inference, in this phase, we used the testing set (10%) on our object detection models to evaluate each model's performance in terms of both accuracy and inference speed. The testing phase is important because it allows us to see how well our models perform on unseen data.

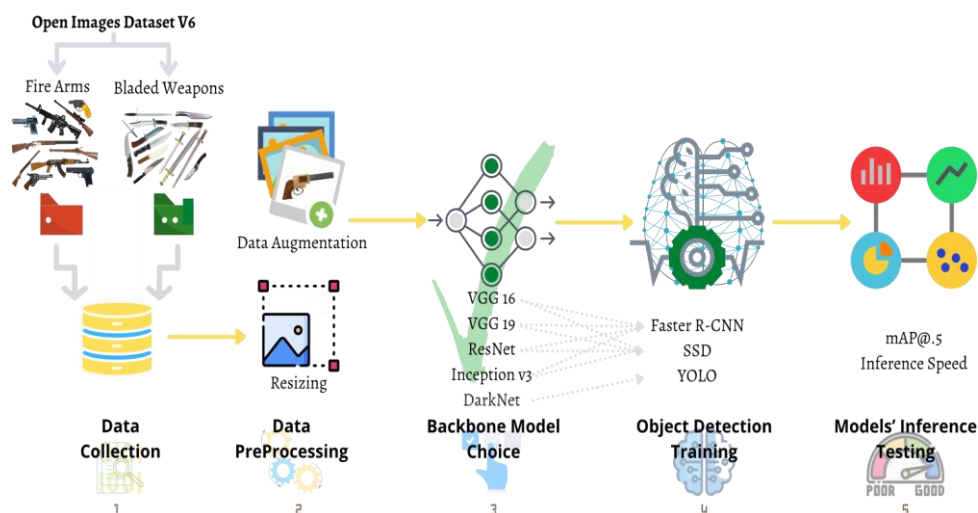


Figure 2. Proposed method

## 4. RESULTS AND DISCUSSION

### 4.1. Hardware characteristics

For our computations and experiments, we used the MARWAN high-performance computing (HPC) infrastructure with the following hardware specifications:

- CPU: 2x Intel Gold 6148 (2.4 GHz/20 cores).
- RAM: 192 GB.
- GPU: 2x NVIDIA Tesla P100 graphics cards, each having 12 GB of RAM.

#### 4.2. Evaluating the results

Table 1 and Figure 3 shows and compare the performance of four different algorithms on a classification task. The objective here is to determine whether an image contains objects such as firearms or bladed weapons. The four algorithms used are VGG 16, VGG 19, ResNet50, and inceptionV3. These two charts below illustrate each algorithm's accuracy show in Figure 3(a), loss show in Figure 3(b), precision show in Figure 3(c), and recall show in Figure 3(d). VGG 16 and VGG 19 both perform well on the classification task, with accuracy scores of 90.01% and 92.81% respectively.

While VGG 19 had also slightly higher precision and recall scores than VGG 16 but also had a higher loss score. ResNet50 outperformed both VGG algorithms, with an accuracy score of 93.44%, higher precision (94.01%) and recall (93.12%), and also less loss (22.68%). Inceptionv3 had the lowest accuracy score of all four algorithms, with only 58.13%, as well as the weakest score in all the other metrics.

As is the case with all DL results in general, these findings are based on a single dataset. It is possible that if a different dataset were used, the results would be different. However, according to these findings, we conclude that ResNet is the optimal algorithm that fits the role of a backbone feature extractor (as shown in Figure 1) in our object detection algorithms.

Table 1. Achieved results for the implemented CNNs backbones models

Algorithm	Accuracy (%)	Loss (%)	Precision (%)	Recall (%)
VGG 16	90.01	31.58	90.25	89.69
VGG 19	92.81	33.48	93.10	92.81
ResNet50	<b>93.44</b>	<b>22.68</b>	<b>94.01</b>	<b>93.12</b>
InceptionV3	58.13	86.93	60.69	49.69

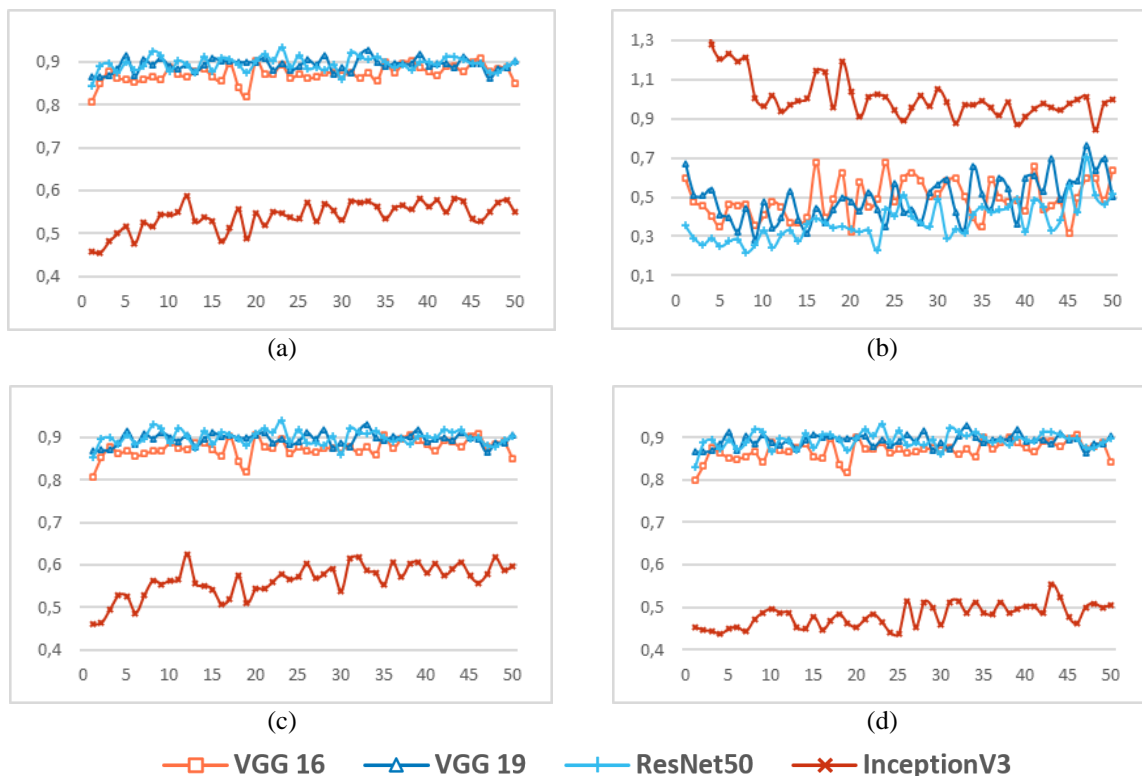


Figure 3. Achieved results for the implemented CNN-based backbones models (a) accuracy, (b) loss, (c) precision, and (d) recall

Table 2 shows the mAP@.5 and the inference speed of three different object detection algorithms faster R-CNN, SSD, and YOLOv5. mAP is a measure used to evaluate object detection models. It is the

mean of the average precision (AP) for each object class. AP is the ratio of the true positives (TP) to the sum of the TP and false positives (FP). The mAP represents the accuracy of an object detector at locating and distinguishing the different classes of objects in an image, thus to compare different object detection systems, the higher the mAP, the better the performance of the system. The @0.5 means that we used 0.5 as an intersection over union (IoU) threshold. IoU is used to determine how well a predicted mask or bounding box corresponds to the ground truth data. The inference speed is the number of FPS that the algorithm can process.

Table 2. Achieved results for the implemented object detection models

Algorithm	MAP@.5 (%)	Inference speed (FPS)
Faster R-CNN	<b>62.65</b>	6
SSD	61.63	14
YOLOv5	56.92	<b>61</b>

All three algorithms have different mAP@.5 scores and inference speed with faster R-CNN having the highest score of 62.65%, however, when looking at the inference speed, faster R-CNN is the slowest algorithm with a speed of 6 FPS. On the other hand, YOLOv5 has the lowest mAP.5 scores with 56.92%, but it is the fastest in terms of inference speed with a speed of 61 FPS. SSD lies in between the two in terms of both mAP@.5 score and inference speed with 61.63% and 14 FPS respectively.

Therefore, it is important to consider both accuracy and inference speed when choosing a model for a particular application. So, if we need accuracy, then faster R-CNN is the best option. On the other hand, if we need speed, then YOLOv5 is the best option. YOLOv5 model is over 10 times faster than the faster R-CNN model. If we need a balance of both (accuracy and speed), then SSD might be the best option. However, for applications such as security and surveillance, where it is critical to detect objects quickly, real-time object detection models enable us to track and recognize objects in a video stream at a very high frame rate (over 30 FPS), allowing us to process images as they are captured by a camera. YOLOv5 may be the optimal algorithm in this case due to its real-time object detection at 61 FPS and high accuracy of 56.92 %. There is no doubt that YOLOv5 is the best real-time object detection algorithm available today. It can detect objects related to crime with great accuracy and speed, and can detect a wide range of objects of different sizes and orientations.

## 5. CONCLUSION

In general, DL object detection systems have the potential to be a highly effective tool for law enforcement agencies and security experts, as they can save both time and resources for forensic activities. However, there are some risks associated with using artificial intelligence (AI), such as using it abusively by law enforcement agencies in accusing innocent individuals or identifying false criminals. Our proposed approach consists to build an object detection model, specifically for detecting crimes' tools, by using YOLOv5. We have chosen YOLOv5 after making a comparison of different object detection algorithms. Our model can accurately detect both firearms and bladed weapons, with a mAP score of 56.92%. It is also able to achieve a very high inference speed of 61 FPS for real-time detection. This makes it an excellent choice for security and law enforcement applications where quick and accurate detection of these kinds of weapons is crucial. Our model is open to more improvement, such as adding symbols or signs of hate and racism to identify and track down individuals or groups who may be involved in such crimes.

## ACKNOWLEDGMENT

This research was supported through computational resources of HPC-MARWAN ([www.marwan.ma/hpc](http://www.marwan.ma/hpc)) provided by the National Center for Scientific and Technical Research (CNRST), Rabat, Morocco.

## REFERENCES

- [1] R. P. Curiel, S. Cresci, C. I. Muntean, and S. R. Bishop, "Crime and its fear in social media," *Palgrave Commun.*, vol. 6, no. 1, pp. 1–12, Apr. 2020, doi: 10.1057/s41599-020-0430-7.
- [2] M. Boukabous and M. Azizi, "Multimodal Sentiment Analysis using Audio and Text for Crime Detection," in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Mar. 2022, pp. 1–5. doi: 10.1109/IRASET52964.2022.9738175.
- [3] T. Wen, J. Cao, and K. H. Cheong, "Gravity-Based Community Vulnerability Evaluation Model in Social Networks: GBCVE,"






- IEEE Trans. Cybern.*, pp. 1–13, 2021, doi: 10.1109/TCYB.2021.3123081.
- [4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning (adaptive computation and machine learning series)*. Cambridge: MIT Press, 2016.
  - [5] A. Kherraki, M. Maqbool, and R. El Ouazzani, "Efficient lightweight residual network for real-time road semantic segmentation," *IAES Int. J. Artif. Intell.*, vol. 12, no. 1, pp. 394–401, Mar. 2023, doi: 10.11591/IJAI.V12.I1.PP394-401.
  - [6] J. Azeez and D. J. Aravindhar, "Hybrid approach to crime prediction using deep learning," 2015. doi: 10.1109/ICACCI.2015.7275858.
  - [7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 580–587. doi: 10.1109/CVPR.2014.81.
  - [8] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.
  - [9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
  - [10] W. Liu *et al.*, "SSD: Single shot multibox detector," in *European Conference on Computer Vision*, Dec. 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0\_2.
  - [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.
  - [12] H. W. Naing, P. Thwe, A. C. Mon, and N. Naw, "Analyzing sentiment level of social media data based on SVM and Naïve Bayes algorithms," 2019. doi: 10.1007/978-981-13-0869-7\_8.
  - [13] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V.-K. Papastathis, and M. G. Strintzis, "Knowledge-assisted semantic video object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1210–1224, Oct. 2005, doi: 10.1109/TCSVT.2005.854238.
  - [14] M. Yandouzi *et al.*, "Forest Fires Detection using Deep Transfer Learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 8, pp. 268–275, 2022, doi: 10.14569/IJACSA.2022.0130832.
  - [15] M. Berrahal and M. Azizi, "Improvement of facial attributes' estimation using Transfer Learning," in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Mar. 2022, pp. 1–7. doi: 10.1109/IRASET52964.2022.9737845.
  - [16] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015. doi: 10.1038/nature14539.
  - [17] A. Kherraki, M. Maqbool, and R. El Ouazzani, "Lightweight and Efficient Convolutional Neural Network for Traffic Signs Classification," *2022 IEEE 9th Int. Conf. Sci. Electron. Technol. Inf. Telecommun. SETIT 2022*, pp. 155–160, 2022, doi: 10.1109/SETIT54465.2022.9875868.
  - [18] M. Yandouzi *et al.*, "REVIEW ON FOREST FIRES DETECTION AND PREDICTION USING DEEP LEARNING AND DRONES," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 12, pp. 4565–4576, Jun. 2022.
  - [19] I. Idrissi, M. Azizi, and O. Moussaoui, "A Stratified IoT Deep Learning based Intrusion Detection System," in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Mar. 2022, pp. 1–8. doi: 10.1109/IRASET52964.2022.9738045.
  - [20] A. Kherraki and R. El Ouazzani, "Deep convolutional neural networks architecture for an efficient emergency vehicle classification in real-time traffic monitoring," *IAES Int. J. Artif. Intell.*, vol. 11, no. 1, pp. 110–120, Mar. 2022, doi: 10.11591/IJAI.V11.I1.PP110-120.
  - [21] A. Kherraki, M. Maqbool, and R. El Ouazzani, "Traffic Scene Semantic Segmentation by Using Several Deep Convolutional Neural Networks," in *2021 3rd IEEE Middle East and North Africa COMMUNICATIONS Conference (MENACOMM)*, Dec. 2021, pp. 1–6. doi: 10.1109/MENACOMM50742.2021.9678270.
  - [22] J. Sun and A. Redei, "Knock Knock, Who's There: Facial Recognition using CNN-based Classifiers," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 1, pp. 9–16, 2022, doi: 10.14569/IJACSA.2022.0130102.
  - [23] P. Siriaraya *et al.*, "Witnessing crime through tweets: A crime investigation tool based on social media," 2019. doi: 10.1145/3347146.3359082.
  - [24] T. Sutikno, H. S. Purnama, R. A. Aprilianto, A. Jusoh, N. S. Widodo, and B. Santosa, "Modernisation of DC-DC converter topologies for solar energy harvesting applications: A review," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 28, no. 3, pp. 1845–1872, Dec. 2022, doi: 10.11591/ijeecs.v28.i3.pp1845-1872.
  - [25] T. Sutikno and D. Thalmann, "Insights on the internet of things: past, present, and future directions," *Telkomnika (Telecommunication Comput. Electron. Control.)*, vol. 20, no. 6, pp. 1399–1420, Dec. 2022, doi: 10.12928/TELKOMNIKA.V20I6.22028.
  - [26] T. Sutikno, H. S. Purnama, A. Pamungkas, A. Fadlil, I. M. Alsofyani, and M. H. Jopri, "Internet of things-based photovoltaics parameter monitoring system using NodeMCU ESP8266," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, pp. 5578–5587, Dec. 2021, doi: 10.11591/ijece.v11i6.pp5578-5587.
  - [27] N. A. M. Aseri *et al.*, "Comparison of meta-heuristic algorithms for fuzzy modelling of COVID-19 illness' severity classification," *IAES Int. J. Artif. Intell.*, vol. 11, no. 1, pp. 50–64, Mar. 2022, doi: 10.11591/ijai.v11.i1.pp50-64.
  - [28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2014, pp. 1–14.
  - [29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
  - [30] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.
  - [31] J. Redmon, "Darknet: Open Source Neural Networks in C." <https://pjreddie.com/darknet/> (accessed May 31, 2022).
  - [32] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "UAV-YOLO: Small object detection on unmanned aerial vehicle perspective," *Sensors*, vol. 20, no. 8, pp. 1–12, Apr. 2020, doi: 10.3390/s20082238.
  - [33] M. Grega, A. Matoriński, P. Guzik, and M. Leszczuk, "Automated detection of firearms and knives in a CCTV image," *Sensors*, vol. 16, no. 1, pp. 1–16, Jan. 2016, doi: 10.3390/s16010047.
  - [34] R. Olmos, S. Tabik, and F. Herrera, "Automatic handgun detection alarm in videos using deep learning," *Neurocomputing*, vol. 275, pp. 66–72, Jan. 2018, doi: 10.1016/j.neucom.2017.05.012.
  - [35] J. Iqbal, M. A. Munir, A. Mahmood, A. R. Ali, and M. Ali, "Leveraging orientation for weakly supervised object detection with application to firearm localization," *Neurocomputing*, vol. 440, pp. 310–320, Jun. 2021, doi: 10.1016/j.neucom.2021.01.075.
  - [36] 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 - ICCCT-2017*, Nov. 2017, pp. 84–88. doi: 10.1145/3154979.3154988.
- [37] P. Mehta, A. Kumar, and S. Bhattacharjee, “Fire and gun violence based anomaly detection system using deep neural networks,” in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Jul. 2020, pp. 199–204. doi: 10.1109/ICESC48915.2020.9155625.
- [38] A. Kuznetsova *et al.*, “The open images dataset V4,” *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1956–1981, Jul. 2020, doi: 10.1007/s11263-020-01316-z.
- [39] C. Wang and Z. Peng, “Design and implementation of an object detection system using faster R-CNN,” in *2019 International Conference on Robots & Intelligent System (ICRIS)*, Jun. 2019, pp. 204–206. doi: 10.1109/ICRIS.2019.00060.
- [40] F. Majeed *et al.*, “Investigating the efficiency of deep learning based security system in a real-time environment using YOLOv5,” *Sustain. Energy Technol. Assessments*, vol. 53, Oct. 2022, doi: 10.1016/J.SETA.2022.102603.

## BIOGRAPHIES OF AUTHORS



**Mohammed Boukabous**    Ph.D. in Computer Science at Mohammed First University in Oujda, Morocco, where he is conducting research in security intelligence using deep learning algorithms in exchanged messages. He holds a M.Sc. degree in internet of things from Sidi Mohamed Ben Abdellah University in Fez, Morocco (2019), as well as a B.Sc. degree in Computer Engineering from Mohammed First University (2016). Furthermore, he holds several certifications in natural language processing, artificial intelligence, security intelligence, big data, and cybersecurity. Additionally, he served as a reviewer for various international conferences. He is currently employed at Mohammed First University as an administrative. He can be contacted at email: m.boukabous@ump.ac.ma.



**Mostafa Azizi**    received a State Engineer degree in Automation and Industrial Computing from the Engineering School EMI of Rabat, Morocco in 1993, then a Master degree in Automation and Industrial Computing from the Faculty of Sciences of Oujda, Morocco in 1995, and a Ph.D. degree in Computer Science from the University of Montreal, Canada in 2001. He earned also tens of online certifications in Programming, Networking, AI, Computer Security. He is currently a Professor at the ESTO, University Mohammed First of Oujda. His research interests include security and networking, AI, software engineering, IoT, and embedded systems. His research findings with his team are published in over 100 peer-reviewed communications and papers. He also served as PC member and reviewer in several international conferences and journals. He can be contacted at email: azizi.mos@ump.ac.ma.