



Gun and Knife Detection Based on Faster R-CNN for Video Surveillance

M. Milagro Fernandez-Carrobles^(✉), Oscar Deniz, and Fernando Maroto

ETSI Industriales, VISILAB, University of Castilla-La Mancha, Ciudad Real, Spain
MMilagro.fernandez@uclm.es
<http://visilab.etsii.uclm.es>

Abstract. Public safety in public areas is nowadays one of the main concerns for governments and companies around the world. Video surveillance systems can take advantage from the emerging techniques of deep learning to improve their performance and accuracy detecting possible threats. This paper presents a system for gun and knife detection based on the Faster R-CNN methodology. Two approaches have been compared taking as CNN base a GoogleNet and a SqueezeNet architecture respectively. The best result for gun detection was obtained using a SqueezeNet architecture achieving a 85.44% AP_{50} . For knife detection, the GoogleNet approach achieved a 46.68% AP_{50} . Both results improve upon previous literature results evidencing the effectiveness of our detectors.

Keywords: Object detection · Guns · Knives · Video surveillance

1 Introduction

The use of weapons in public places has become a major problem in our society. These situations are more frequent in countries where weapons are legally purchased or their use is not controlled [10]. Crowded places are specially vulnerable. Unfortunately, mass shootings have become one of the most dramatic problems we face nowadays [20].

Video surveillance systems, typically based on classic closed circuit television (CCTV) are especially useful for intruder detection and remote alarm verification [6]. However, these systems need to be continuously supervised by a human operator. In this respect, it is estimated that the concentration of a security guard watching a camera panel decreases catastrophically after 20 min.

Security can be increased applying artificial vision algorithms on images obtained from video surveillance systems. Another advantage of these algorithms is the possibility of monitoring larger spaces using fewer devices thus requiring less dependence on the human factor.

Machine learning techniques have been widely used in the field of video surveillance. The prevalent paradigm of deep learning has but increased the potential of machine learning in automatic video surveillance. The objective of

this work is the development of two novel weapon detectors, for guns and knives, applying deep learning techniques and assess their performance.

This paper is organised as follows. Section 2 describes related work for video surveillance. The datasets used are detailed in Sect. 3 where gun and knife datasets are described in detail. Section 4 presents the object detector approaches and Convolutional Neural Networks (CNNs) used for this work. Results for gun and knife detection are shown in Sect. 5. Finally, Sect. 6 is devoted to the Conclusions.

2 Previous Work

The applications of the deep learning paradigm for weapon detection are still rather limited. The seminal work of Olmost et al. [14] presented an automatic handgun detection system for video surveillance. This system was based on a Faster R-CNN with a VGG16 architecture trained using their own gun database. Results provided zero false positives, 100% recall and a precision (IoU=0.5) value of 84.21%.

In Valldor et al. [17] a firearm detector for application to social media was presented. The detector employed a Faster R-CNN and an Inception_v2 network for feature extraction. A public database of images containing several firearms was manually labelled and used for training. Benchmarking was performed on the COCO dataset obtaining a ROC curve that showed usable results.

Verma et al. [18] used the Internet Movie Firearm Database (IMFDB) to generate a handheld gun detector. For that purpose, a Faster R-CNN based on a VGG16 architecture was applied only for feature extraction. Classification was performed using three different classifiers: a Support Vector Machine (SVM), a K-Nearest Neighbor (KNN) and a Ensemble Tree classifier. The best result achieved was 93.1% accuracy, using a Boosted Tree classifier. We have to note that the IMFDB dataset contains mostly profile images of pistols and revolvers at high resolution with homogeneous background, which is not a realistic situation.

The work of Akcay et al. [5] presented a detection and classification system for X-ray baggage security imagery. The work explored the applicability of multiple detection approaches based on sliding window CNN, Faster R-CNN, Region-based Fully Convolutional Networks and YOLO. Their system was composed by images divided into six classes: camera, laptop, gun, gun component, knife and ceramic knife. The best results for firearm detection were achieved with a YOLO architecture obtaining a 97.4% AP_{50} . For knife cases, the best results were obtained using a Faster R-CNN based on a ResNet-101 architecture with a 73.2% AP_{50} .

Finally, in Kanehisa et al. [11] the YOLO algorithm was applied to create a firearm detection system. The firearm dataset used for this study was extracted from the IMFDB website. Detection results obtained a 95.73% of sensitivity, 97.30% of specificity, 96.26% of accuracy and 70% of mAP_{50} .

Regarding knife detection, the most relevant results have been obtained in the context of the COCO (Common Objects in Context) Challenges. COCO is a

large-scale object detection dataset focused on detecting objects in context [13]. Each year COCO launches a challenge based on any of the following artificial vision tasks: detection, segmentation, keypoints or scene recognition. The last object detection challenge using bounding boxes was released in 2017 where the best result for knife detection was obtained by the Intel Lab team. Employing a Faster R-CNN and a HyperNet architecture this team achieved 36.6% AP_{50} . In Yuenyong et al. [19] knife detection was explored using a dataset of 8,527 infrared (IR) images. A GoogleNet architecture was applied to classify IR images as person or person carrying hidden knife. The classification accuracy reported was 97.91%.

In summary, the Faster R-CNN seems to be the prevalent deep architecture for gun and knife detection. This work also focuses on that architecture. As a novelty, this paper uses other CNNs not previously applied for this purpose and focused on the generation of lightweight models specially tailored for embedded, constrained and distributed systems operating in real environments with noisy and sometimes missing data.

3 Materials

In this section we describe the training and test datasets used for both object detection tasks. The data augmentation techniques used are also described.

3.1 Training Datasets

The gun dataset has been extracted from [14]. The dataset is composed by 3,000 images of guns from different views and scenarios. In order to increase the accuracy of the detector, a data augmentation technique was applied to the dataset. The aim is to perform transformations that simulate realistic views of the object to be detected, see Fig. 1:

- Increasing brightness (10%) in order to simulate different illuminations
- Image scaling to simulate different distances to the object
- Mirroring and rotations (5°) to create different canonical views of the object

With these transformations, the dataset was increased to a total of 15,000 images.

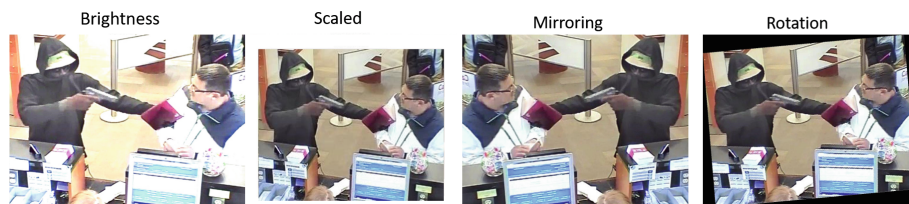


Fig. 1. Data augmentation transformations for the gun dataset

On the other hand, the COCO 2017 dataset [1] has been used to train the knife detector. COCO is a large dataset for object detection and segmentation tasks. The full dataset has a total of 330,000 images with 1.5 million objects divided into 80 classes, one of them knives. In the dataset there are a total of 4,326 images of knives, with a total of 7,770 knives labelled. This dataset has been extended by applying mirroring and scaling transformations for data augmentation so that a total of 12,978 knife images were obtained.

3.2 Benchmarking Datasets

The gun test set was generated leveraging several existing gun datasets with a total of 1,303 images:

- The Olmos et al. test set [14] which is composed by a total of 608 images and 304 weapon images.
- The small_gun category of the Gupta dataset [4] with a total of 80 images of guns.
- The handgun class of the Open Images Dataset V4 [3] with 89 images of guns [12].
- Finally, 526 random images from the COCO dataset [1] without weapon instances.

Regarding knives, the test set was generated using 169 images from the knife and kitchen knife classes of the Open Images Dataset V4 [3,12] and 526 random images from the COCO dataset [1] without knives. Thus, the knife test dataset had a total of 695 images.

4 Methodology

As mentioned above, the main objective of this work is the development of an object detector that efficiently locates guns and knives in real-time video. For that purpose, an approach based on deep learning techniques and more specifically through the Faster R-CNN methodology will be adopted. This object detection approach uses internally a CNN and a Regional Proposal Network (RPN) for the classification and location processes respectively. In order to better understand this methodology, a brief description of its evolution and performance is described below.

4.1 Evolution of R-CNN Object Detectors

The Regions-CNN method was developed in 2014. The processing of a R-CNN can be divided into three steps [8]. Firstly, an algorithm called selective search generates approximately 2,000 region proposals (or regions of interest). These region proposals are independent divisions of the image where an object could be located. Secondly, a CNN extracts features individually from each region

proposal. Finally, the object is classified using a Support Vector Machine (SVM) methodology. Region proposals are considered as positive when their Intersection over Union (IoU) measure against the ground truth exceeds an arbitrary value. Later, the object bounding box localization is calculated by overlapping the selected region proposals.

One of the main disadvantages of the R-CNN was its slow execution time. Fast R-CNN was proposed in 2015 as an improvement of R-CNN [7]. Fast R-CNN is twenty five times faster than its predecessor mainly due to two modifications. Feature extraction is performed using a CNN on the whole input image. Region proposals are selected as in the R-CNN approach by an external selective search method and included in the last layers of the network as projections on the feature map. The SVM classifier in this approach is replaced by a softmax layer. Although the Fast R-CNN was a breakthrough compared to the R-CNN, it still relied on algorithms such as the selective search that formed bottlenecks and slowed down the execution time of the detector.

In 2016 the Faster R-CNN method introduced a new region proposal extraction method called Regional Proposal Network (RPN) [15]. The idea of a RPN is to take advantage of the convolutional layers to obtain region proposals directly. Consequently, a sliding window is applied on the CNN feature map in order to extract region proposals of different sizes. The RPN is not responsible for classifying localized objects, this task is subsequently carried out by a Fast R-CNN. Therefore, a Faster R-CNN is a Fast R-CNN plus a RPN.

4.2 Faster-CNN Base Architectures

The CNN selected as Faster R-CNN base architecture should depend on its final purpose. Many CNNs employ a very deep architecture with the aim of obtaining a higher accuracy at a high computational cost. On the other hand, other architectures can be used that sacrifice precision in order to obtain models that can be integrated into embedded systems. In this work, GoogleNet and SqueezeNet CNN architectures have been tested and compared with the purpose of exposing their advantages and disadvantages for our task of weapon detection in video.

GoogleNet. The GoogleNet network [16] is a CNN developed in 2014. This network demonstrated high accuracy for object detection in the ImageNet contest Large-Scale Visual Recognition Challenge 2014, being the winning architecture with a 6.66% error rate. The architecture of this CNN is mainly composed by Inception layers which are based on covering large image areas while keeping a high resolution for small areas with high feature density. For that, the network applies convolutions in parallel with different filter sizes. The GoogleNet architecture is composed by a total of 22 layers, see Table 1.

Training using GoogleNet for the Faster R-CNN was carried out applying a stochastic gradient descent optimization algorithm with a momentum of 0.9 to accelerate gradient vectors, a L2 regularization method and an initial learning rate of $1e-3$. The optimization was run for 30 epochs.

Table 1. GoogleNet architectural dimensions [16]

Layer name/type	Output size	Filter size/stride	Depth	#1 × 1	#3 × 3 reduce	#3 × 3	#5 × 5 reduce	#5 × 5	Pool proj	Params	Ops
Convolution	112 × 112 × 64	7 × 7/2	1							2.7K	34M
Max pool	56 × 56 × 64	3 × 3/2	0								
Convolution	456 × 56 × 192	3 × 3/1	2		64	192				112K	360M
Max pool	28 × 28 × 192	3 × 3/2	0								
Inception	28 × 28 × 256		2	64	96	128	16	32	32	159K	128M
Inception	28 × 28 × 480		2	128	128	192	32	96	64	380K	304M
Max pool	14 × 14 × 480	3 × 3/2	0								
Inception	14 × 14 × 512		2	192	96	208	16	48	64	364K	73M
Inception	14 × 14 × 512		2	160	112	224	24	64	64	437K	88M
Inception	14 × 14 × 512		2	128	128	256	24	64	64	463K	100M
Inception	14 × 14 × 528		2	112	144	288	32	64	64	580K	119M
Inception	14 × 14 × 832		2	256	160	320	32	128	128	840K	170M
Max pool	7 × 7 × 832	3 × 3/2	0								
Inception	7 × 7 × 832		2	256	160	320	32	128	128	1072K	54M
Inception	7 × 7 × 1024		2	384	192	384	48	128	128	1388K	71M
Avg pool	1 × 1 × 1024	7 × 7/2	0								
Dropout (40%)	1 × 1 × 1024		0								
Linear	1 × 1 × 1000		1							1000K	1M
Softmax	1 × 1 × 1000		0								

SqueezeNet. The SqueezeNet network [9] is a CNN developed in 2016. The main goal of this network was the deployment of a small CNN architecture with fewer parameters instead of improving the accuracy. SqueezeNet achieved the same accuracy than AlexNet on the ImageNet dataset obtaining a model size with 50x fewer parameters. Therefore, it is a valuable alternative for embedded systems, field-programmable gate arrays (FPGAs) and other constrained systems. The SqueezeNet architecture follows three strategies to reduce the number of parameters while maintaining the accuracy level:

- Most convolutions replace 3 × 3 filters for 1 × 1 filters. As a consequence, the number of parameters is reduced 9× by convolution
- Decrease the number of channels using squeeze layers
- Downsample late in the network so that convolution layers have large activation maps. Downsampling is performed reducing the size of the input data or selecting those layers in which downsampling is going to be carried out. Most of the layers have a stride of 1 and layers with stride larger than 1 are accumulated at the end of the network. That produces large activation maps improving accuracy levels

SqueezeNet applies fire modules to achieve the previous strategies. A fire module is composed by a squeeze convolution layer (1 × 1 filters) and an expand layer (mixture of 1 × 1 and 3 × 3 convolution filters). Three parameters are included in a fire module: s1 × 1 (from squeeze layer), e1 × 1, and e3 × 3 (from expanded layer). All are related to the number of filters used in these layers.

Table 2. SqueezeNet v1.0 architectural dimensions [9]

Layer name/type	Output size	Filter size/stride	Depth	s1 × 1 (#1 × 1 squeeze)	e1 × 1 (#1 × 1 expand)	e3 × 3 (#3 × 3 expand)	1 × 1 sparsity	e1 × 1 sparsity	e3 × 3 sparsity	Parameter after pruning
Convolution	111 × 111 × 96	7 × 7/2(×96)	1				100% (7 × 7)			14,208
Max pool	55 × 55 × 96	3 × 3/2	0							
Fire	55 × 55 × 128		2	16	64	64	100%	100%	33%	5,746
Fire	55 × 55 × 128		2	16	64	64	100%	100%	33%	6,258
Fire	55 × 55 × 256		2	32	128	128	100%	100%	33%	20,646
Max pool	27 × 27 × 256	3 × 3/2	0							
Fire	27 × 27 × 256		2	32	128	128	100%	100%	33%	24,742
Fire	27 × 27 × 384		2	48	192	192	100%	50%	33%	44,700
Fire	27 × 27 × 384		2	48	192	192	50%	100%	33%	46,236
Fire	27 × 27 × 512		2	64	256	256	100%	50%	33%	77,581
Max pool	13 × 13 × 512	3 × 3/2	0							
Fire	13 × 13 × 512		2	64	256	256	50%	100%	30%	77,581
Convolution	13 × 13 × 1000	1 × 1/1(×1000)	1				20% (3 × 3)			103,400
Avg pool	1 × 1 × 1000	13 × 13/1	0							
Softmax	1 × 1 × 1000		0							

Fire module sets that $s1 \times 1$ must be less than the sum of $e1 \times 1$ and $e3 \times 3$, so the squeeze layer helps to limit the number of input channels to the 3×3 filters. The SqueezeNet architecture is composed by a total of 13 layers, see details in Table 2.

In this approach, training using a SqueezeNet for the Faster R-CNN was carried out applying a stochastic gradient descent optimization algorithm with a momentum of 0.9 to accelerate gradient vectors, a L2 regularization method and an initial learning rate of $1e-4$. As with the GoogleNet approach, 30 epochs were used to train the classifier.

5 Results

In a detection task there are two possible results, positive and negative. Some positive cases can be classified as negative and vice versa. These cases are called false positives (type I error) and false negatives (type II error), respectively. Thus, the following four cases are considered: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). However, in an object detection task object localization must be considered too. The accuracy achieved in an object detector is commonly evaluated using the mean average precision (mAP). This measurement is defined as the average of the maximum precisions at different recall values. Therefore, the three main concepts considered within this measurement are: precision, recall and Intersection over union (IoU).

- Precision measures the likelihood of a positive case being classified as such. This value is estimated using the amount of real positive cases which were classified as positive. Then, it is the percentage of correct positive predictions.

Precision is calculated as in Eq. 1.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

- Recall (or sensitivity) measures the likelihood of classifying the object as positive. In other words, it measures how good is the network finding positives cases, see Eq. 2.

$$Recall = \frac{TN}{TN + FP} \quad (2)$$

- The Intersection over Union (IoU) quantifies the overlapping between 2 regions. This measures how valuable the prediction is with respect to the ground truth (the real object boundary). A prediction is usually considered to be correct when the IoU is equal or greater than 0.5.
- The Precision-Recall Curve summarizes the trade-off between precision and recall values using different probability thresholds. The area under this curve is known as average precision (AP), a value between 0 and 1 which evaluates the quality of the model. When there is more than one object to be detected, the average precision is calculated for each object resulting in the mean average precision (mAP).

For the problem of gun detection, Faster R-CNN trained using GoogleNet obtained a 55.45% of AP_{50} (AP at IoU = 0.50). Faster R-CNN using a SqueezeNet obtained 85.44% of AP_{50} , a significant difference over GoogleNet. The precision-recall curve acquired for SqueezeNet is shown in Fig. 2. This detector achieved good results, obtaining similar or even improving upon previous results described in the literature. A comparison between our results and another similar work are shown in Table 3.

Table 3. Results comparison for weapon detection based on a Faster R-CNN

<i>Gun Detector performance measurement</i>			
	Our approaches		Olmos et al. [14]
Faster R-CNN methodology	GoogleNet	SqueezeNet	VGG16
AP_{50}	55.45%	85.44%	84.21%
<i>Knife Detector performance measurement</i>			
	Our approaches		COCO Challenge17 (Intel Lab) [2]
Faster R-CNN methodology	GoogleNet	SqueezeNet	HyperNet
AP_{50}	46.68%	1.1%	36.6%

Regarding knife detection, Faster R-CNN based on GoogleNet achieved an AP_{50} of 46.68% and a 1.1% using SqueezeNet, being these results much lower than expected. Nevertheless, results obtained using GoogleNet architecture improve previous knife detection results reported in the literature, see Table 3. Figure 3 shows the precision-recall curve for the GoogleNet approach.

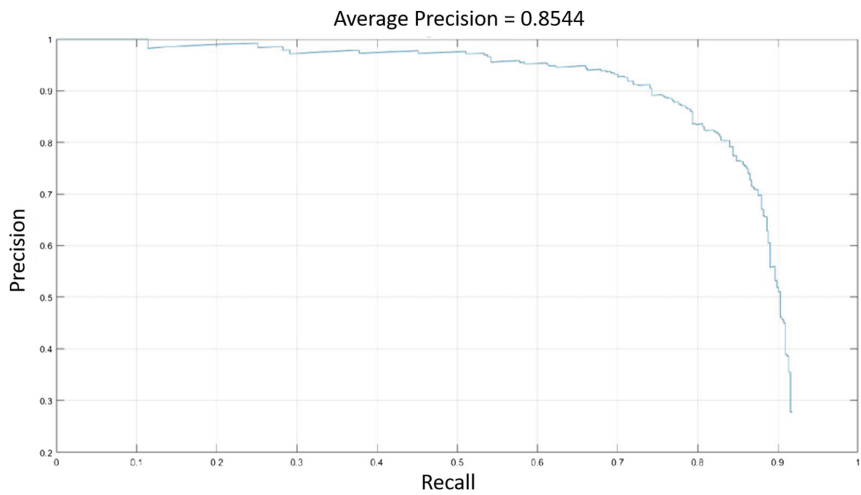


Fig. 2. Precision-recall curve for gun detection using Faster R-CNN and SqueezeNet architecture v1.1

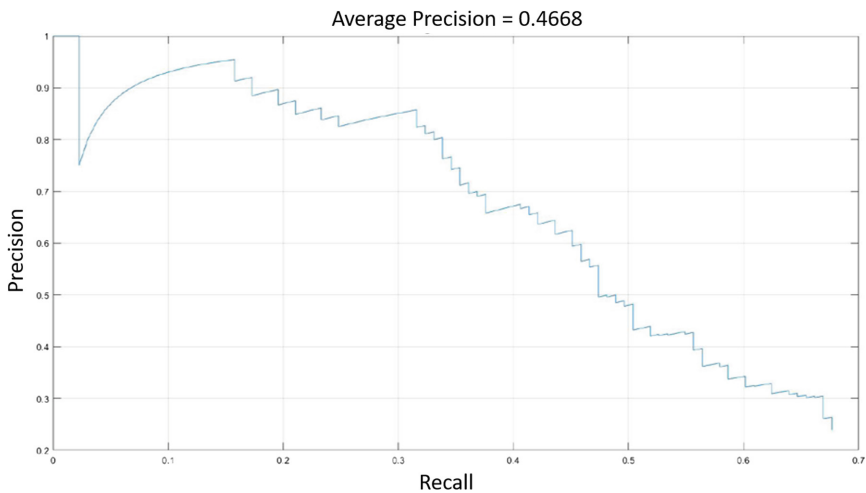


Fig. 3. Precision-recall curve for knife detection using Faster R-CNN and GoogleNet architecture

Some visual results obtained for weapon detection are shown in Fig. 4. These results demonstrate the capability of our weapon detectors to locate guns and knives on the test dataset.

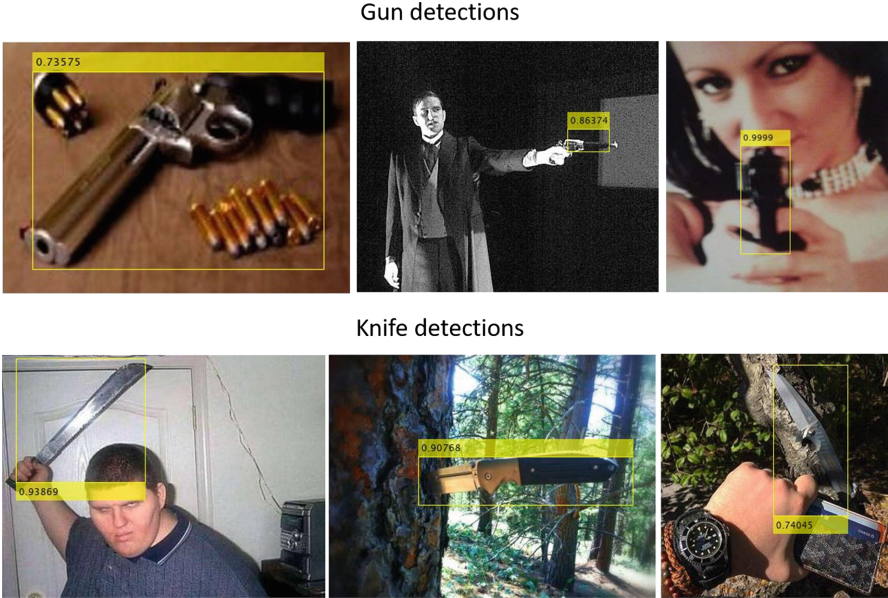


Fig. 4. Scores obtained for gun (Faster R-CNN/SqueezeNet) and knife (Faster R-CNN/GoogleNet) detectors

6 Conclusions

Public and crowded areas are still the target of many violent acts. Video surveillance can be helped by automatic image analysis using artificial vision. This paper describes the implementation of several weapon detectors for video surveillance based on Faster R-CNN methodologies. Several previous studies have applied the Faster R-CNN methodology but, as far as the authors know, none of them have been actually focused on the development of lightweight models that could be later used in constrained and real-time devices. GoogleNet or SqueezeNet are architectures for that purpose. For training, gun and knife images from the work of Olmos et al. and COCO dataset have been used. Several transformations such as rotations, scaling or brightness were applied in order to augment the datasets. Detectors were developed using the GoogleNet and SqueezeNet architectures as CNN base on a Faster R-CNN. The best result for gun detection was obtained using a SqueezeNet architecture achieving a 85.44% AP_{50} . For knife detection, GoogleNet approach accomplished a 46.68% AP_{50} . Both detector results improve upon previous literature from similar studies evidencing the effectiveness of our detectors.

Acknowledgments. This work was partially funded by projects TIN2017-82113-C2-2-R by the Spanish Ministry of Economy and Business and SBPLY/17/180501/000543 by the Autonomous Government of Castilla-La Mancha and the ERDF.

References

1. COCO dataset 2017. <http://cocodataset.org>. Accessed 04 May 2019
2. COCO detection leaderboard. <http://cocodataset.org/#detection-leaderboard>. Accessed 04 May 2019
3. Open images dataset v4. <https://storage.googleapis.com/openimages/web/index.html>. Accessed 04 May 2019
4. Weapon detection by neural network. <https://github.com/Shubham02gupta/Weapon-Detection-by-Neural-network/tree/master/train>. Accessed 04 May 2019
5. Akcay, S., Kundegorski, M.E., Willcocks, C.G., Breckon, T.P.: Using deep convolutional neural network architectures for object classification and detection within X-ray baggage security imagery. *IEEE Trans. Inf. Forensics Secur.* **13**(9), 2203–2215 (2018). <https://doi.org/10.1109/TIFS.2018.2812196>
6. Dastidar, J.G., Biswas, R.: Tracking human intrusion through a CCTV. In: 2015 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 461–465, December 2015. <https://doi.org/10.1109/CICN.2015.95>
7. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448, December 2015. <https://doi.org/10.1109/ICCV.2015.169>
8. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, pp. 580–587. IEEE Computer Society, Washington, DC (2014). <https://doi.org/10.1109/CVPR.2014.81>
9. Iandola, F.N., Moskewicz, M.W., Ashraf, K., Han, S., Dally, W.J., Keutzer, K.: SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR abs/1602.07360* (2016). <http://arxiv.org/abs/1602.07360>
10. Jehan, F., et al.: The burden of firearm violence in the United States: stricter laws result in safer states. *J. Inj. Violence Res.* **10**(1), 11–16 (2018). <https://doi.org/10.5249/jivr.v10i1.951>
11. Kanehisa, R., Neto, A.: Firearm detection using convolutional neural networks. In: Proceedings of the 11th International Conference on Agents and Artificial Intelligence, ICAART, vol. 2, pp. 707–714, January 2019. <https://doi.org/10.5220/0007397707070714>
12. Kuznetsova, A., et al.: The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale. *CoRR abs/1811.00982* (2018). <http://arxiv.org/abs/1811.00982>
13. Lin, T., et al.: Microsoft COCO: common objects in context. *CoRR abs/1405.0312* (2015). <http://arxiv.org/abs/1405.0312>
14. Olmos, R., Tabik, S., Herrera, F.: Automatic handgun detection alarm in videos using deep learning. *Neurocomputing* **275**, 66–72 (2018). <https://doi.org/10.1016/j.neucom.2017.05.012>. <http://www.sciencedirect.com/science/article/pii/S0925231217308196>
15. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: 28th International Conference on Neural Information Processing Systems, NIPS 2015, vol. 1, pp. 91–99. MIT Press, Cambridge (2015). <http://dl.acm.org/citation.cfm?id=2969239.2969250>
16. Szegedy, C., et al.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9, June 2015. <https://doi.org/10.1109/CVPR.2015.7298594>

17. Valldor, E., Stenborg, K.G., Gustavsson, D.: Firearm detection in social media images. In: Swedish Symposium on Deep Learning 2018, September 2018
18. Verma, G.K., Dhillon, A.: A handheld gun detection using Faster R-CNN deep learning. In: Proceedings of the 7th International Conference on Computer and Communication Technology, ICCCT-2017, pp. 84–88. ACM, New York (2017). <https://doi.org/10.1145/3154979.3154988>
19. Yuenyong, S., Hnoohom, N., Wongpatikaseree, K.: Automatic detection of knives in infrared images. In: 2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON), pp. 65–68, February 2018. <https://doi.org/10.1109/ECTI-NCON.2018.8378283>
20. Zhang, Y., Wang, Y., Foley, J., Suk, J., Conathan, D.: Tweeting mass shootings: the dynamics of issue attention on social media. In: Proceedings of the 8th International Conference on Social Media and Society, #SMSociety 2017, pp. 59:1–59:5. ACM, New York (2017). <https://doi.org/10.1145/3097286.3097345>