Article

Journal of Computational Technologies

June 2022, Vol.6 No.17 22-27

Transfer learning for handgun detection

Transferencia de aprendizaje para la detección de armas de fuego

MARTÍNEZ-DÍAZ, Saúl†*

Tecnológico Nacional de México/Instituto Tecnológico de La Paz, División de estudios de Posgrado e Investigación, México.

ID 1st Author: *Saúl, Martínez-Díaz* / **ORC ID:** 0000-0003-4962-5995, **Researcher ID Thomson**: Q-7112-2019, **CVU CONACYT ID**: 175255

DOI: 10.35429/JOCT.2022.17.6.22.27

Received: January 25, 2022; Accepted: June 30, 2022

Abstract

Insecurity is a growing up problem affecting many cities around the world. Among others, firearm assault is one of the most common crimes committed. Although in some places have been installed video surveillance cameras, human intervention is still required to analyze the captured scenes. To prevent crime, a system that automatically detects dangerous situations is required. However, several problems arise when detecting objects from low-cost video surveillance systems. Some of these problems are poor quality of images, non-homogeneous illumination, background noise, occluded and rotated objects. In this paper, we propose a method to detect handguns by adapting a previously trained Convolutional Neural Network (transfer learning). The system was tested with images obtained from three video sequence captured with a low-cost webcam, under not controlled conditions. The detection errors were 8.3%, 15.7 and 11.7%, respectively. These results are comparable with other state of the art methods tested with higher quality images.

Transfer learning, Crime prevention, Convolutional neural network

Resumen

La inseguridad es un problema creciente que afecta a muchas ciudades del mundo. Entre otros, el asalto con arma de fuego es uno de los delitos que se cometen más comúnmente. Aunque en algunos lugares se han instalado cámaras de videovigilancia, aún se requiere la intervención humana para analizar las escenas captadas. Para prevenir la delincuencia se requiere un sistema que detecte automáticamente las situaciones de peligro. Sin embargo, surgen varios problemas a la hora de detectar objetos con sistemas de videovigilancia de bajo costo. Algunos de estos problemas son la mala calidad de las imágenes, la iluminación no homogénea, el ruido de fondo y objetos girados o parcialmente ocultos. En este artículo, proponemos un método para detectar armas cortas mediante la adaptación de una red neuronal convolucional previamente entrenada (transferencia de aprendizaje). El sistema se probó con imágenes obtenidas a partir de tres secuencias de video capturadas con una cámara web de bajo costo, en condiciones no controladas. Los errores de detección fueron del 8,3%, 15,7 y 11,7%, respectivamente. Estos resultados son comparables con otros métodos de última generación probados con imágenes de mayor calidad.

Transferencia de aprendizaje, Prevención del crimen, Redes neuronales convolucionales

Citation: MARTÍNEZ-DÍAZ, Saúl. Transfer learning for handgun detection. Journal of Computational Technologies. 2022. 6-17:22-27.

* Author's Correspondence (E-mail: saul.md@lapaz.tecnm.mx)

† Contributing researcher as first author.

Introduction

In many cities around the world, most crimes are committed by using firearms. This type of crime takes life of many victims annually. For example, in México City, according to the ICESI (Institute of Citizen Insecurity Studies, 2010), firearm assault is the most common crime. Although in some places have been installed video surveillance cameras, most of them require human intervention to analyze videos. Unfortunately, due to the enormous number of images to be analyzed, it is almost impossible for a person to watch simultaneously all scenes, detect suspicious activities and activate a preventive alarm system. Moreover, in many cities, the installed video surveillance systems provide low quality images, making difficult gun identification even for a human. One way to reduce this kind of crimes is early detection, so that the security agents or policemen can prevent violent acts. Therefore, to prevent and reduce crime, automatic detection systems capable of operating in real environments with poor quality images and under not controlled conditions, are required.

To detect suspicious activities, at first, it is necessary to recognize if a gun is present in the analyzed scene. Recognition of objects from video surveillance systems is a challenging problem due to poor quality images, non-homogeneous illumination, background noise distorted (rotated, scaled, blurred, etc.) and occluded objects. Literature offers several methods to classify guns: In (Kasemsan, 2014) proposed a method based on corner detection and template matching is presented; additionally, in (Grega, Matiolanski, Guzik, & Lesczuk, 2016) is proposed a method that uses Principal Component Analysis (PCA) and Artificial Neural Networks (ANN); besides, (Martinez-Diaz, Palacios-Alvarado, & Martinez-Chavelas, 2017) used invariant features and ANN, developed in parallel processing hardware, to detect pistols.

On the other hand, the last few years, Convolutional Neural Network (CNN) have achieved superior results than other classical machine learning methods in image classification, detection, and segmentation, for numerous applications.

For weapon detection, in (Olmos, Tabik, & Herrera, 2018) is introduced a detection system for both, surveillance and control purposes, based on a CNN; also, in (Xu & Hung, 2020) an AI-based system for automatic detection and recognition of weapons in surveillance videos is proposed, reporting precisions of 0.85 and 0.70 at intersection over union values of 0.5 and 0.75, respectively. However, it seems that the problem of guns recognition has not been totally solved yet.

In this paper, we propose a method to detect handguns in a scene. The method is based on deep learning techniques, region proposal CNNs and transfer learning. Transfer learning is very useful to save time and computational resources, due to a pre-trained CNN can be adjusted to recognize other objects. We retrain CNN to recognize rotated and occluded low-quality images. The system was tested with images obtained from three video sequence captured with a low-cost webcam, under not controlled conditions. The detection errors were 8.3%, 15.7 and 11.7%, respectively. The paper is organized as follows: in section 2 we review some basic concepts; in section 3 we introduce the proposed method; in section 4 we provide and discuss computer simulations; finally, in section 5 we summarize our conclusions.

Basic concepts

Convolutional neural networks

Unlike traditional machine learning methods for classification, in which features must be chosen manually and extracted with specialized algorithms, deep learning networks automatically discover relevant features from data. CNNs are composed with an input layer, an output layer, and many hidden layers in between (LeCun, Bengio, & Hinton, 2015). Each hidden layer can be one of the following types:

Convolutional layer. Units in a convolutional layer are organized in feature maps. Within them, each unit is connected to local patches in the feature maps of the previous layer through a set of weights, called a filter bank. Each filter activates certain features from the images. For two-dimensional discrete functions $f(x, y)$ and $g(x, y)$, convolution is defined as:

$$\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(i,j)g(x+i,y+j) \qquad (1)$$

Pooling layer. This layer is used to merge semantically similar features into one, to simplify the output, by performing nonlinear down sampling, which reduces the number of parameters that the network needs to learn. The pooling layer takes a pool size as a hyperparameter, usually 2 by 2. Processing of the input image is as follows: divide the image in a grid of 2 by 2 areas and take from each four-pixel a representative value (normally the maximum value is used).

Rectified linear unit (ReLU). The result of the convolution is then passed through a non-linearity, usually rectifying the input signal by mapping negative values to zero and maintaining positive values This rectification allows a faster and more effective training.

These three operations are repeated over tens or hundreds of layers, with each layer learning to detect different features. After feature detection, the architecture of a CNN shifts to classification. The next-to-last layer is a fully connected layer that outputs a vector of K dimensions, where K is the number of classes that the network will be able to predict. This vector contains the probabilities for each class of any image being classified. The final layer of the CNN architecture uses a softmax function, to provide the classification output.

Region-based CNN (R-CNN)

Contrasting with classification, detection requires the accurate localization of objects (probably many) into images. Compared to image classification, object detection is a more challenging task that requires more complex methods to be solved. Complexity arises because detection requires the accurate localization and refinement of many objects. Several algorithms have been proposed for training R-CNN´s, some methods use multi-stage pipelines, but they are slow. Other methods use a sliding window technique to generate region proposals.

For this task, region-based methods have shown better performance than other methods. In this case, the first step is proposing several candidate object localizations; then, proposals must be refined to achieve precise localization. Each region must be evaluated and its membership to any class of object vs. background is scored. From the latter, the most popular algorithms are regions with CNN (R-CNN) (Girshick, Donahue, Darrell, & Malik, 2014), Fast Region-based Convolutional Network (Fast R-CNN) (Girshick, Fast R-CNN, 2015) and Faster Region-based Convolutional Network (Faster R-CNN) (Ren, He, Girshick, & Sun, 2015).

Transfer learning

The success of CNNs depends on the number of images used to train them. Usually, thousands or millions of images are required to achieve good results. Training a CNN from scratch may require weeks of computation in a high-performance computer. Besides, preparing manually images for training is a very high time-consuming task. A better option is using a pre-trained network, which is fine-tuned with a few images, to make it work in the new desired task. This method is called transfer learning.

Transfer learning refers to the situation where what has been learned in one setting is exploited to improve generalization in another setting (Goodfellow, Bengio, & Courville, 2016). For example, one CNN trained to recognize one set of visual categories, such as cars, then is fine-tuned to learn about a different set of visual categories, such as trucks. Here, a key point about image data is that the extracted features from a data set are highly reusable across other data sources. Usually, only the deeper layers are fine-tuned, and the weights of the early layers are fixed. The reason for training only the deeper layers, while keeping the early layers fixed, is that the earlier layers capture only primitive features like edges, whereas the deeper layers capture more complex features. The primitive features do not change too much with the application at hand, whereas the deeper features might be sensitive to the application at hand. Some popular pre-trained CNNs, available for transfer learning are AlexNet (Krizhevsky, Sutskever, & Hinton, 2017), ResNet (He, Zhang, Ren, & Sun, 2016) and GoogLeNet (Szegedy, et al., 2016).

Proposed Method

Pre-trained network

For our implementation we selected AlexNet. This network was the winner of the 2012 ILSVRC competition. Figure 1 shows AlexNet architecture. As can be seen, the net contains eight layers with weights; the first five are convolutional and the remaining three are fully connected. The output of the last fully connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels. The kernels of the second, fourth, and fifth convolutional layers are connected only to those kernel maps in the previous layer, which reside on the same graphics processing unit (GPU). The kernels of the third convolutional layer are connected to all kernel maps in the second layer.

Training algorithm

For training, we tested three algorithms: R-CNN, Fast R-CNN and Faster R-CNN. The best results were achieved with Fast R-CNN algorithm. Fig. 2 illustrates the Fast R-CNN operation. The network takes as input an entire image and a set of object proposals. Then, the whole image with several convolutional and max pooling layers is processed to produce a map feature. Then, for each object proposed a pooling layer extracts a fixed-length feature vector from the map. Each feature vector is fed into a sequence of fully connected layers that end in two output layers: one that produces softmax probability estimates over all object classes and a background.

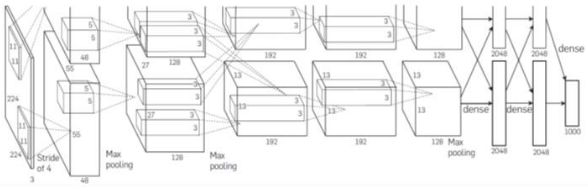


Figure 1 AlexNet architecture
Source: Figure taken from Goodfellow, Bengio & Courville, 2016

All weights of the network are adjusted using backpropagation algorithm. The loss function (L) used is:

$$L(p,u,t^u,v) = L_{cls}(p,u) + \lambda[u \geq 1]L_{loc}(t^u,v) \qquad (2)$$

Where p is a discrete probability distribution, computed by a softmax function; t^u are offsets for each of the K classes; $L_{cls}(p, u)$ = -logp_u is the log loss for the true class u; L_{loc} is defined over a tuple of true bounding-box regression targets for class u, $v = (v_x; v_y; v_w; v_h)$, and a predicted tuple tu = ($t^u{}_x$; $t^u{}_y$; $t^u{}_w$; $t^u{}_h$), again for class u. [u ≥ 1] evaluates to 1 when $u \geq 1$ and 0 otherwise. By convention the background class is $u = 0$.

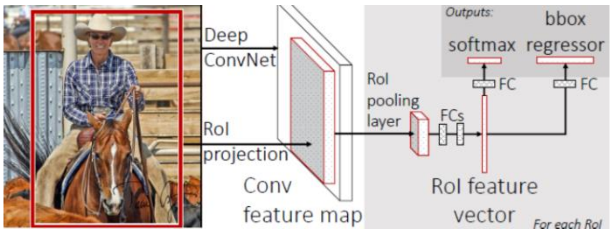


Figure 2 Fast R-CNN operation
Source: Figure taken from Girshick, 2015

Modified network

AlexNet has been trained on over a million images and can classify images into 1000 object categories. The network has learned rich feature representations for a wide range of images. It takes an image as input and outputs the probabilities for each of the object categories. The first layer (input layer) requires input images of size 227-by-227-by-3, where 3 is the number of color channels; then, each input image must be resized to such size. The last three layers of the pretrained network net are configured for 1000 classes. These three layers must be fine-tuned for the new classification problem. To retrain the selected net, we replaced the last three layers of the network. The new added layers were a fully connected layer, a softmax Layer and a classification layer. The final fully connected layer was set to have the same size as the number of classes in the new data set (one class: handgun). To learn faster in the new layers than in the transferred layers, we increased the learning rate factors of the fully connected layer.

Results

CNN

In this section, we illustrate the performance of the proposed technique. We tested the system with images obtained from a low-cost web camera.

Due to federal regulations, it is difficult to obtain firearms in the country; in consequence, we use a Walther Airsoft pistol, which meets color, size, and shape specifications of a real handgun. The video was captured indoor, under non-controlled conditions. The artificial illumination was produced by a non-homogeneous source of light. All images were 640x480x3 pixels. Figure 3 shows an example of an input image.

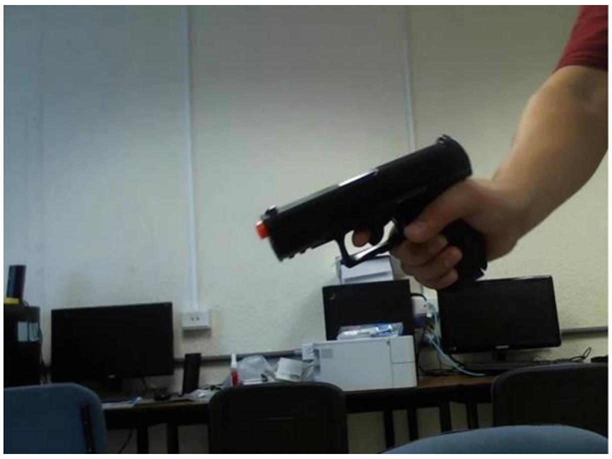
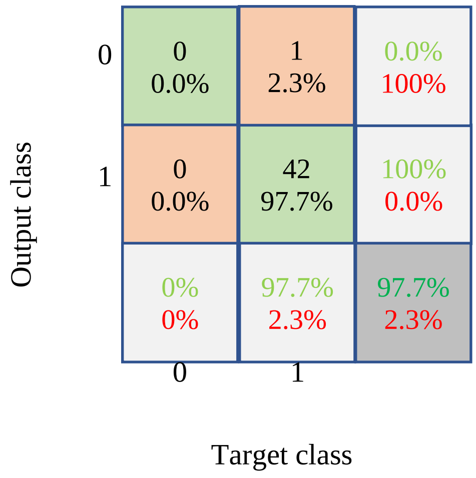


Figure 3 Example of training image

To train the network, we used 105 images containing a handgun. Each handgun was manually enclosed into a box, and coordinates of each box were used as ground truth, 60% of the images were used for training and 40% for testing. The network was trained in a low-performance GPU (GeForce GTX 960M). The main parameters selected for training were gradient method L2-norm, epochs 50, minibatch size 8, momentum 0.9000 and initial learning rate 1.0000e-03.

Figure 4 shows the confusion matrix of results, where class one is for the handguns and class zero is for the background. As can be seen, the system misclassifies only 2.3% of images (one false positive). Moreover, figure 5 illustrates the detection of a handgun with 0.99 of confidence, in an image where the handgun is blurred.



Output class	0	0 0.0%	1 2.3%	0.0% 100%
	1	0 0.0%	42 97.7%	100% 0.0%
		0% 0%	97.7% 2.3%	97.7% 2.3%
		0	1	

Target class

Figure 4 Confusion matrix

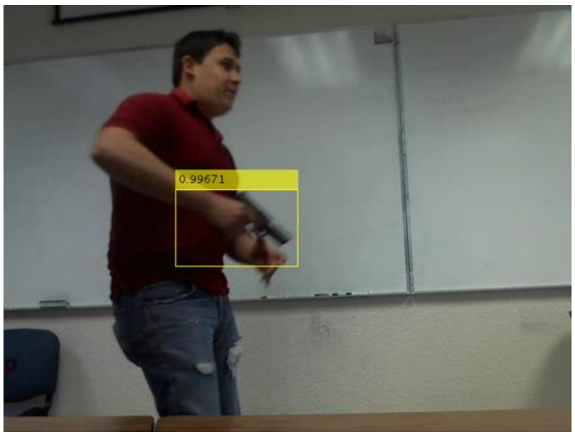


Figure 5 Example of handgun detection

Conclusions

In this paper, we proposed a method to detect handguns in real scenes. The proposed method uses convolutional neural networks to perform handgun detection. With this technique is not required to choose manually the relevant features of objects nor preprocessing of images in the detection stage. Besides, by using transfer learning, we can take advantage of a pre-trained net and adjust it to our application. Results shown a good performance of system, even when it was trained with just a few images, a reduced number of epochs and was tested with poor quality video sequence. This is a first step toward reduce crime in big cities. Future work includes recognition of other kind of weapons, tracking of detected objects and a system to analyze scenes and quantify the risk for people near such scene.

Acknowledgment

This work has been funded by Tecnológico Nacional de México [grant number 11432.21-P].

References

Girshick, R. (2015). Fast R-CNN. IEEE International Conference on Computer Vision (ICCV) (págs. 1-9). IEEE. https://ieeexplore.ieee.org/document/7410526 DOI: 10.1109/ICCV.2015.169.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Features hierarchies for accurate object detection and semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (págs. 1-21). IEEE. https://ieeexplore.ieee.org/document/6909475. DOI: 10.1109/CVPR.2014.81

Goodfellow, I., Bengio, I., & Courville, A. (2016). Deep Learning. MIT Press. https://link.springer.com/article/10.1007/s10710-017-9314-z. DOI: https://doi.org/10.1007/s10710-017-9314-z

Grega, M., Matiolanski, A., Guzik, P., & Lesczuk, M. (2016). Automated detection of firearms and knives in a CCTV image. Sensors, 1-16. https://www.mdpi.com/1424-8220/16/1/47. DOI: https://doi.org/10.3390/s16010047

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual lerning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (págs. 770-778). IEEE. https://arxiv.org/abs/1512.03385. DOI: https://doi.org/10.48550/arXiv.1512.03385

Institute of Citizen Insecurity Studies. (2010). ICESI-ENSI Results by state. México: Institute of Citizen Insecurity Studies. https://seguridadenperspectiva.blogspot.com/2011/03/septima-encuesta-nacional-de.html

Kasemsan, M. L. (2014). The classification of gun's type using image recognition theory. International Journal of Information and Electronics Engineering, 54-58. http://www.ijiee.org/index.php?m=content&c=index&a=show&catid=43&id=433. DOI: 10.7763/IJIEE.2014.V4.407

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 84-90. https://dl.acm.org/doi/10.1145/3065386. DOI: https://doi.org/10.1145/3065386

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 436-444. https://pubmed.ncbi.nlm.nih.gov/26017442/. DOI: 10.1038/nature14539

Martinez-Diaz, S., Palacios-Alvarado, C. A., & Martinez-Chavelas, S. (2017). Accelerated pistols recognition by using a GPU device. IEE 2017, INTERCON (págs. 1-4). Cuzco: IEEE. https://ieeexplore.ieee.org/document/8079659. DOI: 10.1109/INTERCON.2017.8079659

Olmos, R., Tabik, S., & Herrera, F. (2018). Automatic handgun detection alarm in videos using deep learning soft. Neurocomputing, 66-72. https://www.sciencedirect.com/science/article/abs/pii/S0925231217308196. DOI: https://doi.org/10.1016/j.neucom.2017.05.012

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. Neural Information Processing Systyems (NIPS), (págs. 1-9). https://arxiv.org/abs/1506.01497. DOI: https://doi.org/10.48550/arXiv.1506.01497

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2016). Going deeper with convolutions. IEEE 2015 Conference on Computer Vision and Pattern Recognition (CVPR) (págs. 10059-10066). IEEE. https://ieeexplore.ieee.org/document/7298594. DOI: 10.1109/CVPR.2015.7298594

Xu, S., & Hung, K. (2020). Development of an AI-based system for automatic detection and recognition of weapons in surveillance videos. 2020 IEEE 10th Symposium of Computer Applications & Industrial Electronics (ISCAIE) (págs. 48-52). IEEE. https://ieeexplore.ieee.org/document/9108816. DOI: 10.1109/ISCAIE47305.2020.9108816