Automatic Handgun Detection in images using Deep and Traditional Machine Learning

Sandipan Sarma,[1] Souvik Saha[2] and Pushan Banerjee[3]
sandipan.sarma7@gmail.com[1] souviks.saha8@gmail.com[2]
pushanbjee@gmail.com[3] *

July 11, 2017

Abstract

Using real-time object detection to improve surveillance is a promising application of classification methods. One particular application is the detection of the hand-held firearms. Thus far, previous work has mostly focused on the detection of concealed weapons within infrared data. By contrast, we are particularly interested in the rapid detection and identification of weapons from images which can be extended to surveillance videos. Our first approach involves the use of Faster R-CNN to build a handgun detector for images, where the precision value remains low, possibly due to the limited number of CNN layers used. Then our second approach included the use of SVM classifier where we achieved better precision as well as recall.

Keywords - Faster R-CNN, SIFT, SVM, Region Proposal Network, Keypoint descriptors

1 INTRODUCTION

Security and surveillance issues are a major setback in our society. It requires human intervention to detect weapons in public places. If we use machine learning we can replace this human intervention and use machines to detect such weapons from images. We can then extend the problem to videos which can help us track weapons in real time videos and also activate alarm to inform the security forces. In particular, one solution to this problem is to equip

*This work was done as a part of the project in the proceedings of the Fourth Summer School on Computer Vision, Graphics and Image processing, May 31-July 14, 2017, ISI Kolkata, India

surveillance or control cameras with an accurate automatic handgun detection alert system. Existing studies address the detection of guns but only for X-ray or millimetric wave images. In the last five years, deep learning in general and Convolution Neural Networks (CNNs) in particular have achieved competitive results compared to all the classical machine learning methods in image classification, detection and segmentation in several applications. Instead of manually selecting features, deep learning CNNs automatically discover increasingly higher-level features from data. We aim at developing a good gun detector in photos using CNNs as well as traditional machine learning techniques. A proper training of deep CNNs, which contain millions of parameters, requires very large datasets, in the order of millions of samples, as well as High Performance Computing (HPC) resources, e.g., multi-processor systems accelerated with GPUs. Transfer learning through fine-tuning is becoming a widely accepted alternative to overcome these constraints. It consists of re-utilizing the knowledge learnt from one problem to another related one. There were several challenges pertaining to this specific task that we faced during the design of our project, including (but not limited to):

- the event in which a weapon or part of a weapon is hidden from view by another object (e.g. another hand) or an issue pertaining to the surrounding environment (e.g. poor lighting)
- variety of different types, shapes, and sizes of weapons, leading to a variety of different image sizes and bounding box sizes.
- due to the lack of availability of GPUs, we have had to restrict our training image dataset to 100 images and only three CNN layers for faster computation and learning process.

2 METHODOLOGY

Object detection consists of recognizing an object and finding its location in an input image. The existing methods address the detection problem by reformulating it into a classification problem, they first train the classifier then during the detection process they run it on a number of areas of the input image using the region proposals approach. The approaches used and compared in our project includes:

- Faster R-CNN as a deep learning model (classification and regression used for object classification and bounding box-detection).
- Support Vector Machine (SVM) classifier on Scale Invariant Feature Transform (SIFT) descriptors corresponding to an image.

2.1 Creation of database

We have built a database of 100 images that contains pistols in different contexts and scenarios, downloaded from diverse web-sites, and considered 70% of it to

be the training set and the rest being the test set. We considered a two class model and labeled the pistols for training purpose by providing its localization, i.e., bounding box, in each individual training image. The rest of objects in the image are considered background.

2.2 The Faster R-CNN method

We train a Faster R-CNN object detector for detecting handguns. Faster R-CNN is an extension of the R-CNN and Fast R-CNN object detection techniques. All three of these techniques use convolutional neural networks (CNN). The difference between them is how they select regions to process and how those regions are classified.

In the region proposals approach, instead of considering all the possible windows of the input image as candidates, this approach selects actual candidate regions using detection proposal methods. The first detection model that introduced CNNs under this approach was Region-based CNNs (R-CNN). It generates around 2000 potential bounding boxes using the selective search method, warps the obtained regions into images of the same size then, feeds them to a powerful CNN-based classifier to extract their features, scores the boxes using SVM, adjusts the bounding boxes using a linear model, and eliminates duplicate detections.R-CNN and Fast R-CNN use a region proposal algorithm as a preprocessing step before running the CNN. The proposal algorithms are typically techniques such as Edge Boxes or Selective Search, which are independent of the CNN. In the case of Fast R-CNN, the use of these techniques becomes the processing bottleneck compared to running the CNN. Faster R-CNN addresses this issue by implementing the region proposal mechanism using the CNN and thereby making region proposal a part of the CNN training and prediction steps. The scheme of the experiment is shown in Figs. 1 and 2.

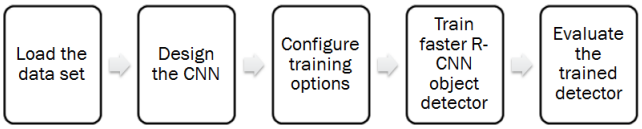

Figure 1: Steps involved in the Faster R-CNN model

2.3 SVM Classifier using extracted SIFT descriptors

A SIFT feature is a selected image region (also called keypoint) with an associated descriptor. Keypoints are extracted by the by the SIFT descriptor. It is also common to use the SIFT descriptor independently (i.e. computing descriptors of custom keypoints).

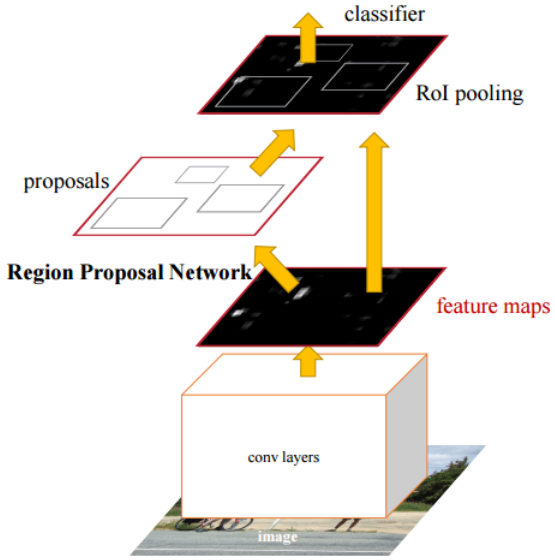

Figure 2: Faster R-CNN with region proposals approach

2.3.1 SIFT

A SIFT keypoint is a circular image region with an orientation. It is described by a geometric frame of four parameters: the keypoint center coordinates x and y, its scale (the radius of the region), and its orientation (an angle expressed in radians). The SIFT descriptors use as keypoints image structures which resemble blobs. By searching for blobs at multiple scales and positions, the SIFT detector is invariant (or, more accurately, covariant) to translation, rotations, and re scaling of the image. The keypoint orientation is also determined from the local image appearance and is covariant to image rotations. Depending on the symmetry of the keypoint appearance, determining the orientation can be ambiguous. In this case, the SIFT descriptors returns a list of up to four possible orientations, constructing up to four frames (differing only by their orientation) for each detected image blob.

2.3.2 SVM

We can use a SVM when our data has exactly two classes. An SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes. Margin means the maximal width of the slab parallel to the hyperplane that has no interior unregularized data points.

The support vectors are the unregularized data points that are closest to the separating hyperplane; these points are on the boundary of the slab. Fig.

3 illustrates these definitions, with + indicating data points of type 1, and indicating data points of type 1.There are different types of kernels: Linear kernel,Gaussian kernel(RBF) and Polynomial kernel to name a few. In machine learning, the (Gaussian) radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. Fig. 4 shows flow of experiment using SVM.

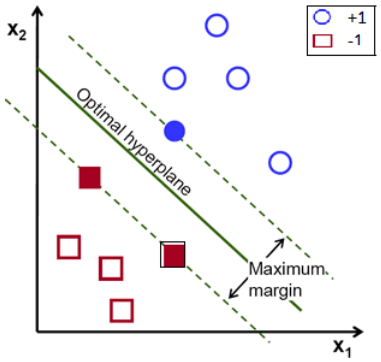

Figure 3: Illustration of SVM

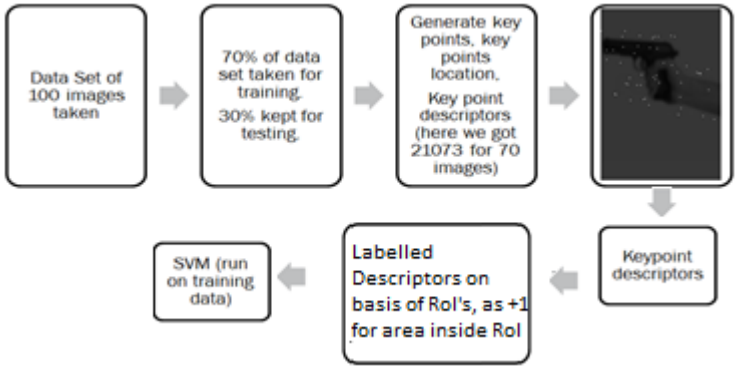

Figure 4: Flow of experiment using SVM

3 EXPERIMENTS

3.1 Faster R-CNN method

Our experiment uses a small handgun data set that contains 100 images. Each image contains 1 labelled instance of a handgun. A small data set is useful for exploring the Faster R-CNN training procedure, but in practice, more labelled images are needed to train a robust detector. The training data is stored in a

table. The first column contains the path to the image files. The remaining columns contain the RoI (Region-of-Interest) labels for handguns. A CNN is the basis of the Faster R-CNN object detector. We created the CNN layer by layer using Neural Network Toolbox in MATLAB. For classification tasks, the input size is typically the size of the training images. For detection tasks, the CNN needs to analyze smaller sections of the image, so the input size must be similar in size to the smallest object in the data set. In this data set all the objects are larger than [16 16], so select an input size of [32 32]. This input size is a balance between processing time and the amount of spatial detail the CNN needs to resolve. The final layers of a CNN are typically composed of *fully connected layers* and a *softmax loss layer*.

An object detector function trains the detector in four steps. The first two steps train the region proposal and detection networks used in Faster R-CNN. The final two steps combine the networks from the first two steps such that a single network is created for detection. Each training step can have different convergence rates, so it is beneficial to specify independent training options for each step. Here, the learning rate for the first two steps is set higher than the last two steps. Because the last two steps are fine-tuning steps, the network weights can be modified more slowly than in the first two steps. During training, image patches are extracted from the training data. After the training is done, testing is done on the test data.

3.2 SVM Classifier on extracted descriptors

In our experiment we used SIFT descriptor for the key points to be used in the SVM classifier. The dataset used in our experiment consists of 70 images where 70% of the images used for the training dataset and 30% used for the test data. Here 70 images are our training set from where the SIFT descriptor extracts the descriptor for the key points. A matrix of 21073 rows and 128 columns of the descriptor values is obtained (each descriptor is a 128-element vector). Fig. 5 shows an example of a training image with the keypoints plotted on it.

Among the detected keypoints for a training image, the ones falling within the RoI are labelled as foreground descriptors, while the rest are labelled as background descriptors. Since the number of background descriptors is very high compared to that of foreground descriptors, we randomly select 1800 descriptors from each of the classes to train the SVM classifier. The chosen descriptors are fed to an SVM classifier and various kernels were tested with. The keypoint precision (KP) and keypoint recall (KR) for different kernels are noted as shown in Table 1 for the training keypoints. The best result was obtained in the polynomial kernel. Other kernels are seen to be not that much efficient in classifying the images. Hence, the polynomial kernel is used for further testing. The SVM classifier classifies the positive samples if there exist the presence of handgun in the images. The predicted bounding box is constructed so as to enclose the predicted foreground keypoints between the 15th and the 85th percentiles to impart some noise immunity.

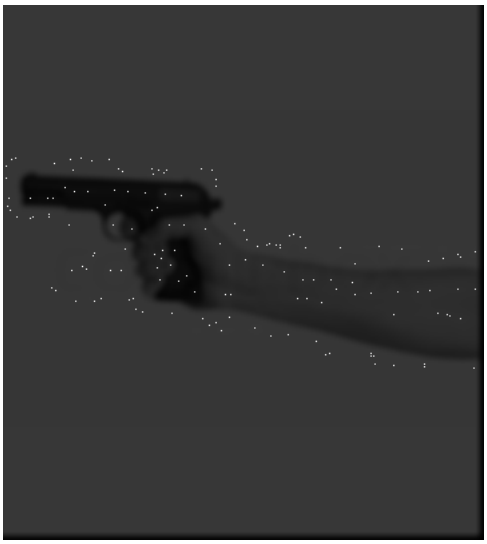

Figure 5: Detected keypoints on a training image

Table 1: Keypoint precision and keypoint recall values with different choices of kernels for SVM

Kernel	**KP**	**KR**	**KP * KR**
Linear	0.16	0.40	0.064
Polynomial (d=2)	**0.25**	**0.85**	**0.213**
RBF (sigma = 0.1)	0.16	0.99	0.158
RBF (sigma = 1)	0.20	0.94	0.188
RBF (sigma = 50)	0.15	0.23	0.034

4 RESULTS

4.1 Faster R-CNN

The method gives us the bounding boxes for our desired object (handgun in this case), along with a score which indicates the probability of being a handgun indeed. But, surprisingly, the performance of this model was not up to the mark. Performance on two sample test images is shown in Figs. 6 and 7. It is seen that the Faster R-CNN often detects multiple RoIs. This results in a lower precision as seen in Table 2.

Figure 6: Detected handgun in a test image using Faster-RCNN approach

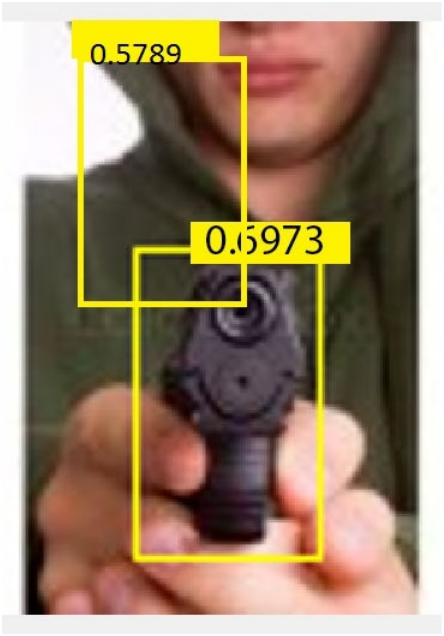

Figure 7: Detected handgun in a test image using Faster-RCNN approach

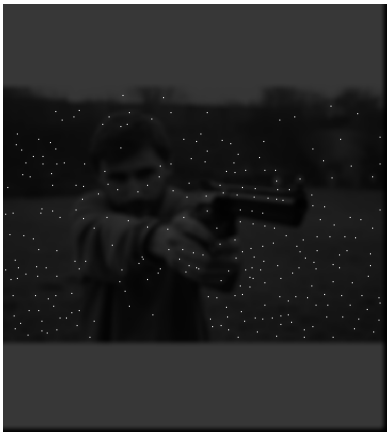

Figure 8: Keypoints as found by SIFT on test image 1

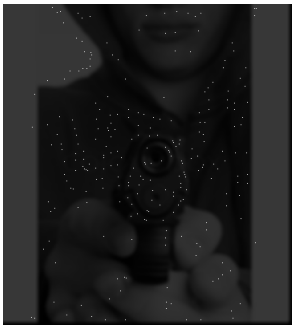

Figure 9: Keypoints as found by SIFT on test image 2

Figure 10: Predicted keypoints after SVM on test image 1

Figure 11: Predicted keypoints after SVM on test image 2

Figure 12: Detected bounding box on test image 1

Figure 13: Detected bounding box on test image 2

4.2 SVM with SIFT descriptors

SVM classified the descriptors quite well, and then bounding boxes were generated for each test image, as in the previous case. The resulting boxes were somewhat larger than the handguns present in the images, but almost all of them detected the handguns inside them. Polynomial kernel is used for this classification and 61.9% accuracy is obtained.

Figs. 8 and 9 show two sample test images with their keypoints as found by SIFT detector. Figs. 10 and 11 show the predicted keypoints of the respective images plotted with red dots. Figs. 12 and 13 show the detected bounding box for respective images.

Table 2: Keypoint precision and keypoint recall values with different choices of kernels for SVM

Model	**Precision**	**Recall**
Faster R-CNN	0.09	0.58
SVM + SIFT	**0.62**	**0.62**

5 CONCLUSIONS

Both the models- Faster R-CNN and SVM with SIFT were ran on the same environment and their properties were compared. Both approaches are independent of the image size. The poor performance observed for Faster R-CNN is due to the small training dataset and due to less number of convolutional layers. But one advantage of this method is that it is likely to be able to detect multiple handguns if trained properly. On the other hand, the SVM and SIFT descriptor reliant approach, while producing good results on the test images, cannot be directly applied for images with multiple guns. A future research direction may be to combine these approaches to overcome the short-comings of both.

References

[1] Roberto Olmos, Siham Tabik and Francisco Herrera, "Automatic handgun detection alarm in videos using deep learning", Soft Computing and Intelligent Information Systems research group, February, 2017.

[2] Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", in Advances in Neural Information Processing Systems 28 (NIPS 2015).

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition", in European Conference on Computer Vision (ECCV), 2014.

[4] R. Girshick, "Fast R-CNN", in IEEE International Conference on Computer Vision(ICCV), 2015.

[5] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", in International Journal of Computer Vision, volume 60, issue 2, pp 91-110, November 2004.