Detection and Segmentation of Firearms using Deep Transfer Learning

Qaiser Abbas
Department of Computer Science and Engineering
University of Engineering and Technology Lahore
Lahore, Pakistan
mqaiser617@gmail.com

Ali Zain
Department of Computer Science and Engineering
University of Engineering and Technology Lahore
Lahore, Pakistan
alizain15@live.com

***Abstract*—Convolutional Neural Networks have been proved as a very useful tool in the real time detection of objects to improve the automatic surveillance. One of this application of CNN is the detection of life-threatening equipment such as firearms (rifles and pistols etc.). Previous work is related to detection of these weapons using hardware solutions. We are more interested in the software solution of this problem by detecting and segmenting the firearms in images and videos using popular image segmentation framework Mask R-CNN. Experiments have proven the effectiveness of our proposed approach for firearms detection and segmentation**

***Index Terms*—CNNs, Firearms, Semantic Segmentation, Weapon Detection, Deep Learning**

I. Introduction

From their inception the CNNs have been used in variety of object detection tasks in computer vision domain. The researchers have used these CNNs in detecting various objects in various fields. This makes CNNs a great tool for coping with the object detection task. In recent times the mass shooting events in different countries have caused huge loss of the precious lives of the people. Most notable of these mass shootings events are the Columbine massacre in USA where 37 persons were attacked, the attack on Uotya Island in Norway affecting 179 persons and the Realengo massacre in Brazil which caused 13 causalities.

To tackle this situation the mainly used systems are CCTVs to inspect the people and detect the weapons to avoid any bad incidents. Despite the installation of these CCTVs systems these unpleasant incidents are occurring at an alarming rate and no proper solution devised yet. This research project tries to tackle the issue of these unpleasant incidents using software approach. The video streams from camera can directly detect the presence of weapons in real time and then this information can be shared with the concerned authority to respond to such events.

Our solution is based on deep convolutional neural network framework called Mask R-CNN which is a semantic segmentation framework based on Faster R-CNN. Our baseline model is ResNet 50 and ResNet 101 for feature extraction which is trained on Coco weights. As the learned features can be used to deduce the important details about the new objects so these coco weights will be used to train our model on our own dataset. As Mask R-CNN is semantic segmentation framework so no proper annotated dataset is available so we have gathered a total of 700 images of weapons for this purpose. We used a python script to automatically download these images from Google. We have trained our TensorFlow based Mask R-CNN model on this collected dataset.

We faced following challenges during this project which are

- No proper annotated dataset is available in firearms domain.
- The framework works on pixel level classification and segmentation of the object so some of the pixels in the object are not well masked.
- We have trained our model on only 520 images which are not enough for a deep learning model although we have used transfer learning approach.

Besides these challenges one main challenge was the slow speed of the framework because this Mask R-CNN is based on faster R-CNN which is slow in speed due to its feature extraction architecture. Based on experiments we found that our maximum performance achieved was using ResNet101 as baseline model with pretrained COCO weights and learning rate of 0.001.

II. Related Work

An automatic system for pistol detection in videos for control and surveillance purposes was presented by the authors. In this work the main focus was to minimize the problem of false positives. The best results were obtained from the Faster R-CNN based model which provides zero false positive with 84.21 % precision [1].

Authors proposed an approach to detect the visible and hidden weapons in the terahertz images. A system was designed with two deep learning based classifiers. The first stage classifier detects the visible weapon and second stage classifier detects the hidden weapons. The GoogLeNet was modified for this two stages classification system in which the last connection layer is replaced with the three layer deed forward network. They achieved 1.72 % for first classifier and 0.12 % for second classifier [2].

An approach was presented to detect handheld guns using Faster R-CNN. In thier work 93 % accuracy was achieved using Faster R-CNN approach. The main objective of this

research was to improve accuracy in detection of handheld guns in real time [3].

In this research work the real time detection of handheld weapons i.e. rifles and pistols is done using CNN approach. They used a Tensorflow based implementation used to detect and classify of weapons in pictures. The training accuracy achieved was 93 % and test accuracy was 89 % [4].

Authors used CNNs to detect Firearm. The model is trained in such a way as it can correctly detect the weapons which are not given in the training dataset. The 96.26 % accuracy has been achieved in this work [5].

They detected firearms using Faster R-CNN approach. To avoid the false detection in the images the objects included skin and cell phones is added in the training dataset. The model is trained on 200 images in which atleast one firearm is visualized and to evaluate the false positive detection rate on 5000 images without firearm [6].

III. Methodology

A. Dataset Preparation

We have downloaded nearly 700 raw images from google using a python script. We have used 520 images from training the model and rest of the images are used for validation and testing purposes. The raw images which were of no use were manually separated and deleted from dataset so model can learn relevant features. As the proper annotated images datasets are not available for Mask R-CNN so we have to download and annotate the images ourselves which is time consuming process. We have used the Oxford's VGG Group's VIA 1.0 annotation tool which is available at the following URL (http://www.robots.ox.ac.uk/ vgg/software/via/via-1.0.0.html). This tool can produce the pixel level annotation instead of bounding box annotations. These annotated images then are used to train our Mask R-CNN model.

B. Dataset Division

In our dataset 520 images are used to train the model and rest of the images are used as validation while we have downloaded another 50 images for testing our model. Following is the division of our dataset

- 70% train set
- 20% validation set
- 10% test set

C. Data Preprocessing

As we are using the deep learning (CNNs) so we don't need to worry about the feature extraction as the CNNs extract features itself. That's why we used raw images directly without any preprocessing.

D. Classification Method

We have used region based Convolutional Neural Network for the detection purpose. This R-CNN works on the region proposals and then works on these regions to identify and detect the targeted objects. We have used Mask R-CNN framework for detection and segmentation purposes.

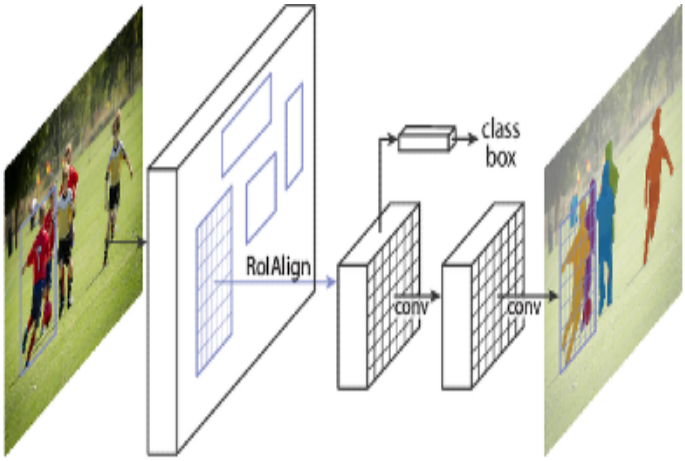


Fig. 1. Architecture of Mask R-CNN.

E. Mask R-CNN

Mask R-CNN is based on the Faster R-CNN but with an additional mask output. Here is the simple working of the Mask R-CNN framework.

- The Mask R-CNN framework is built on top of Faster R-CNN. So, for a given image, Mask R-CNN, in addition to the class label and bounding box coordinates for each object, will also return the object mask.
- Similar to the ConvNet that we use in Faster R-CNN to extract feature maps from the image, we use the ResNet 101 architecture to extract features from the images in Mask R-CNN. So, the first step is to take an image and extract features using the ResNet 101 architecture. These features act as an input for the next layer.
- Now, we take the feature maps obtained in the previous step and apply a region proposal network (RPM). This basically predicts if an object is present in that region (or not). In this step, we get those regions or feature maps which the model predicts contain some object.
- Then we apply a pooling layer and convert all the regions to the same shape. Next, these regions are passed through a fully connected network so that the class label and bounding boxes are predicted.
- Once we have the RoIs based on the IoU values, we can add a mask branch to the existing architecture. This returns the segmentation mask for each region that contains an object. It returns a mask of size 28 X 28 for each region which is then scaled up for inference.

F. Training

Once we have collected, annotated and divided the data in test, train split we can now finally start training the model to detect the guns in images. For this purpose the awesome Matter-port GitHub repository provides us the command line tool to train our model. Depending upon different parameters such as learning rate, weight decay and baseline architecture, the model can take from 2 to 3 hours to train. Our model took 2 and 1.5 hours respectively for training. We used Google

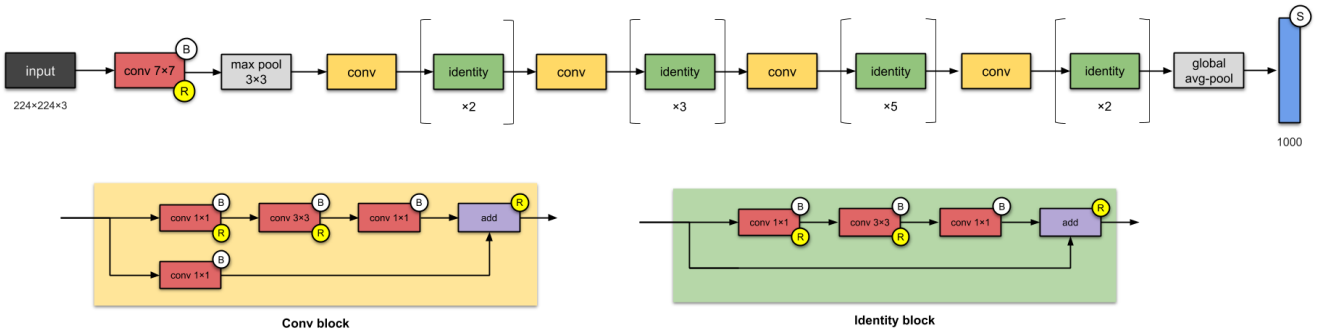


Fig. 2. ResNet 50 Architecture

Colaboratory for training purpose which provides free access to use Tesla K80 GPU it also provides a total of 12 GB of ram , and one can use it up to 12 consecutive hours.

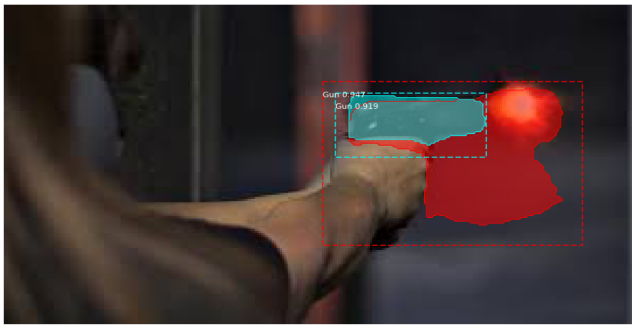
IV. Results

Our detection and segmentation results were far better than we expected. With ResNet 101 as baseline model and learning rate of 0.001 our Mask R-CNN model achieved a great segmentation and detection accuracy. Although we have a small dataset of nearly 700 images our results are amazing. Some of the results that we achieved are displayed in the following images.

Fig. 3. Detection Results

Our detection accuracy is remarkable on validation data and segmentation accuracy is also very good. But as it is machine learning algorithm so some of the true negatives are recorded in which mask generation is also not better. Besides this we added some noise to images so we can analyze the system behaviour. Although the system detected the guns in noisy images but segmentation mask was not noticeable. Following are some examples of our negative results.

Fig. 4. False Results.



In the following image the noise was added to test the system and evaluate performance. The result is not so good.

V. Experiments

We trained these models which are

- ResNet 101 with 20 epochs and 0.001 learning rate using pretrained COCO weights
- ResNet 50 with 10 epochs and 0.0001 learning rate using pretrained imagenet weights
- ResNet 101 with 20 epochs and 0.001 learning rate using pretrained imagenet weights
- ResNet 50 with 10 epochs and 0.0001 learning rate using pretrained COCO weights

Other parameters were untouched as the number of parameters are a little bit high so we only tested with theses experiments.

Baseline	Weights	Learning Rate	Epochs
ResNet101	COCO	0.001	20
ResNet50	Imagenet	0.0001	10
ResNet101	Imagenet	0.001	20
ResNet50	COCO	0.0001	10

Our results shows that best performance was achieved with ResNet 101 with 0.001 Learning rate using pretrained COCO weights.

VI. Comparison

As per our best knowledge no author previously has worked on firearm segmentation task. So we can not compare our segmentation results with some existing literature although our accuracy and segmentation is quite good.

VII. Conclusion

In this project we developed a firearm detector which is based on Mask R-CNN framework. This framework is based on Faster R-CNN. The algorithm we used is a two-stage detector in which first the target objects are passed as proposed regions to the system. The second stage of algorithm does the detection and segmentation task. The ResNet 101 is used as baseline model for feature extraction and then Mask R-CNN generates the masks and bounding boxes where the firearms are detected. As there is no publicly available firearm dataset for segmentation so we have collected and annotated the dataset.The model performed better by using pretrained COCO weights along with ResNet 101 as baseline model and 0.001 learning rate with 20 epochs. The model's segmentation and detection accuracy is satisfactory and can be used in real time with some enhancements. We used only 700 images so if we increase the dataset and then train model for more epochs then accuracy and segmentation can be enhanced.

VIII. Future Work

In future we aim to develop a large dataset for firearm segmentation for improving its accuracy and we aim to train model for more epochs along with some other parameter fine tuning for better performance.

References

[1] R. Olmos, S. Tabik and F. Herrera, "Automatic Handgun Detection Alarm in Videos Using Deep Learning", Soft Computing and Intelligent Information Systems research group, University of Granada, 18071 Granada, Spain.

[2] J. Yuan and C. Guo, "A Deep Learning Method for Detection of Dangerous Equipment", 8th International Conference on Information Science and Technology June 30-July 6, 2018; Granada, Cordoba, and Seville, Spain.

[3] G. Verma and Anamika Dhillon, "A Handheld Gun Detection using Faster R-CNN Deep Learning", National Institute of Technology, Kurukshetra, November 2017.

[4] Justin Lai and Sydney Maples, "Developing a Real-Time Gun Detection Classifier", Stanford University.

[5] R. Fumihiro, A. Kanehisa and A. Almeida, "Firearm Detection using Convolutional Neural Networks", Federal University of Maranhao (UFMA), São Luís, Brazil.

[6] E. Valldor, K. Stenborg and D. Gustafsson, "Firearm Detection in Social Media Images", Swedish Defence Research Agency, Sweden.