A semantic segmentation approach to recognize assault rifles in ISIS propaganda images

Abeysinghe, B.; Veilleux-Lepage, Y.D.; Bloom, M.; Sunderraman, R.

A Semantic Segmentation Approach to Recognize Assault Rifles in ISIS Propaganda Images

Bhashithe Abeysinghe
Department of Computer Science
Georgia State University
Atlanta, USA
tabeysinghe2@student.gsu.edu

Yannick Veilleux-Lepage
Institute of Security and Global Affairs
Leiden University
Leiden, Netherlands
y.d.veilleux-lepage@fgga.leidenuniv.nl

Mia M. Bloom
Department of Communication
Georgia State University
Atlanta, USA
mbloom3@gsu.edu

Rajshekhar Sunderraman
Department of Computer Science
Georgia State University
Atlanta, USA
raj@gsu.edu

***Abstract*—Between 2014 and 2018, the Islamic State in Iraq and Syria (ISIS) perfected the use of social media for its propaganda. To understand and counter these efforts by ISIS, it is critical to analyze their propaganda materials. During the past few years, a systematic effort has been made to catalog and annotate these materials which appear in the form of images, video and text. However due to the sheer volume of the material, it is an extremely onerous task to maintain. In this work, we present a deep learning solution to automatically identify and tag images for assault rifles. We present our experiments of a semantic segmentation approach to localization of assault rifles in a self-collected and maintained data set. Our goal is to consume minimal amount of data and cater to an analysis platform. The state of the art for object localization is the Convolutional Neural Network (CNN). A limitation of CNN is that it only handles images of fixed dimensions. One way to deal with this limitation is to re-size the input images, however this is not an ideal solution. A more flexible approach is to use a Fully Convolutional Network (FCN), which provides a robust solution for varied sizes of input images. We show that FCNs can achieve high performance in detecting and localizing objects in a real world setting, with non-curated data. We also show that by using a step wise training pipeline it is possible to learn a representation of the object using a bounding box annotation.**

***Index Terms*—localization, Fully Convolutional Networks, assault rifles, semantic segmentation.**

I. Introduction

The Islamic State in Iraq and Syria's (ISIS) Virtual Caliphate [1] is still active and disseminating propaganda despite recent aggressive actions taken by social media companies. Since the declaration of the "Caliphate" in 2014, ISIS has taken jihadist media production to new heights, releasing a steady stream of images and videos to circulate Jihadi messages and attract new recruits from the region and globally.

This research is partly funded by the Department of Defense's Minerva Research Initiative (Documenting the Virtual Caliphate Minerva #N00014-16-1-3174) and the Office of Naval Research. All opinions are exclusively those of the authors and do not represent the Department of Defense or the Navy

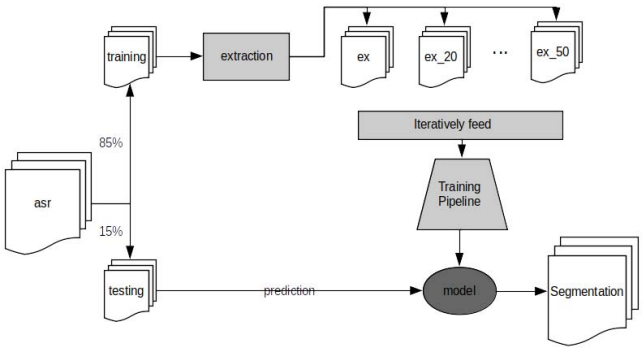

Fig. 1. Brief illustration of the methodology

Ubiquitous tools and technology help in content creation and propagation, and all this is happening at very large volumes and frequencies. It is important to note that analyzing these propaganda materials will help in turn regulating and ultimately creating a safe environment [2]. Due to the volume of multimedia generated, it is humanly impossible to analyze or even to study the material systematically. Usually the requirement to analyze the propaganda material assumed time constraints. In order to handle the volume and velocity of data, a seamless system is needed from which it is possible to observe all the platforms where these propaganda images/videos/texts are circulated. Also when said media is graphic and can incite violence, adverse effects for the attitude/mentality of scholars studying these materials is inevitable. Since there are many outlets from where the propaganda is shared it becomes rather tedious and time consuming to analyze all this material manually. These factors dictate that an automated and seamless system should be designed and implemented to analyze the materials. Besides, it is evident that the system should be constructed using the principles of data science.

978-1-7281-6332-1/20/$31.00 ©2020 IEEE
DOI 10.1109/ICSC.2020.00082

A. Overview

Due to more aggressive screening and social media companies implementing their terms of service to delete propaganda material from traditional applications such as Twitter, Facebook and YouTube, much of this terrorist propaganda is becoming scarcer by the day, but more privacy and anonymity-centric social media are being used as tools for the dissemination of propaganda. One such media platform is Telegram. On Telegram, ISIS manipulates an environment rich with addictive properties, creating online spaces that encourage group identity, shared opinions, and dominant ideologies, while exploiting the individual "need" to be a part of the group. "Documenting the Virtual Caliphate" is the first up to date, multilingual, and searchable database of archived ISIS visual propaganda. This database represents a living archive of ISIS' official multimedia content and provides the grounds for a systematic analysis of ISIS materials by researchers.

Our research team has collected data from these networks which distributes a form of material known as *photo reports* and created a file system database. This is usually a set of images related thematically and by location compiled into a PDF document. The photo reports are released by ISIS regulated outlets therefore while authenticity is important we believe that the data is authentic and outside intervention on these images is minimal. It is necessary to extract images from these photo reports, while there could be insignificant images ('word arts', 'arrows', 'insignia' etc.) in these documents in order to feed them into the deep learning framework; presently we need to filter the images generated by the PDF image extractor. We break down the data set into images and store them using the date and an index given to the image according to the order ISIS organized them in the photo report. The data set includes numerous objects of interest. During the first phase as a proof of concept, we determined to localize assault rifles in the propaganda images. Since we are creating a platform for a analysis framework, we will need the counts of objects appearing in the image. Therefore we need a object detection or a semantic segmentation or a hybrid model to satisfy this requirement rather than just a classification model. To this end we create a precise deep learning model so that it is possible to distinguish comparable types of weapons in the images such as large machine guns and various other hand guns versus assault rifles. The framework should be accurate enough to detect the difference between these objects while being fast. The main difficulty in using deep learning models is the limitation of annotated data. Even though we have 19,000+ images at hand, the annotated subset is quite small in size; Only 905 images.

Typically these photo reports are formed using different photos taken by different devices, this results in images with a variety of dimensions. Due to this it is not possible to use CNNs as the architecture for a deep learning solution [3], [4]. CNNs use a fully connected layer at the end and the number of neurons in the fully connected layer is computed using the dimensions of the previous feature vector. Dimensions of the feature vector depends on the input image and the convolution and pooling operations in the CNN [4]. This led us to use FCNs in our experiments due to its robustness for the image size. In an FCN, we have one convolution layer and an up-sampling layer instead of a fully connected layer at the end [3].

We discuss how the various models of FCN as proposed in [5] are behaving and performing in our data set. We also conduct experiments in different types of data generated and extracted from the same data set.

B. Problem Statement

The goal of the research is to design and implement an AI framework based on Deep Learning techniques to document and analyze terrorist propaganda images. The deep learning platform should be fast enough and accurate enough to support the requirement of the AI framework.

A subset of the 19,000+ images data set was extracted that contained images of assault rifles. We shall call this subset as the assault rifle data set (asr). In the assault rifle data set there are images of various dimensions and we were required to create a deep learning framework to localize assault rifles. While the framework should have an acceptable accuracy rate, it should also be fast enough to handle our large 19,000+ data set. The framework should also be able to provide analysis of how many assault rifles are in use in all of these images.

The rest of the paper is organized as follows: In Section II we present a survey of significant methods used in object detection and localization. Then, in Section IV a brief summary of the procedures and methods are laid out. Finally, in V we report the results and present a discussion.

II. Related Work

Object localization is not a new topic for computer vision. First use of neural networks goes back to [6]. We conduct experiments of the behaviour of FCNs with limited annotations on a new and noisy data set. Therefore we take inspiration from transfer learning [7] and already established and state of the art of object recognition principles [8] [4] [5]. It is a well known fact that stacking layers into deep architectures will allow a neural network to learn complex patterns. This was first proven by [8] and then again by [9]. It is important to note that both of these models are CNN based models and intuitively, stacking more layers will increase the number of

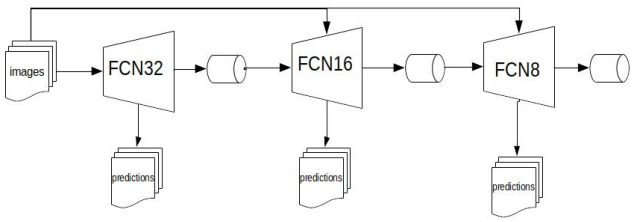

Fig. 2. Training pipeline

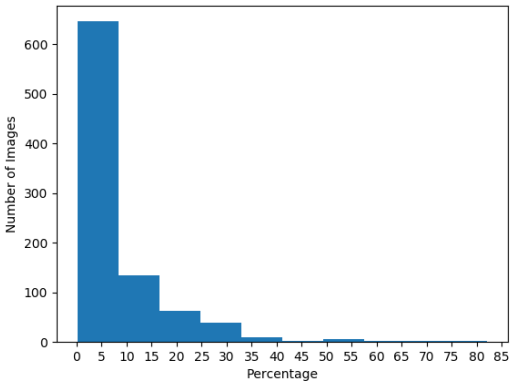

Fig. 3. This shows the distribution of object of interests area to the whole image area as a percentage

parameters to train. However, the architecture of CNNs will significantly decrease the number since they are learning the filter weights using back propagation, instead of the pixel wise relations [4].

Region based Convolutional Neural Networks (RCNN) [10] and its derivations are now a canonical standard for object detection; they improve the speed of detection by introducing a regression based solution. RCNNs using selective search [11] extracts proposed regions and instead of using a *window scan* mechanism, RCNNs use a combination of simple machine learning techniques such as Support Vector Machines or Linear Regression [3] to come up with the four corners of a bounding box . Fast R-CNN [12] inspired by RCNNs incorporates two augmentations for the RCNN algorithm which makes it much faster compared to RCNNs. Fast R-CNNs use CNN to extract features before the selective search which makes the search space smaller and instead of the traditional machine learning techniques, this uses a *softmax* layer to compute the regression output. There is another variation for this model which is known as Faster R-CNN [13]. Among other augmentations the significant speed improvement was due to Faster R-CNN got rid of the slow selective search and incorporated a region proposal network [13] [14].

The major limitation of CNNs which prevents us from applying CNNs for this data set is that CNNs require that the images be in a fixed dimension; Our data set consists of images of various sizes and resolutions. This is because as explained earlier, the images are acquired, edited, and circulated by different sources. Fixing or standardizing the dimension of the image is also a sub-optimal solution because figure 3 shows that most images have the object of interest concentrated in a small area and this will make it difficult for the network to learn the actual representation of the assault rifles and detect, rather it will learn the background information which is abundant in the image. In [15] this can be seen as they detect both 'L' shaped objects and firearms at the same time. Also, since [15] is a CNN based network it is not possible to use this to detect assault rifles in our data set. There is other research done on the same topic [16] [17] with CNNs achieving good performance, since authors are using CNNs the techniques used there is not applicable in our data set. While specific work done on related fields are not applicable, general detection platforms based on CNNs such as RCNN, Fast-RCNN, Faster-RCNN and YOLO [18] are not applicable as well.

Fully Convolutional Neural Networks are almost identical to the architecture to the CNNs. The difference is at the final layers. FCN architecture allows it to process arbitrary sized input [5]. The theory behind the architecture is that instead of using a fully connected layer to classify the pixels of an image, we use the pooled smaller images [5] and feed it into an up-sampling layer. This process brings the image back to its original proportions. Also, this provides the pixel wise output, which makes it very easy to apply in semantic segmentation. This up-sampling can be achieved by using up-sampling layers (also known as deconvolution layers). It is possible to obtain this up-sampling at any point of convolution in the network, if it is done at a shallow layer one needs to up-sample using a higher factor [5]. This is due to the fact that complex features are learnt in deeper levels [8] [9]. [5] showing that at different depths of their network up-sampling and fusing the outputs it is possible to get various granularities for the output.

There are variants of FCN such as Region based Fully Convolutional Network (R-FCN) [19]. This also improves the speed of detection by incorporating region proposal into the FCN.

III. Data

We are using several data sets to train and test the model. These were created because in initial experiments, the network failed to learn from the original images. After conducting additional experiments, we discovered that the model learns object representations if it received objects without distractions, such as background and other complex objects. Hence we created a series of data sets which are shown in Table I. There are different ways of extracting the object of interest from the image. You can extract the object using a buffer zone and save the image. If there are multiple objects in the image, you can extract each object separately and save them each as separate images. Also, you could create one image which includes all of the objects in the same image. Doing so the network will learn that there are more than one object of interest in the image while learning its object representation. Also, by changing the buffer size of the image we can control the context of the object of interest the model see. We created several data sets having different buffer sizes as well.

Moreover, since there are similar objects for interested object; grenade launchers, light machine guns etc. we needed to teach the network what comprises the features of a non-assault rifle. To this end we added negative images data set.

IV. Methodology

The assault rifles data set consists of 905 images with annotated assault rifles using a bounding box. There are images of various dimensions and in most images the object of interest is small. The distribution of the area is as illustrated in figure 3. This presented a significant problem for the training of the model. The initial deployments of the model failed to learn the distributed representation of the assault rifle. We took two measures, using pre-trained weights and then to augment the data set in several steps to make it easier for the model to give attention to the salient features of interest. Worth noting that before augmenting the data set, we first divide the assault rifle data set into training and testing sets and the following augmentations were done only for the training set. The testing was done on the data subset which was not modified, hence our claim of learning the object representation still stands. An illustration of the training procedure is shown in figure 1.

Training in our experiments is a pipeline procedure, there are different components and each component is a FCN model. First the data set is split into training and testing sets and augmentations are applied to the training set. The training set then will be used to extract RoIs in the image. Likewise several types of data sets will be created as explained in IV-A, each data set will be fed into the pipeline to learn a specific task. Each component of the pipeline is discussed below.

A. Extraction of the object

In initial training we extracted the object with a small buffer zone of 10 pixels around the object. This allows the neural network to see more pixels of 'assault_rifle' and less 'background'. This will give an image with at least 80% of the area is the object of interest. This grants the network to see the object of interest without getting distracted by the surrounding context. The network will be trained and predictions and the model weights are saved as outputs.

In some images, there are multiple assault rifles, because of data limitations we extracted all rifles as individual 'extracted' images and we left images with multiple rifle so that the model eventually learns to detect multiple objects. This exponentially increased the data set size to a 2306.

B. Context learning

After that we changed the size of a buffer zone, and then increased the buffer zone with iterative steps of 10 pixels and create different data sets. Buffer zone is the area surrounding the bounding box. Note that we are not changing our annotations with the buffer zone increase and this does not change the size of the data set because we will only be changing the size of the buffer to generate new data, not mixing data with the previously generated data. We will be using the previously learned and saved model outputs in this step transferred into this model as start up weights. This allows us to reserve the previously learned features in the model while learning new features. Since here we are increasing the buffer zone, while the model sees the object from the previously learned weights, it also detects the context of the objects location. Iterative steps of this buffer zone should not increase more than a certain point because it will again have the same problem of small object of interest and we identified that to be 50 pixels.

TABLE I
GENERATED DATASETS

Data set	Description
asr	Assault rifle dataset
asr_ex	asr + extracted images, 10 pixel buffer
asr_ex_mul	asr_ex + extraction for each object
asr_ex_mul_ [number]	[number] size buffer

C. Training

In preliminary testing we found out that starting with weights from [8] yields the best results at the later stages of the pipeline. Therefore complete testing and evaluation here onwards is done only using VGG16 weights. In any stage, the training will go through 3 different phases illustrated in figure 2. There are 3 granularity levels for FCN model proposed in [5], FCN32 is more coarse or rough while FCN16 and FCN8 are more finer and tighter when detecting the object. We created a pipeline for training where any image data set generated in any of the previous stages go through all FCN32, FCN16 and FCN8 models. FCN32 will be started off with VGG16 [8] weights. VGG16 here is pre-trained on ImageNet [20] data set. In each phase, the model weights will be transferred into the next level while all predictions are being recorded. Which means that FCN32 model starting with VGG16 will be trained and the model which performs the best in all the epochs will be stored as the best model, this model weights are then being transferred to FCN16. This is possible because the configuration of each model are very similar. For further reading authors direct the readers to [21]. Then advancing along the pipeline the are finer and tighter around the edges and will represent a correct object than a bounding box (semantic segmentation). And when it is learning the context of the object it learns to omit human hands and other similar objects which occlude the object of interest (refer to figure 4).

The underlying network is VGG16, and similar to [8] in all convolutional layers we have rectified linear unit type

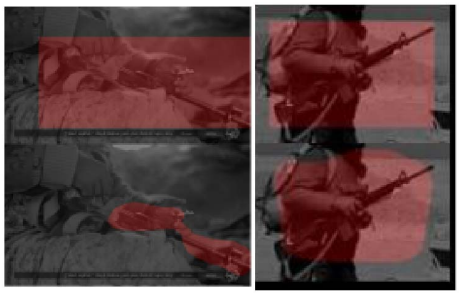

Fig. 4. This shows two examples of tight fitting when extracted images used in training

TABLE II
EXPERIMENTS OVER DIFFERENT DATA SETS

Data set	Accuracy	IOU
asr	71.3	63.9
asr_ex_mul	79.3	47.9
asr_ex_mul_50	**80.3**	**55.7**
asr_neg_ex_mul	50.0	63.7

of activation. And all ROI pooling will be done with a max pooling type of operation. In each up-scoring (or de-convolutional) layer, we have a relevant stride parameter too. For FCN32, we are using 32 stride in one of the last up-scoring layer (out of 2 up-scoring layers), in FCN16 we have 16 stride in the last up-scoring layer (out of 2 up-scoring layers) and for FCN8 we have 3 up-scoring layers and from which the middle one will have 8 stride. All up-scoring layers will not implement a bias.

Fully convolutional layers in between the convolutional layers and the up-scoring layers will have a 0.5 drop out to handle generalization of the model. And we are optimizing the network with stochastic gradient descent. Since we are transferring the weights of [20] from [8], the network converges quickly. Authors used about 150 epochs to train the network in all experiments. And authors are using cross entropy loss fixed for the 2-dimensional feature vectors.

D. Negative images

A negative image is an image where you do not find an assault rifle, these images could be any form of other firearms, any other scene from our data set. A data set with negative images incorporated into the original training data set was also created.

E. Testing

In each stage of training using validation data sets and mini testing sets to evaluate the model performance in each stage and to log them. Using the initial testing set created from the original assault rifles data set, we test each model at the end of the pipeline and report the performance. In each stage of the training phase we evaluate the model using the mini testing sets created using the main training set.

Fig. 5. Evaluation outputs on the pipeline, from left FCN32, FCN16 and FCN8

V. EXPERIMENTS

As discussed in Section IV-C, we have different phases of training and each phase will output evaluations using mini testing sets, which are from the respective data set in which the model is being trained. These results can be used to explain how the model is behaving at each stage or component of the pipeline, such as to update the weights taking the model performance at that particular stage. We can use the trained FCN8 to detect object without having to go through the pipeline. This makes the test time quite fast.

Metrics of the results are as follows, Accuracy is set to be pixel based accuracy per class, meaning that each pixel of the image is compared with the annotation to test whether the network predicted the correct output. Since the problem is defined as semantic segmentation pixel wise accuracy is a good measure. IOU here in table II is the intersection over union, since we are ultimately looking for an object we also have to consider and evaluate the proportion of the area of the object which is correctly detected.

A. Faster-RCNN on the data set

In order to compare our FCN model performance, we used Faster-RCNN [13], [22] as a baseline on the data set. We re-trained [13] to detect assault rifles using our annotations then used that model to perform detection on our testing data. We also implemented and measured the pixel based accuracy on Faster-RCNN to compare the two models. Faster-RCNN achieves much higher accuracy levels than FCN model. It achieves 92.72 pixel accuracy against the 80.3 that the FCN generates.

While Faster-RCNN consumes 3-8 images per second to generate full bounding boxes, FCN model consumes 16-19 images per second and generates a localization map, which is more than double the speed. But, to compute full bounding boxes, the FCN model should be set inside a pipeline where the final outputs are bounding boxes rather than a segmentation.

B. Segmentaion with FCN

We again tested FCN model for a segmentation task with sample images annotated with segmentation of rifles. This yields a class based accuracy of 66.48 and an IoU of 42.27.

All models have been created and tested with Pytorch [23] and Pytorch-fcn [24] using two NVIDIA 1080Ti totalling up to 18GB GPU RAM.

C. Discussion

Our best model accuracy was 80.31%, due to the fact we are evaluating against bounding boxes and our predictions are tighter around the borders we are at a disadvantage. It is interesting to see the model behaviour throughout each stage of the training; first we feed original images without any modification to the network, resulting in an accuracy of 71.3. This is a good benchmark to test our other models. IOU here is larger because the predicted objects were also bigger and ultimately making the intersection larger. Then we extracted

objects out of the images and fed it into the network, this improves the accuracy by 8%, but as a trade off we can see the IOU values decrease. The explanation for that is from now on our predictions become much more smaller making our intersection smaller and ultimately IOU value smaller.

Then we train a model with larger buffer zone, this gives the best results we have obtained which is 80.31%. We can also observe an increase of the IOU, Now the predictions are much larger. When we mix in negative images into the training samples our model gets lost in the gradients and fails to learn anything. And it seems like that it fails to keep the previous knowledge as well.

This FCN model which is trained for localization does not perform well with a classification task for assault rifles because we first start our training process with all images given has an assault rifle present. Therefore the network expects all the images to have assault rifles in as well. This network can act as a part of a pipeline where the network extracts the interest areas of a assault rifle similar to [12].

Another interesting fact about the model is that even though we are mostly accurate in localizing the object there are disconnected and isolated smaller pixels in the area around the detected region. A post processing stage is inevitable in a case like this, because for the system to count the number of objects there should not be isolated objects predicted incorrectly. We could obviously use some image processing techniques to remove these.

The best model can process an average of 9 images per second, depending on the size of the image the speed of processing changes. There are instances where the model processed 16 images per second. Comparing with a human annotator this is a rather significant increase in speed.

VI. Conclusions

The initial problem statement was to create a seamless system to localize assault rifles in images. We have proven that even though not as accurate as a human annotator, we can localize assault rifles with a substantial accuracy rate. Our method is also reasonably fast, with a rate of 9 images per second. To this end we need to have another model which can classify an image as an image with a assault rifle or not. But by generalizing the model it would be possible to remove the redundant model. In order to do this we must train the model with more classes, ultimately generalizing the model to all the classes.

References

[1] M. Bloom and C. Daymon, "Assessing the future threat: Isis's virtual caliphate," *Orbis*, vol. 62, no. 3, pp. 372 – 388, 2018. Stabilizing the Fertile Crescent after the Fall of the Caliphate.

[2] C. K. Winkler, K. E. Damanhoury, A. Dicker, and A. F. Lemieux, "The medium is terrorism: Transformation of the about to die trope in dabiq," *Terrorism and Political Violence*, vol. 31, no. 2, pp. 224–243, 2019.

[3] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recogn.*, vol. 77, pp. 354–377, May 2018.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, p. 2012.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[6] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proceedings of the 2Nd International Conference on Neural Information Processing Systems*, NIPS'89, (Cambridge, MA, USA), pp. 396–404, MIT Press, 1989.

[7] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition."

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[9] 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 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, 2016.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, (Washington, DC, USA), pp. 580–587, IEEE Computer Society, 2014.

[11] J. R. Uijlings, K. E. Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vision*, vol. 104, pp. 154–171, Sept. 2013.

[12] R. Girshick, "Fast r-cnn," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, (Washington, DC, USA), pp. 1440–1448, IEEE Computer Society, 2015.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, (Cambridge, MA, USA), pp. 91–99, MIT Press, 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015.

[15] A. d. A. N. Rodrigo Fumihiro de Azevedo Kanehisa, "Firearm detectio using convolutional neural networks," 2019.

[16] D. G. Erik Valldor, K-G Stenborg, "Firearm detection in social media images," 2018.

[17] 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, (New York, NY, USA), pp. 84–88, ACM, 2017.

[18] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[19] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, (USA), pp. 379–387, Curran Associates Inc., 2016.

[20] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei, "Imagenet: A large-scale hierarchical image database," in *In CVPR*, 2009.

[21] O. Matan, C. J. C. Burges, Y. Le Cun, and J. S. Denker, "Multi-digit recognition using a space displacement neural network," in *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, (San Francisco, CA, USA), pp. 488–495, Morgan Kaufmann Publishers Inc., 1991.

[22] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron." https://github.com/facebookresearch/detectron, 2018.

[23] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[24] K. Wada, "pytorch-fcn: PyTorch Implementation of Fully Convolutional Networks." https://github.com/wkentaro/pytorch-fcn, 2017.