Firearm Detection in Social Media

Erik Valldor and David Gustafsson
Swedish Defence Research Agency (FOI)
C4ISR, Sensor Informatics
Linköping
SWEDEN

erik.valldor@foi.se david.gustafsson@foi.se

ABSTRACT

Manual analysis of large data sets is often not feasible due to time and cost constraints. In many cases, images make up a large part of the data, and often contain lots of valuable information. Objects and symbols in images provide important information about links with criminal groups and terror organizations. Deep learning has recently achieved remarkable results in tasks such as finding and classifying objects within an image, but often require huge sets of training data to be effective.

We present a deep neural network that is able to detect firearms in images, with the intention of being able to automatically detect weapon posers (i.e. a person that wants to brag about their access to weapons) in social media. The detector is based on an "off-the-shelf" deep neural network that has been pretrained on an open data set to detect common objects. We retrain this network on the task of detecting firearms. To obtain a sufficient amount of training data we use images extracted from feature films where firearms are visible.

We found the detector to perform well on the evaluation data and generalize well to the application domain.

1.0 INTRODUCTION

Images are traditionally more opaque to automated analysis compared to, for example, text data. However, the past few years have seen an immense improvement in the performance of automated image analysis methods. The main source of this improvement are the machine learning techniques known as Deep Learning. Deep Learning has become an extremely popular research field, and as a result, there exists a large number of tools, in terms of software frameworks, to easily get started with Deep Learning based applications.

One specific type of image analysis method where Deep Learning has been shown to excel is *object detection*. Object detection can be defined as the task of determining whether an image contains any objects out of a set of given classes, as well as determining each objects location and size within the image.

From the perspective of information retrieval, object detection gives information about what objects are present in an image, how many objects there are of a given class, as well as the spatial relationship between these objects.

Today, object detection can be considered a relatively mature technique, and there exists good support for this type of analysis in popular Deep Learning frameworks. There also exists a number of "off-the-shelf" models that can be downloaded and used for inference without performing any training on your own. These models are however limited to detecting the types of objects that exists on the dataset they were trained on. This is typically some large public dataset such as MS-COCO [4] or ImageNet [1]. Many of these classes can certainly be useful for the present application, but there is also a need to be able to extend this list of

classes to other object types that are of interest.

As a case study, we develop a detector for firearms. The goal is to investigate what is required in terms of time and resources in order to develop a detector for a new object type.

2.0 BACKGROUND

Our firearm detector should be considered a part of a larger analysis framework, where other object types or visual information, as well as data of other modalities (such as text) is analysed and aggregated.

Practical applications of such a framework in general, and the firearm detector in particular includes:

- **General forensic analysis of images**. Here we consider both images from social media and images found on hard drives and mobile phones. Detection of different objects, including general objects and weapons, and store the result in a searchable database is very important in many investigations. The general investigation usability will not be discussed more in this paper but is one important application of the developed detector.

- **Disinformation and troll detection on internet.** Internet troll commonly post similar, but not identical, information on different forums. Images containing details that will provoke strong feelings (and potential reactions) are commonly posted. Images containing weapon often cause strong feelings for different reasons; it can be images from an ongoing violent conflict or images from criminal activities. Detection of similar images containing firearms posted or shared during a shorter time-span can give information about a disinformation operation.

- **Detection of lone actor (lone wolf) terrorists.** Detection of lone actor terrorists is of crucial importance for many law enforcement agencies, and is the main application discussed in this article. Object detection by itself is not sufficient to detect these persons. Instead it should be considered to support detection of so-called weak signals that (combined with other information) can be used to detect possible lone actors.

2.1 Lone actor detection

Early detection of lone actor terrorists is of crucial importance for law enforcement agencies. Unfortunately, detection of such persons is a very hard problem that requires information gathering using many different sources and techniques. In this section we will discuss how image analysis and more specifically, detection of firearms, can be used in the search for lone actor terrorists. Image analysis by itself is not sufficient to solve this problem. Instead, it should be seen as a complement to other information sources and processing methods that aims at detecting high risk persons. The following presentation is based on research by Kaati & Johansson [2], and focuses on weak signal detection based on text analysis in social media. Our focus is on how image information can be integrated in such a weak signal framework.

A weak signal is defined as information that is not sufficient by itself to detect a lone actor. Instead, a large number of weak signals can increase or decrease the probability of a hypothesis about a person's potential activities. Meloy et al. [3] proposed a list of warning behaviours that precede acts of violence (see Table 1). Both the response to other peoples posted images (e.g. likes), and images posted by the actor are considered potential weak signal warnings.

Table 1: List of warning behaviours proposed by Meloy et al. [3]. The column to the right indicate if image – posted or response – are of relevance for the specific warning.

Warning signal name	Description	Relevance for weapon detection
Pathway	Behaviours including the planning, preparation and implementing of an attack.	Minor
Fixation	Pathological preoccupation of a person or a cause	Minor
Identification	Warrior mentality, associating closely with weapon.	Highly
Novel aggression	Capacity of violence	Highly
Energy burst	Increased activities close to the day of the attack	Minor
Leakage warning	Information about the planned attack is communicated to third parties.	Minor
Directly communicated	Direct threat	Minor
Last resort behaviour	Increasing desperation or distress.	

Some of the behaviours/warnings relates to the *capability*, *intent* and *opportunity* to carry out an attack. Responses on firearm images and posting of firearm images mainly contribute to identification and aggression warnings.

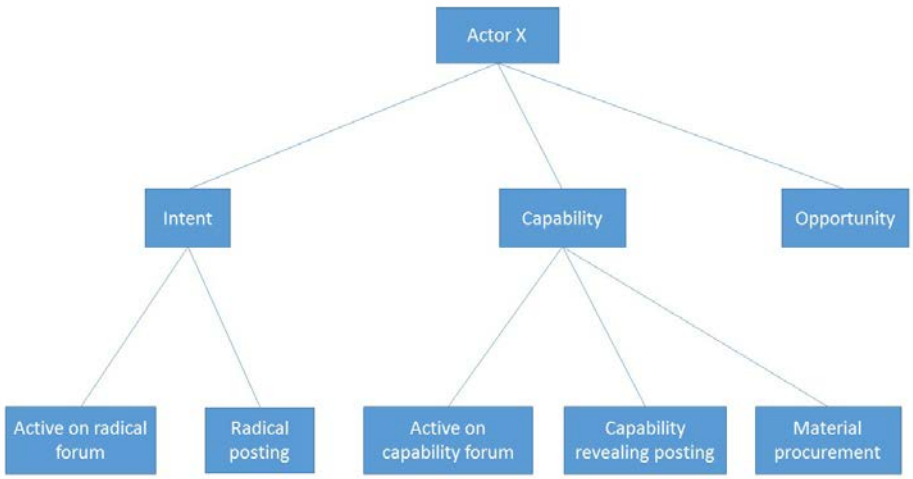

Figure 1: Decomposition of indication factors for an actor.

The weak signals, the warnings, can also be decomposed into different factors. A key factor is motivation or intention, which indicate a will to perform a terror act. Another key factor is the capability to perform a terror act. Possession of firearms indicate capability both directly (i.e. by being able to use the firearm), and indirectly, by connection to networks that can deliver firearms.

3.0 DETECTOR DEVELOPMENT

The workflow for developing a Deep Learning detector for a new object type is depicted in Figure 2.

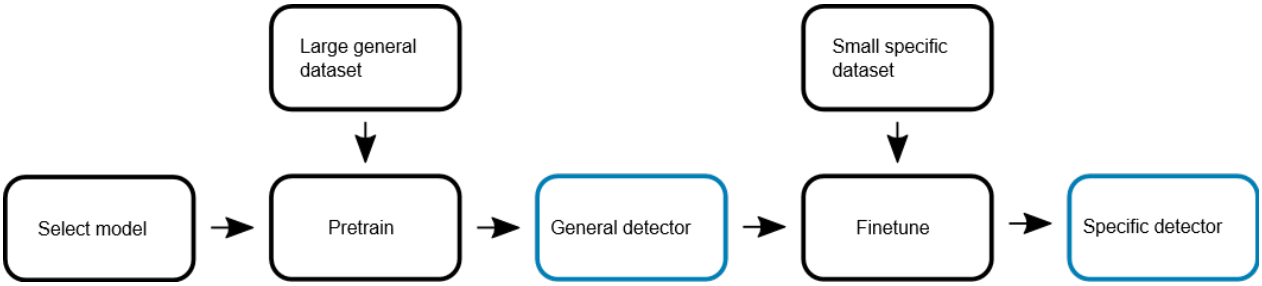

Figure 2: The general workflow for developing a detector for new object types.

The first stage consists of determining what type of Deep Learning model is suitable. The model usually consists of a deep convolutional neural network augmented with an object detection classifier.

In the next stage, we pretrain this model on a large dataset. This dataset does not necessarily have anything in common with the types of objects that the final detector is supposed to detect. Nevertheless, it has been shown that this type of pretraining greatly increases the performance of the final detector. The reason for this is that the convolutional neural network tends to learn to extract very general features that are also useful for detecting other types of objects. If we had a large enough dataset for the actual objects that are to be detected, this type of pretraining could most likely be skipped. This is often not the case however, because obtaining large quantities of annotated data for a new object type normally involves lots of manual labour.

The result from the pretraining is a detector that is able to detect all objects present in the pretraining dataset. This detector is then used as a starting point for creating the detector for the intended types of objects. This is the finetuning stage. Here, the classifier part detector resulting from the pretraining stage is reconfigured to accommodate the new classes. The convolutional network is kept intact. This reconfigured network is then trained on our dataset that contains the specific objects of interest. The result is a detector for the types of objects present in the dataset used during the finetuning stage.

4.0 SOFTWARE FRAMEWORK

Since we want to rely as much as possible on existing tools, it is important that a framework is used that incorporates as much functionality as possible related to the development of a new detector. There is a large number of software frameworks available for Deep Learning based development. In this study we use Tensorflow [6] in conjunction with its Object Detection API [8]. This framework has functionality that supports all stages of described workflow for developing a detector for new object types.

The framework also provides a number of pretrained models that can be used to detect objects from the MS-COCO dataset. For training a new detector, the framework only requires that a configuration file is created that contains all necessary parameters of the model. The only additional thing that has to be provided by the user is an annotated dataset of examples containing object of the specific types that are to be detected.

5.0 DATA ACQUISITION

Obtaining a sufficiently large amount of data to train the model is probably the most challenging task when applying Deep Learning to a new problem. Using a common search engine to collect images may result in a couple of hundreds of unique examples. However, often a couple of thousand or even tens of thousands of examples is needed.

For our weapon detector, we found a public database of images containing various firearms, originating from movies. We spent a total of 100 hours on annotation, resulting in roughly 40k annotated images. The annotations consists of bounding boxes (rectangular regions) that encapsulate each weapon present in the image.

We divide the firearm category into two broad labels; “handgun” and “rifle”. Firearms that is held in a single hand is considered to be handguns, and two-handed firearms are considered rifles. We conjecture that this makes the learning easier, as these two types of firearms appear visually very different.

The fact that the training set consists of images from movies raises the question about how well the model will be able generalize to the application domain, in our case images from social media. To this end, we also collect a test set of images originating from social media. These images had to be collected manually via search engines, as no public dataset was found. In total we were able to collect 200 such images, and they contain at least one firearm in varying visual difficulty. To be able to also evaluate the false positive detection rate, we use 5000 images from the MS-COCO validation set.

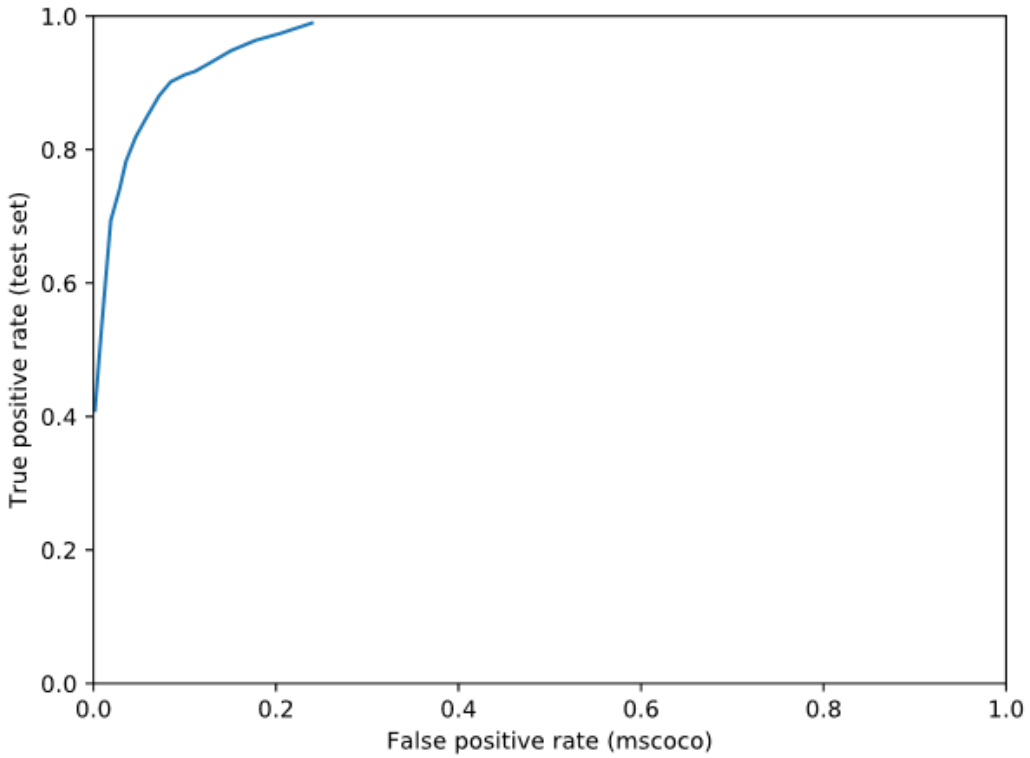

Figure 3: ROC curve for the firearm detector. The true positive rate is measured on a test set of 200 images containing firearms. The false positive rate is measured on 5000 images of the MS-COCO 2017 validation set.

6.0 DETECTION MODEL

The detector is based on Faster R-CNN [7] and uses the Inception v2 [5] network for feature extraction. The model is pretrained on the MS-COCO dataset, which helps alleviate the need for large amounts of training data. This is a standard model that is part of the Tensorflow Object Detection API *Model Zoo*. While there are more advanced models available, this one is chosen because of its relatively fast training and inference speeds.

The final training of the detector took about 40 hours on a single high end graphics card. Although the actual training does not need any human intervention, the need to manually tune hyperparameters of the model (which is essentially a trial-and-error process) means that this involves some manual labor as well. We estimate that the amount of active work spend on training the detector was about 40 hours (one working week).

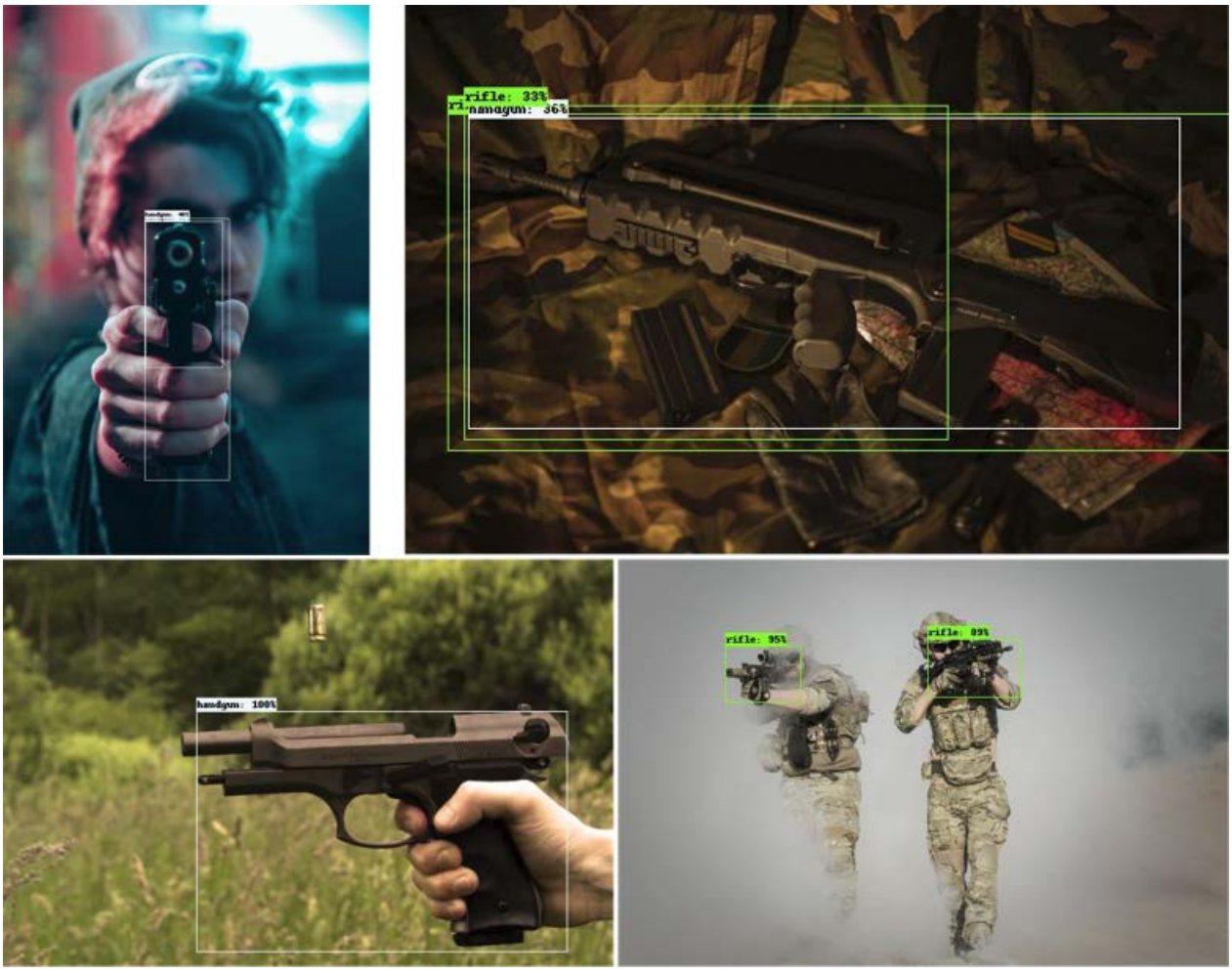

Figure 4: (Best viewed digitally) Example images for some true positive detections.

7.0 RESULTS

The evaluation is performed in an “image classification” manner, where the maximum detection score for a firearm in a given image is set as the total score for the image. Essentially, this means that we discard the localization part of the detection during evaluation, and focus only on whether the image as a whole contains any firearms. This is done to mirror the actual use-case, where the localization of objects is not as important as simply being able to flag an image as containing a firearm.

For a given detection to be considered correct, the produced bounding box by the detector has to have an Intersection-Over-Union overlap of at least 0.5. Otherwise the detection is considered a false positive. Multiple detections of the same object is not punished. We do not make any distinction between the two classes “handgun” and “rifle” when scoring the detector. E.g. a rifle that is classified as a handgun is considered a correct detection.

The ROC curve for the final detector is shown in Figure 3, Figure 4 presents some example images of correct detections and Figure 5 shows some examples from the MS-COCO validation set where the detector made false positive detections. Commonly occurring false positive detections include handheld gadgets and skiing gear.

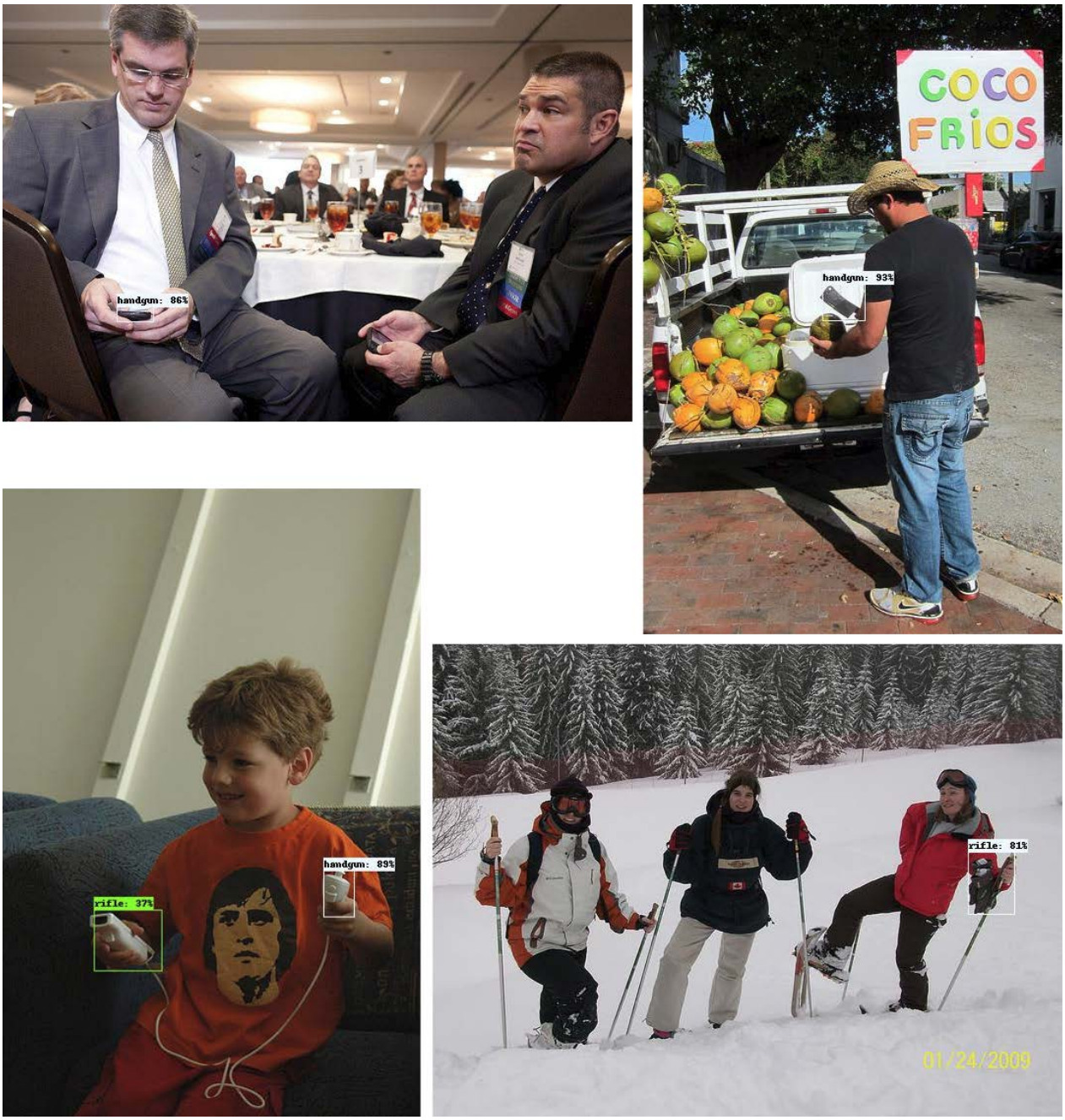

Figure 5: (Best viewed digitally) False positive detections from the MS-COCO validation set. Handheld "gadgets" and skiing gear, are some of the objects that are most often misclassified as firearms.

8.0 CONCLUSIONS

We have presented the development of a detector for firearms in images, intended to be used in scenarios such as information retrieval from social media and forensic investigations of image material. The work shows that a detector for a new object type can be developed relatively quickly, given that there is a sufficient supply of example data.

There are still many things that can be done to improve the performance of the produced detector. To avoid getting false detections for objects such as skis and cell phones etc., images from the MS-COCO training set

could be used to mine these types of hard examples during training. As with any machine learning task, the results will also most likely improve with more training data, and specifically, training data from the application domain.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li och L. Fei-Fei, ”ImageNet: A large-scale hierarchical image database,” *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[2] L. Kaati och F. Johansson, ”Countering lone actor terrorism: weak signals and online activities,” FOI-S--5372--SE (2016).

[3] J. R. Meloy, J. Hoffman, A. Guldimann och D. James, ”The Role of Warning Behaviors in Threat Assessment: An Exploration and Suggested Typology,” *Behavioral Sciences & the Law*, vol. 30, pp. 256-279, 2012.

[4] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár och C. L. Zitnick, ”Microsoft COCO: Common Objects in Context,” *arXiv*, 2014.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens och Z. Wojna, ”Rethinking the Inception Architecture for Computer Vision,” *arXiv*, vol. abs/1512.00567, 2015.

[6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz och L. Kaise, ”TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed,” *arXiv*, 2016.

[7] S. Ren, K. He, R. Girshick och J. Sun, ”Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, 2015.

[8] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song och S. Guadarrama, ”Speed/accuracy trade-offs for modern convolutional object detectors,” *IEEE CVPR*, 2017.