Developing a Real-Time Gun Detection Classifier

Justin Lai
Stanford University
jzlai@stanford.edu

Sydney Maples
Stanford University
smaples@stanford.edu

1. Abstract

Using real-time object detection to improve surveillance methods is a promising application of Convolutional Neural Networks (CNNs). One particular application is the detection of hand-held weapons (such as pistols and rifles). Thus far, previous work has mostly focused on weapon-based detection within infrared data for concealed weapons. By contrast, we are particularly interested in the rapid detection and identification of weapons from images and surveillance data. For our project, we used a Tensorflow-based implementation of the Overfeat network as an integrated network for detecting and classifying weapons in images. Our best performance was achieved on Overfeat-3 with 93% training accuracy and 89% test accuracy with adjusted hyperparameters.

2. Introduction

Gun violence in metropolitan areas of the United States remains stubbornly high, despite decades of gun control efforts. Many gun-related crimes, including armed assault and robbery, occur in public places with existing surveillance systems. However, most systems rely on constant human supervision; this often demands an unrealistic amount of vigilance, and can be expensive and ineffective when multiple video streams are present.

The increase of video surveillance in public spaces and the proliferation of body cameras for police can potentially be leveraged for gun detection systems. Video systems could alert police and surveillance personnel when a gun is detected in real time, resulting in prompter action.

Our approach addresses the problem of real-time detection. To guide our project, we first set out to find the most efficient metric for our task. For our baseline model, we used a VGG-16 based classification model pre-trained on the ImageNet dataset. [6] The ImageNet dataset is an object detection dataset comprised of about 1.3 million images with approximately 1,000 object classes. In tandem with the ImageNet dataset, we fine-tuned our own dataset of about 3,000 weapon-based images, pulled from the Internet Movie Firearm Database, [citation] home-made videos, and frames from select surveillance- or news-based YouTube videos. This was so that each image would contain weapons in a variety of different contexts, situations, orientations, and distances. We trained these on a Tensorflow-based version of the Overfeat Convolutional Neural Network architecture. [7]

There are 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)
- There are a variety of different types, shapes, and sizes of weapons, leading to a variety of different image sizes and bounding box sizes

In addition, when working with video specifically, there is the added difficulty of locating an object in real-time. Doing so is reliant on the confidence of the classifier. In other words, the classifier does not register an object as a weapon unless its confidence that that the object is a weapon is above a certain threshold. This often slows down real-time classification.

We found that our best model was the Overfeat network after adjusting the learning rate from 0.001 to 0.0003 and the confidence threshold from 50% down to 30%.

3. Related Work

Our project differs from some other Object Classification problems in two respects: firstly, we are concerned with real-time detection, and thus are especially concerned with the speed of classification; secondly, we are especially concerned with accuracy, since a false alarm could result in very adverse responses. In choosing the right approach, we need to make the proper trade-off between accuracy and speed.

Weapon Detection Research

There is some existing research on weapon detection in

image and video data. We have reviewed image classification research including the analysis of infrared data for concealed weapons [2], the detection of violent scenes in movie data [4], and gun detection in body camera videos [**?**]. While most research in the field have not employed deep learning/neural net techniques, this paper has: Olmos, Tabik, and Herrera investigate automatic gun detection in surveillance videos, triggering an alarm if the gun is detected (Automatic Handgun Detection Alarm in Videos Using Deep Learning) [6]. The project implements an R-CNN in order to detect the action sequence of a handgun being drawn. Their implementation confirms the findings of Moez Baccouche and his team, who found that video action sequences could be effectively trained with 3-D CNNs (spatio-temporal features), and then classified using R-CNNs.[1] Olmos, Tabik, and Herreras research specifies further areas for improvement, including video preprocessing (adjusting contrast and luminosity to improve results).

Fast R-CNN Olmos et. al have proposed a detection system that uses a trained Faster R-CNN. They note two classification approaches: the sliding window approach and the region proposals approach.

As a result, Olmos et. al use a region proposals approach with a much higher classification speed of 2s/image (with a Fast R-CNN). Olmost et. al have also created a useful dataset of 3000 gun images, which we used in our own implementation. They also conclude that additional work needs to be done to test other classifiers such as GoogLeNet Overfeat.

YOLO YOLO treats the object detection problem as a regression problem, instead of having a normal pipeline of region proposals and classification. This leads YOLO to perform extremely quickly in real-time, but at the cost of some accuracy: YOLO is able to achieve 63.4% mAP at 22 ms latency. [12] However, the research done by Wang et. al has noted that YOLO has been noted to have major accuracy issues in real-time object detection, especially with "skinnier objects like traffic signs, bottles and plants". [13]

GoogLeNet OverFeat GoogLeNet Overfeat uses a sliding-window approach to detection. Thought it has been criticized for its speed (which could be a major problem for detection). Overfeat uses the sliding window approach, which considers a very large (10^4) number of windows in a single image. While it achieves high accuracy, it can be impractical for detection problems due to its speed (at 14s per image, it would result in a very delayed detection).[8]

Tensorbox : Tensorbox is an implementation of GoogLeNet-Overfeat with two independent FC layers for softmax and regression. It has shown promising applications for real-time object detection in videos, and player-tracking in basketball. The project is developed and maintained by Stanford PhD student Russell Stewart.[7]

4. Methods

Data Collection and Preprocessing: We first downloaded 3,000 images for the training set and 500 images for the validation set from IMFDB, a movie internet firearms database. Approximately 2,735 of these images were useable, as a number of these images had weapons that were occluded by darkness or rendered unseeable due to blurriness or scale. We sorted 2,535 of these images into our training set and 218 of these images into our test set. We resized all images to 640 x 480 such that they were compatible with the provided TensorBox framework. We then used a script to set bounding boxes around each firearm within each image. The script output the x, width and y, height of each respective bounding box to an output script. We then reformatted the output script to be used as a json using a separate script. To find the names of the images that ultimately werent used, a separate script was run that located the difference between images in training set and images used in the training set json file. Those images were moved to an unused images folder so as not to stall the training process.

Set	class no.	total images
all	2	2753
train	2	2535
test	2	218

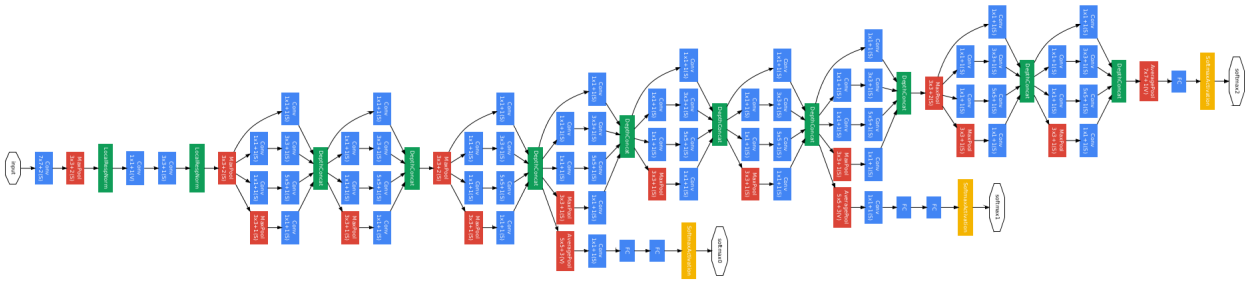
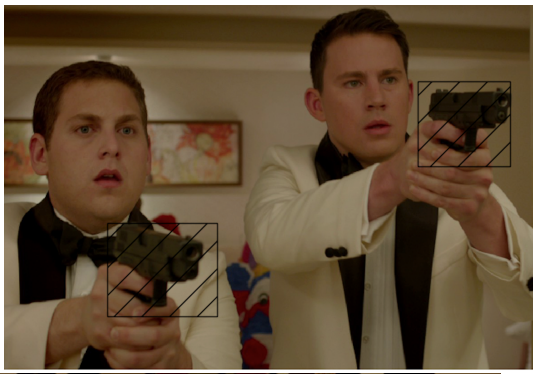


Figure 1. GoogLeNet Architecture

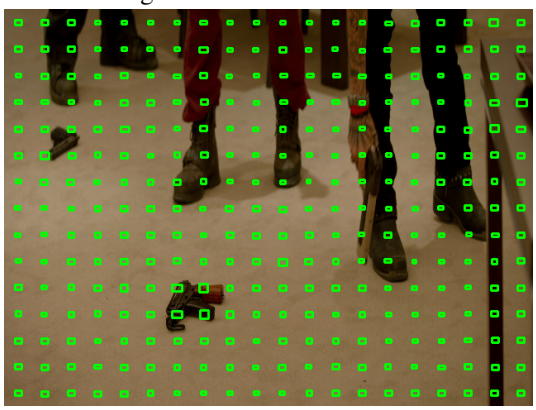


Baseline: For our baseline, we used the VGG-16 classifier, the largest VGGNet architecture, pre-trained on ImageNet as a baseline (courtesy of the theano library, with Keras on the front-end). We tested the classifier on 200 self-collected images of various guns (the final dataset is currently being compiled). 79 of these images were classified as rifles (class score of 762 in VGG), and 121 of these images were classified as revolvers (class score of 763 in VGG). Our baseline achieved a 58% accuracy on revolver classification and 46% on rifle classification.

Overfeat Implementation / Parameters We decided to implement Overfeat through Tensorbox. Then, we trained by adjusting our hyperparameters in three distinct ways:

- Overfeat 1: 50% confidence threshold, learning rate 0.001
- Overfeat 2: 50% confidence threshold, learning rate 0.0003
- Overfeat 3: 30% confidence threshold, learning rate 0.0003

We noticed that decreasing our confidence threshold resulted in a more rapid change of bounding box sizes as our data was training.



Results: Overfeat-3 consistently performed the best on training with an accuracy of 93%

Model	Train Acc	Test Acc
VGGNet	0.57	0.46
Overfeat-1	0.62	0.56
Overfeat-2	0.69	0.64
Overfeat-3	0.93	0.89

We believe that this drastic difference was directly attributable to the confidence threshold used. When the classifier required more confidence to output a bounding box, it would have a higher likelihood of not outputting the bounding box at all, whereas with a lower confidence threshold came a higher likelihood of outputting a correct bounding box and correctly classifying the weapon in the image.

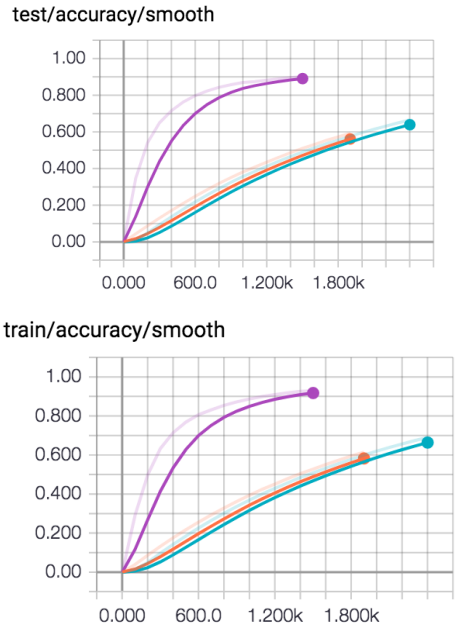


Figure 2. Orange: Overfeat-1, Blue: Overfeat-2, Purple: Overfeat-3

5. Conclusion

Our best performance was achieved on Overfeat-3 with 93% training accuracy and 89% test accuracy with adjusted hyperparameters. Our decreased threshold (0.5 to 0.3 confidence) and increased learning rate gave the classifier the necessary 'loosening-up' to make more predictions.

6. Future Work

Unfortunately due to time constraints, we were unable to optimize for the time performance of classification; each classification took around 1.3 seconds, which is likely too long for any live video feed through a surveillance or body camera; more work is needed to test the available classifiers and optimize performance for both accuracy and speed.

References

[1] Moez Baccouche, et al. *Sequential deep learning for human action recognition*. International Workshop on Human Behavior Understanding. Springer Berlin Heidelberg, 2011. 2

[2] Samir K. Bandyopadhyay, Biswajita Datta, and-Sudipta Roy *Identifications of concealed weapon in a Human Body* Department of Computer Science and Engineer, University of Calcutta, 2012. 2

[3] Aaron Damashek and John Doherty *Detecting guns using parametric edge matching* Project for Computer Vision Course: CS231A, Stanford University, 2015.

[4] Claire-Hlne Demarty, et. al *The MediaEval 2012 affect task: violent scenes detection* Working Notes Proceedings of the MediaEval 2012 Workshop. 2012. 2

[5] Roberto Olmos, Siham Tabik, and Francisco Herrera *Automatic Handgun Detection Alarm in Videos Using Deep Learning* Soft Computing and Intelligent Information Systems research group, Department of Computer Science and Artificial Intelligence, University of Granada, 2017. 1, 2

[6] K. Simonyan, A. Zisserman *Very Deep Convolutional Networks for Large-Scale Image Recognition* Visual Geometry Group, Department of Engineering Science, University of Oxford, 2015. 1, 2

[7] Russell Stewart *Tensorbox* https://github.com/TensorBox/TensorBox. 1, 2

[8] Pierre Sermanet, et. al *Overfeat: Integrated recognition, localization and detection using convolutional networks* Courant Institute of Mathematical Sciences, New York University, 2013. 2

[9] Pierre Sermanet, et. al *You only look once: Unified, real-time object detection* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. 2

[10] Dumitru Erhan, et al *Scalable object detection using deep neural networks* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun *Faster r-cnn: Towards real-time object detection with region proposal networks* Advances in neural information processing systems, 2015. 2

[12] Joseph Redmon and Anelia Angelova *Real-time Grasp Detection Using Convolutional Neural Networks* Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015. 2

[13] Chung Yu Wang and Cheng-Yue Royce *Traffic Sign Detection using You Only Look Once Framework* Technical Report Project for Computer Vision Course: CS231N, Stanford University, 2015. 2

[14] Deng, Jia, et al. *Imagenet: A large-scale hierarchical image database* Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[15] The IMFDB *Internet Movie Firearms Database, Online*