CS CAPSTONE TECHNOLOGY REVIEW

DECEMBER 4, 2018

PEDESTRIAN COUNTING AND PRIVACY PRESERVATION

PREPARED FOR

OREGON STATE UNIVERSITY

DR. FUXIN LI

PREPARED BY

GROUP 9

PAVEMENT PROMETHEUS

MILES DAVIES

Abstract

The City of Portland is updating their data gathering system to better integrate data and technology into the decisions made by the city. One issue that arises is that privacy preservation is often at odds with data gathering. Our task is to provide data on, and hopefully a solution to, this issue. Mainly our concern is manipulation of data so the collected data can be stored and analyzed without violating privacy portions of the city's social contract. Our solution uses YOLOv3 and masking to remove identifying information about the citizens in the videos.

CONTENTS

1 Introduction

Pavement Prometheus is a senior capstone team working on the Pedestrian Counting and Privacy Preservation project. As our project will be focusing on Machine Learning, technologies of interest will primarily be focused on the field of computer science. We will need to select a good source control system to allow multiple parties to work on a single code base with ample transparency with regards to any changes made. We will require a base object detection model to work with and adapt for our purpose of ultimately identifying pedestrians and obfuscating any identifying information. We will also require a neural network that the object detection model will employ, preferably one that is translation-invariant given the chaotic requirements of image processing. By selecting, hopefully, the best technology to be employed for our project, our team will have less work to perform in the future.

1.1 Work Log

Project Section	**Maintainer**
Simluation Setup	Mazen Alotaibi
Data Collection	All team members
Pedestrian Detection Algorithm	Mazen Alotaibi and Miles Davies
Face Detection Algorithm	Mazen Alotaibi and Stephanie Allison Hughes
Obfuscator Algorithm	Ian McQuoid
Source Control and Software Design	Ian McQuoid

2 Source Control

Our project involves multiple team members across a couple timezones. Picking the right source control is not only desirable, but necessary. Likewise evaluating the benefits of a given source control choice is important, whether it's diagnostic tools, allowed private repositories, or browsing layout.

2.1 GitHub

GitHub is one of the first widely adopted online source control platforms. This source control repository is is appealing in no small part because we need to maintain an account for grading in our Senior Capstone class, but also because all parties within our organization has some historical experience with the tool. The platform is intuitive with its own UI client and can be accessed from a command line terminal, desktop, and web interface[1]. For some of its more advanced features the platform needs a subscription, with private repositories offered on a somewhat limited basis depending on your account status.

2.2 BitBucket

BitBucket is very similar to GitHub, and is a repository that is owned and developed by Atlassian. Like GitHub, it uses the Git revision control systems, making it potentially very accessible for anybody familiar with GitHub and Git. Also like GitHub, the source control tool can be accessed from a command line terminal, desktop, and web interface[1]. The service also allows unlimited private repositories for free, with file storage increasing by 1GB a month. They don't offer features such as SSL or Two-factor authentication, but then for the purposes of our project this is unnecessary. Perhaps most importantly our client has designated this tool the source control of choice given he has preexisting material on an account there.

2.3 Google Cloud Source Repository

Google Cloud Source Repository (CSR) is relatively new to the field of source control, as such they currently have no UI client for user options- and instead rely on tools like Sourcetree or the GitHub UI client. The platform is free for an account with up to 5-project users, with 50GB of free storage total initially and an additional 50GB free egress each month[1]. Perhaps most appealing, the Google Cloud Platform offers free diagnostic tools for debugging and error reporting, and so code can be analyzed directly from your git repository to track any issues or errors. This advantage is primarily useful in a deployed code environment with users, and given the exploratory nature of our research project isn't applicable. However, CSR can integrate with BitBucket or GitHub with relative ease.

3 Object Detection Model

Our project will require an Object Detection Model for the purposes of detecting pedestrians in a time sensitive environment. There are several open source object detection models offered online that might be applicable. We will be focusing on an object detection model which can identify objects with a good degree of precision and accuracy in real time. Accomplishing both of these requirements will be difficult, given precision comes at the cost of speed and speed the cost of precision.

3.1 Faster R-CNN

Faster R-CNN is an object detection model that aims to detect objects in a real-time environment. This model is itself an improvement on the older Fast R-CNN object detection model, which uses a selective search to generate region proposals[2]. Instead of selective search, Faster R-CNN uses a region proposal network to generate region proposals which are used in conjunction with a network in order to detect objects. The way the region proposal network works is by ranking region boxes and proposing which contain objects. These region boxes are then run against a trained network to classify the objects.

3.2 Single Shot MultiBox Detector

Single Shot MultiBox Detector (SSD) is an object detection model designed for real-time applications. Designed as an improvement on Faster R-CNN, it eliminates the aforementioned's need for a region proposal network. This allows the technology to enjoy real-time prediction speeds and an accuracy of 80% on average, making it the most accurate real time object detection model currently available[3]. SSD however does fall behind in the time it requires to process images, as it can only process images at around 20 frames-per-second. The object detection model ultimately consists of two parts: extract feature maps from the image, and apply convolution filters on the maps to detect objects.

3.3 You Only Live Once v3

You Only Look Once v3 (YOLOv3) is an object detection model that is both very accurate and fast. After being trained on the COCO dataset the algorithm can achieve a 60-70% accuracy rate in real time at 30+ frames-per-second (Given a Pascal Titan X graphics card)[4]. YOLOv3 works by applying only a single neural netowrk on a given image. This image in turn is divided into regions, with each region provided predicted bounding boxes and probabilities. Where bounding boxes are weighted by predicted probabilities. This allows the model certain advantages against other classifier based systems, as its predictions are performed over the entire image (giving it a global context) and use only a single network evaluation instead of thousands.

4 Neural Networks

For our project we will be interested in implementing a neural network for our object detection model which will be used to train data against until it can be deployed to detect objects with a high degree of precision and accuracy. Neural networks in general are designed to mimic the activity of the human brain, and are created using layers of neurons with each doing data transformations, ordering and sorting, to a provided input. This is sometimes referred to as a feature hierarchy. Most often neural network variations defer in the number of layers employed and how these layers interact with one another in the process of producing an output for a given input.

4.1 Deep Neural Network

A Deep Neural Network (DNN) is a neural network with at least three layers: an input layer, a hidden layer, and an output layer. Whereas all neural networks require an input and output layer, the DNN is referred to as deep because of its hidden layer (of which there can be multiple), they are employed when just a single input and output layer do not suffice for ordering and classifying a provided data set[5]. Most neural networks designed to perform some non-trivial task are actually a classification of DNN. These neural networks being overtly generic, DNNs are useful for a wide range of applications, but fall behind other neural network frameworks when it comes to performing a specific task.

4.2 Convolution Neural Network

A Convolution Neural Network (CNN) is a type of DNN which allows the usage of convolutional and pooling layers, these are better for problems that are translation-invariant such as images. The convolution layer consists of a set of learnable filters which convolve across the dimension of the input volume during the initial forward pass of a given input[5]. This ultimately creates a 2-dimensional activation map which allows the filters to activate whenever a specific feature is detected at a certain input. This makes CNNs particularly adept at image processing.

4.3 Recurrent Neural Network

A Recurrent Neural Network (RNN) is a type of neural network where connections between nodes within layers form a directed graph. This architecture allows the RNN to display time variant behavior for a given input. This means that the output of one iteration of the neural network will impact the next iteration. These neural networks also take an arbitrary input/output length whereas CNNs generally take a fixed input size and to produced a fixed sized output[5]. Given these properties, RNN are particularly good for speech and text analysis but less so when it comes to image processing.

5 Conclusion

Given the above technologies my team will be employing a mix of GitHub and Bitbucket for our source control, the former primarily for the purposes of grading and the latter for the needs of our client. For a base object detection model to work from, our team will use the YOLOv3 system as it provides an extremely fast processing speed without sacrificing much accuracy. This makes the system ideal for the purpose of identifying (and obfuscating) identifying details from a camera feed. Likewise, we will be employing a modified CNN for our requisite neural network, as it is designed to handle translation-invariant problems such as object detection within images.

6 REFERENCES

[1] S. Sharma, "Google cloud source repository vs. github vs. bitbucket — comparative analysis," Available at https://www.mediaagility.com/google-cloud-source-repository-vs-github-vs-bitbucket-comparative-analysis/ (2018/11/01).

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *ArXiv e-prints*, Jan. 2016.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *ArXiv e-prints*, Dec. 2015.

[4] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *ArXiv e-prints*, Apr. 2018.

[5] H. Choi, "Introduction to neural networks: Dnn / cnn / rnn," Available at http://alinlab.kaist.ac.kr/resource/Lec1_Introduction_to_NN.pdf (2018/11/01).