CS CAPSTONE DESIGN DOCUMENT

APRIL 18, 2019

PEDESTRIAN COUNTING AND PRIVACY PRESERVATION

PREPARED FOR

OREGON STATE UNIVERSITY

DR. FUXIN LI

PREPARED BY

GROUP 9

PAVEMENT PROMETHEUS

IAN MCQUOID
MAZEN ALOTAIBI
MILES BIGELOW DAVIES
STEPHANIE ALLISON HUGHES

Abstract

The City of Portland needs a tool for gathering data on traffic and pedestrians for internal and public use. The main issue that arose was privacy preservation and the removal of personally identifiable information from the data. The Pedestrian Counting and Privacy Preservation project serves to provide the city with stripped data and analysis of the data. This document's purpose is to present the methods and pieces of the project and the way that the pieces will be developed. The information covered will include the object detection model, facial detection and obfuscation, tracking models, and how the data will be analyzed and made accessible.

CONTENTS

1 Change Log

Section	Original	New
Multi-Object Tracking	• Using Chanho's algorithm to applying tracking on objects on live-streaming videos. • Applying	• We have removed this section because it is an extra element that we thought before Code Freeze but we couldn't. However, we will be adding it to our Final Release. • We haven't added this part because the code doesn't output testable information for the TAs to evaluate our work.

2 Overview

2.1 Purpose

Our team will design a pedestrian/vehicle detection model which is able to obfuscate all identifying features of pedestrians and vehicles for a given video feed. This will allow for storage of the video data without storing identifying information on the pedestrians.

2.2 Scope

The scope for this project is immediately to have a system that results in information on pedestrian movements that can be stored for open access by the public. An update that is not necessary, but is desirable, is the ability to provide data on traffic as well.

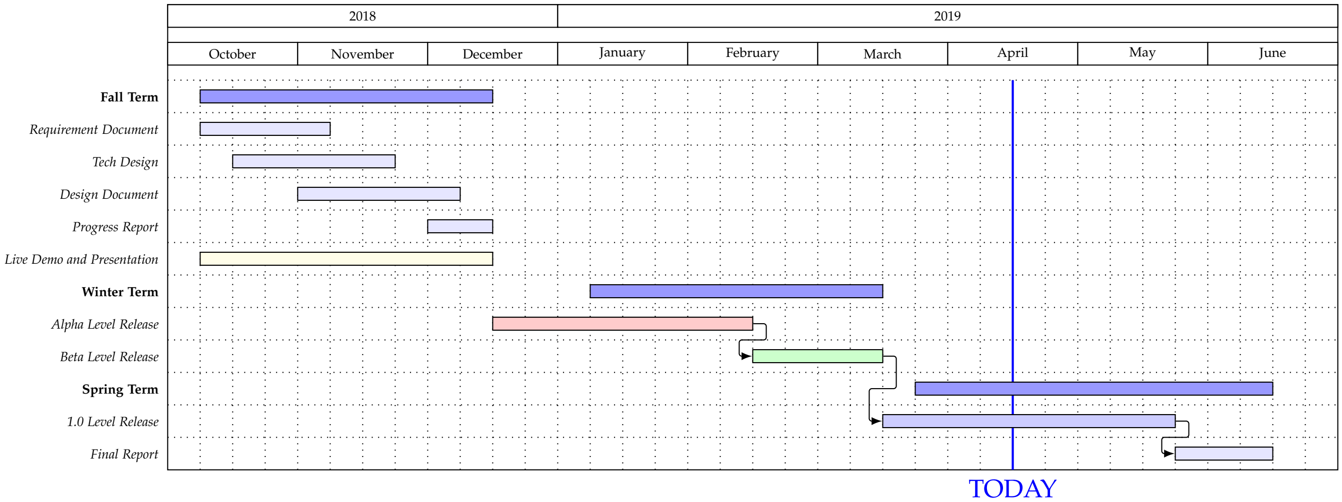
2.3 Glossary

Term	Definition
Convolutional Neural Network (CNN)	A class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery [1].
Recurrent Neural Network (RNN)	A class of artificial neural network where connections between units form a directed graph along a sequence [2].
Long Short-Term Memory networks (LSTMs)	A special kind of RNN, capable of learning long-term dependencies [3].
You Only Look Once (YOLOv3)	A state of the art object detection model which can classify objects with a high degree of fidelity in a time sensitive environment [4].
mean Average Precision (mAP)	The mean for a metric denoting percentage of objects precisely identified, a ubiquitous standard used by object detection models [4].
Car Learning to Act (CARLA)	An open simulator for urban driving. CARLA has been developed from the ground up to support training, prototyping, and validation of autonomous driving models, including both perception and control [5].
OpenCV	OpenCV is a library of programming functions mainly aimed at real-time computer vision. [6].
Facial Keypoints Detection	Facial detection through the use of multiple key points on a person's face [**?**].
Obfuscation and Mangling	Used interchangeably. The irreparable destruction of data. Specifically used in relation to identifying features of objects.

2.4 Design Stakeholders

The software described in this document, Facial Detector, and Obfuscator, is a project under the advisement of Chanho Kim (Georgia Institute of Technology) and Dr. Fuxin Li (Oregon State University). The client for this project is the City of Portland, which wants a proof of concept for a way to transform the data from their traffic cameras so the city may store the data without storing identifying information about the citizens in the footage.

2.5 Design Timeline



.

3 References

[1] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales, "When Face Recognition Meets with Deep Learning: an Evaluation of Convolutional Neural Networks for Face Recognition," *ArXiv e-prints*, Apr. 2015.

[2] A. Shekhar, "Understanding the recurrent neural network," Available at https://medium.com/mindorks/understanding-the-recurrent-neural-network-44d593f112a2 (2018/04/14).

[3] C. Olah, "Understanding lstm networks," Available at http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (2015/08/27).

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

[5] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," *ArXiv e-prints*, Nov. 2017.

[6] "Realtime computer vision with opencv," Available at https://dl.acm.org/citation.cfm?id=2206309 (2018/10/30).

4 Design Viewpoints

4.1 Introduction

The Pedestrian Counting and Privacy Preservation project has four primary design viewpoints which will need to be considered and implemented. Our project will require a robust object detection system, which is able to identify objects in both an accurate and time-efficient manner. This object detection in turn will feed into a face recognition system that will ultimately be responsible for both improving detection accuracy and meeting privacy preservation goals. We will also be implementing a multi-object tracking system that will allow identity retention of an object across multiple frames. In addition to collecting important data on pedestrian and vehicle behavioral patterns, this tracking system will also assure validity in such measures as aggregate object counts; preventing a single object from being counted twice between frames. All of these components will feed directly into a data analysis and access system, which will be responsible for the extraction and serialization of any pertinent information filtered for storage on a server.

4.2 Viewpoint: Real-time Object Detection

4.2.1 Viewpoint Description

Real-time object detection is a necessary component of the pedestrian counting and privacy preservation system. Through the implementation a proper object detection algorithm, it is possible to aggregate data on pedestrian and vehicle traffic using a camera without requiring any video feed storage for further analysis. This would effectively maintain privacy for all parties picked up in a camera feed. Real-time object detection will be part of a collective effort by Miles Davies and Stephanie Allison Hughes, who will both be responsible for systems pertaining to object detection and face detection. Object and face detection, in turn, will be broken down into subsections involving setting up detection algorithm, and the requisite deep learning framework for its training and validation. Criteria involved for evaluating the success of object detection will relate to both the speed and accuracy of said detection. This will ultimately be a negotiation between the two criteria, as a faster processing time will temper our possible accuracy, while an improved accuracy will require a longer processing time.

4.2.2 Design Concerns

Our primary design concerns for our real-time object detection relate to both the speed and accuracy of our model. Because our system will primarily involve interesting information extraction while simultaneously excluding any identifying details, we require an object detection algorithm that maximizes both detection accuracy and detection speed. An object detection algorithm which meets both requirements will allow us to eschew any form of video storage for more conventional methods of processing, as all necessary information can be extracted in real-time. By eschewing video storage, this system will have no persistent data that contains personally identifying information which might violate the privacy of any party recorded in the camera feed.

4.2.3 Design Elements

Design elements include the setup and choice of a base object detection algorithm, a deep learning framework for both the training and validation of our selected detection system, a comprehensive dataset to train the model on, and any requisite computer hardware for training the model. After selecting and implementing a base object detection algorithm in our chosen deep learning framework, our team will need to download a dataset to begin training and validation our detection algorithm against. Our team can then investigate methods to improve either the accuracy or speed of our model to compare against the base setup; whether through the reduction of object classes available for detection or through some novel restructuring of the model's neural networks or weights. For the selection of a base object detection algorithm to implement, our team research has indicated You Only Look Once (YOLOv3) is the best choice for maintaining an adequate detection accuracy with an impressive framerate speed[4]. The speed and accuracy of this detection system appears to be the result of a union between the DarkNet 53 system, for feature extraction, and Feature Pyramid Networks (FPN), which uses a bottom-up and top-down pathway for improved small object detection accuracy. Our selected deep learning framework, in turn, will be the open-source machine learning library PyTorch. We selected this deep learning framework for a variety of reasons, including ease of use, active community support, and the customizability of its Neural Networks.

4.2.4 Relationship

The real-time object detection system will be the first layer of processing performed on a camera feed for data extraction; and will be interfaced directly with the facial detection, tracking, and data analysis and access systems. Through a combination of the object, facial, and tracking detection we will be able to present information for data analysis and serialization for extraction and storage to a server. Through the interfacing with the face and tracking subsystems, any shortfall in the accuracy of the real-time object detection system will ideally be assuaged. Allowing for a comprehensive, accurate, detection system performing in a real-time environment.

4.3 Viewpoint: Face Detection

4.3.1 Viewpoint Description

Our group will conduct facial detection using a feature-based method by using a pre-trained convolutional neural network. For the feature-based method, the facial features are detected through examining the edge, intensity, color, shape, etc. of a feature. Using a pre-trained facial detection model allows our group to use a model that has been trained on a large dataset. It is important that we use images from different perspectives so that we can make sure faces are detected in varied situations.

4.3.2 Design Concerns

Main concerns of the design are making sure that the faces are detected from different perspectives and angles. The pre-trained CNN will allow us to detect pedestrian faces with a model that is based on a large dataset with a variation of faces. The City of Portland footage will be taken from different perspectives, the so the model must be able to detect multiple sides of the face from several different camera angles. Another design concern is ensuring high confidence of results within various times of day and weather conditions. Footage may be taken from outside environments where daylight, crowds, or other outside factors may alter the visibility of the scene. This stresses the importance of testing the program with different images to ensure confidence in the detection.

4.3.3 Design Elements

Design elements of facial detection includes determining a facial detection method, determining a pre-trained model to use, detect and obfuscate faces within images, and testing the program with a variety of pedestrian situations. When determining a facial detection method, it is important to weigh the pros and cons of each method. For our particular case, a feature-based detection method will allow us the best flexibility for our needs. Feature-based detection used the key facial features of a face to identify a face, rather than the face as a whole. Using the feature-based method is more likely to detect facial features despite the orientation of the face. If a person is faced to the side or in a different direction, the program is not fazed as it is not looking at the overall face but the distinctive features. This makes the speed of the program quicker as it would not need to run extra functions to detect faces from different angles, just the primary program. Using a pre-trained facial detection model with OpenCV features, we are able to detect faces and display the confidence of their detection. The main focus of pedestrian detection is on preserving the privacy of pedestrians through obfuscation. In our face detection program we will use a blurring tool that will obfuscate the faces of pedestrians within the scene. The hope is that through obfuscation, City of Portland workers can continue to use the traffic footage and data without needing to worry about specific privacy violations.

Once the face detection and obfuscation program is developed, it will be tested with images of people from different angles and perspectives so that we can ensure the detection works in different real-world situations. The detection program should be able to detect different types of people from different depths of perspective doing different tasks. Whether someone is looking in a different direction or is farther away from the camera, their faces should be able to be detected. The output of the program should be of the original image with the faces of pedestrians detected and obfuscated.

4.3.4 Relationship

Facial detection plays in integral part in our project as the faces must be detected to protect the privacy of the pedestrians. A pedestrians face is a unique, identifying aspect of a person, so if it is not obscured, the person could be easily tracked.

If just the pedestrians were detected and the entire bodies were obscured as a block, larger amounts of video footage would be lost. By just detecting the faces, the crucial identifying information is obscured without altering too much of the footage. The point of the project is to keep the privacy of the pedestrians while retrieving of traffic data. For people to keep their privacy and live free lives, faces must be detected to be obscured.

4.4 Viewpoint: Data Analysis and Access

4.4.1 Viewpoint Description

Data analysis and access is a key subsystem in fulfilling the end goal presented to our group by the City of Portland. Ian McQuoid and Mazen Alotaibi will be the two designers of the subsystems relating to data aggregation, analysis, and access. The main purpose of this system is to provide the City with reasonable access to the data provided by sensors around the city, after stripping all personally identifying information, and to make the data provided more intelligible by parsing the video or photographic information gathered into a serialized format. This formatted data will allow the City to make decisions about the roadways and traffic with greater speed and accuracy. The criteria our group will use to interpret and evaluate the system will be concerned with the ergonomics relating to the access of the information and the accuracy to ground truth models for the analysis and representation of the serialized data.

4.4.2 Design Concerns

Our main concerns in the Data subsystems are related to the persistent data structure, the data access scheme, and the data content. The City of Portland largely uses the JavaScript Object Notation (JSON) as a key standard for transmission of data objects between departments. Because of the City's JSON use, the City of Portland requires analyzed data to be in JSON form with a web-based interface. The first concern raised is the requirement relating to the basic data format, while the second concern relates to the structure the analyzed data will be stored in. The data content is the broadest concern in the design process. The content of the analyzed data must be verifiable, accurate, and interesting. The basis for the content concern relates directly to the applications for the analyzed data. Information relating to traffic makeup, size, lane usage, and speed are all directly applicable to traffic analysis, so final data content in the database must provide comparable or directly related information.

4.4.3 Design Elements

The direct data access format must be in JSON form as defined by the City's concern. The format specification affords the team the opportunity to use the MongoDB persistent data structure for storing final data-sets. The MongoDB database program has default interfaces in place which will be the base for access to stripped and analyzed data. The resulting system will be presented as a web-based interface backed by the MongoDB program for storing and accessing data. The interface will be implemented using a RESTful API written in Python which will create a simple HTML web page for accessing and downloading JSON-formatted data from a MongoDB database. The presented solution will directly address both of the primary concerns presented by the City of Portland. The final design concern is related to the final content of our stored data. The primary pictorial and video data will be presented directly to the user and will not be stored in the database; however, analyzed and stripped information needs to be stored in the database. Before data can be stored in the database, pertinent information needs to be taken from the photographic or video data collected by the City's sensors. The data system will use a mixture of direct storage of information and inferential stripping techniques to gather the required information. Primarily, the object detection and tracking systems will provide concrete data on number, makeup, and trajectory of the vehicles and pedestrians. The concrete data will be stored directly in the MongoDB structure. Further, the data analysis subsystem will be comprised of categorization algorithms. Incoming data-sets will be split into geographic and time-stamped categories. The final results will be stored in the database structure for access.

4.4.4 Relationship

The data analysis and access system will directly interface with the three detection and tracking systems. All three of these systems will present the data system with uniform information in the form of both photographic or video data as well as serialized information about the geographic and temporal placement of the data-sets. When possible, serialized information such as the number of detected objects will be passed to the data system and parsed, categorized, and stored by the access subsystem. All non-serialized data will be passed directly to the analysis subsystem, as the photographic and video data will not be accessible through the MongoDB interface. The two subsystems of access and analysis will interface with each other in a symbiotic relation with the analysis system pulling data from the access system as well as pushing analyzed and serialized data into the database.

4.5 Conclusion

By addressing and implementing each of these viewpoints we will create a robust system capable of collecting census data on pedestrians, vehicles, and their traffic patterns while simultaneously protecting any personally identifying information from being stored and put at risk. This system will comprise of an object detection system which first identifies pedestrians and vehicles alike. While a face recognition will be responsible for detecting pedestrians for masking and improved detection accuracy, and a multi-object detection system will retain memory of unique objects to both maintain data integrity and collect pedestrian and vehicle behavior. The data in turn will be piped though a data analysis and access system responsible for the serialization and ultimately storage of data on a server. This system, fully realized, could be implemented on a city by city basis to provide informed decisions on transit planning, census counts, and pedestrian behavior while simultaneously preserving the privacy of all parties involved.