CS Capstone

December 4, 2018

Pedestrian Counting and Privacy Preservation

Prepared for

Oregon State University

Dr. Fuxin Li ____________ Signature ____ Date

Prepared by

Group 9

Pavement Prometheus

Ian McQuoid ____________ Signature ____ Date

Mazen Alotaibi ____________ Signature ____ Date

Miles Davies ____________ Signature ____ Date

Stephanie Allison Hughes ____________ Signature ____ Date

Contents

Problem Statement

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 drawing the ire of the population or violating privacy portions of the city's social contract. One such solution may arise from using existing tools SSD [1], YOLOv3 [2], Mask R-CNN [3], and Dr. Li's Comupter Vision model [4] to extract usable information from the data.

1 Problem

Data collection has always been at odds with privacy. Up until 2018, Portland had rectified this difference by relying on training individuals to stand on city corners and collect census data in lieu of more modern (and invasive) techniques involving cameras. Despite the antiquated and error prone nature of this method, Portland persisted in its use until the Smart City PDX program was initiated by Mayor Ted Wheeler. This program, a broad initiative designed to redress societal inequities and improve under-served communities, relies on updated infrastructure for data collection to inform decisions on enacting policies. To this end, the City of Portland has partnered with Intel and General Electric to procure a wide array of sensors that will serve in gathering transit data on pedestrians and vehicles. This data can be used for many purposes including planning city transit to be more efficient, reducing traffic related accidents, and having general census information for any future community projects. One problem is that using unadulterated sensor information is problematic with regards to protecting civilian privacy.

Portland's primary constraint is that they must make sure technology serves the interests of its city residents. As such, the city requires that identifying-features of any data collected be removed. This would imply that all data operations need to be performed in real time with sensor collection. This is a prohibitive requirement since nearly any form of statistical analysis is a computationally intensive process. Modern limitations being what they are, it is nearly impossible to perform an analysis of any depth under these constraints – especially on a budget. We then must preserve some information so we can perform future metrics against the data. Any realistic solution will need to balance two factors: the amount of processing power available, and the amount of information preserved. The former factor is a definite metric determined entirely by the hardware provided, while the latter is negotiable. What one person deems permissible information another might consider invasive. This preservation of privacy, in turn, needs to be balanced against the amount of information necessary to perform any useful analysis.

2 Proposed Solution

There are two key questions we must answer when creating a solution. The first is "How far can we mangle the video while still preserving the important data?" This is key because the further we can alter the video data, the more likely that saving the data will be considered "OK" by the community. Google has implemented ideas in this area for their Streetview software for Google Earth. Their software attempts to blur faces, license plates, and various other information. The issue is that Streetview often over compensates, or misses information. This is a huge problem because the cost for false negatives, not blurring pedestrians' faces, is much greater than the cost for false positives, blurring different objects rather than pedestrians' faces. Four key computer vision models have been introduced to our team by Dr. Li [5]: You Only Look Once (YOLOv3) [2] and Single Shot MultiBox Detector (SSD) [1] (for real-time object detection), Mask Region-Convolution Neural Networks (Mask R-CNN) [3] (for instance segmentation), and Dr. Li's Computer Vision model [4] (for assigning an ID for each pedestrian for continuous identification of assigned masks).

According to Dr. Li [5], our team is required to find a middle point allowing us to preserve the privacy of pedestrians while generating useful traffic data. We will build a computer vision model for detecting pedestrians within each frame. State-of-the-art computer vision models, such as You Only Look Once (YOLOv3) [2] and Single Shot MultiBox Detector (SSD) [1], have poor accuracy in detecting objects. These models average %63 at 78FPS and %74 at 59FPS, respectively, compared to other Computer Vision models that have a higher accuracy, but need more computation time for object detection. Therefore, pedestrian detection using YOLOv3 and SSD increases the chances of having more false negatives, not blurring pedestrians' faces, due to the low accuracy of real-time analysis. To reduce the chances of having false negatives generated by our model we will need to separate the object detection model into two stages: pedestrian detection and face detection.

We have decided to separate the project into three phases for the development pipeline: pedestrians detection, object mangling, and extracting public opinion. For the first phase, we have three detection methods that we will be testing: pedestrian detection only using YOLOv3 [2], pedestrian detection and face detection using YOLOv3, and pedestrian detection using YOLOv3 and applying instance segmentation using Mask R-CNN [3]. Each pedestrian method has pros and cons. When applying object mangling on pedestrian detection only using YOLOv3 we might secure the privacy of the pedestrian, but we will lose a lot of information about the traffic because more data will be lost to mangling. Therefore, by applying object mangling on face detection after pedestrian detection, we will preserve the privacy of pedestrians while still having some information about the traffic. This solution will require a lot of computation time and power since it means applying YOLOv3 twice in sequence. Lastly, if the general public requires our model to mangle more than their faces, we will need to use Mask R-CNN rather than face detection. Once again, this method require a lot of computation power and time. Another problem is that we will lose a lot of spatial information of the traffic due to frames lost by Mask R-CNN.

For the second phase -object mangling-, we will need to build mangling methods to mangle pedestrians' images without the ability to reverse the image transformations. Therefore, we have three methods for object mangling that we will be testing: mangling pedestrians using the same mask and color, same mask and different color, and different mask and color. Mangling pedestrian images using the same mask is easy to compute and we guarantee the loss of specialty of the detected pedestrian. However, we will lose some information on the pedestrian's behavior and the traffic. In contrast, mangling using the same mask and different color will solve the lost information problem, but it will require us to use Dr. Li's Computer Vision model [4], which will increase computation time. In addition, if we wanted to extract more information about pedestrians' behavior and the traffic we would need to mangle using a different mask and color. Doing so will increase the computation time and may result in an increased security risk since the mask will have some kind of correlation with the detected pedestrian. For the last phase, we will gauge the public opinion of the each of the mangling methods. Then we will use the model with the median privacy level accepted by the general public and the highest level of information preservation.

3 Performance Metrics

In the hopes that our group can deliver tangible traffic efficiency suggestions to the city of Portland, the project will be evaluated in multiple stages with major requirements focused on accuracy of tracking and privacy coverage of pedestrians. The first area of the project actively measured is the accuracy of the program's ability to recognize and track pedestrians. The end goal is achieving at least 70 percent accuracy. Ensuring a high accuracy in pedestrian tracking is important since the data can affect the level of safety that can be achieved through new changes in pedestrian traffic flow. Another important measured aspect is the level of privacy coverage people feel the program has. This measurement will first be recorded through gauging pedestrians' responses on which level of privacy people are most comfortable with when it comes to the video data used. Based on the responses of the people surveyed, our group will develop the software to maintain that level of privacy. Each level of privacy will have a certain amount of hiding of ones identity. People will be surveyed throughout the build cycle so that our group will design the software with the customers needs always in mind. Consistent feedback and evaluation of privacy standards should meet the pedestrian's expectations by the time we have a final working prototype. The program should also be efficient enough to be used with the amount of processing power the street light provides. Through the accuracy checks, privacy checks, and efficiency checks, our group hopes to have a deliverable solution that provides suggestions for areas of traffic that need to be altered to optimize traffic safety.

Requirements Document

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.

1 Introduction

The software described in this document, Facial Detector and Obfuscator, is a project under the advisement of Chanho Kim (Georgia Tech) 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. The software will be based largely on *YOLOv3* [2].

1.1 System 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.

1.2 System Scope

The scope for this project isimmediately 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.

1.3 Definitions

Term	Definition
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 [6].
Convolutional Neural Network (CNN)	A class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery [7].
Facial Keypoints Detection	Facial detection through the use of multiple key points on a persons face [8].
mean Average Precision (mAP)	The mean for a metric denoting percentage of objects precisely identified, a ubiquitous standard used by object detection models [2].
Obfuscation and Mangling	Used interchangeably. The irreparable destruction of data. Specifically used in relation to identifying features of objects.
VGGFace2	A large-scale face recognition dataset [8].
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 [2].

2 Dataset

2.1 Description

Our team will generate a dataset for the pedestrian detection algorithm using Car Learning in Act (CARLA) [6]. This entails gathering datasets for face detection algorithm from VGGFace2 [8] and applying data pre-processing to clean the data of unwanted noises.

2.2 User Stories

Generate Pedestrians Dataset

As a developer, I want to generate a dataset, using CARLA, of pedestrians so I can train the pedestrian detection algorithm.

Gather Faces Dataset

As a developer, I want to collect a dataset from VGGFace2 to train the face detection algorithm.

Data Pre-processing

As a developer, I want the data to be clean to suppress unwanted distortions or enhance some image features important to further processing.

3 Pedestrian Detection

3.1 Description

Our team will employ the You Only Look Once (YOLOv3) [2] object detection model for identifying pedestrians. By employing this model we will aim to balance mean Average Precision (mAP) with the processing speed necessary to be effective in a real time detection environment.

4 User Stories

Setup the Configuration of YOLOv3

As a developer, I want to configure the *YOLOv3* convolutional network layers, confidence thresholds, input and output pipelines, so that I can begin feeding data to the model to begin training.

Training Model

As a developer, I want to use the selected (and labeled) data set to train *YOLOv3*. This will be a reiterative process as I interpret the output and discover how best to balance the mAP of the object detection model with processing speed.

Adapting Dataset

As a developer, I want to select a data set to train *YOLOv3*. This data set will need to be properly labeled and adapted to be fed into the object detection model.

Deploying Model

As a developer, I want to deploy the adequately trained *YOLOv3* object detection model.

Storing Data

As a city official, I want the resulting data to be useful for open machine learning for the public.

5 Face Detection and Obfuscation

5.1 Description

Our team will train a convolutional neural network (CNN) with Facial Keypoints Detection to find pedestrian faces using the *VGGFace2* dataset. By having a threshold of the number of key points detected, we will then label that area as a detected face. Our team will finally apply an obfuscation technique to remove identifying information of the face from the data.

5.2 User Stories

Determine Facial Keypoints

As a developer, I want to determine facial keypoints that will be used so that I can have a set group of points to focus on within the *VGGFace2* data.

Create a Simple Neural Network

As a developer, I want to create a simple neural network so that I have a base model to build my program off of.

Form Convolutional Neural Network

As a developer, I want to form a convolutional neural network so that I can better analyze the dataset.

Train Convolutional Neural Network

As a developer, I want to train my *CNN*s so that I can use the facial keypoints to train the program to detect other faces than the ones tested with.

Data Removal

As a city official, I want the tool to destroy personally identifiable information from the data in an irreparable manner.

6 SURVEY

6.1 Description

Our team will perform data collection through polling and surveys to ascertain public opinion on privacy. This is specifically in relation to the facial obfuscation/mangling our system will implement. We will judge the correctness of our system using our analysis of this data as a key marker.

6.2 User Stories

Query the General Public

As a city official, I want a baseline of data that any inferences can be checked against. This data should come from a breadth of people.

Options of Obfuscation

As a developer, I want the queries to relate to ways of obfuscating personally identifiable data so I can concretely know what method to use.

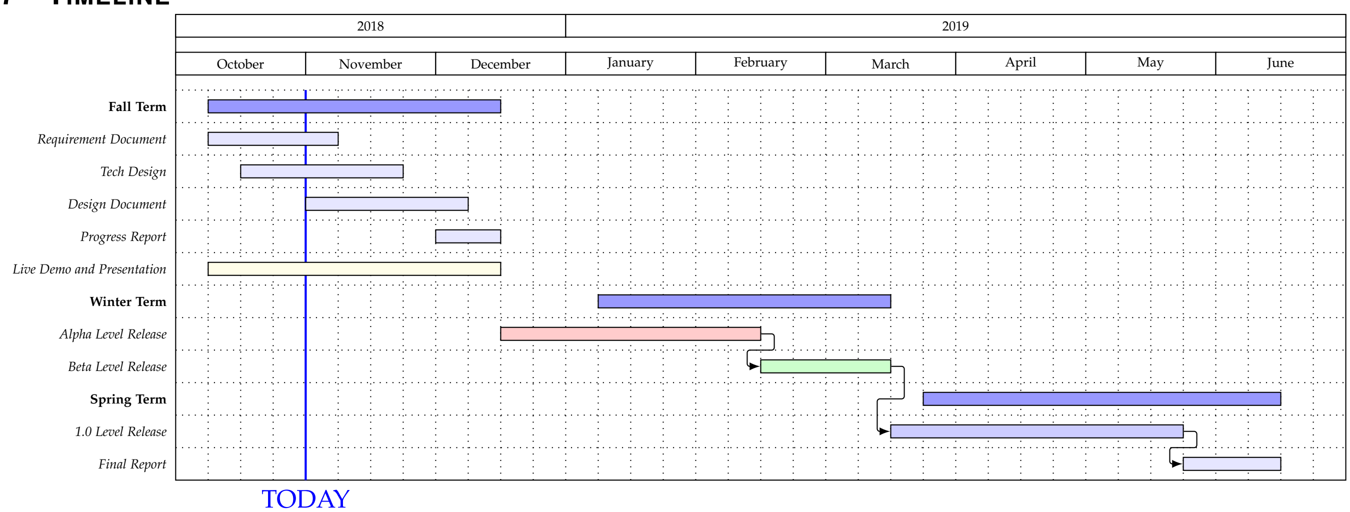
Simple But Comprehensive Results

As a city official, I want questions that are not open ended to draw quick conclusions from. Questions should be varied, but relevant.

Unbiased Tests

As a participant, I want polls and surveys that don't misrepresent my feelings or push me to answer in a way inconsistent with my views on the issue.

7 TIMELINE



.

Design Document

1 Overview

1.1 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.

1.2 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.

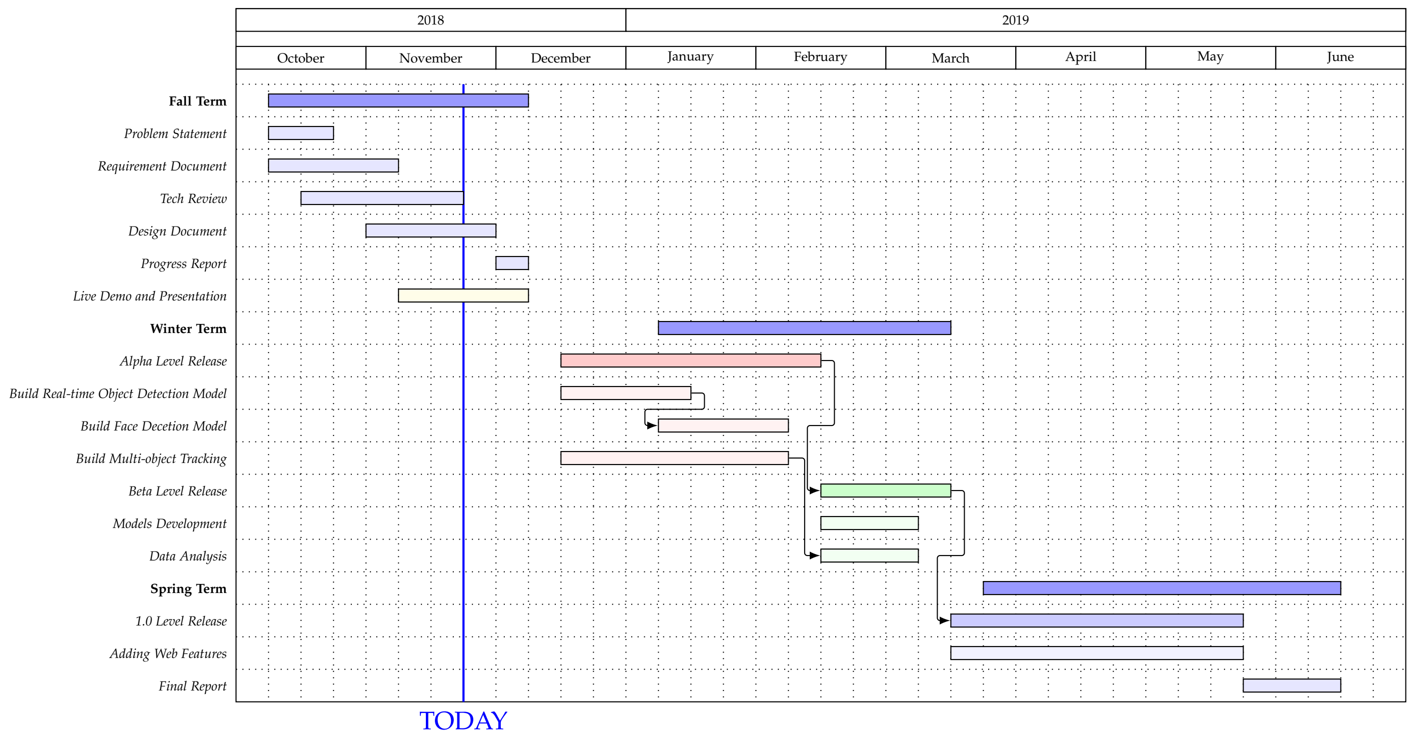
1.3 Glossary

Term	Definition
Convolutional Neural Network (CNN)	A class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery [9].
Recurrent Neural Network (RNN)	A class of artificial neural network where connections between units form a directed graph along a sequence [**?**].
Long Short-Term Memory networks (LSTMs)	A special kind of RNN, capable of learning long-term dependencies [**?**].
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 [2].
mean Average Precision (mAP)	The mean for a metric denoting percentage of objects precisely identified, a ubiquitous standard used by object detection models [2].
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 [6].
VGGFace2	A large-scale face recognition dataset [8].
Facial Keypoints Detection	Facial detection through the use of multiple key points on a person's face [8].
Obfuscation and Mangling	Used interchangeably. The irreparable destruction of data. Specifically used in relation to identifying features of objects.

1.4 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.

1.5 Design Timeline



2 Design Viewpoints

2.1 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.

2.2 Viewpoint: Real-time Object Detection

2.2.1 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 subsubsections 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.

2.2.2 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.

2.2.3 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[2]. 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.

2.2.4 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.

2.3 Viewpoint: Face Detection

2.3.1 Viewpoint Description

Our group will conduct facial detection using a feature-based method by training a convolutional neural network (CNN) and validating our results. For the feature-based method, the facial features are detected through examining the edge, intensity, color, shape, etc. of a feature. Training of a facial detection model allows our group to use the video footage from the city of Portland and define the faces we are looking for. The data we use is from low-quality footage where the environment can take different forms, so it is important that we can train the data in those various settings. The validation of facial detection is responsible for ensuring that the results from our program are true to the information that is actually within the video footage. The final criteria the facial detection program is to achieve at least 70 percent accuracy of detecting faces.

2.3.2 Design Concerns

Main concerns of the design are the the quality of video to work with and ensuring consistent detection accuracy. The video footage provided by the city of Portland is low quality and is taken from various perspectives. Low quality footage may not allow us to make out distinct facial features, so we will need to make sure we train our model with different quality face images. With the video taken from different perspectives, the CNN must also be trained with multiple sides of the face from several different camera angles. Another design concern is ensuring high accuracy of results within various times of day and weather conditions. The video footage is taken from outside environments where rain, fog, snow, and other elements may alter the visibility of the scene. This stresses the importance of training the program with many different environments and consistently testing the accuracy of the detection to understand what may need to be adjusted. To avoid these design concerns, our group must have in-depth CNN training and validation.

2.3.3 Design Elements

Design elements of creating a facial detection program includes training a CNN and validation of results. To implement feature-based facial detection, first the key facial features that will be tracked must be determined. Our group will take snapshots of the traffic video footage and use those frames to train with initially. The image is examined to find common variances in the face, grouping the components together to ensure they match. By having a threshold of the number of key points of features detected, we can then label that area as a detected face. 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. The results of the program should output a JSON file containing the detected facial feature data.

Once the CNN is trained to detect faces, the results are validated by testing the accuracy between our program versus that of industry-leading facial detection software, Face by Microsoft. Face uses the distinctive features of a face to mark a section of an image to be a person.The software examines 27 different elements of a face including the face edges, hair, eyes, eyebrows, mouth, nose, chin,and more. Each facial feature is measured with an x and y value delivered in a JSON file. This technology has a 70 percent accuracy, focusing on the facial features, making it a very accurate implementation. With the same accuracy goals and output type as our program, it makes it perfect to test against.

2.3.4 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.

2.4 Viewpoint: Multi-object Tracking

2.4.1 Viewpoint Description

Multi-object tracking tasks to localize multiple objects, maintaining their identities, and predict their individual movements given an input video. Mazen Alotaibi will be the main designer of the subsystem relating to building the multi-object tracking system supervised by Dr. Fuxin Li and Chanho Kim because of their research experience in developing multi-object tracking computer vision systems. The main purpose of this system is to collect and pre-process data to describes the behavior of pedestrians, bikers, and cars in traffic. The collected data should include the bounding boxes of all relevant objects as well as unique identifiers for each object in the image. The expected outputs of the computer vision system are heat maps of objects, pose estimate, and movement tracking to be used in the data analysis stage.

2.4.2 Design Concerns

Our main concerns in the multi-object tracking system are related to generating data-sets for training the computer vision system, building the computer vision system, and validating the result of the computer vision system over rain, snow, and other weather conditions as well as changes of lighting conditions. Generating data-sets is challenging because of two reasons: legality of using videos without the consent of captured pedestrians and labeling the captured objects within video frames.Because of the complexity of recording unconsenting individuals, it is crucial that preserving privacy is our highest priority. Our system should seek to not only localize people within the image, but also to censor their faces in order to preserve their anonymity. Because we might participate in a Multiple Object Tracking competition to test our computer vision system, we expect our computer vision system to have high accuracy and speed when tracking multiple objects within any video format. Lastly, because our computer vision system will be used on surveillance cameras that monitor inner cities and highways, our computer vision system's accuracy of detection and tracking stay be high over rain, snow, and other weather conditions as well as changes of lighting conditions.

2.4.3 Design Elements

Due to the requirement of having the computer vision system ready for production after develop, our optimal solution for the development framework is PyTorch. The preview version PyTorch 1.0 was chosen as our framework for our computer vision system due to its speed and simplicity. Pytorch 1.0 utilizes the machine learning framework Caffe2 for its backend operations. PyTorch and Caffe2 are collaboratively developed and managed by Facebook and Microsoft and is built with scalability and simplicity in mind. In addition, PyTorch allows us to import our computer vision system into production C++ code without the need to rewrite or change any of the PyTorch code that has been written in Python. Moreover, the expected output of the computer vision system should be in JavaScript Object Notation (JSON) format because that what the City of Portland expect when obtaining the data for production use. For the development of the computer vision system structure, our computer vision system will use YOLOv3 for object tracking and Dr. Li's computer vision system [4], which uses LSTM cells to obtain the spatial information of detected object over time.

2.4.4 Relationship

Multi-object tracking is the most complex part in our project because obtaining more useful information that describes pedestrians, bikers, and cars in traffic is needed for the data analysis stage. In addition, developing the a decent multi-object tracking system for our project requires a lot of development and understanding of building computer vision models.

2.5 Viewpoint: Data Analysis and Access

2.5.1 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.

2.5.2 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.

2.5.3 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 general back-end written in Node JS 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, comparative, statistical, traffic theory based, and neural network algorithms. Incoming data-sets will first be split into geographic and time-stamped categories and will be passed into the predictive algorithmic structures for analysis. The outputs from the system will be comprised of predictions on congestion, traffic incidents, and recommendations for traffic flow organization. The final results will be stored in the database structure for access.

2.5.4 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.

2.6 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.

Tech Review

This subsection outlines the technolgoies investigated and chosen for the execution of Pedestrian Counting and Privacy Preservation project. The subsection will discuss about the selection of programming language, deep learning framework, and real-time object detection algorithm. In addition, the document will compare and contrast multiple options within each category based on certain criteria.

0.1 Introduction

The software described in this document, Facial Detector, and Obfuscator, is a project under the advisement of Chanho Kim (Georgia Tech) 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.

0.1.1 System 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.

0.1.2 System 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.

0.1.3 Work Log

subsection	**Author**
Source Control	Ian McQuoid
Programming Language	Mazen Alotaibi
Framework	Mazen Alotaibi
Facial Detection	Stephanie Allison Hughes
Obfuscation	Ian McQuoid
Real-time Object Detection	Miles Davies

0.2 Source Control

...

0.2.1 Options

...

0.2.2 Criteria Being Evaluated

...

	C++	**Java**	**R**	**MATLAB**	**Python**
Speed	1	0	1	1	0
Learning Curve	0	1	1	1	1
Costing	1	1	1	0	1
Community Support	1	0	1	1	1
Production Ready	1	1	0	0	1
DNN Frameworks Support	1	0	0	0	1
Total	**5**	**3**	**4**	**3**	**5**

TABLE 1: ...

0.2.3 Discussion

...

0.3 Programming Language

0.3.1 Options

We have four options for the selection of the programming language that will be used, C++, Java, R, MATLAB, and Python. According to a survey of 2,000+ data scientists and machine learning developers about which languages they use and what project they are working on [10], 58% use Python, 44% use C/C++, 33% use R, and 13% use Java in general. When asked about prioritization for development, 33% prioritize Python, 19% prioritize C/C++, 5% prioritize R, and less than 1% prioritize Java. Moreover, for general usage, Python is usually used because it is an easier and faster way to build highly-performing algorithms because of its libraries support. C/C++ is mostly used in AI in games, robot locomotion, and embedded computing because of its level of control, high performance, and efficiency. R is usually used in bioengineering and bioinformatics because of its long history with biomedical statistics. Java is usually used in network security, fraud detection, and enterprise focus projects.

0.3.2 Criteria Being Evaluated

We will evaluate each option based on speed, learning curve, costing, community support, production ready, and Deep Neural Networks (DNN) frameworks support.

	C++	**Java**	**R**	**MATLAB**	**Python**
Speed	1	0	1	1	0
Learning Curve	0	1	1	1	1
Costing	1	1	1	0	1
Community Support	1	0	1	1	1
Production Ready	1	1	0	0	1
DNN Frameworks Support	1	0	0	0	1
Total	**5**	**3**	**4**	**3**	**5**

TABLE 2: Comparison between Programming Languages based on selected criteria

0.3.3 Discussion

C++ is the most powerful programming language out of these options because it is the only low-level programming language selected, which means high performance and efficiency. C++ is supported by the community and most of Deep Neural Networks frameworks, such as PyTorch and Caffe. However, C++ has a sharp learning curve, which makes the development time slower. Java is easy to learn to learn and production ready, but Java isn't supported by any of the Deep Neural Networks frameworks. R is designed for statistical analysis and visualizations, but R isn't supported by any of the Deep Neural Networks frameworks and it isn't ready for production. MATLAB is designed for mathematical modeling and computer vision. MATLAB is supported by the community and Caffe, a Deep Neural Networks framework. However, MATLAB is the only language that you need to pay for its license. Python is the only programming language from this list that has the largest community support and Deep Neural Networks framework support. In addition, Python is easy to implement and production ready. However, Python is slow compared to C++ [11][12].

From my research, the best two programming languages based on our criteria are C++ and Python. C++ is fast but hard to develop. Python has every criterion checked besides speed. However, our client, City of Portland, wants a proof of concept of a way to transform the data from their traffic cameras so the city may store the data with storing identifying information about the citizens in the footage. Therefore, we will be using Python as our best option.

0.4 Framework

0.4.1 Options

We have four options for the selection of the deep learning framework that will be used, TensorFlow, Keras, and PyTorch. TensorFlow, Keras, and PyTorch created by Google in November 2015, a Google engineer (Francois Chollet) in December 2017, and Facebook in July 2018 respectively. TensorFlow and PyTorch are adopted by several giant companies, such as Twitter, IBM, AirBus because of their highly flexible system architecture [13]. TensorFlow is mostly used in voice/image recognition, text classification/summarizing, and natural language processing because of the development tools supported by Google Translate. Keras is used for prototyping because its high-level syntax and architecture, however Keras isn't always easy to customize the Neural Network because of its high-level architecture. Therefore, researchers tend to use Keras as front-end and TensorFlow as back-end for fast development and easy to customize the Neural Networks. In other hand, PyTorch already has these features as it is used for training deep learning models quickly and effectively so it is the framework of choice for a large number of researchers. In addition, TensorFlow, Keras, and PyTorch are fully support Python only [14].

0.4.2 Criteria Being Evaluated

We will evaluate each option based on speed, learning curve, documentation, community support, framework logic structure, and full customization of Neural Networks (NN).

	Keras	**TensorFlow**	**Keras/TensorFlow**	**PyTorch**
Speed	0	1	1	1
Learning Curve	1	0	1	1
Documentation	1	1	1	0
Community Support	1	1	1	1
Framework Logic Structure	0	0	0	1
Full Customization of NN	0	0	0	1
Total	**4**	**3**	**5**	**5**

TABLE 3: Comparison between Deep Learning Frameworks based on selected criteria

0.4.3 Discussion

Keras is the easiest to learn and implement compared to TensorFlow and PyTorch and it is supported by the community. However, Keras isn't fast and doesn't allow full customization of Neural Networks. In the other hand, TensorFlow is fast and has the largest community support. However, TensorFlow is hard to implement because TensorFlow's calls structure and it doesn't allow full customization of Neural Networks. Moreover, Keras and TensorFlow can be used together, which will allow us to use Keras easy syntax and TensorFlow speed. But we can't fully customize the Neural Networks. PyTorch has all TensorFlow and Keras features besides documentation and. Although PyTorch is created by Facebook AI Research Lab, it is still young, 2018, compared to TensorFlow, 2015. Nonetheless, PyTorch is the only framework that allows full customization of Neural Networks and has the best logical structure of the framework of the data pipeline [13] [14] [15].

From my research, the best two framework structure based on our criteria are Keras/TensorFlow and PyTorch. Keras/TensorFlow has simple syntax and easy to implement, and PyTorch has the best logical structure of the framework and the only that allow full customization of the Neural Networks. Although Keras/TensorFlow will help us start the project fast as our advisor, Chanho Kim, uses TensorFlow mainly, selecting PyTorch will be the optimal option because PyTorch allows us to fully customize the Neural Networks, which we might need to when we change our current computer vision system for research purposes.

0.5 Facial Detection Methodology

...

0.5.1 Options

...

0.5.2 Criteria Being Evaluated

...

	C++	Java	R	MATLAB	Python
Speed	1	0	1	1	0
Learning Curve	0	1	1	1	1
Costing	1	1	1	0	1
Community Support	1	0	1	1	1
Production Ready	1	1	0	0	1
DNN Frameworks Support	1	0	0	0	1
Total	**5**	**3**	**4**	**3**	**5**

TABLE 4: ...

0.5.3 Discussion

...

0.6 Facial Detection Technology

...

0.6.1 Options

...

0.6.2 Criteria Being Evaluated

...

	C++	Java	R	MATLAB	Python
Speed	1	0	1	1	0
Learning Curve	0	1	1	1	1
Costing	1	1	1	0	1
Community Support	1	0	1	1	1
Production Ready	1	1	0	0	1
DNN Frameworks Support	1	0	0	0	1
Total	**5**	**3**	**4**	**3**	**5**

TABLE 5: ...

0.6.3 Discussion

...

0.7 Obfuscetion

...

0.7.1 Options

...

0.7.2 Criteria Being Evaluated

...

	C++	Java	R	MATLAB	Python
Speed	1	0	1	1	0
Learning Curve	0	1	1	1	1
Costing	1	1	1	0	1
Community Support	1	0	1	1	1
Production Ready	1	1	0	0	1
DNN Frameworks Support	1	0	0	0	1
Total	**5**	**3**	**4**	**3**	**5**

TABLE 6: ...

0.7.3 Discussion

...

0.8 Real-time Object Detection

...

0.8.1 Options

We have four options for the selection of the real-time object detection algorithm that will be used, Faster Regional-Convolution Neural Networks (Faster R-CNN)[16], Mask R-CNN[3], Single Shot MultiBox Detector (SSD)[1], and Yon Only Look Once (YOLOv3)[2]. Faster R-CNN uses Convolution Neural Networks as feature extractor and Region Proposal Network (RPN), a region proposal method by an internal Deep Network and the Region of Interest (RoIs) are derived from the feature maps instead. SSD uses VGG19 netowrk as feature extractor and convolution layers followed by convolution filters to make the prediction. YOLOv3 uses DarkNet 53 as backbone for feature extraction and Feature Pyramid Networks (FPN), composes of a bottom-up and a top-down pathway, to detect small object better [17].

According to a research comparison [18], Faster R-CNN has a small accuracy advantage if real-time speed isn't needed. SSD has a good frames per seconds for lower resolution images at cost of accuracy in real-time processing. However, SSD performs much worse on small object detection. For both algorithms, they both perform better when the input image resolution is high and SSD accuracy for detecting small object increases. In the other hand, when decreasing the resolution of the input image into half, the accuracy is decreased by 15.88% on average for both Faster R-CNN and SSD.

0.8.2 Criteria Being Evaluated

We will evaluate each option based on implementation, real-time detection of small, medium, and large objects, segmentation, frames, and mean average precision (mAP).

	Faster R-CNN	**Mask R-CNN**	**SSD**	**YOLOv3**
Implementation Complexity	1	0	0	0
Detection of Small Objects	1	1	0	0
Detection of medium Objects	1	1	1	1
Detection of Large Objects	1	1	1	1
Segmentation	0	1	0	0
Speed	0	0	1	1
Total	**4**	**4**	**3**	**3**
Frames	17	0	59	91
mAP	34.9	0	26.8	33

TABLE 7: Comparison between Real-time Object Detection Algorithms based on selected criteria

0.8.3 Discussion

Faster R-CNN [16] has the highest accuracy compared to SSD [1] and YOLOv3 [2], and Faster R-CNN is easy to implement and the only one that can detect small objects. However, Faster R-CNN has the lowest frames per second, which means a lot of spatial information of continue object detection lost due to this. In addition, SSD and YOLOv3 have similar results, but YOLOv3 have more frames per second and higher accuracy compared to SSD [18] [19] [17].

According to Dr. Li [5], our team is required to find a middle point allowing us to preserve the privacy of pedestrians while generating useful traffic data. We will build a computer vision model for detecting pedestrians within each frame. State-of-the-art computer vision models, such as You Only Look Once (YOLOv3) [2] and Single Shot MultiBox Detector (SSD) [1], have poor accuracy in detecting objects. These models average 33% at 91 frames per seconds and 26.8% at 59 frames per seconds, respectively, compared to other Computer Vision models that have a higher accuracy, but need more computation time for object detection. Therefore, pedestrian detection using YOLOv3 and SSD increases the chances of having more false negatives, not blurring pedestrians' faces, due to the low accuracy of real-time analysis. To reduce the chances of having false negatives generated by our model we will need to separate the object detection model into two stages: pedestrian detection and face detection.

From my research, I conclude that YOLOv3 is the optimal real-time object detection algorithm because the project is mostly focused on real-time object detection. In addition, by implementing YOLOv3 using PyTorch, we will be able to change the backbone of YOLOv3 with better backbone that suits our problem. Moreover, we will be detecting pedestrians within each frame, which means we will insure a higher accuracy than 33% as pedestrians are medium size objects.

0.9 Conclusion

To sum up, we have decided to select Python, PyTorch, and YOLOv3 as our main programming language, Deep Learning framework, and real-time object detection algorithm respectively. Python because it is easy to learn and develop, supported by the community, and supported by Deep Learning frameworks. PyTorch because it allows us to develop and fully customize the Neural Networks. Finally, YOLOv3 because it is the best real-time object detection algorithm from my research.

Progress Report

The City of Portland is attempting to gather data on traffic patterns to increase pedestrian and driver safety and to create an open-source data-set on the topic. The Pedestrian Counting and Privacy Preservation project aids this goal by allowing the city to store video and photo information without storing identifying information about the people in the data. Serving as an archive of the steps and plans for Pavement Prometheus project, the following document provides an overview of the project's past, current state, and goals for the future. The document will also cover the overarching design for the project and the problems that the team had encountered over a three-month time-frame.

1 Introduction

1.1 The Problem

The City of Portland has partnered with companies such as AT&T as part of the Smart City PDX initiative. The goal of the project is to use technology to better the lives of the city's citizens, specifically helping bridge the technological divide and help under-served communities. One aspect of the initiative is to use traffic and roadside cameras to develop open-source data-sets for the community to use for individual projects and for the city to use to collect data on traffic and pedestrian patterns. The data will be used to inform legislative and construction decisions as well as traffic decisions including traffic-light timing. The main problem that the city has run into is in storing the data. Currently, the data captured by the cameras has identifying information on people pictured in the media and, as a result, can not be stored for further analysis or use. The Pedestrian Counting and Privacy Preservation project primarily serves to fix this problem by stripping the incoming data of personally identifiable information so that it can be stored and further analyzed.

1.2 Design

Our project has been split into four different components that will each be worked on individually and combined to form an overarching solution to privacy preservation in the face of data collection. The first component will be a real-time object detection system which, as its name entails, will involve detecting objects in real-time from a given camera feed. For the purposes of this project we will be primarily interested in identifying both vehicles and pedestrians. The second component will be face detection, which will both serve to reinforce the accuracy of the object detection in identifying pedestrians, but will also be vitally important in obfuscation and masking for privacy preservation. Our third component will involve an object tracking system, which will primarily serve a purpose for extracting interesting metrics in serving the data collection aspect of this project. Our object tracking system will also help ensure accuracy for such things as aggregate data counts, as real-time object detection detects objects on a frame-by-frame basis and has no awareness of identifying a unique object between frames. Finally, we will have a data analysis and access component that will ultimately involve an online API for both storing and accessing the data collected to a database owned by the City of Portland.

1.3 Goals

The goals of our project are to develop a program to detect pedestrians, deliver data in a JSON format to the City of Portland, and develop a research paper based on our findings. When it comes to pedestrian detection, we hope to train a convolutional neural network to detect the bodies and faces of pedestrians in real-time. The results of this program will be analyzed and validated so pedestrians are detected at at least a 70 percent accuracy. Along with the face detection, the program will obscure the faces of those pedestrians so their identity is not compromised.

Our next goal is to deliver the pedestrian detection data in the form of a JSON file to the City of Portland. Our group will create an API that can be used to output pedestrian data from our program in a JSON format. This will allow developers for the City of Portland to easily access and use the data from our program. The last goal is to devise research paper focusing on pedestrian privacy and detection. Our group will investigate pedestrians preferred level of privacy for detection and base the development of our program on those preferences. Our research will focus on the detection of pedestrians while ensuring the privacy of ones identity through obscuring their faces. Our group will work to fulfill the desires of the general public while keeping as much data as possible so it can be used for future analysis.

2 Plans

2.1 Object Detection

Real-time object detection will be the first stage of analysis performed on a provided camera feed. This portion of the project is designed to implement an algorithm which can detect classes of objects in a timely manner which can then be fed to other subsystems, like face detection and object tracking, for further analysis. To reach this objective, we will be implementing the You Only Look Once object detection algorithm on a PyTorch deep learning framework, our choices being predicated on both speed and accuracy of a given object detection algorithm, and on the utility of a deep learning framework. While we are still looking for potential datasets to train our model against, we will likely use CARLA (an open source vehicle and pedestrian simulator which labels objects recorded automatically) to initially produce footage for training and later validating our model. We also have some interest in tweaking the YOLO object detection algorithm to potentially improve accuracy and speed, possible solutions including reducing object classes recognized.

2.1.1 Face Detection/Obfuscation

Face detection will be used to detect the faces of pedestrians using a feature-based method so their identities can be protected. Our group will develop a program to train a convolutional neural network to detect the facial features of a pedestrian. We decided to implement a feature-based method as it allows for faces to be detected easily from different perspectives and environments. While we detect the full bodies of pedestrians for motion and location data, the faces are detected for privacy matters. A human face is the most highly identifying part of a person, so our group detects it to be obscured. Through obscuring the face, all vital pedestrian data is recorded except for the pedestrians distinctive features. Our project goal is to research the detection of pedestrians while ensuring privacy protection, so face detection allows us to accomplish this.

2.2 Data Analysis

A large portion of the data that the project will aggregate will be in a pure format e.g. the number of vehicles in a frame, their speed and trajectory, what lanes of highway contain the most traffic. This information, including the raw photographic and video data, is useful for constructing data sets, but is less useful for providing answers to decision problems. Analysis of the amassed information will be essential in providing concrete advisement to the city. The data will be analyzed using a mixture of comparative, statistical, traffic theory based, and neural network algorithms with the goal of providing information concerning traffic flow suggestions and traffic incidents. Analyzed data will be made accessible through a web interface. The interface will serve data in a JSON format including geographic and chronological markers for the data.

3 PROBLEMS

3.0.1 General Datasets

Our group had multiple obstacles in our search to determine the best dataset to use for our project. First, we had to determine which format of data would work the best to serve all our needs, yet there seemed to be issues with each option. The City of Portland provided video feeds of different streets around Portland to take images from, but the image quality was poor and the live feed was too fragmented. Our group then looked into using a well-known, large dataset of faces to train our model, but just having that data would still lead to a lot of work. While our group would have a variety of face data to work with, we would need to individually label each pedestrian in an image. Finally, we investigated using a simulated environment to train our program with, but the software would need to be installed on the OSU server.

3.0.2 CARLA Datasets

CARLA is a system for supporting the training and validation of autonomous driving systems. The great draws to CARLA are that the data created is a fair representation of traffic and pedestrian movement including pre-labeling of objects and the freedom that the system affords. Our team hopes to use the system to help train our project. One problem that the team ran into while using the CARLA system is that since the purpose of CARLA is for autonomous driving, the camera was meant to be situated in a car. Moving the camera to a stationary position broke much of the usability of CARLA, including removing segmentation for cars and pedestrians, and causing the system to pause. For a decent amount of time, the automated pathing systems used for determining the trajectory and interaction between agents in the CARLA system was irregular and caused cars to crash unexpectedly, and pedestrians to walk into walls continuously and clash with each other in crowded situations. This could have unforeseen effects on the finished system as CARLA data would form the base for our training.

3.1 Goals

The outline of our project as initially described was that we would be working with hardware provided by General Electrics and Intel for the Smart City PDX initiative in Portland, where both companies had provided a cumulative total of 200 sensors along four of the citys busies traffic avenues. Because of Portlands privacy concerns, storing unadulterated video footage from the cameras was not allowed, and so they required a method to extract interesting information from the footage while simultaneously not compromising any personal information. After our first meeting with the City however, we discovered that GE and Intel had sole control over the footage recorded by these cameras, and that gaining access to that hardware would likely be impossible as nothing in their contract with the city stipulated shared use. This apparently inspired, in part, by the citys reluctance to handle any details which might violate the privacy of its citizens. Instead we have access to somewhat more basic cameras which refresh an image once every three minutes, from which the city is only interested in our ability to detect vehicles.

4 Solutions

4.1 Datasets

4.1.1 General

To tackle our issues involving determining the best dataset to use, our group focused in on what we really needed. Our group developed criteria for narrowing down the dataset forms based on accuracy, ease of use, and labor intensity. By comparing and contrasting our options along with discussions with our mentor, we determined to use the simulation software called CARLA. If we used the video data provided by the City of Portland, the low quality of the footage may decrease the accuracy of detection. Our group would need to individually screenshot many frames of the video and label pedestrians one-by-one, making data collection difficult and time consuming. This setup would not allow us to send the City of Portland a program that could be easily developed by others in the future. Similar obstacles are faced when using the large face dataset to work with as people will still need to be individually labelled within images. This narrowed our debate down to the simulated environment as the software has people pre-labelled and has clear resolution. This makes training and development with the simulation to be accurate, simple to use, and low intensity, meeting all of our requirements. By setting clear criteria, talking with mentors, and comparing practicality, our group was able to overcome our obstacles to determine the best dataset.

4.1.2 Carla

The problems identified with the CARLA training and verification system have been noticed by the development team and a patch that addresses the issues is currently being worked on. This leaves our team in a position reliant on the CARLA development team. In the situation that the CARLA team does not address the issues in a timely manner, our team will need to find a separate data-set generator, or use a static data-set for training our neural networks.

4.2 Goals

Given the obvious discrepancy from the initial goals of our primary client, Dr. Li, and the requested objective from the City of Portland, we have had somewhat of a need to rectify the difference. As it currently stands we plan on meeting all of Portlands requested objectives, presenting a demo at the beginning of this December, in the hopes that theyll be able to expand the scope of their requested project. With some potential coming from other companies looking to do business with Portland who might provide access to better camera hardware then what is currently available. Barring that, however, Dr. Li still wishes to meet our initial goals of devising a system that can detect and obfuscate personally identifying information from a camera feed in real time. As the project would serve as both an excellent learning opportunity and be of immediate use to any city interested in preserving privacy while taking advantage of large scale data collection operations.

5 Weekly Summary

5.1 Week 1

The first week of our senior capstone class covered the production of a resume for the purposes of future employment. The class also had us write a fictional autobiography regarding what route wed like to take through our lives; putting us in a mindset that had us planning for the future. Otherwise this week was mostly introductory material for how the course would proceed.

5.2 Week 2

The second week of our senior capstone class we had to choose a project to get involved with. Selection involved reading through a list of potential project details and selecting, in ranked order, our top ten preferences. We would then select a project that we explicitly did not want to be a part of, so that it would be removed from the list of potential candidates. Lectures consisted of working on voice tone in constructing emails to be sent to your client.

5.3 Week 3

The third week we were able to discover which team we had been assigned to, in our case the Pedestrian Counting and Privacy Preservation project. We were able to meet up with our client, Dr. Fuxin Li, who in turn imparted us the goals for this project and how it tied into the Portland Smart City PDX initiative. While the entire team was enthusiastic about the project some of us werent that familiar with Machine Learning, presenting an important subject to learn over the course of the quarter. Ian this week setup a github repository with a Kanban board with defined program flow and work goals, while Maze took the initiative to become our point of contact with the professor.

5.4 Week 4

This week my our team made progress on getting the Github working, met with the T.A and planned out our project through the group problem statement. For the joint Github, each member practiced accessing the shared repository and uploaded our latex files to the Documents folder. We also added the kanban board extension to the Github in order to plot out the progress of our tasks. This week my group also met with the T.A. in the Kelley Engineering Center, discussing our progress and learning about the layout of the T.A. meetings. Along with this, my group worked through the group Problem Statement, which allowed us to all get on the same page for the problem, plan, and solution that were going towards.

5.5 Week 5

This week our was able to meet with the T.A. and our client Dr.Li to go over the Requirements Document and get greater clarification on our project outline. In our meeting with our T.A., we were able to get important feedback on our Problem Statement to understand where we can improve on future papers. We also were able to go through examples of the Requirements Document from previous years so we could have a better understanding of the layout of the document. Our group worked to setup our project with the right connections by emailing Kevin McGrath about available GPUs, Dr. Bailey to schedule a meeting for the next week, and Mr. Kim to schedule a weekly meeting time. We also met with our client Dr.Li and met his other graduate student doing research under him. We talked to them about the requirements for our project and got good insight as to the software we could work with. The meeting allowed us to align our program goals and resources to those that our clients had researched with in the past in order to track our progress.

5.6 Week 6

This week our team met with our Dr. Fuxin Li and City of Portland Representative to discuss about the project goals. Our team encountered a difference in direction and project outline from our team's initial understanding with the City of Portlands desires. Going into the meeting with the City of Portland representative, our team believed that they were looking for a program that would focus more purely on pedestrian movements and that their highest priority was to protect the privacy of the pedestrians. Coming out of the meeting we learned that the representative is looking to detect vehicles as well and make more of the programs focus to be on high accuracy, with privacy as an added benefit. Due to this, our team met up after the meeting and fully wrote out our outline for the project with the desires of the City of Portland in mind and I believe we are now all on the same page. In addition, we have meet with Chanho Kim to discuss about our issues with our attempt to extract useful samples for CARLA to be used in later in training.

5.7 Week 7

This week our team met with our grading TA Richard to discuss the formatting of the final technology review document along with the next design paper coming up. In addition, we also video-chatted our mentor Chanho to discuss future steps to get data to be used to train a CNN including the CARLA simulator. Based on the advice from our mentor we will continue to explore different sources for our data other than the current simulation we have been looking at. Moreover, our team went through our technical review documents and further discussed each persons role within the team. We met up and went through each of our documents to understand other team members perspective of their work and see how each of our contributions comes together with the full project.

5.8 Week 8

This week our team met with our grading TA Richard to discuss the upcoming Design Document. Our team began work on an online course on machine learning provided through Facebook to help better our understanding of the tools used for our project and to know how the tools interact. One problem that we ran into occurred when one team team member downloaded a large data-set, but while trying to overclock his processor, he accidentally crashed his operating system. The crash caused a loss of the data set, but didn't cause irreparable damage to the project. We ended the week by starting work on a demo for our clients.

5.9 Week 9

This week our team member, Mazen, finished his work on a demo by applying YOLO to a couple of traffic videos he downloaded from YouTube as well as photos downloaded from the ODOT and City of Portland web-pages. One issue that our team found was that the partitioning of roles for the project had become more difficult than we originally expected with overlaps being common. We created final roles where every team member would have a role that aligned with their interests and had a reasonable amount of work relating to it. The issue relating to the CARLA system persisted into this week.

5.10 Week 10

This week our team worked on the progress report document and presentation. The progress report required our team to distill project and the past ten weeks into a single information sink. Our team also began outlining the data-analysis portion of our project including how we will implement the system and what its concrete goals are.

6 Retrospective

Week	Positives	Deltas	Actions
1	Covered introductory material for the senior capstone class.	Do not yet know what project to apply for.	Completed resume and fictional autobiography to help career preparation.
2	Capstone class lecture covered the importance of voice tone, and how best to go about addressing your client in a professional setting.	There is uncertainty about which project we will ultimately be assigned to.	We picked which projects we wished to apply for in the upcoming year.
3	Our team discovered we had been assigned to the Pedestrian Counting and Privacy Preservation project.	We need to organize a meeting time with our client to discuss the specifications and particulars about the project.	Met with our team, exchanged contact information and decided on what meetup times would work best for us.
4	Team set up the foundations of our project by creating a shared Github repo for our work. Went through project purpose of project to be better aligned on our project goals.	Do not have proper equipment to run Computer Vision model so will need to contact teacher to gain proper resources.	Conducted meeting to go over Problem Statement document to allow for everyone to be on the same page for the project outline.
5	Team worked with our client to communicate clear project outline and requirements. Received helpful feedback from T.A. to improve future papers and worked with Dr.Li's graduate student to get advice for our project's technology.	All members still trying to fully understand project implementation and will need to do further research. While we are clear on our goal, we realized the way to get there is still being determined.	Held meeting with our client to discuss project feasibility along with meeting with his graduate student to get advice. Researched into possible software and approaches to accomplishing our project's needs.
6	Team met with all clients to discuss about the the project direction and final goal.	Team needed to communicate more with the main client to understand the final goal more.	Have a shared Google Doc with Dr. Li and Dr. Dominguez to be reviewed every even week.
7	Team started working on build YOLOv3 and setting up CARLA.	All team members should contribute on doing something to balance work.	Follow Agile method to separate tasks and have Agile goals for the entire team to contribute to.
8	Team began work on Facebook's machine learning class. Gained better understanding of data models and training sets including the ODOT data from traffic cameras.	Team needed to better understand the separation of duties and the distribution of expertise.	Have a team meeting in which we partition the goals between the members.

9	Finished work on a demo showing that YOLO is working and is performing as expected on video and still images.	Team needed to better understand the goal of the data analysis portion of the project. CARLA was still not completely working.	Ian was tasked with understanding and creating goals and methods for the data analysis system.
10	Team finalized the goals for the project and began work on the progress report document.	CARLA issues persisted into this week.	Team will contact the CARLA development team to find out where the patches are in progress.

7 Conclusion

In Conclusion, our group hopes to implement a program that can aid the City of Portland in their work to increase pedestrian safety. Under the guidance of our mentor Dr. Fuxin Li, our team has made clear progress this term to setup the foundation of our project. The outline of our project is to detect pedestrians through object and face detection, obscure pedestrians identities, and perform data analysis. This allows pedestrian traffic to be tracked while ensuring privacy protection, enabling better city planning efforts in the future. Pedestrian safety is a growing concern with increasing vehicle traffic and privacy becoming an issue with public cameras involved. Our project covers both needs through developing a program that will gather all possible traffic information while obscuring any identifying pedestrian information. Through our research, we hope to pave the way for greater safety and privacy for the people of Portland.

References

[1] 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.

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

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *ArXiv e-prints*, Mar. 2017.

[4] C. kim, F. li, and J. Rehg, "Multi-object Tracking with Neural Gating Using Bilinear LSTM," *Oregon State Univesity*, Mar. 2018.

[5] F. Li, Group meeting, October 2018.

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

[7] D. Cornelisse, "An intuitive guide to convolutional neural networks," Available at https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050 (2018/04/24).

[8] "Vggface2 about," Available at http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/ (2018/10/30).

[9] 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.

[10] C. Voskoglou, "What is the best programming language for machine learning?" Available at https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7 (2017/05/05).

[11] P. Kanada, "Which programming language is considered to be best for machine learning?" Available at https://www.datasciencecentral.com/profiles/blogs/which-programming-language-is-considered-to-be-best-for-machine (2018/07/28).

[12] BizDevCorp, "The 5 best programming languages for ai development," Available at https://www.futureproofing.io/blog/the-5-best-programming-languages-for-ai-development (2018/06/05).

[13] J. Hale, "Deep learning framework power scores 2018," Available at https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a (2018/09/19).

[14] M. Opala, "Deep learning frameworks comparison tensorflow, pytorch, keras, mxnet, the microsoft cognitive toolkit, caffe, deeplearning4j, chainer," Available at https://www.netguru.co/blog/deep-learning-frameworks-comparison (2018/09/06).

[15] A. Ahmed, "Choosing a machine learning framework in 2018," Available at https://agi.io/2018/02/09/survey-machine-learning-frameworks/ (2018/02/09).

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *ArXiv e-prints*, Jun. 2015.

[17] J. Hui, "What do we learn from region based object detectors (faster r-cnn, r-fcn, fpn)?" Available at https://medium.com/@jonathan_hui/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9 (2018/03/28).

[18] ——, "Object detection: speed and accuracy comparison (faster r-cnn, r-fcn, ssd, fpn, retinanet and yolov3)," Available at https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359 (2018/06/19).

[19] ——, "Design choices, lessons learned and trends for object detections?" Available at https://medium.com/@jonathan_hui/design-choices-lessons-learned-and-trends-for-object-detections-4f48b59ec5ff (2018/03/28).