

# Computer Vision and Internet of Things Application to Enhance Pedestrian Safety

Ujjwal Khanna

A Thesis  
in  
The Concordia Institute  
for  
Information Systems Engineering

Presented in Partial Fulfillment of the Requirements  
For the Degree of  
Master of Applied Science (Quality Systems Engineering) at  
Concordia University  
Montreal, Quebec, Canada

September 2020

© Ujjwal Khanna, 2020

# CONCORDIA UNIVERSITY

## School of Graduate Studies

This is to certify that the thesis prepared

By **Ujjwal Khanna**

Entitled **Computer Vision and Internet of Things Application to Enhance  
Pedestrian Safety**

and submitted in partial fulfilment of the requirements for the degree of

**Master of Applied Science (Quality Systems Engineering)**

complies with the regulations of the University and meets the accepted standards with respect to originality and quality

Signed by the final Examining Committee:

\_\_\_\_\_  
*Dr. Chun Wang* Chair

\_\_\_\_\_  
*Dr. Akshay Kumar Rathore* Examiner

\_\_\_\_\_  
*Dr. Farnoosh Naderkhani* Examiner

\_\_\_\_\_  
*Dr. Anjali Awasthi* Thesis Supervisor

Approved by \_\_\_\_\_  
Dr. Mohammad Mannan, Graduate Program Director

September 8, 2020 \_\_\_\_\_  
Dr. Amir Asif, Dean of faculty  
Gina Cody School of Engineering and Computer Science

# ABSTRACT

## Computer Vision and Internet of Things Application to Enhance Pedestrian Safety

Ujjwal Khanna

With the increasing population, the issue of pedestrian safety is currently of major concern in most cities of the world. Pedestrian safety is concerned with ensuring the well-being of pedestrians and reducing the potential risk areas as well as implementing measures to reduce accidents. The aim of this study is to propose a computer vision and cloud-based solution that enhances pedestrian safety by collecting, visualizing and analyzing pedestrian and vehicular data across different intersections in the city of Montreal. In the past, the rate of accidents in the City of Montreal involving pedestrians has been quite high, therefore a method to solve this problem has led to this study.

About 200,000 images were collected across 43 intersections in the city of Montreal from the Traffic cameras – Ville de Montreal website. The data was collected from March 8, 2020, up until March 22, 2020 and then from May 1st, 2020 to 11th May 2020. An object detection and classification model using Faster RCNN algorithm to identify pedestrian and vehicles at the intersection was implemented. Further, this model was used to obtain a dataset showing the number of pedestrians and vehicles at the intersections. The information obtained from this data set was used for visualization and in-depth analysis of the pedestrian and vehicle data in order to derive patterns of peak and non-peak hours and high-risk intersections.

Furthermore, zero inflation poisson distribution model was implemented on our dataset to display the timings and intersections which had zero pedestrian counts for long hours of the day as compared to the vehicle count. A heat map was generated to visualize the dataset and to assist data viewers to identify which areas should get most attention.

Finally, we created a prototype solution that mimicked the traffic control system by utilizing LEDs and microcontrollers (IoT device), cloud services, publish/subscribe model, and object detection. To implement this prototype, the data obtained through the object detection model was sent onto the cloud (Cloud MQTT), from where it was used to control the programmed microcontrollers (IoT devices) present at the different intersections based on the vehicle and pedestrian counts. The system managed to show excellent accuracy for detection of vehicles and pedestrians on the dataset, and the delay experienced in controlling the microcontroller was also negligible, thus making our system effective and reliable.

# Acknowledgements

First and foremost, I would like to thank Professor Anjali Awasthi for her supervision and constant guidance in working on this research. Her perseverance and support allowed me to take steps in the right direction, stay on track, and progress on time with this chapter.

I would further like to express my gratitude towards my colleagues at Concordia University's lab for their constant encouragement and cooperation. In particular, I would like to thank Mohsen Amoie for his constant help and support in challenging situations encountered during my research. I must thank my dear Jasmeen for always being there and helping me with her unconditional support. I also want to thank my friend Akhil for being there during hard times.

Finally, I would like to thank my parents Dr. Rajesh Khanna and Ms. Ritu Khanna and my brother Rohan Khanna for their unwavering support during this challenging and enriching journey. Without their encouragement it would not have been possible for me keep working hard throughout this journey.

# Computer Vision and Internet of Things Application to Enhance Pedestrian Safety

## TABLE OF CONTENTS

<b>List of Figures</b>		X
<b>List of Tables</b>		XIV
<b>List of Abbreviations and Symbols</b>		XV
<b>Chapter 1</b>	<b>Introduction</b>	1
1.1	Research Objective	2
1.2	Contributions	3
1.3	Organization of Thesis	4
<b>Chapter 2</b>	<b>Background</b>	6
2.1	Introduction	6
2.2	Key Terminologies and Technologies	6
	2.2.1 Internet of Things	6
	2.2.2 Neural Networks	7
	2.2.3 Cloud MQTT	8
2.3	TensorFlow	9
2.4	TensorFlow object detection API	10
2.5	NumPy	10
2.6	Pandas	10
2.7	Seaborn	11

2.8	MQTT (publish/subscribe, client/broker)		11
2.9	LabelImg		12
2.10	Computer Vision		12
2.11	ESP8266		13
2.12	NodeMCU firmware		13
2.13	ESPlorer IDE		14
2.14	2.14 Lua		14
2.15	2.15 Conclusion		15
<b>Chapter 3</b>	<b>Literature Review</b>		16
3.1	Introduction		16
	3.1.1	Pedestrian Volume by Identification at Intersections	17
	3.1.2	Signalized and Unsignalized intersections	18
	3.1.3	Object Detection algorithm	20
	3.1.4	Cloud Based Services	21
3.2	Case Study 1: Montreal Gazette		25
3.3	Case Study 2: SPVM (Service de police de la Ville Montreal)		39
3.4	Problem Discussion		49
3.5	Conclusion		54
<b>Chapter 4</b>	<b>Working with Object Detection</b>		55
4.1	Introduction		55
4.2	Neural Networks		55
	4.2.1	Steps involved in Neural Network Training	58
	4.2.2	How a NN works	58
	4.2.3	Tasks Involved in Computer Vision	65
	4.2.4	Convolutional Neural Network	66
	4.2.5	Computer vision and video analysis	70
	4.2.6	How an object detection algorithm works	72
		4.2.6.1 The convolution layer	72

		4.2.6.2	Stride and Padding	74
		4.2.6.3	The Pooling Layer	75
		4.2.6.4	The output layer	77
	4.2.7	Region-based convolutional neural network		79
		4.2.7.1	Intuition of RCNN	79
		4.2.7.2	Problems with RCNN	82
	4.2.8	Fast RCNN		83
		4.2.8.1	Problems with Fast RCNN	85
	4.2.9	Faster RCNN		86
		4.2.9.1	Problems with Faster RCNN	89
		4.2.9.2	RCNN Training	91
4.3	Conclusion			92
<b>Chapter 5</b>	<b>Implementation Methodologies</b>			94
5.1	Introduction			94
5.2	Step 1: Acquisition of Data			95
5.3	Step 2: Training Data			102
5.4	Step 3: Connecting with Cloud			102
5.5	Outputs and Results			104
5.6	Visualization and Analysis of Data			105
	5.6.1	Pedestrian analysis		105
	5.6.2	Vehicle Analysis		117
5.7	Zero Inflated Poisson Distribution			126
	5.7.1	Zero Inflated Poisson Distribution Results		132
5.8	Heat Map			135
5.9	Architecture & Outputs			138
5.10	Conclusion			144
<b>Chapter 6</b>	<b>Conclusion and Future Scope</b>			145
6.1	Contributions Summary			145
6.2	Future Scope and Research Direction			146



## LIST OF FIGURES

<b>Fig. No.</b>		<b>Page. No.</b>
1.3	Flowchart for the thesis organization	6
3.1	Number of fatal accidents at intersections	28
3.2	Serious injuries at intersections	29
3.3	Pedestrian accidents at intersections	29
3.4	Bike accidents at intersections	30
3.5	Showing Highest Danger Indices at various intersections	31
3.6	Age of drivers involved in Fatal accidents	31
3.7	Age of drivers involved in serious injuries	31
3.8	Age of drivers involved in Bike accidents	32
3.9	Age of pedestrians involved in Pedestrian accidents	32
3.10	Fatal accidents during the hour of the day	33
3.11	Serious injuries during the hour of the day	33
3.12	Bike accidents during the hour of the day	34
3.13	Pedestrian accidents at the hour of the day	35
3.14	Fatal accidents during the day of the week	36
3.15	Serious injuries caused during the day of the week	36
3.16	Number of Bikes involved in accidents during the day of the week	37
3.17	Number of accidents based on pedestrians on the day of the week	39

3.18	Number of Serious Injuries by Month	39
3.19	Number of Fatal Accidents by Month	39
3.20	Number of Fatal Accidents by Month pedestrians	39
3.21	Number of monthly accidents involving	38
3.22	Number of monthly accidents involving	40
3.23	Percentage of accidents involving pedestrians and their levels of alcohol consumption	41
3.24	Distribution of the percentage of accident victims based on causes of accidents and type of victim in the year 2011 to 2015	43
3.25	Number of offences observed related to driving vehicles with respect to safety of pedestrians and cyclists	44
3.26	Distribution of percentage based on movement and type of accident from 2011 to 2015	45
3.27	Advanced Sidewalks for pedestrians	51
3.28	Visible pedestrian crossing	51
3.29 (a)	Various issues on the Montreal streets causing a threat to the safety of pedestrians and cyclists	52
3.29 (b)	Various issues on the Montreal streets causing a threat to the safety of pedestrians and cyclists	52
3.29 (c)	Various issues on the Montreal streets causing a threat to the safety of pedestrians and cyclists	53
3.29 (d)	Various issues on the Montreal streets causing a threat to the safety of pedestrians and cyclists	54
3.29 (e)	Various issues on the Montreal streets causing a threat to the safety of pedestrians and cyclists	54
4.1	Layers that define a Neural Network	57
4.2	Artificial Neural	58

4.3	Rectified Linear Unit	60
4.4	Neural network, where input is provided and output (risk factor) is achieved	60
4.5	2D array	62
4.6	Steps involved in classification of the image.	66
4.7	CNN showing feature extraction	71
4.8	Output image from the convolution layer	74
4.9	Output image with Stride as 2	75
4.10	Padding	76
4.11	Padding to obtain output image of the same size as input image	76
4.12	Max pooling	77
4.13	Network showing the entire convolutional layer and working	78
4.14	Screen box signifying Region of Interest	81
4.15	Bounding box regressors	81
4.16	Final Convolutional layer giving SoftMax loss	83
4.17	Convolution and pooling output	86
4.18	RPN	88
4.19	Faster RCNN	90
4.20	RCNN Training	92
5.1	Some sample images of our data set before classification	99
5.2	Images collected from traffic cameras	101
5.3	Architecture showing the working of the proposed model	104
5.4 (a)	Neural network output images of our data set	105

5.4 (b)	Neural network output images of our data set	105
5.4 (c)	Neural network output images of our data set	106
5.5	Graph showing average number of pedestrians vs hour of the day for 32e Avenue and rue Provost intersection	107
5.6	Graph showing average number of pedestrians vs hour of the day for Boulevard Robert-Bourassa and Rue Sherbrooke intersection	108
5.7	Graph showing average number of pedestrians vs hour of the day for Rue Guy and Rue St. Catherine intersection	109
5.8	Graph showing average number of pedestrian vs hour of the day for Boulevard de Maisonneuve and Rue Guy intersection	110
5.9	Variation of average pedestrian count vs hour of the day	113
5.10	Pedestrian count for each intersection calculated over the entire data set	117
5.11	Variation of vehicle density with hour of day	119
5.12	Vehicle count for each intersection calculated over the entire data set	127
5.13		
5.14	Zero Inflated Poisson distribution Results	133
5.15	Heat Map showing low and high pedestrian counts as color codes	137
5.16	Circuit with the microcontroller, LED, Male to female jumper wires, 3.3V resistor and a breadboard	138
5.17	Prototype Architecture	139
5.18	Esplorer IDE	141
5.19	Cloud MQTT instance receiving messages on different intersection topics	142
6.1	Future scope of the the proposed system	146

## LIST OF TABLETES

<b>Fig. No.</b>		<b>Page. No.</b>
2.1	ESP8266 pinout	13
3.1	Summary of Literature Review regarding Pedestrian Safety	23
3.2	Number of pedestrian victims in each age group based on type of injury	46
3.3	Percentages of accidents involving pedestrians in different regions of Quebec	49
4.1	Comparing the characteristics of CNN, RCNN, Fast RCNN, and Faster RCNN	91
5.1	Names of dangerous intersections in Montreal along with their locations for this study	96
5.2	Information that can be deduced from the images	100

## List of Abbreviations and Symbols

CNN	Convolutional Neural Network
ACE	Automatic Color Enhancement
MLE	Maximum likelihood estimation
JAAD	Joint Attention for Autonomous Driving
MQTT	Mosquito Telemetry Transport
AWS	Amazon Web Services
API	Application Platform Interface
LED	Light Emitting Diode
IoT	Internet of Things
V2P	Vehicle-to-pedestrian
HOG	Histogram of gradients
Node MCU	Node Microcontroller Unit
LTE	Long Term Evolution
4G	Fourth Generation
ROI	Region of interest
Wi-Fi	Wireless Fidelity
ReLU	Rectified Linear Unit
IoU	Intersection Over Union

# Chapter 1

## Introduction

With the increase in population and ongoing construction in Montreal, controlling the traffic and ensuring pedestrian safety has become a challenge. The current traffic systems in place are unequipped to deal with the sudden increase in the number of pedestrians and vehicles on the road. Moreover, in recent years, the number of accidents involving pedestrians has also risen, thus creating a dire need for a system to enhance pedestrian safety.

At present, the city of Montreal has 2,300 intersections that have traffic lights, out of these 315 consist of pedestrian signals that have hand and pedestrian icons indicating to the people when to cross and also providing a countdown of the seconds left to cross the street. In addition to this, currently, the only mode of pedestrian safety in the city is calling '911' during emergencies or fixing a larger time frame on pedestrian signals in crowded intersections. As such, there is no dynamic system in place which can ensure pedestrians' safety, or which can automatically adjust based on pedestrian densities during rush hours. With fluctuating densities and increase in the number of accidents, there is a dire need to lay down a system that can aid in preventing accidents and ensuring the safety of pedestrians.

About 200,000 images were collected across 43 intersections in the city of Montreal from the Traffic cameras – Ville de Montreal website. The data was collected from March 8, 2020, up until March 22, 2020 and then from May 1st, 2020 to 11th May 2020. An object detection and classification model using Faster RCNN algorithm to identify pedestrian and vehicles at the

intersection was implemented. Further, this model was used to obtain a dataset showing the number of pedestrians and vehicles at the intersections. The information obtained from this data set was used for visualization and in-depth analysis of the pedestrian and vehicle data in order to derive patterns of peak and non-peak hours and high-risk intersections.

Furthermore, zero inflation poisson distribution model was implemented on our dataset to display the timings and intersections which had zero pedestrian count for long hours of the day as compared to the vehicle count. A heat map was generated to visualize the dataset and to assist data viewers to identify which areas should get most attention.

Finally, we created a prototype solution that mimicked the traffic control system by utilizing LEDs and microcontrollers (IoT device), cloud services, publish/subscribe model, and object detection. To implement this prototype, the data obtained through the object detection model was sent onto the cloud (Cloud MQTT), from where it was used to control the programmed microcontrollers (IoT devices) present at the different intersections based on the vehicle and pedestrian counts. The system managed to show excellent accuracy for detection of vehicles and pedestrians on the dataset, and the delay experienced in controlling the microcontroller was also negligible, thus making our system effective and reliable.

## **1.1 Research Objective**

The future of smart cities is discussed often but without any development by implementing solutions. Various smart city initiatives and proposed models need to be explored to create a viewpoint as to how such initiatives can match up to the framework criteria [76] (Chourabi, et al,



2012). An effective implementation of smart cities would involve a smart and efficient traffic management system, which can help in reducing accidents and streamlining the traffic flow.

The objective of this research is to build a system for enhancing pedestrian safety as well as improving the current traffic management system. We collected and processed image data for 43 core intersections of Montreal to study and analyze the pedestrian and vehicle counts. The aim was to utilize this data to build an efficient cloud-based solution that can enhance the traffic management in these intersections.

## 1.2 Contributions

The contributions of this thesis are as follows:

- **Collecting, Processing, and visualizing data:** The first step towards the implementation of the Pedestrian Safety Enhancement system involved collecting relevant data in order to analyze the situation of pedestrians in the city. To do this, we first collected a total of 200,000 images for 43 different intersections in the city of Montreal. These images were first preprocessed by removing blurry, unavailable, and distorted images, after which our trained object detection model was applied onto it to detect pedestrians and vehicles. Our trained object detection model involved using the Faster RCNN algorithm for detection and classification of pedestrians and vehicles. This information obtained by applying this object detection model was stored for further analysis. It was then thoroughly analyzed and visualized in order to better understand the threats posed to pedestrian safety as well as to broaden our knowledge of the problem statement. The main contribution of this step was to obtain the relevant dataset, process it, analyze it, and visualize it to highlight the problem areas.

- **Utilizing Cloud Services for Implementing the Pedestrian Safety Enhancement**

**System:** The next step involved sending the data obtained in the previous step onto the cloud and using it to create a prototype. This involved sending the visualized data onto the cloud and using this cloud platform to control microcontrollers with attached LEDs present at different intersections using the Cloud MQTT Publish/Subscribe protocol. This depicted how this system can be further extended on a larger scale to control the Traffic Lights system in a real-time and efficient manner.

### **1.3 Organization of Thesis**

This thesis has been organized in six chapters. In chapter 1, introduction and need of the current research with its objectives are discussed. Chapter 2 covers the detailed literature review and case studies related to the work being proposed in this thesis in order to analyze and determine how this study compares with the existing solutions. Chapter 3 provides the methodology used for proposed system as well as its implementation. Chapter 4 and Chapter 5 present working with object detection and implementation of methodologies. The results of implementation of present study are also well depicted. Chapter 6 presents discussions of analysis carried out, conclusions and scope of future study.

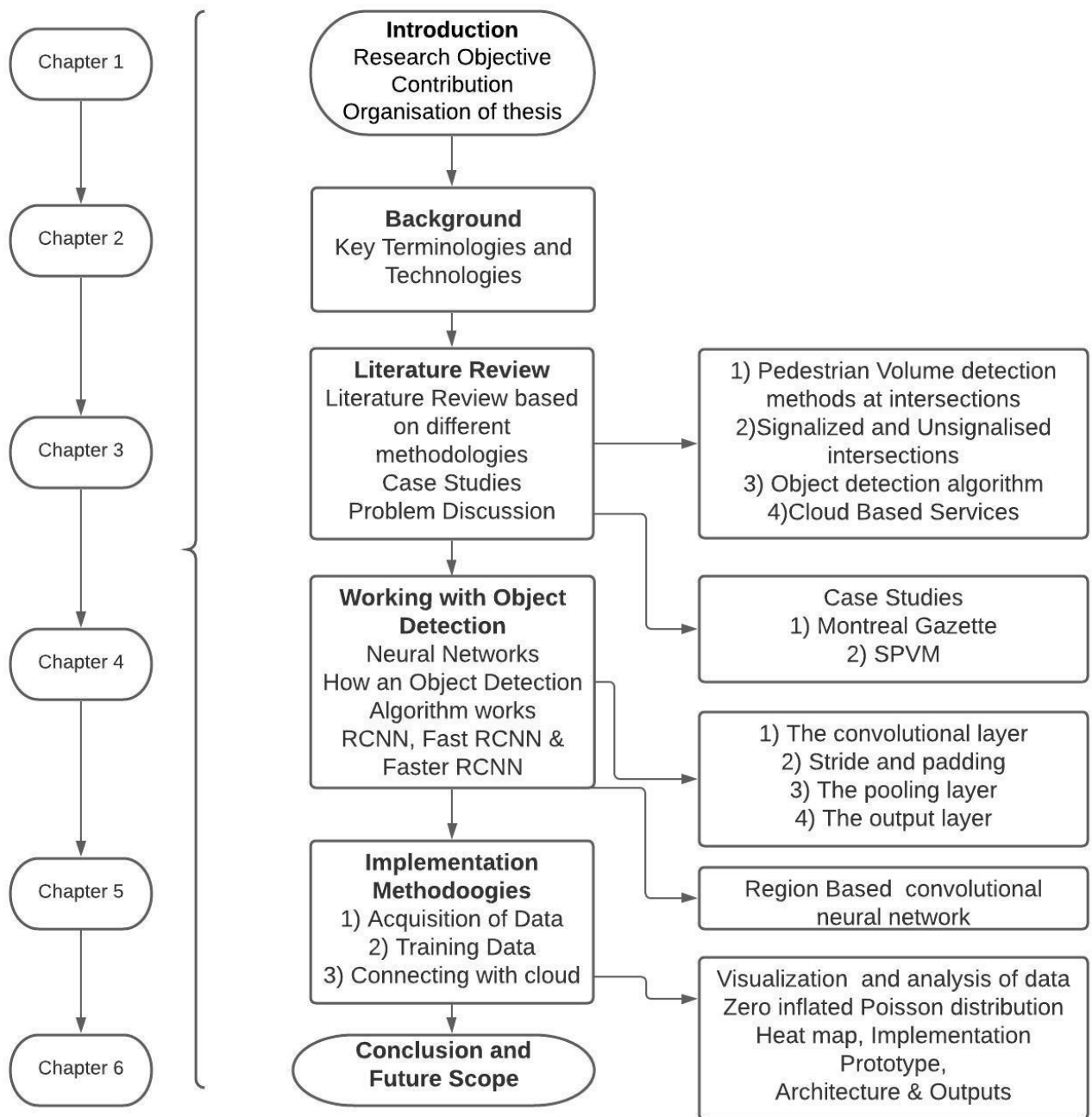


Fig.1.1. Flowchart for the thesis organization

# Chapter 2

## Background

### 2.1. Introduction

This chapter focusses on describing the technologies, libraries, and tools used while developing this thesis. First, the main python libraries used for neural network training purposes are described. This is followed by detailed explanation of the cloud platforms used in the prototype implementation, hardware sensors and development boards, Integrated Development Environments (IDEs), and language used while developing the proposed system.

### 2.2 Key Terminologies and Technologies

Some key technologies and terminologies essential to the work proposed in this chapter are defined below.

#### 2.2.1 Internet of Things

The use of Internet of Things in this research is based on interaction of data with devices. Internet of things is a concept for processing and utilizing data based on the ability of network devices that sense and collect data which is fed to them. It does not require human-to-human or human-to-computer interaction. There are several keywords that are used in the concept of Internet of Things such as Network Devices, Sensors with ability to collect data, Actuators, Internet, Embedded computing devices and processing and utilizing data which is received from sensors.

The various components involved in Internet of Things are Sensors which are used to collect data from the environment. Examples of some sensors can be Motion Sensors, Temperature and Humidity etc. Actuator, which is a component of a machine, is responsible for moving or controlling a mechanism, for example an LED. Embedded computing devices have the ability to read data from sensors, ability to receive and send instruction to actuators, for example if motion is detected switch on the LED. Next is networks, to provide connectivity, IoT devices are tied together by networks using various wireless technologies, standards and protocols. All of these devices need not be connected to the internet independently. Rather a cluster of devices could be created and the base station or the cluster head could be connected to internet. There are several languages that are currently used in Internet of things, for our research Lua Script is used. There are several cloud based IoT services and solutions available. There is ThingSpeak, CloudMQTT, ThingWorx, IBM Watson, AWS IoT (this is by Amazon web services), Azure IoT (this is by Microsoft), PubNub and many more. In this research some of them will be used.

### **2.2.2 Neural Networks**

A neural network is a way of machine learning that models itself like the human brain, it involves creating an algorithm allowing the computer to learn by incorporating data. Just like the human brain has a basic unit called neuron, the essential building block of an artificially intelligent neural network is called a perceptron. Just like a child's brain, algorithms need information to learn. Hence, unlike other algorithms, deep learning algorithms cannot be programmed directly for the task. There are three learning strategies for neural networks namely supervised, unsupervised and reinforced learning.

Supervised learning is the simplest as it contains labelled dataset. The computer goes through it and processes the algorithm until the desired result is achieved.

Unsupervised learning does not have any labelled dataset to learn from. In this, the neural network analyzes the dataset, a cost function alerts the neural network as to how far the target is from the expectation. Thus, it adjusts accordingly to increase the accuracy of the applied algorithm.

Reinforced learning is an algorithm where if the algorithm is not giving accurate results it is reinforced till it learns to give positive results.

### **2.2.3 Cloud MQTT**

MQTT stands for Message Queuing Telemetry Transport. MQTT is a protocol for M2M (Machine to Machine) and IoT applications. This is based on Publish/Subscribe model. There can be multiple clients that can be created eg: Client1, Client2... (eg: ESP8266) (Unique Id). MQTT is defined as machine to machine protocol and client could be any machine. For instance, an ESP8266 development board with a motion sensor could be a MQTT client and a Raspberry Pie with temperature sensor could be another client, each client must have a unique ID. Then there is a publisher, a subscriber, topic name, and a message, which will be sent and received on the topic. Then we have the MQTT Broker which lets thousands of topics run concurrently. The features of MQTT are as follows: It Pushes messages instead of Pull messages, it is reliable even when used with unreliable networks, ideal for constraint networks (Low bandwidth, high latency etc.), very simple and easy to implement and has 3 QoS levels. QoS 0 means at most one delivery of message, QoS 1 means at least once delivery of message, QoS 2 means exactly once delivery of message. Cloud MQTT server is a free service provided by [cloudmqtt.com](http://cloudmqtt.com).

## 2.3 Tensor Flow

One of the most important concept of deep learning is object detection. API or Application Programming Interface provides a set of operations that save time. TensorFlow, an end-to-end open source platform for machine learning created by google, is a python library used for faster numerical computations. Deployment of computations are easily allowed as per its architecture. Model Zoo is a framework that already exists with the pre trained models which include COCO dataset, KITTI dataset and the Open images dataset. Tensors are a de facto for representing the data in deep learning. Thus, TensorFlow is a library based on Python that provides different types of functionality for the implementation of deep learning models. The term Tensor represents a data of multi-dimensional array, whereas flow represents a series of operations performed on the tensors. ConvNet produces a feature map of image based on the input received about the image. The estimators of tensor flow are fully supported in TensorFlow and can be created using tf.keras models. An input function using the Datasets API to scale, or train large dataset is conducted. Estimators need control as to when and how the input pipeline is built, input\_fn is used as an estimator to call the function with no arguments and returns tf.data. dataset. Thus, the tf.keras. model can be trained with tf.estimator API. In the next step the Region proposed network is applied to these feature mas, the ConvNet thus returns the proposals of the object along the object score. After this step the RoI pooling layer is also applied to these proposals to produce a small feature map of fixed size. The proposals then are passed onto a fully connected layer that includes the softmax layer and a linear regression layer. This process classifies and outputs the bounding boxes for objects.

## **2.4 TensorFlow object detection API**

TensorFlow object detection API is basically a framework which is used for pretraining models on the go like YOLO, SSD, RCNN and Faster-RCNN etc. A TensorFlow Object Detection API environment was set up. This data was converted into TF Record file format that is the Pascal Voc format dataset to Tensor Flow record format. A record file is created and then the Training is started. A faster R-CNN pre-trained model from Tensor Flow detection model is extracted to the model folder and the pedestrians and vehicles are identified.

## **2.5 NumPy**

It is an open source numerical python library that consists of multi-dimensional array and matrix data structures. Besides this, the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays. It is an extension of Numeric and Numarray and is a wrapper around a library that is implemented in C. Its interoperability supports a wide range of hardware and computing platforms, and plays well with distributed, GPU and Sparse array libraries. NumPy arrays have a performance feature called vectorization. It is fast as it replaces the loop with an operation that runs on items parallelly.

## **2.6 Pandas**

Developed by Wes McKinney, a software library that is written for python programming language, Pandas is used for data manipulation and analysis. Data structures and operations for the manipulation of numerical tables and time series are specialty of pandas. It is built on NumPy package and provides two important data structures namely series and dataframe. DataFrames allow storing and manipulation of tabular data in rows of observations and columns of variables.



## 2.7 Seaborn

A python data visualization library, seaborn provides high-level interface for drawing attractive and informative statistical graphics. Matplotlib is also used for better insights of our data and works with different shapes and configurations. The seaborn package was developed based on the Matplotlib library and is used to create more attractive and informative statistical graphics.

## 2.8 MQTT (publish/subscribe, client/broker)

A light weight, open, simple and easy to implement protocol, MQTT is a publish/subscribe messaging protocol. From a Machine to Machine (M2M) and Internet of Things (IoT) context this protocol has been essential for communication in such constrained environments. Invented in 1999 by Andy Stanford-Clark (IBM) and Arlen Nipper (Arcom) for connecting oil pipelines via satellite, a need of a protocol with minimal battery loss and minimal bandwidth was required. This protocol serves various features like simpler to implement, Quality of service data delivery, lightweight and bandwidth efficient, data agnostic and continuous session awareness. Here, “MQ” refers to the MQ series which is an IBM product developed to support MQ telemetry transport.

The pub/sub or Publish/Subscribe pattern provides an alternative to a client/server architecture that was used traditionally. In a publish/subscribe model, there is no connection between the publisher and the subscriber, the pub/sub model decouples the client that sends a message, also called the publisher, from the clients that receive the message, also called Subscribers. There is no direct interaction between the publisher and subscriber, this connection is handled by the broker, which is the third component. The broker filters the incoming messages and distributes them

correctly to subscribers. Decoupling of the publisher from the subscriber is the most important aspect of pub/sub. A few decoupling dimensions are:

**Space decoupling:** Where the publisher and subscriber do not need to know each other, or there is no exchange of IP address and port. Time decoupling in which publisher and subscriber don't run at the same time, Synchronization decoupling in which operations on both components need not interrupt during publishing or receiving. Hence, direct communication is removed by the pub/sub model. The decision as to which client/subscriber will be controlled is made possible by the filtering activity of the broker.

## **2.9 Labelling**

Annotation of the vital information present in our system is necessary for our model to identify the exact coordinates it should look for. Labelling is written in python programming language and uses the QT library for its graphical interface, making the annotation easier by recognizing the pedestrians and vehicles at the intersection.

## **2.10 Computer Vision**

Computer vision is the study of replicating the human vision processes, passing them on to machines and automating them. It is a field of artificial intelligence that teaches computers to evaluate the visual world. It is the process of extracting, understanding and learning from a single image. It is a process of a machine to receive and do the analysis of visual data, the aim is to enable a computer to perform tasks with the accuracy as close as humans. It is thus closely related to machine learning and image processing. The image can take in many forms, such as video

sequences, depth images, view from multiple cameras, or multi-dimensional data from a medical scanner. It allows to interpret and understand the visual world in a better way.

## 2.11 ESP8266

The ESP8266 is an independent system on chip (SOC) wifi module. It is incorporated with TCP/IP protocol stack allowing it to connect to any wifi network. It was first manufactured by the Chinese company called Espressif. It consists of a 32-bit microcontroller unit (MCU), 11 GPIO pins (General Purpose Input/Output pins) and an analog input as well. It can be programmed like Arduino or any other microcontroller. There is an official SDK available to program it in C, it is used to run LUA scripts on it, which is used in this thesis as well. The ESP8266 has 1MiB of built-in-flash that allows the device to be capable of connecting to Wi-Fi.

The pinout of ESP8266 is shown in table 2.1 as:

1. VCC, Voltage (+3.3V and can handle up to 3.6V)	5. CH_PD, Chip power-down
2. GND, Ground (0V)	6. RST, Reset
3. RX, Receive data bit X	7. GPIO 0, General-purpose input/output No. 0
4. TX, Transmit data bit X	8. GPIO 2, General-purpose input/output No.2

**Table 2.1. ESP8266 pinout**

The VCC and GND are the pins to power, RX and TX are used for communication.

## 2.12 NodeMCU firmware

The term NodeMCU combines the “node” and the micro-controller unit. This firmware uses the Lua scripting language, based on the eLua project, it is built on Espressif Non-OS SDK for ESP8266. The NodeMcu is a development board with an ESP8266 and shares the same name as the firmware. A NodeMCU development board is with an esp-12F esp8266 which is soldered to the OCB of the NodeMCU. The boot configuration pull-up and pull-down and enable pin pull-up

is added by the required circuits by the development board across the module. A USB chip to connect the ESP8266 to USB and an auto reset circuit to enable the upload tool into flashing the ESP8266. The 3.3V power of the ESP8266 is converted from 5V of USB. NodeMcu firmware is a Lua language interpreter.

## **2.13 ESPlorer IDE**

ESPlorer IDE is an IDE for ESP development, supported by platforms like Windows (x86, x86-64) , Linux(x86, x86-64, ARM soft & hard float), Solaris(x86, x86-64), Mac OS X(x86, x86-64, PPC, PPC6). The purpose of this software is to establish a serial communication with your ESP8266, send commands and control your applications. It is a Lua based IDE which is used to develop applications for the NodeMCU. The ESPlorer IDE is an IDE for ESP developers. To run this Java needs to be installed in the system.

## **2.14 Lua**

Designed and implemented by a team at PUC-Rio, a Pontifical Catholic University of Rio de Janeiro in Brazil, it is a powerful, efficient, lightweight, scripting language. The NodeMCU is programmed with the Lua language. It is an open source, lightweight and an embeddable scripting language built on top of C language. It is robust and a free open source software distributed under the MIT license.

## **2.15 Conclusion**

This chapter played a crucial role in getting familiar with the tools and technologies to be utilized in implementing the solution as well as getting accustomed to the basic concepts. This step was indeed essential before reviewing other literature in order to better understand the solutions currently available to tackle the problem statement and think of innovative solutions that could be implemented using these latest tools and technologies. Thus, in this chapter we explored key terms with respect to libraries, cloud platforms, hardware and sensors, IDEs, and languages currently available to develop novel solutions with regards to the problem at hand, i.e. risks to pedestrian safety.

# Chapter 3

## Literature Review

### 3.1. Introduction

This chapter provides a critical appraisal of literature and case studies relevant to the objective of current research. Several research works have already been done about pedestrian safety. In this chapter a comparison has been made with past research and current research to bridge the gap for optimum pedestrian safety.

A review of the literature about the exploration about researches and methodologies that aim to identify and visualize pedestrians' volumes and risks associated with them at intersections were studied. Further various studies about object detection algorithms used for calculations of the above-mentioned parameters at signalized and unsignalized as well as controlled and uncontrolled intersections were studied. Also included are studies about researches which involved approaches to enhance pedestrian safety at intersections using cloud-based solutions. Subsequently, we compare these methodologies based on what technologies have proven to be efficient in pedestrian detection to count them at intersections, and lastly, we explore how automated streetlights have enhanced pedestrian safety. This detailed literature review is then followed by an analysis of the two major case studies conducted in Montreal on collecting data with regards to pedestrian safety. The first case study by Montreal Gazette provides data on the most dangerous intersections in the city of Montreal, while the second study by SPVM (Service de Police de la Ville Montreal) provides data obtained from accidents recorded in the Police database, which we use to analyze pedestrian safety and risks posed to it. The final section of the chapter contains a brief problem discussion

highlighting the current issues that pose a threat to pedestrians in Montreal, as well as the current shortcomings of the road safety system that can not only decrease pedestrian safety but can also affect the overall functioning of the traffic system.

### **3.1.1. Pedestrian Volume by Identification at Intersections**

The aim of [65,b] is to investigate the accuracy of three common counting methods like manual counting using sheets, manual count using clickers and manual count using video cameras, the analysis of all the three methods were done at the same time. The results indicate that manual counts with either sheets or clickers systematically underestimated pedestrian volumes. The error rate grew exponentially due to lack of familiarity with the tasks or fatigue. The most significant results of this study were that pedestrian counts taken in the fields were lower than counts taken by observing video recordings.

[65,a] comprises a sample of fifty intersections, weekly pedestrian volumes with surrounding land uses, transportation attributes and neighbourhood socioeconomic characteristics. Ordinary least squares (OLS) regression was used to develop the pedestrian crossing volume model. The model can be used to develop rough estimates of pedestrian intersection crossing volumes. Improved pedestrian volume estimates will enable planners, designers, engineers, public health professionals, and others to improve the safety and convenience of pedestrian transportation. It is demonstrated pedestrian volumes can be routinely integrated into transportation safety and planning projects. The research shows how different factors are taken for different lengths of time, at different times of the day, in different locations, under varied weather conditions as these adjustment factors allow to extrapolate short counts to a full day, week, month or year. Field data collectors took manual counts of locations and for some intersections infrared sensors were rotated

to capture variations in pedestrian volume patterns due to time of day, location and weather conditions.

### **3.1.2. Signalized and Unsignalized intersections**

[35] presents a model named potential pedestrian risky event for analysis where two CCTV camera footage were used. The system calculates and extracts frame-level behavioural features by detecting pedestrians and vehicles. Classification of the risky events into six clusters was done by using k-means clustering and decision tree algorithms. Video footage from unsignalized crosswalks were used for feasibility of the model.

This research aims to create a safer mobility environment by identifying dangerous driving patterns beforehand. The research achieves this by analysing the data obtained from existing repurposed security footages. However, there are certain limitations to the information derived from such data, and thus it can be seen as the first step towards safer mobility. In [74] the author discusses that yielding rates of vehicles were 94% in the sites where red signal or beacons were used as compared to “active when present” devices and enhanced and high-visibility treatment.

Roadway warning lights provide amber flashing lights, mounted flush to the pavement surface at the crossing location. The flashing light can be activated by a push button or a passive detection technology such as microwave sensors. Thus, the study shows that midblock signals, half signals and HAWK signals had 95% of compliance rates to the proposed system.

Objective of the [30] was to diagnose pedestrian safety issues and traffic conflict techniques. Gait analysis results showed that a single pedestrian had 9% higher walking rate than those who walked in groups. Study showed that males tend to be faster than females with higher step length but lower step frequency. The study also showed how the violators of signals at intersections tend



to have higher walking speeds compared with non-violators. Computer vision and conflict indicators like TTC (time-to-collision) indicators were used for this study.

Signalized intersections [73] comprise of 40% of crashes involving pedestrians. Survival analyses statistical methodology was used to produce Kaplan-Meier survival curves for waiting time prior to crossing unsafely, separately for males and females, and for each intersection. The mean observed waiting time for about 90% pedestrian showed that waiting time of females before crossing the road was 44% more than males. The waiting time varies with the probability of a pedestrian to cross the road when it is unsafe i.e motor vehicles still have green or yellow. With the increase in signal waiting time pedestrians get impatient and cross the roads violating the traffic signal according to the study, thus increasing the risk of getting being struck by a motor vehicle. The study concludes that reducing the waiting time will decrease the probability of pedestrian crossers being hit by a motor vehicle.

Like in medical sciences survival analysis is used to study the effectiveness of different drugs on cancer patients, similarly this work used survival analysis to estimate how the risk that occurs to pedestrians varies with time.

[43] suggests two classes for illegal crossing behaviours at signalized intersections namely “crossing at a red light” and “crossing outside a crosswalk” were formed and two datasets using video based observation and survey based observation was obtained respectively. Further two Bayesian network-based behaviour models were developed to investigate the characteristics and their impacts on the two data sets. Waiting time, location of traffic attractions has a significant influence on crossing outside of a crosswalk.

### 3.1.3. Object Detection algorithm

[59] suggests a modified faster R-CNN method with automatic color enhancement (ACE) that can improve sample contrast by calculating the relative light and dark relationship to correct the final pixel value. A calibration method based on sample categories reduction is presented to accurately locate the target, which is the pedestrian for detection, after which the R-CNN target detection was applied on the experimental dataset. Improving accuracy for pedestrian detection by 3.2%. The Faster R-CNN has an average miss rate of 12.7% compared to Faster-CNN + ACE at the subway.

In Estimation of intersection traffic density on decentralized architectures with deep networks - IEEE Conference Publication. (2020), an approach for chaotic traffic scenarios by proposing a decentralised approach is mentioned using image processing and deep learning which was found to be effective in managing a large traffic network in real-time. Implementing video -based ROI creation and deep networks for detection allows this method to generalize across deployments with minimal reconfiguration. To help taking traffic signal cycle timings at the intersection NVIDIA Jetson TX2 edge computing device was installed at each intersection.

In [55] discussion about RetinaNet based pedestrian detection and action recognition, where a recurrent neural network to estimate the time to cross the street for multiple pedestrians was the focus. Four pedestrian action labels namely (PPC) Pedestrian is Preparing to Cross the Street, Pedestrian is crossing the street (PC), Pedestrian is about to cross the street (PAC) and Pedestrian intention is ambiguous (PA) were formed out of the (JAAD) Joint Attention for Autonomous Driving data set. In this work, two distinct approaches were used. The first approach simply tagged the samples as pedestrian or not pedestrian, whereas the second approach involved using multiple

such tags. In terms of performance measurements, even though the first approach performed better, the second approach is more realistic since it incorporates not just pedestrian/not pedestrian tags, but multiple other tags capturing pedestrian actions.

[83] focusses on an algorithm for object-detection and tracking by utilizing the algorithm for moving-object detection and tracking with the given sequence of laser scan data of an intersection. The goal is to detect all moving objects entering the intersection, estimate state parameters and track motion trajectories. Laser scanning is used to accomplish this goal. Reliability items can be associated with feature parameters to discriminate direct observations from occluded features. Detection algorithm was used on grouping measurements of moving object.

In [85] an Unmanned aerial vehicle-based method is proposed in this research. It is used for tracking and simulating pedestrian movements at intersections. It involves high resolution videos acquired by UAV using CNN for recognizing and positioning moving targets, including pedestrians, cyclists and vehicles. A social force model for high-precision pedestrian trajectories at complex targets was the goal of the research. However, during flight the UAV is subject to drift reducing accuracy of the trajectory data. The research suggests that identifying pedestrian density can be beneficial for future search of pedestrian safety.

#### **3.1.4. Cloud Based Services**

In [9] the focus is on developing of wireless-based vehicle-to-pedestrian (V2P) collision avoidance using energy efficient methods and non-dedicated existing technologies namely smartphones, cellular networks and cloud. This road safety app can be set to driver or pedestrian mode and can frequently send vehicle and pedestrian geolocation data (beacons) to cloud servers. Cloud performs threat analysis and sends alerts to users who are in risky situation. Constant

pedestrian to cloud (P2C) beaconing can quickly drain smartphone battery making the system impractical. Adaptive multi-mode (AMM) built on situation-adaptive beaconing was employed as it reduces the power consumption of beacon rate control, keeping data freshness required for timely vehicle-to-pedestrian collision prediction and also commands the mobile apps to change P2C beaconing frequency according to collision risk level from the surrounding vehicular traffic.

Like our research, this study explains that in comparison to ad-hoc technologies (e.g. WiFi, IEEE 802.11p) using cellular technology (e.g.: 3G, LTE) together with smartphones is a better fit for pedestrian safety applications in terms of reducing user adoption costs and market penetration time.

[1] proposes a Pedestrian – Safe Smart Crosswalk System based on IoT using a CCTV with object tracking to reduce and save lives from accident. The system detects the accidents and rescues casualties from severe injury. Sensors and floodlights were used to light up the pedestrian, that could allow the vehicle driver to easily detect using sensors and camera. The research shows that considering the average stopping distance, a collision force can be reduced or become zero by early discovery of a vehicle driver. The psychological effects that are expected are that the driver tries not to speed up near the crosswalk area because of the surveillance camera.

In [25], a fuzzy-based approach is used to allow dynamic control of the green traffic lights, allowing pedestrian crossing times to be increased or decreased based on pedestrian accumulation. The paper provides an in-depth exploration of this scenario by making use of a simulation technique. The FLC configuration and application scenario is also analyzed thoroughly. Table 3.1 summarizes literature review carried out for present study.

<b>Authors</b>	<b>Paper Topic</b>	<b>Category</b>	<b>Model</b>	<b>Approach</b>
[55]	Multi-Task Deep Learning for Pedestrian Detection, Action Recognition and Time to Cross Prediction	Object Detection	Object Detection Recurrent Neural Network	Tags
[50]	Vision-Based Potential Pedestrian Risk Analysis on Unsignalized Crosswalk Using Data Mining Techniques	Unsignalized crosswalks	k-means clustering and decision tree algorithms	Bayesian network
[85]	Tracking and Simulating Pedestrian Movements at Intersections Using Unmanned Aerial Vehicles. Remote Sensing	Object Detection	High res CNN	UAV
[22]	Estimation of intersection traffic density on decentralized architectures with deep networks	Object Detection	Edge computing	Decentralized
[65, a]	Pilot Model for Estimating Pedestrian Intersection	Pedestrian Volume	OLS	Data collection Crossing Volumes
[74]	Motorist Yielding to Pedestrians at Unsignalized Intersections: Findings from a National Study on Improving Pedestrian Safety	Signalized intersection	Bayesian Network	Beacons

[73]	Survival analysis: Pedestrian risk exposure at signalized intersections	Signalized intersection	Kaplan-Meier	Survival analysis
[43]	Analysis on Illegal Crossing Behavior of Pedestrians at Signalized Intersections Based on Bayesian Network	Signalized Intersection	Bayesian – N/w	Survey and video
[59]	A Study on Faster R- CNN-Based Subway Pedestrian Detection with ACE Enhancement	Object Detection	Faster RCNN+ ACE	Pixel value, calibration
[83]	Detection and Tracking of Moving Objects at Intersections Using a Network of Laser Scanners	Object Detection	Laser Scanner	Detecting and tracking moving object
[9]	Cloud-Based Pedestrian Road-Safety with Situation-Adaptive Energy-Efficient Communication	Cloud Services	Wireless V2P, AMM Beacons	Beacons
[35]	Pedestrian – safe smart crossing system based on Iot	IoT	CCTV	Sensors and Floodlights
[25]	Smart Pedestrian Crossing Management at Traffic Light Junctions through a Fuzzy-Based Approach	Signalized crossings	Fuzzy logic, Vissim	Simulation
[65, b]	Pedestrian Counting Methods at Intersections: A comparative study	Signalized and Unsignalized	Counting	Manual counts sing sheets, clickers and video cameras

**Table 3.1: Summary of Literature Review regarding Pedestrian Safety**

### **3.2. Case Study 1: Montreal Gazette**

As per the Service de police de la Ville Montreal (SPVM) the rules and the guidelines are publicly available for the pedestrians to follow. A pedestrian must walk in between the white lines running from one signal to the signal across. As soon as the hand flashes, pedestrian cannot leave to cross the intersection, if the pedestrians are in the middle of crossing, they must hurry up. The numeric countdown is depicted by a hand, when that hand goes solid pedestrians cannot cross if they have not started already. The green light is the correct way of crossing on a crosswalk. Several intersections in the city do not have the traffic signals, the city thus has two types of crosswalks, those at intersections and those between intersections. It has the same parallel white lines across or diagonal to the intersection, however instead of traffic signals it is controlled by a STOP sign. Not just pedestrians but the traffic signals are always highly related to the vehicles and their users. There are rules and regulations that limit breaking the rules. In every offence there is a procedure that is followed by the Transports Quebec. A speeding vehicle or skipping a red light is detected by a photo radar device also called red light camera. The following data appears as the camera takes the pictures from the device: location, date, time, license plate number, position of the vehicle, speed of the vehicle and color of the traffic light. These pictures are highly confidential and are directly sent to the Surete du Quebec and then to evidence processing center (CTP) for further processing.

If the information observed by the officer at the CTP is complete the owner of the vehicle is identified based on their vehicle number plate by the law enforcement officer. The Bureau des Infractions et Amendes (BIA) of the minister de la justice receives the offense report electronically, further the (DPCP) Directeur des pour suites criminelles et penales receives the statement and sends it to vehicle owner, the owner receiving the report must plead guilty or not guilty within 30 days.

The increase in the numbers of drivers, cyclists, and pedestrians on the streets often leads to numerous situations involving accidents or near misses. Some streets are more challenging to drive or walk on than others and are thus more dangerous and accident-prone.

Montreal Gazette has provided data for the most dangerous and accident-prone areas for 6 years (2006-2011). The data comprises of accident histories taken from police reports across all the intersections and segments of Montreal. This data was obtained from the Société d'assurance automobile du Québec, and it contains a detailed account of each accident including the location, time, date, number of victims and vehicles, number of cyclists, pedestrians and possibly the ages of drivers and those involved. However, this data only comprises of those accidents for which the police officers were at the scene to report the specific circumstances. Thus, all the accidents that take place in the city of Montreal are not covered in this data.

A lot of this obtained data was discarded from the Montreal Gazette data set. For instance, accidents involving vehicles were taken only if they resulted in casualties. Due to the difficulty in obtaining coordinates, accidents taking place on highways were discarded, and the records missing street names or other essential details were also not taken into consideration.

The Addresses Québec street directory and Montreal Intersection Database were used to standardize the street names. Groups of accidents were created for this data based on the closeness of the intersections (within a 50-meter radius), or by road segments present between the intersections within 80 meters. The accidents grouped based on road segments are represented at the segment mid-points. The accident records which did not fall into any groups are represented at their exact locations. PostGIS was used for conducting the geospatial analysis.



For each intersection or segment, the danger index was calculated using the formula shown below:

$$\frac{6V_f + 3V_s + V_m}{V}$$

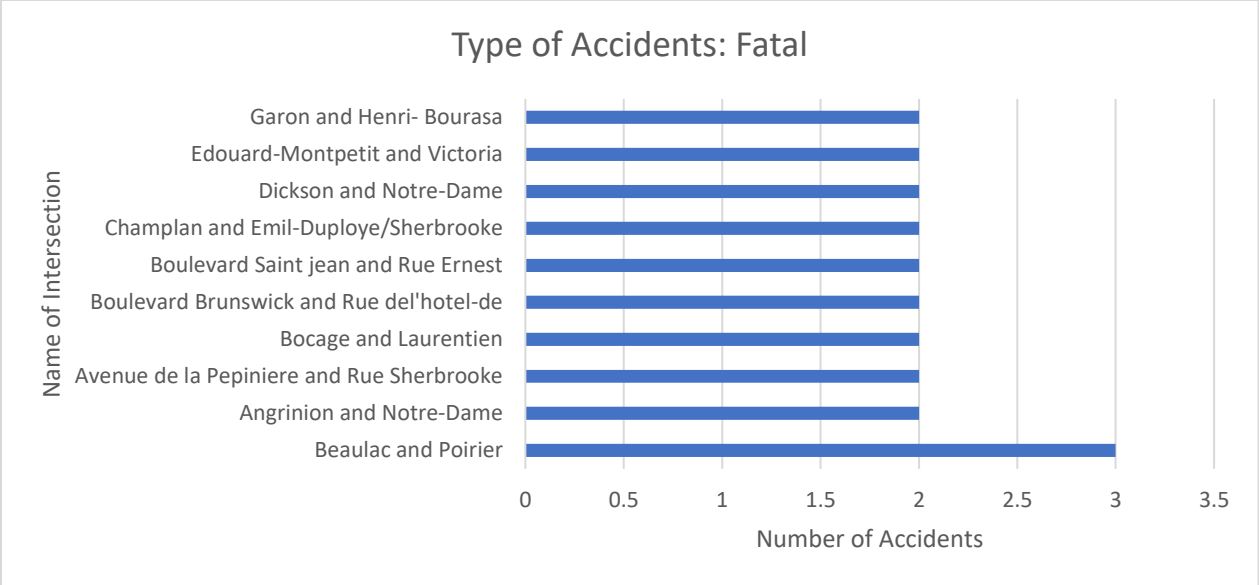
Where,

$V_f$  = Number of Killed Victims,  $V_s$  = Number of victims with serious injuries,  $V_m$  = Number of victims with minor injuries,  $V$  = Total number of victims on the intersection

The data collected over these six years depicts fatal accidents, serious injuries, accidents related to bikes and pedestrians. The motive of the study is to enable information showing intersections with the most dangerous intersections with the greatest number of accidents. These are further followed by representation of Danger index on the following intersections.

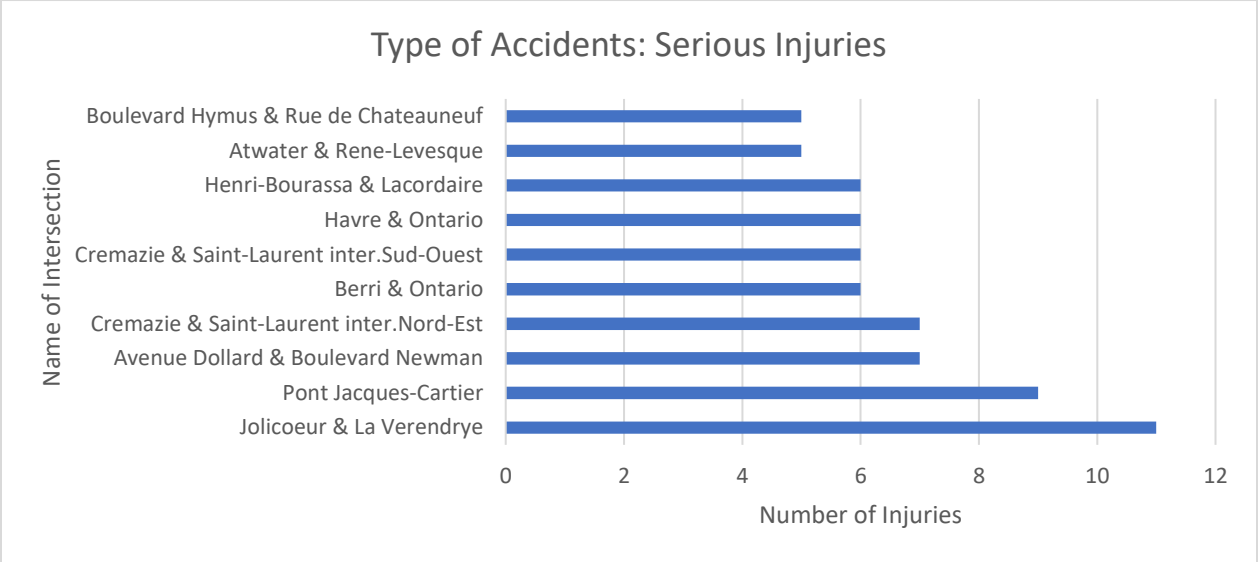
Analysis of the study based on four broad categories of accident types at different intersections:

Fatal, Serious Injuries, Bikes, Pedestrians are presented in figures 3.1 to 3.4.



**Fig 3.1. Number of fatal accidents at intersections (Courtesy of [3])**

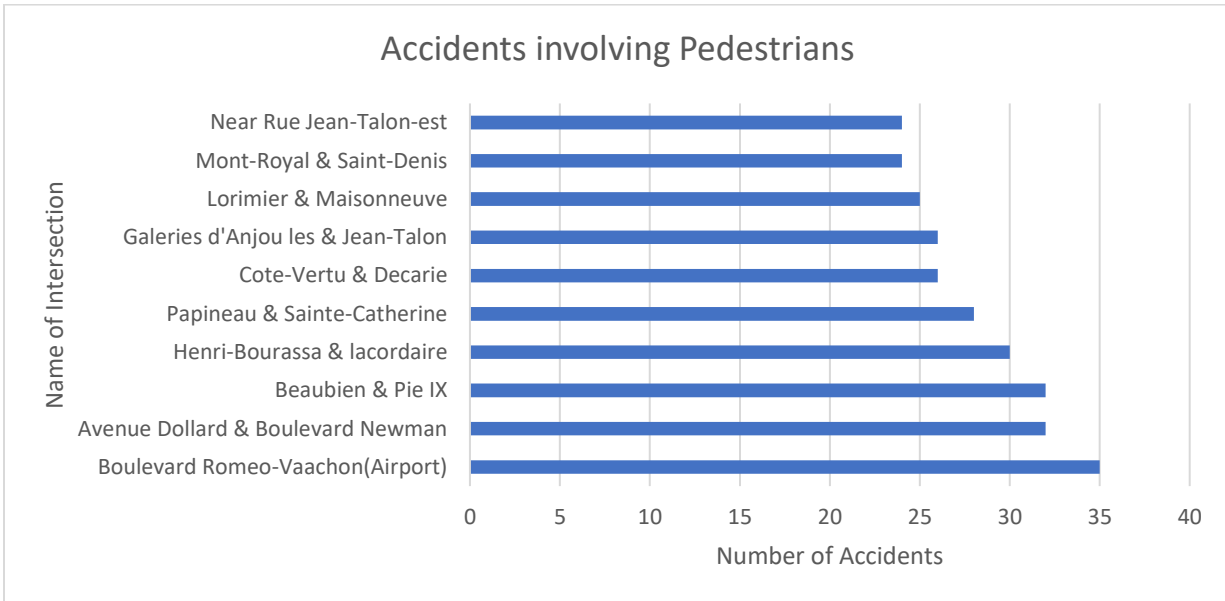
In these 6 years it can be observed that the Beaulac and Poirier intersection had the greatest number of accidents. Remarkably all the other intersection is observed to have an average of 2 accidents.



**Fig 3.2. Serious injuries at intersections (Courtesy of [3])**

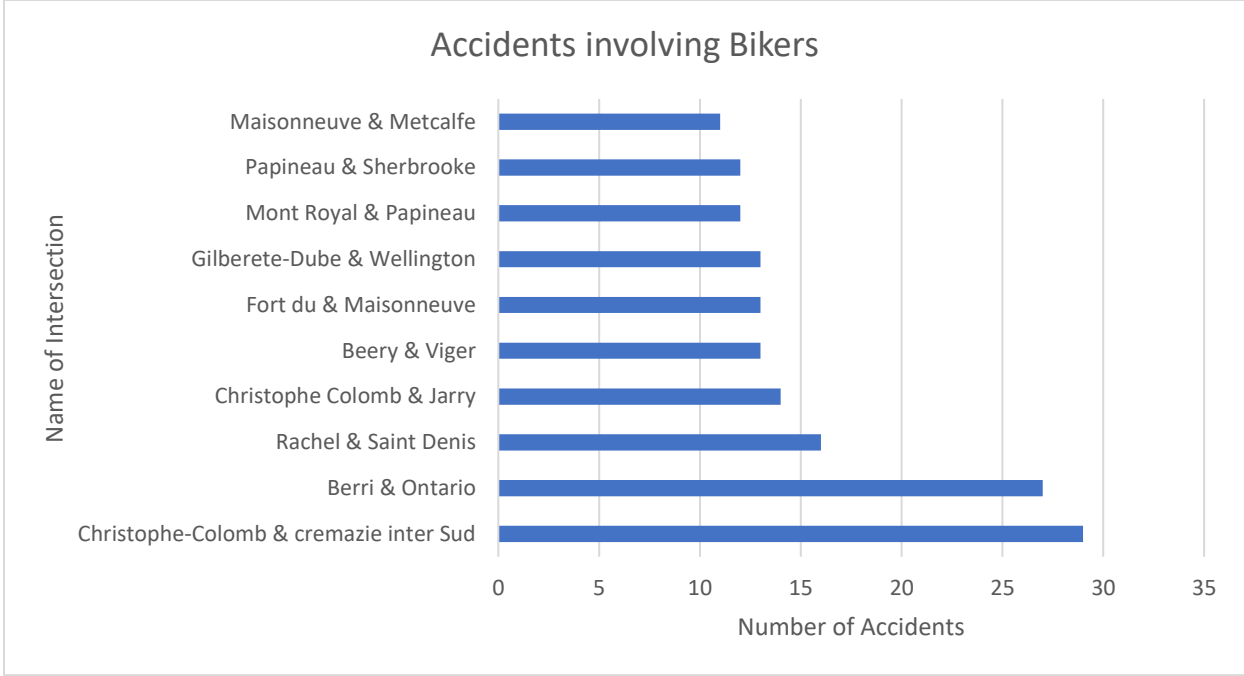
Jolicoeur and La Verendrye are observed to have the greatest number of accidents. Pont Jacques-Cartier has about 9 accidents making it the second most dangerous intersection. 6 serious injuries

were noted on Henri-Bourassa & Lacordaire, Havre & Ontario, Cremazie & Saint Laurent inter.Sud-Ouest, Berri & Ontario.



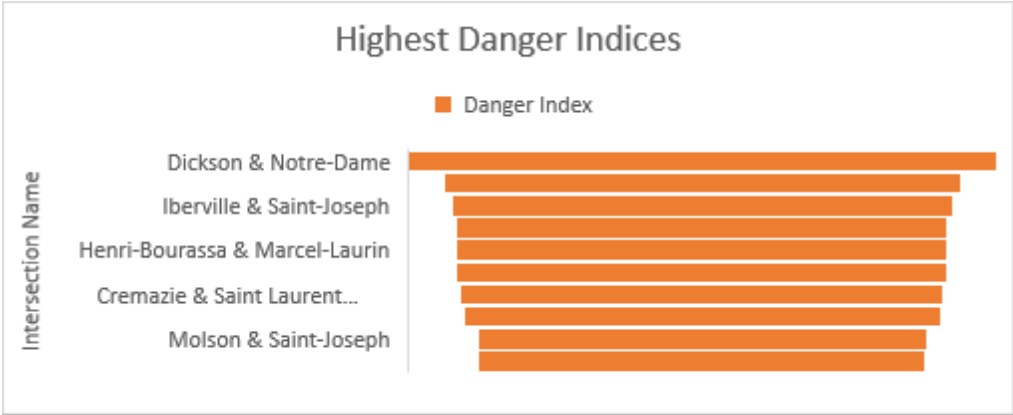
**Fig 3.3. Pedestrian accidents at intersections (Courtesy of [3])**

There were several accidents involving pedestrians over these six years. Boulevard Romeo-Vaachon (Airport) had 35 pedestrians involved in an accident. All the other intersections as shown in the fig. have accidents ranging from 23 to 33.



**Fig 3.4. Bike accidents at intersections (Courtesy of [3])**

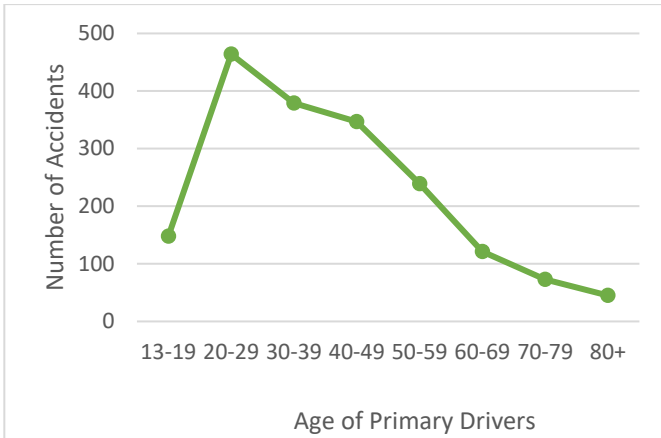
On the Christophe-Colom & Cremazie inter sud and Berri Ontario intersections more than 25 bike users were involved in accidents. This is a large number compared to all the other intersections as there were less than 15 accidents involved.



**Fig 3.5. Showing Highest Danger Indices at various intersections (Courtesy of [3])**

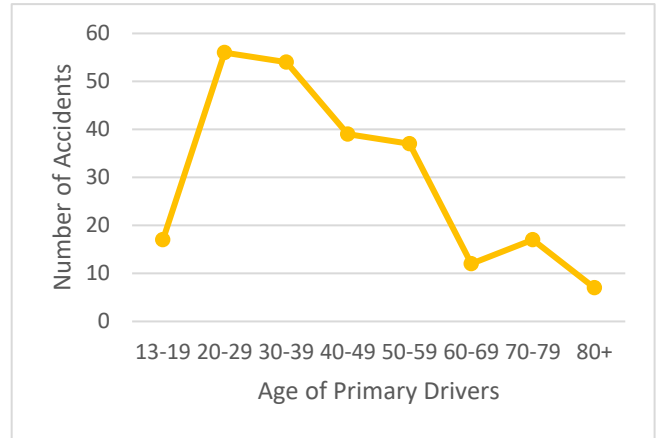
On observing the highest Danger indices, Dickson & Notre-Dame, as shown in figure 3.5, were found to be the riskiest intersections. Followed by Iberville & Saint-Joseph having 2.56 danger

index. Garon & Henri-Bourassa, Henri Bourassa & Marcel-Laurin, Notre Dame & Pie-IX have 2.5 danger index each. All other intersections like Cremazie & Saint laurentinter. Nord-Est, Rene-Levesque & Sanguinet, Molson & Saint-Joseph, Pont-Jacques-Cartier have less than 2.5.



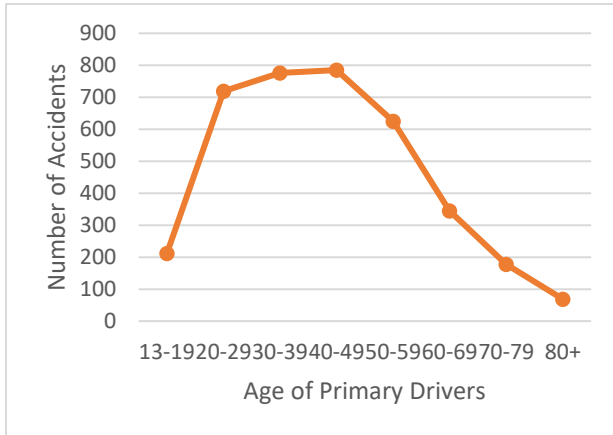
**Fig 3.6. Age of drivers involved in Fatal accidents**

(Courtesy of [3])



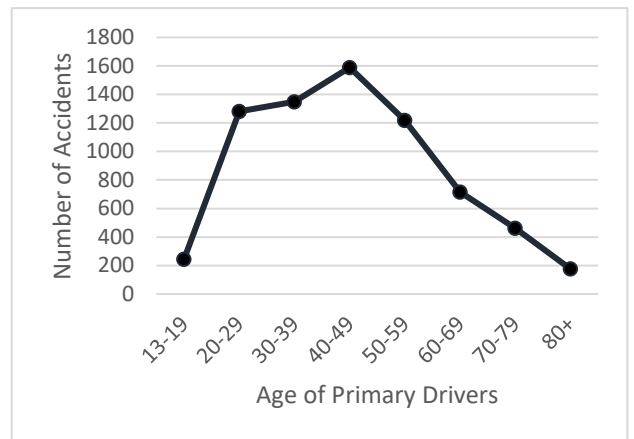
**Fig 3.7. Age of drivers involved in serious injuries**

(Courtesy of [3])



**Fig 3.8. Age of drivers involved in Bike accidents**

(Courtesy of [3])



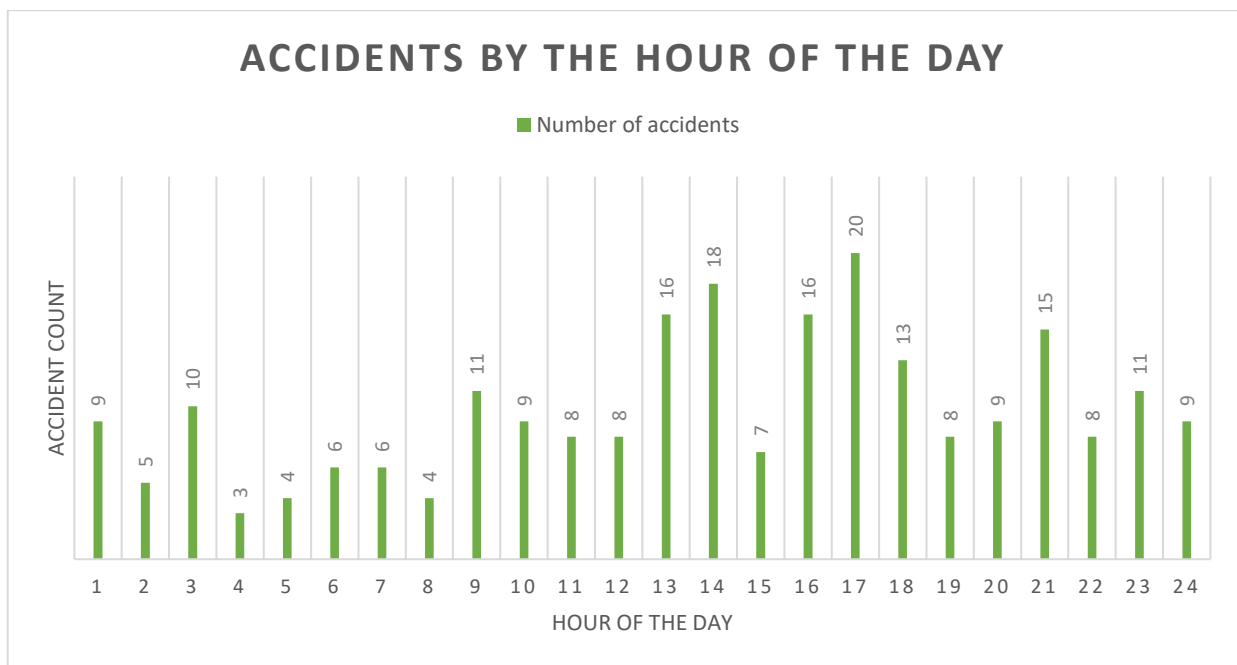
**Fig 3.9. Age of pedestrians involved in Pedestrian accidents**

(Courtesy of [3])

From the above figures 3.6 to 3.9, it can be observed that about 450 accidents took place where the age of the driver was between 20 to 29 years. This age group was primarily involved in fatal accidents and in serious injuries. Drivers aged between 20 to 60 years contributed the most making

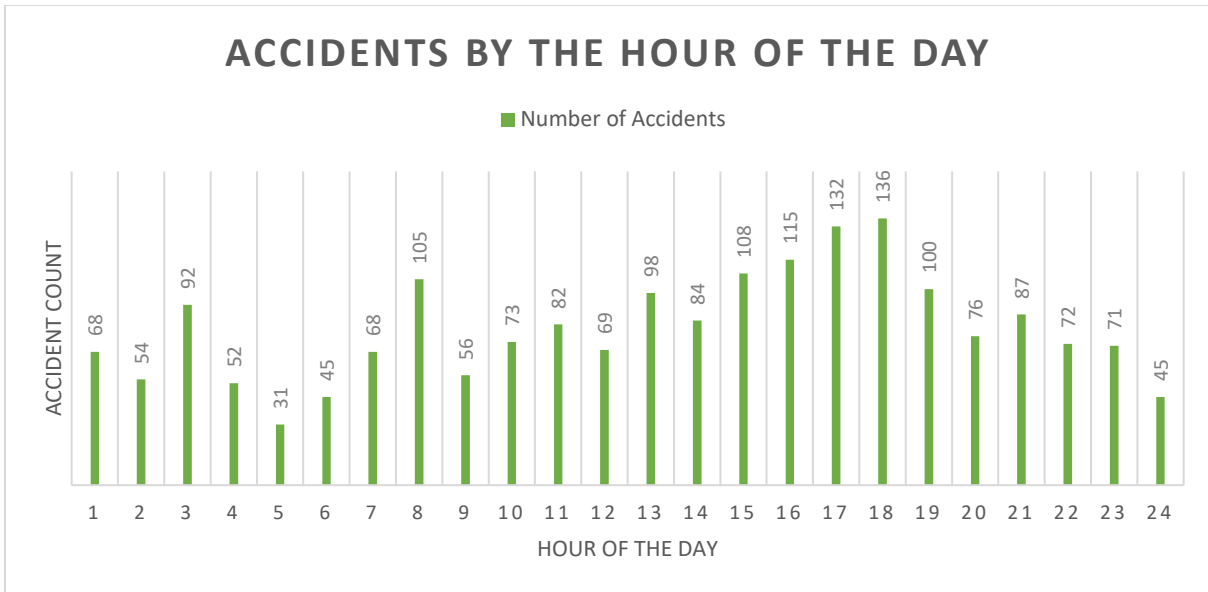
the number of accidents involving bikes approximately 800. Age of primary drivers involved in pedestrian accidents comprises of the greatest number of accidents out of any accident type of up to 1600 within the age group of 40-49.

Through this research it can be observed that pedestrian accidents are at the highest rate. For our research we need to consider that the age group of 40 to 50 contributes highly to accidents at intersections. Hence a study to estimate the risk factor and to optimize a solution is necessary.



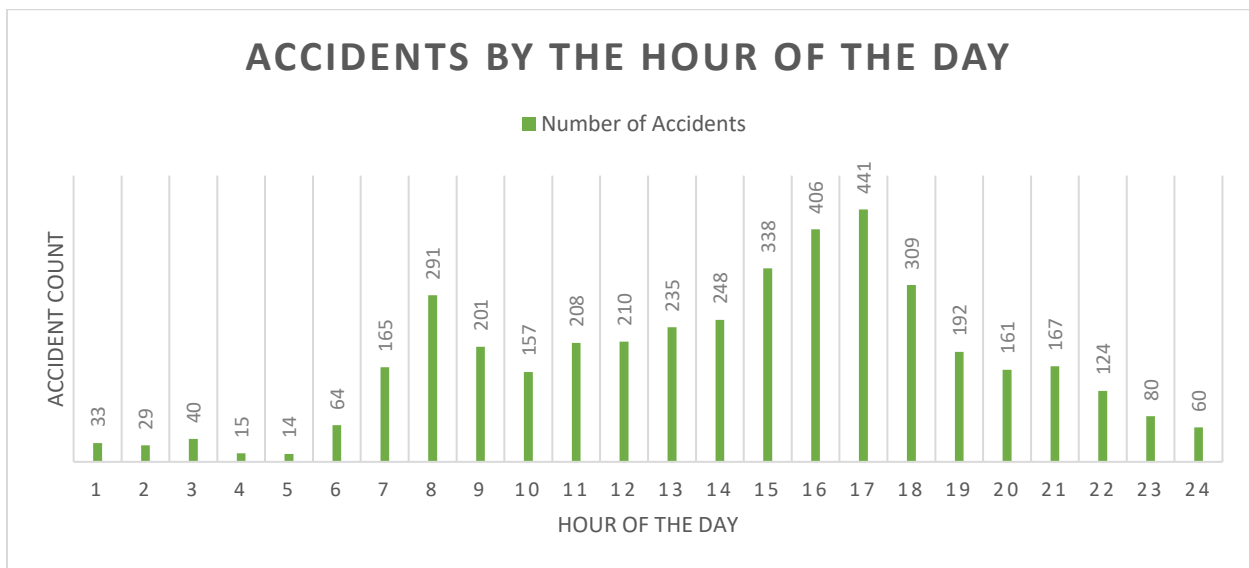
**Fig 3.10. Fatal accidents during the hour of the day (Courtesy of [3])**

It can be observed from figure 3.10 that during 17:00 in the evening there are about 20 accidents which is the maximum out of any hour during the day. The 16:00-17:00 hour sums up to be the maximum number, 36, of fatal accidents during the day. Moreover, during 13:00-14:00 the number of fatal accidents turn out to be the second highest, accounting to 34 fatal accidents.



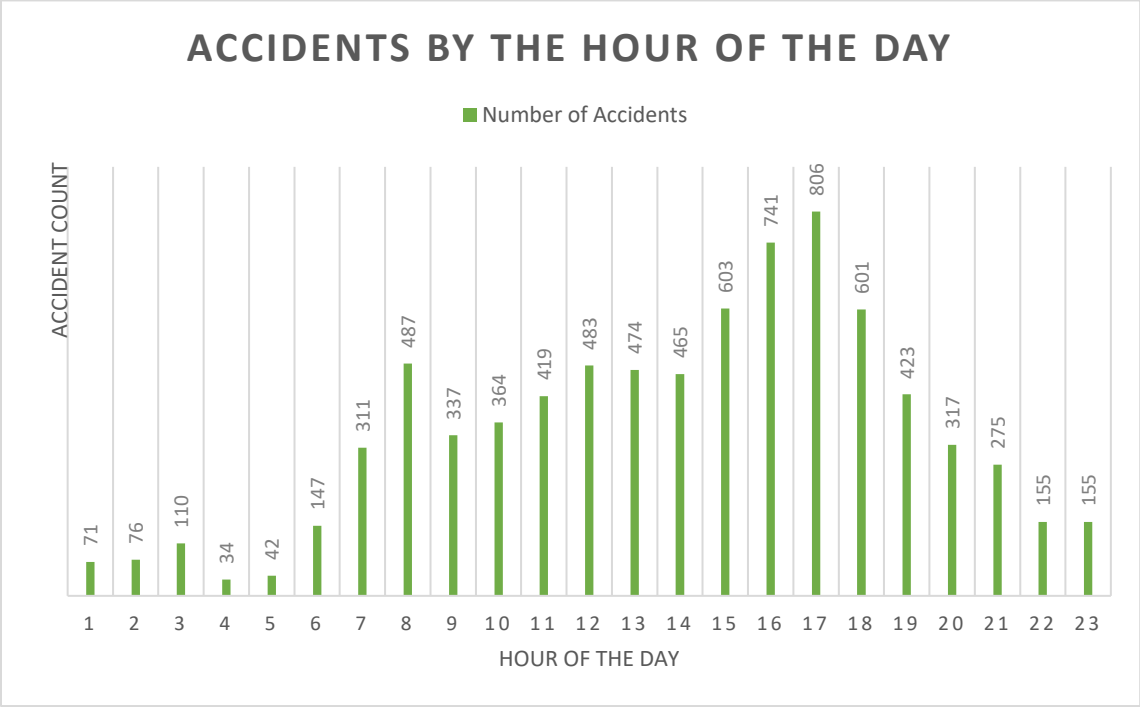
**Fig 3.11. Serious injuries during the hour of the day (Courtesy of [3])**

It can be observed from figure 3.11 that at 18:00 in the evening there were the maximum number of serious injuries, of about 136. From 15:00-19:00 there were more than 100 serious injuries. Apart from this at 8:00a.m we can observe 105 accidents.



**Fig 3.12. Bike accidents during the hour of the day (Courtesy of [3])**

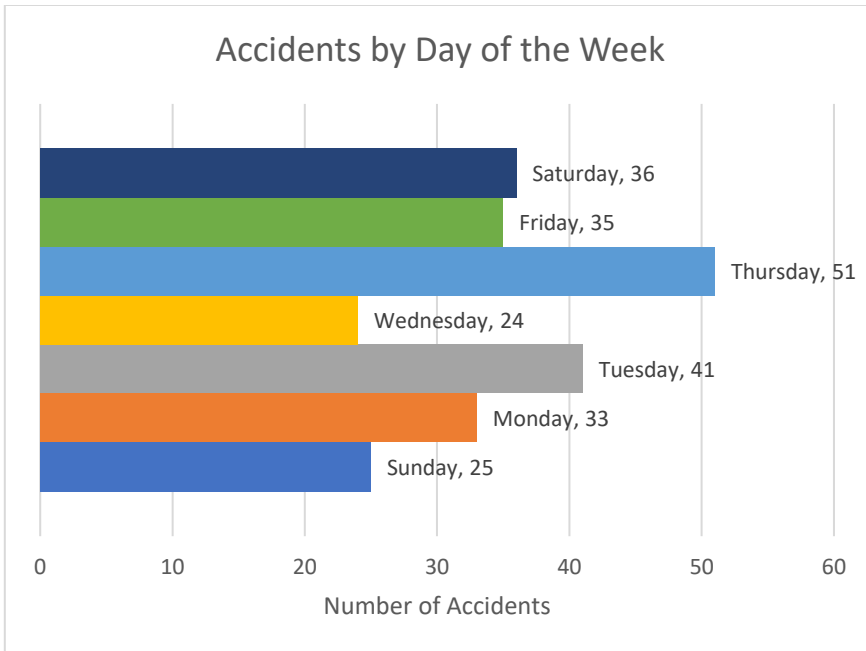
Again, as observed from figure 3.12, in the previous figures, at 17:00 there were maximum number of accidents involving bikes. From 15:00-18:00 about 1500 accidents took place.



**Fig 3.13. Pedestrian accidents at the hour of the day (Courtesy of [3])**

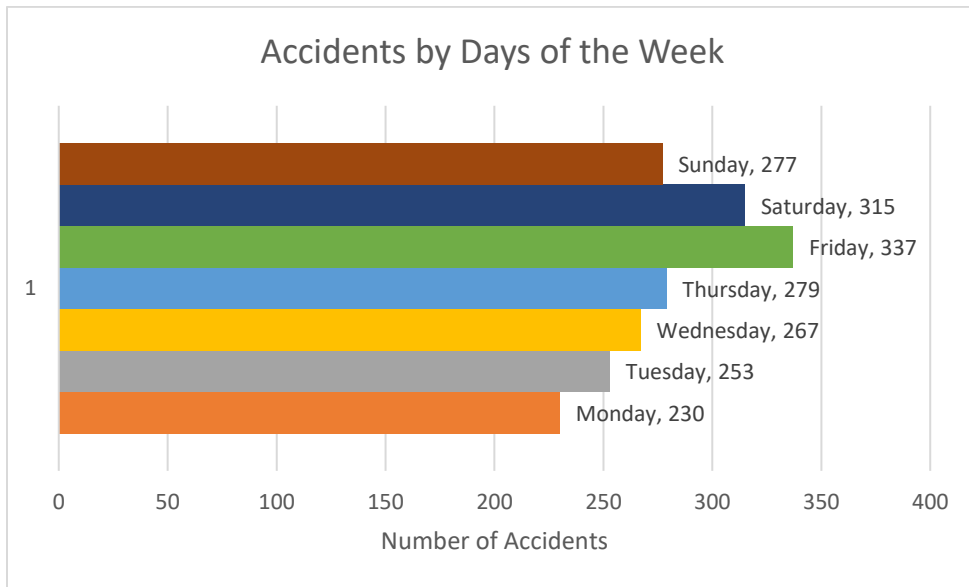
On observing figure 3.13, the data for number of accidents involving pedestrians, the highest danger time is observed from 15:00-18:00. However, about 487 pedestrians were involved in an accident at 8:00 am. This is a result of high volume of pedestrians involving high volumes of students and workers.





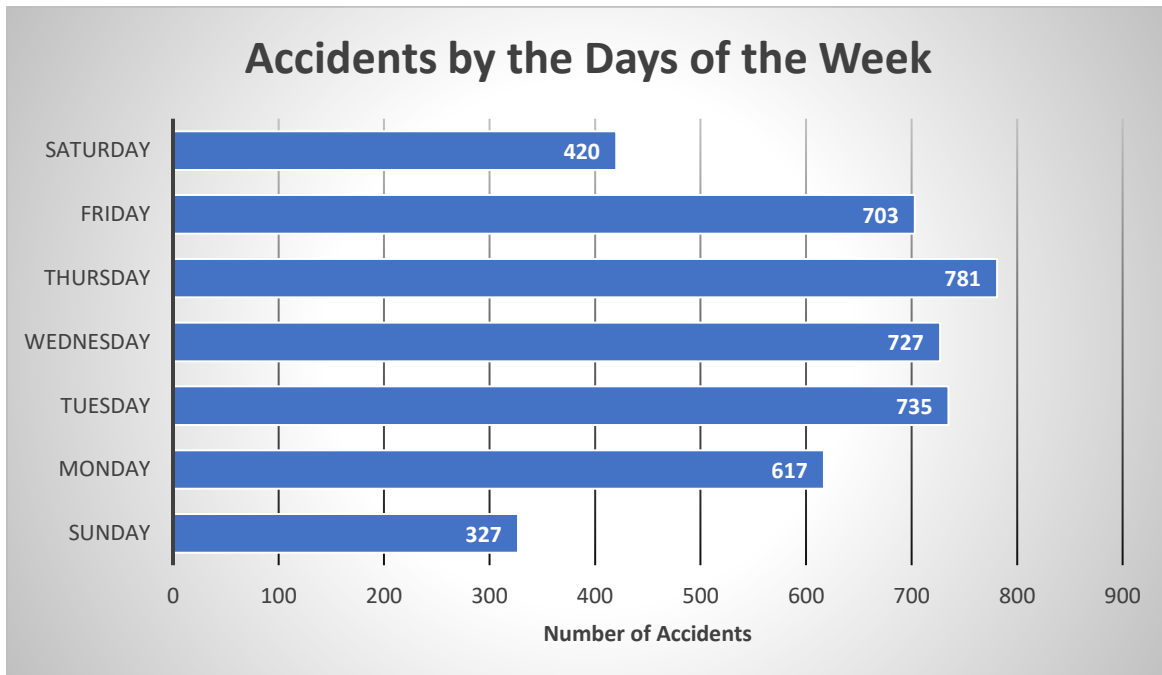
From the following figure it can be observed that 51 fatal accidents took place during Thursdays and 41 on Tuesdays.

**Fig 3.14. Fatal accidents during the day of the week (Courtesy of [3])**



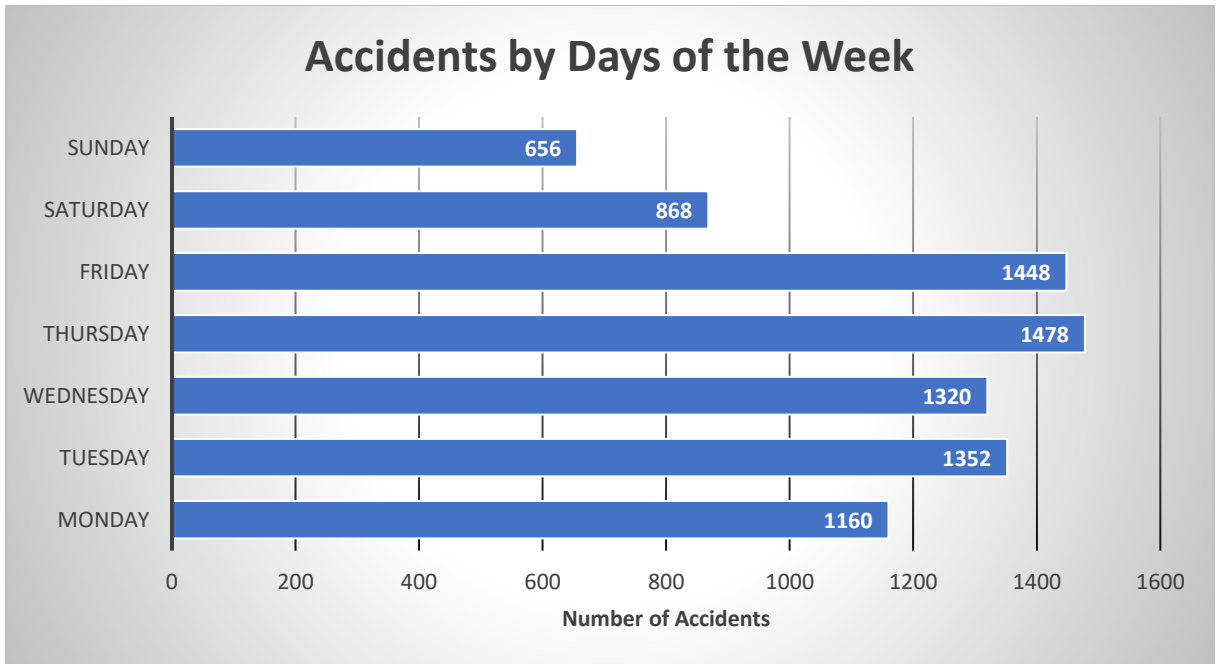
**Fig 3.15. Serious injuries caused during the day of the week (Courtesy of [3])**

From figures 3.14 and 3.15, it is observed that serious injuries were caused mostly on Friday's due to the weekend rush hours. The data clearly depicts how Friday, Saturday and Sunday have contributed to be the most dangerous days for serious injuries.



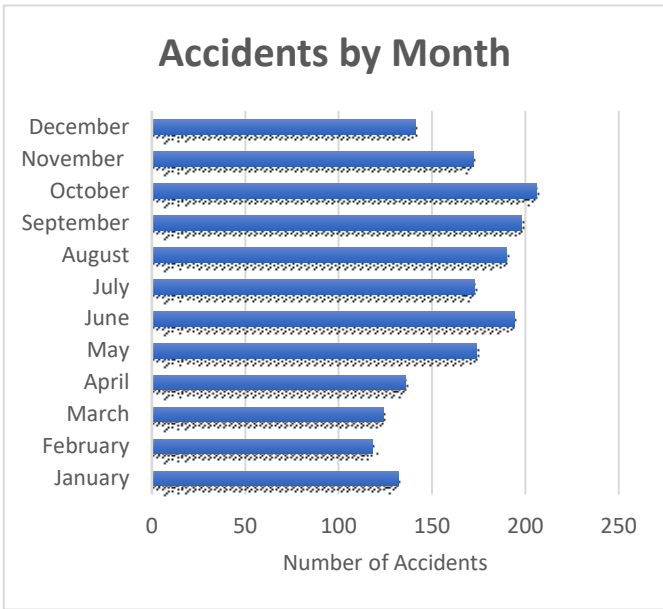
**Fig 3.16. Number of Bikes involved in accidents during the day of the week (Courtesy of [3])**

It is observed from figure 3.16 that Thursday tends to have the maximum number of accidents that involved bikes. Friday, Wednesday and Thursday showed similar number of accidents during the week. Sunday on the other hand showed the least amount of accidents as people prefer to stay at home and there is not a lot of rush.



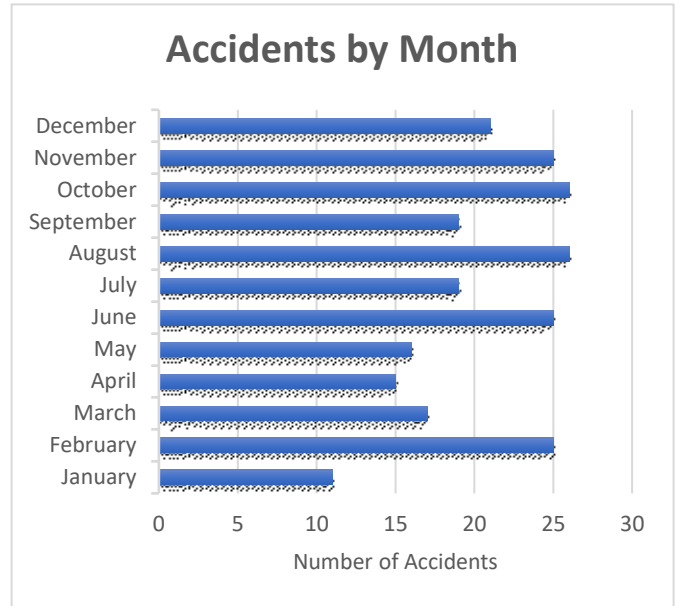
**Fig 3.17. Number of accidents based on pedestrians on the day of the week (Courtesy of [3])**

As shown in figure 3.17, Friday and Thursday tend to be the riskiest days of the week for accidents involving pedestrians. Monthly, October turned out to be the most dangerous month for accidents leading to serious injuries, as seen in figure 3.18. The month of February had least number of bike accidents as depicted in figure 3.20. February, June, August, October being the months of seasonal changes in Quebec saw massive fatal accidents as shown in the graph below in figure 3.19.



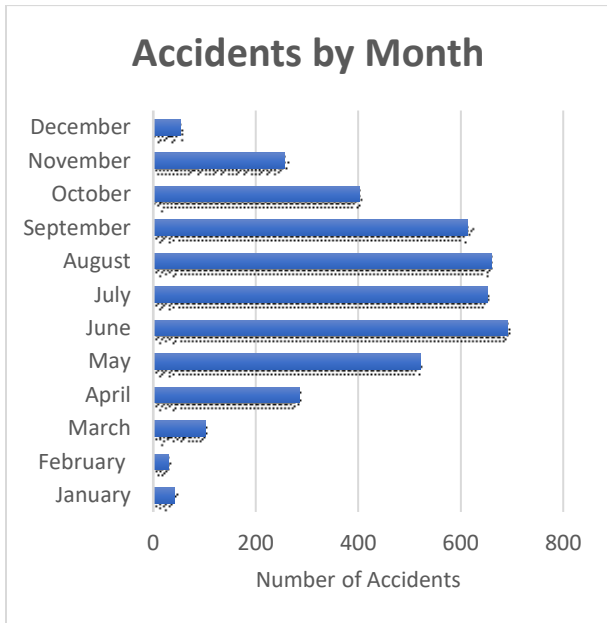
**Fig. 3.18 Number of Serious Injuries by Month**

(Courtesy of [3])



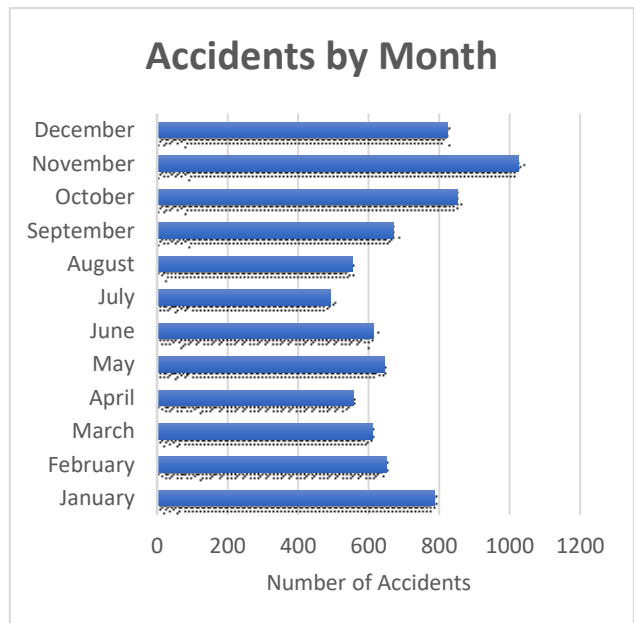
**Fig.3. 19 Number of Fatal Accidents by Month**

(Courtesy of [3])



**Fig. 3.20 Number of Bike Accidents by Month**

(Courtesy of [3])



**Fig.3.21 Number of monthly accidents involving pedestrians**

(Courtesy of [3])

In the graphs above, number of accidents involving bikes were maximum in the month of June and number of accidents involving pedestrians were highest in November.

### 3.3. Case Study 2: SPVM (Service de police de la Ville Montreal)

Bibliothèque et Archives nationales du Québec, consists of detailed profile of facts and statistics about pedestrians. This source shows the reports of accidents from 2000 to 2015. It is segregated into four parts: What, Who, When and Where. Although this research is based on Quebec, it gives us an idea of what went wrong in the City of Montreal.

The research begins by answering WHAT. The number of accidents involving pedestrian accidents minimized variably from 2000 to 2015 as shown in figure 3.22.

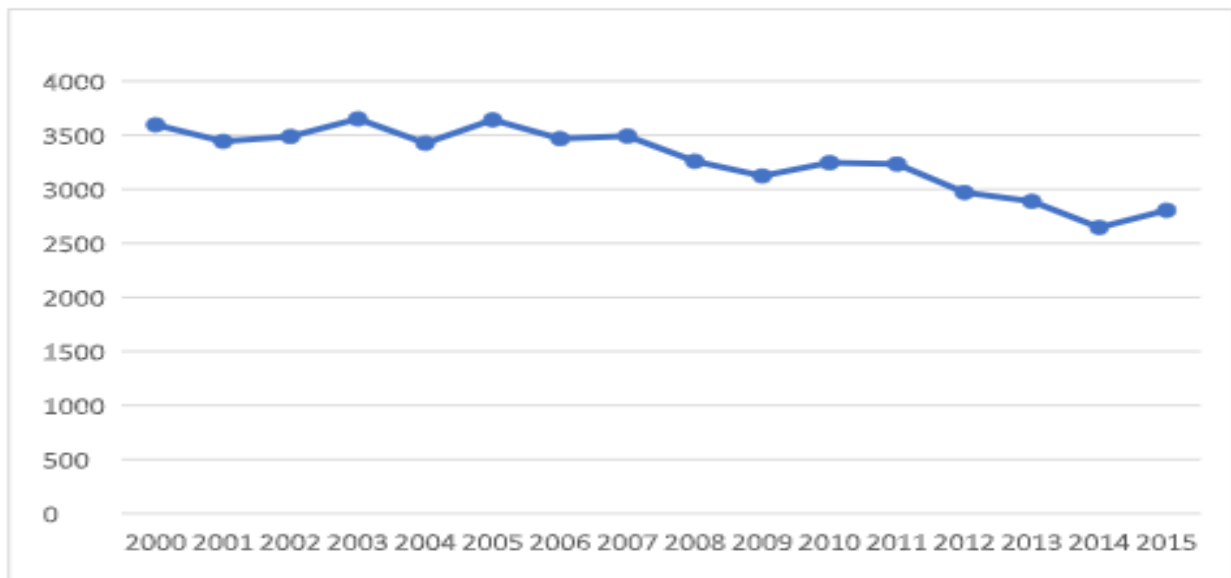
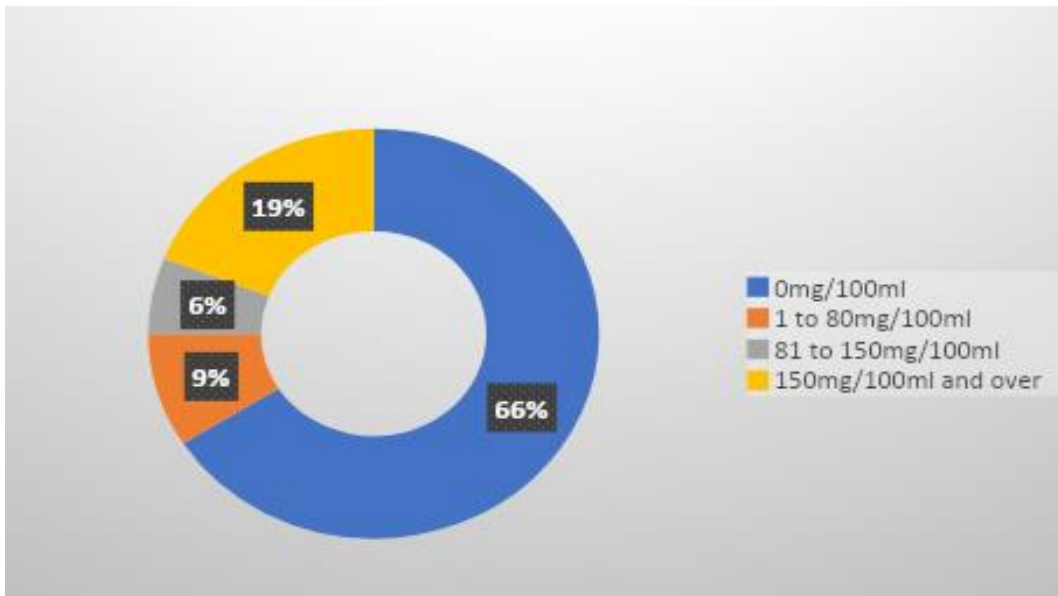


Fig 3.22. Number of accidents involving pedestrians from 2000 to 2015 (Courtesy of [84])

There had been a drop in the number of pedestrian accidents causing bodily injury. It can be observed that during the period of year 2000 to 2015, the number of victims associated with accident causing bodily injury dropped from 47,577 in 2000 to 37,351 in 2015, this represents a decrease of 21%. In the year 2015, there were 44 pedestrians who were killed in Quebec. This led to the

conclusion that pedestrians are a vital part of the road. As there are more and more vehicles on the road, suggestions for public to get around on bicycle and public transport were raised.

The number of ways to protect pedestrians under the influence of drugs and alcohol from vulnerability is very less. As shown in figure 3.23, overall, in Canada about 24% people had blood alcohol consumption above 160mg/100ml. However, in Quebec, from 2010 to 2014, 281 pedestrians were killed with blood alcohol consumption of about 119 (42%) out of which 34% had been drinking and 19% had a blood alcohol concentration above 150 mg/100ml.



**Fig. 3.23. Percentage of accidents involving pedestrians and their levels of alcohol consumption**

(Courtesy of [84])

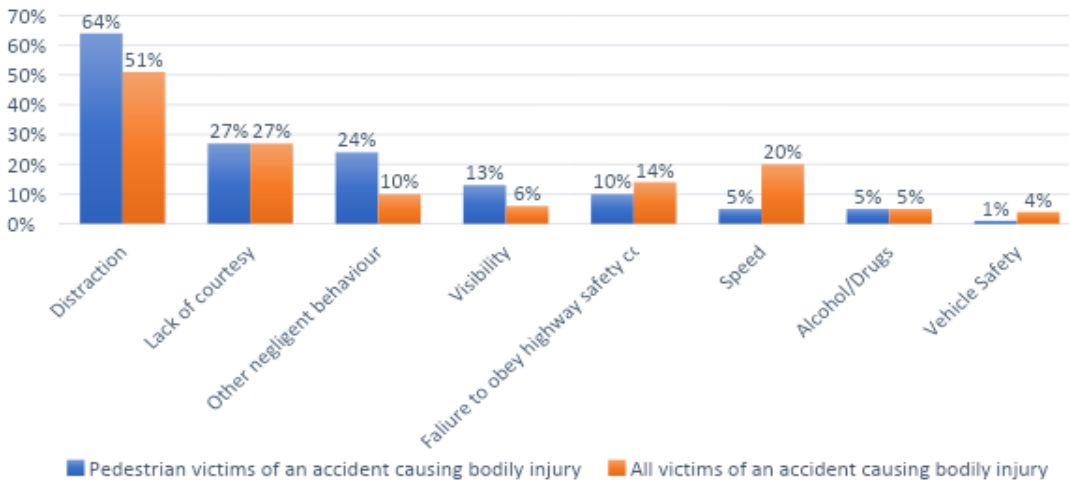
The speed of the vehicle is directly proportional to the injury of the pedestrian. As the speed rises so does the probability of the pedestrian getting killed. The probability in the range of 30km/h to 50km/h range increases suddenly. A pedestrian is 10% likely to be killed if the vehicle is at a speed of 30km/h and it rises to 75% if the speed is 50km/h. If the speed rises to 70km/h this leads to a death rate of nearly 100%.

The timings of sunrise and sunset vary across Canada. City of Montreal experiences fewer daylight hours during winters and with the snow the pedestrian's path gets even worse to walk on. Blomberg et al. (1986) suggested ways like wearing of reflective material, battery-powered flashlight, etc. for pedestrians for increasing visibility for the vehicle drivers during nighttime. This would decrease the risk of pedestrian-vehicle collision at dark hours of the day. Another study by Owens et al., 1994 showed a runner wearing reflective clothing such as a band, strap attached to body parts in motion while running and another runner without any such accessories. The reflective accessories gave a better detection ability. Six times faster detection was experienced with the pedestrians having a flashlight and pedestrians without a flashlight.

One of the most alarming situations is the number of pedestrians using mobile phones. As per the study, one out of five pedestrians look at their cell phones while crossing an intersection. According to a study in Australia done in 2007 on distraction of pedestrians using cell phones while crossing a street, it was observed that more than 20% cell phone users look at their screen. This also affected the walking speed of the pedestrians and their analysis of traffic before crossing.

Another study by Nasal et al. conducted in 2008 explained pedestrians using cell phone, I-pod or no device at all at crosswalks. 48% of the cell phone users crossed unsafely at the intersections compared to 16% of pedestrians using I-Pod type devices and 25% using no devices at all. Thus, cell phone use can be a great threat for the pedestrians.

**Distribution of the percentage of an accident victims based on probable causes of accident and type of victim, 2011 to 2015**



**Fig. 3.24. Distribution of the percentage of accident victims based on causes of accidents and type of victim in the year 2011 to 2015 (Courtesy of [84])**

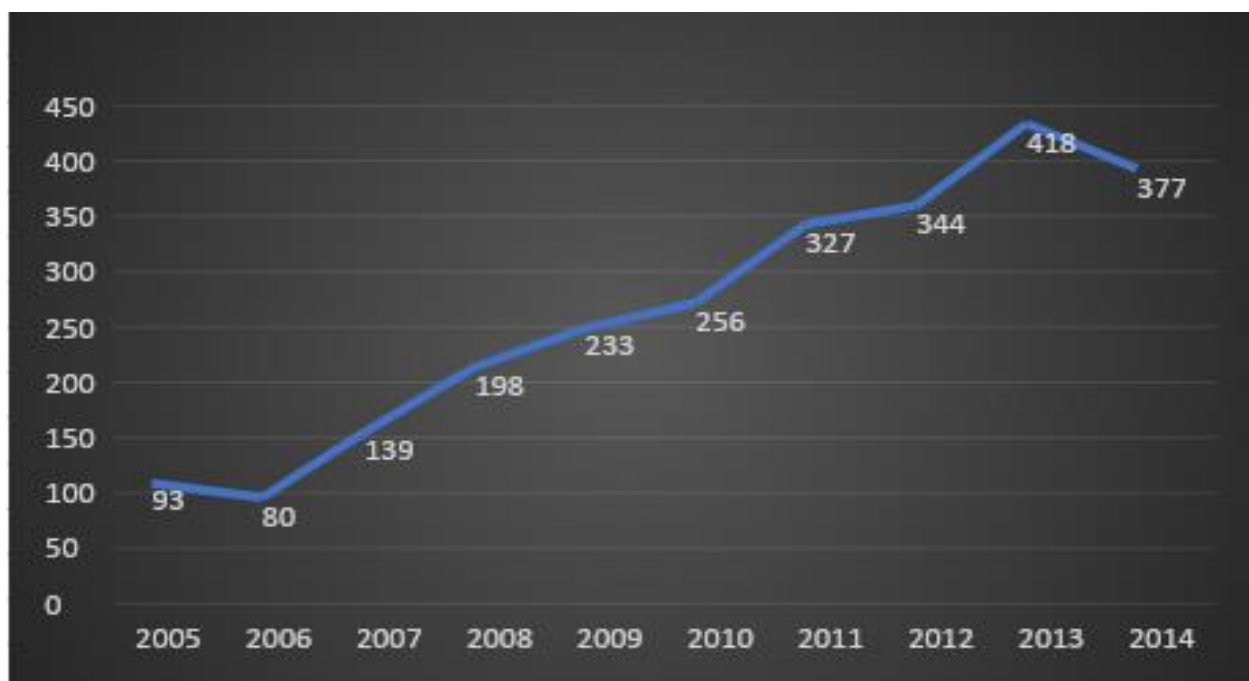
This data is based on the reports presented by police officers, figure 3.24 shows the probable causes of accidents from 2011 to 2015, the percentages of victims based on victim type can be compared. The maximum number of causes and percentages of victims were distraction (64%), lack of courtesy (27%), other negligent behavior (24%) and visibility (13%).

As per the study, the pedestrians who enter from the left side of the vehicle are at higher risk and are seen or observed lesser than those coming from the right when distances are equal. Researches have shown that drivers at night are not easily able to see the pedestrians. Photometric measurements have proved that the headlight can only reach to pedestrian’s feet from a certain distance and does not cover the body. All distances remaining equal, and given the direction of traffic, the cars travel on the right of the road. Hence pedestrians entering from the right side of the road are more prominent to be visualized. On the other hand, drivers might not be able to see who is entering from the left. This study showed that a pedestrian walking from the left side of a vehicle



at a distance of 250ft (76.2m) has about 45% chances of being seen by a driver compared to 68% probability of a pedestrian coming in from the right.

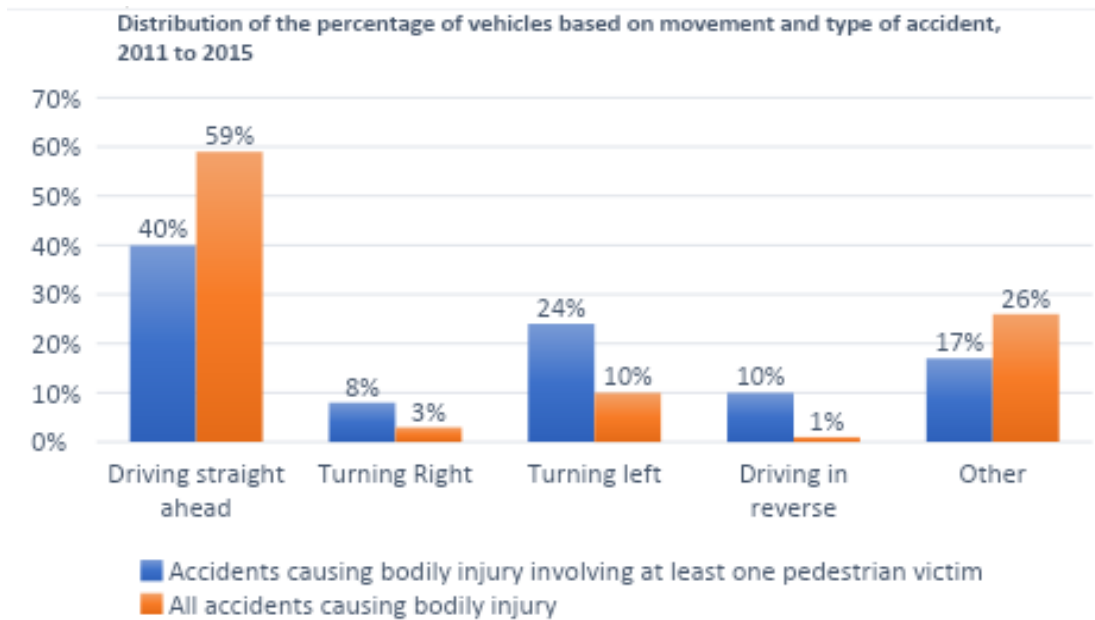
As per the research as shown in figure 3.25, there had been a number of offences observed that were related to driving a vehicle with respect to pedestrians and cyclists. Under the section 349 of the Highway Safety Code – Failure to respect the pedestrian and cyclist rights have been on a rise. About 93 were observed in 2005 to 418 in 2013 and 377 were observed in 2014



**Fig. 3.25. Number of offences observed related to driving vehicles with respect to safety of pedestrians and cyclists (Courtesy of [84])**

The study explains that the collisions can be divided into two main categories: parallel collisions, where a pedestrian walks with or against the traffic flow, and traversal collisions, where a pedestrian travels across the vehicle's trajectory. In case of parallel collisions pedestrian is not always aware that a vehicle is approaching or is assuming that the driver figured out their presence.

On the other hand, in traversal collision, the field of vision of the driver is restricted, giving them lesser time to react as shown in figure 3.26.



**Fig. 3.26. Distribution of percentage based on movement and type of accident from 2011 to 2015 (Courtesy of [84])**

About 37% Quebecers believe that pedestrians obey traffic rules. According to a survey (2014) Pedestrian Safety Campaign, the Société de l'assurance automobile du Québec asked the perspective of pedestrians and vehicles with each other. About 52% believe that drivers are respectful to pedestrians. Only 37% were generally respectful to traffic rules. 80% Quebecers are concerned with pedestrian safety. The survey showed that 84% respondents believe that police officers should be strict or very strict towards those pedestrians who fail to obey traffic rules. On the other hand, 94% believed that police officers should be strict towards those who fail to obey traffic rules with regards to pedestrians. Blind spots were a major factor for accidents according to pedestrians (94%) who were in the vicinity of heavy vehicles. As far as pedestrian safety is concerned, about 24% pedestrians said that they often take measures to be visible to the drivers by

wearing bright clothes or visible accessories. The varied mobility devices with different size and speeds are increasing as well. This diversity of modes will further lead to changes and increase in pedestrian accidents. MPMDs or Motorized personal mobility devices (four wheeled mobility scooters, electric scooters), MPMDs like three wheeled scooters or motorized wheelchairs come in the category of pedestrian transportation and non-motorized personal mobility devices (e.g. Inline skates and skateboards) add to this diversity. Some MPMDs are capable of travelling as fast as 40km/hr. and weigh 90kgs. There are no specific rules, licenses, permits, registrations required to use these MPMDs. Although most of the cases are due to the falling of person from the device, collision of motor vehicles and pedestrian's wheels can lead to serious injuries. This comes from the fact that these pedestrians can travel on roadways.

#### WHO?

This study showed that the population of young and the elderly are at the maximum risk of accidents as pedestrians. The study compared pedestrian accidents involved from (age 0 to 14) and elderly (aged 55 and older). Active adults (age 15 to 64) are most prone to accidents with a frequency of more than 250.

Age Bracket	Number of pedestrian victims of an accident causing bodily injury (n)	Percentage of pedestrian victims of an accident causing bodily injury over all victims
0 to 4	61	11.0%
5 to 9	67	8.8%
10 to 14	127	13.0%
15 to 19	303	7.2%
20 to 24	257	5.2%
25 to 34	386	5.8%
35 to 44	299	5.5%
45 to 54	344	6.6%
55 to 64	367	8.9%
65 to 74	249	10.0%
75 to 84	204	15.7%
85 to 89	59	20.5%
90 and older	14	18.7%
Total	2737	7.4%

**Table 3.2. Number of pedestrian victims in each age group based on type of injury (Courtesy of 84)**

A study done in 2001 showed that pedestrians aged 60 and older sustained more serious injuries than victims in other age brackets at lower impact speeds. A study from 2007 gave a detailed description of capabilities and processes that deteriorate with age and become a major factor in road safety. These may include hearing loss, physical movement, loss of balance, many types of visual functions, ability to react to slipping, and tripping.

Another study from 1988 explained various factors that explain the cognitive issues faced by the older generation. Elderly pedestrians can face issues such as misjudging of gaps due to poor evaluation of incoming vehicles and the distance between them, distraction that can lead to absent minded following of other pedestrians who might be alert, visual attention which means putting more attention on a traffic signal or light rather than the traffic itself, expectations from the drivers, and impatience that can be due to long waits before crossing the street, or crossing between parked cars. According to studies the pedestrians aged 65 and older constitute of the major subgroups that are prone to accidents. Another category that is vulnerable to accidents is children. That's because they are still developing their cognitive skills. As crossing the streets require complex processes and behaviors, children are considered as undeveloped to these, for instance planning their route, detecting vehicles, evaluating speed and distance of incoming vehicles, deciding the right time to cross the street. They are vulnerable due to their small size and lower eye level which could be dangerous if a vehicle is approaching.

Different types of vehicles are involved and are responsible for fatal accidents from 2011 to 2015. 75% are automobiles or light trucks, 17% are heavy vehicles, 5% are buses, and 3% of other kinds of vehicles were involved

When?

When we look upon this thesis, time is one of the major aspects considered for pedestrian safety. The study shows that between 3:00 p.m. and 9:00 p.m. more than four out of ten accidents causing bodily injury involving at least one pedestrian (43%) occurred. About 26% accidents took place between 3p.m. and 6p.m., compared to 25% for all accidents causing bodily injury and 17% between 6p.m. and 9 p.m. Around 10.4% of accidents that involved at least one pedestrian occurred in October, 11.96% in November and 10.8% in December. The analysis also shows that on an average pedestrian accident that cause bodily injury is more frequent on weekdays than on weekends. It was observed that around 10% pedestrians on weekends (Saturday and Sunday) were involved in an accident as compared to 13% that cause bodily injury. However, on weekdays (Monday to Friday), the average daily pedestrian victim was around 16%. Weather conditions were another factor that played a role in pedestrian accidents, clear weather conditions experienced around 63.8% accidents causing bodily injury involving at least one pedestrian victim. 12.1% of at least one pedestrian was involved in an accident during rain or heavy showers.

Where?

The regions where speed limit is 50 km/h or less, accidents involving at least one pedestrian victim are more frequent. About 95% accidents involving at least one pedestrian victim occurred in 2015. These zones had a speed limit of 50km/h or less. The study shows that accidents involving pedestrians have been more frequent on main roads and residential areas. While more than 70% accidents occurred involving at least one victim pedestrian 44.0% were on main roads and 27.7% were in residential streets. About 53% business and commercial, and about 38% commercial accidents involving at least one pedestrian took place in 2015. Rural environments involved serious pedestrian injuries as well.

Table 3.3 shows that Montreal is the administrative region where bodily accidents involving at least one pedestrian victim occur most frequently. The research reveals that Montreal is the most risky and pedestrian accident-prone region (with a rate of 63.5) as compared to Nord-du-Quebec (16.4) and Laval (37.4). About 42.9% bodily injuries involved at least one pedestrian victim in the year 2011 to 2015.

Administrative Region	Accidents causing bodily injury involving at least one pedestrian victim	All accidents causing bodily injury	Average annual pedestrian victim rate per 100,000 residents
Montréal	42.9%	21.1%	63.5
Montérégie	13.1%	18.5%	25.5
Capitale-Nationale	7.9%	8.1%	31.7
Laurentides	5.5%	7.6%	27.3
Laval	5.3%	5.1%	37.4
Lanaudière	4.3%	6.7%	26.1
Outaouais	3.8%	4.2%	28.8
Chaudière-Appalaches	2.8%	5.4%	19.1
Mauricie	3.8%	4.3%	29.7
Saguenay – Lac-St-Jean	2.8%	3.6%	26.2
Estrie	2.3%	4.0%	20.5
Centre-du-Québec	2.0%	3.6%	2.2
Abitibi-Témiscamingue	1.8%	2.1%	34.2
Bas-Saint-Laurent	1.5%	2.8%	20.6
Coté-Nord	0.9%	1.2%	29.6
Gaspésie- Îles-de-la-Madeleine	0.6%	1.3%	19.1
Nord-du-Quebec	0.2%	0.3%	16.4

**Table 3.3: Percentages of accidents involving pedestrians in different regions of Quebec (Courtesy of 84)**

### **3.4. Problem Discussion**

The issue of pedestrian safety is of major concern and requires reliable and sturdy solutions that can effectively reduce the risks posed to pedestrians by enhancing the current road safety and traffic management solutions. The system proposed in this thesis, when compared with other existing technologies, provides a more promising solution due to its low-cost, efficiency, and reliability.

The figures below highlight a few problems that are currently a major concern for pedestrians as well as overall traffic handling. Advanced sidewalks (Figure 4) are built for the ease of pedestrians crossing the streets. They might be effective for small crossings but are dangerous for intersections, also psychologically it does not give pedestrians the feeling of being safe.

Speeding vehicles like the HMV's could cause harm to the pedestrians waiting to cross the streets on the advanced sidewalks. It creates a bottleneck for the approaching vehicles and thus is not an effective measure for the safety of pedestrians.

In figure 3.27 to 3.28 the sidewalks are enhanced for pedestrian safety. Fig. 3.27 comprises of advanced sidewalks, whereas Fig. 3.28 accentuates the visibility of the crossing by painting it with bright colors. It might give an attractive path to the pedestrians to cross, but for the consequent vehicles approaching, it might be distracting, and not serve the purpose they are intended to serve, and accidents may still take place.



**Fig 3.27. Advanced Sidewalks for pedestrians (Courtesy of [75])**



**Fig 3.28. Visible pedestrian crossing (Courtesy of [24])**

Figure 3.29 (a, b, c, d, e) highlights more pressing issues that are a safety concern for pedestrians. These images highlight the problems currently observed in downtown Montreal, which



need urgent attention. In the figures, there is ongoing unenclosed construction taking place, and the sign board has fallen due to precipitation and wind flow. These have damaged the streets and people who are unaware of this can be in danger. The figure also shows how waste from restaurants is drawn towards the cycle track and can also be drawn towards the main road, thus causing fatal accidents.

A lot of these major streets in the city of Montreal experience varied issues when it comes to the safety of pedestrians walking or using any mode of transport, and cyclists.



**Fig. 3.29 (a).** Various issues on the Montreal streets causing a threat to the safety of pedestrians and cyclists



**Fig. 3.29 (b).** Various issues on the Montreal streets causing a threat to the safety of pedestrians and cyclists



**Fig. 3.29 (c).** Various issues on the Montreal streets causing a threat to the safety of pedestrians and cyclists



Fig.3.29



(d).

Various issues on the Montreal streets causing a threat to the safety of pedestrians and cyclists



Fig. 3.29 (e). Various issues on the Montreal streets causing a threat to the safety of pedestrians and cyclists

These issues highlight various small and big issues that affect the safety of the pedestrians across the city. However, despite the fact that the city has been taking several measures to minimize the accidents and fatalities, these issues keep coming back. This can be due to lack of proper solutions

that can minimize risks of pedestrians getting hurt or involved in accidents. Due to efforts on decreasing accidents, the total number of accidents decreases annually.

### **3.5. Conclusion**

This chapter provides a detailed review of the literature in order to identify the current solutions and systems proposed to enhance pedestrian and road safety. For this analysis, we divide the review of literature into three research categories. The first category consists of researches which study pedestrian volumes at signalized or unsignalized intersections. The second category involves an analysis of researches involving different object detection algorithms to identify pedestrians, vehicles, and other aspects of traffic management system. The last category of researches consists of works that provide cloud-based solutions to aid the current traffic safety systems. The review of literature is followed by a thorough analysis of two major case studies conducted in the city of Montreal that provide relevant data in terms of pedestrian accidents and pedestrian counts at dangerous intersections. These case studies are studied in detail and the analysis is used to identify the various risks that threaten pedestrian safety. We conclude this chapter with a final section that discusses the problem at hand with regards to pedestrian safety. This is the problem that we will be tackling in this thesis and for which we will be providing an effective solution.

# Chapter 4

## Working with Object Detection

### 4.1. Introduction

In this chapter, we provide an in-depth description of how neural networks work, and how we use them for object detection. The chapter also provides details of the various convolutional neural networks and their working. Information on how computer vision takes place and how object detection algorithms function are also given in this chapter. Finally, this chapter analyzes the various techniques of Region-based Convolutional Neural Networks, their shortcomings and advantages. The aim of this chapter is to explain in detail the main concepts involved in computer vision and efficient object detection.

### 4.2. Neural Networks

#### What are Neural Networks?

Neural networks are a family of functions which take a very loose inspiration from the frames that are used to approximate functions that have a large number of inputs. They are an example of nonlinear hypothesis where the model can learn to classify much more complex relations. It also scales better with logistic regression that did not have large number of features. In the figure 4.1, the Input layer, the hidden layer, and the output layer are the three layers that define any neural network. The input layer will have the input given to the model, the hidden layers are particular

layers that try to learn a representation from the data, and output layer gives the output of a particular solution that the neural network is designed to give

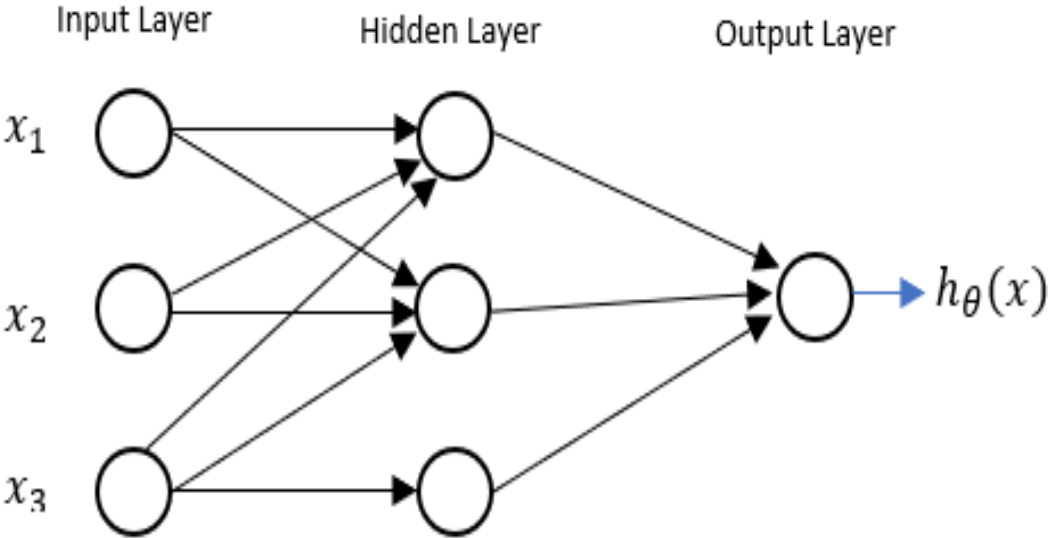


Fig. 4.1. Layers that define a Neural Network (Courtesy of [4])

This figure 4.1 represents an artificial neuron. It has weights and biases which are multiplied with a dot product and are passed through some kind of nonlinear activation function. What a non-linear activation function does is help us learn the different variations of a same class object. These variations can be learnt better as we use the output from the previous layers rather than a linear classifier where we use same set of weights and try to train it on a particular set of data points.

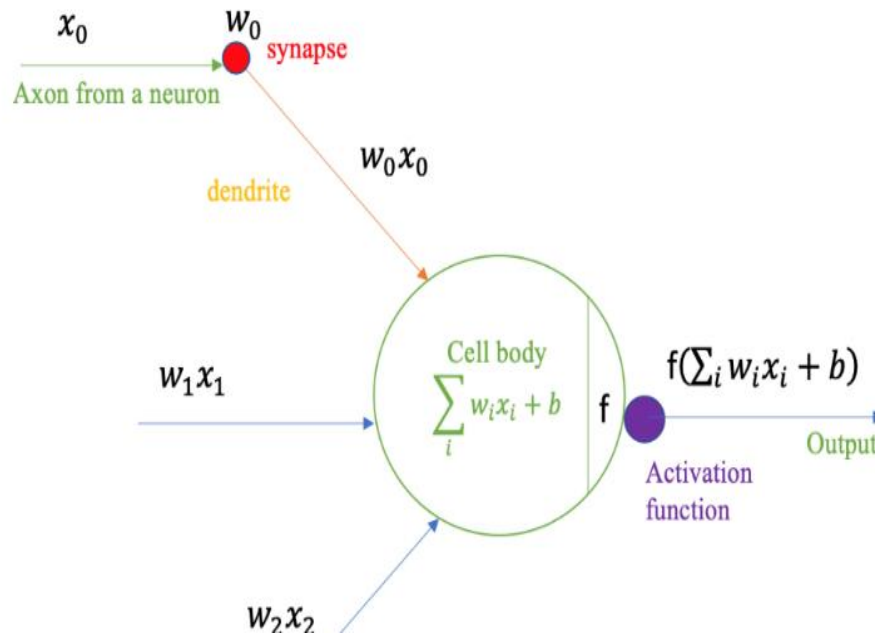


Fig. 4.2. (a) Artificial Neural Network (Courtesy of [17])

**Sigmoid**

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

**tanh**       $\tanh(x)$

**ReLU**       $\max(0, x)$

Fig. 4.2. (b) Various Activation function (Courtesy of [17])

Figure 4.2 shows various activation functions that can be used. The first is a sigmoid function, second is a tan h, and the third is a ReLU function. Sigmoid is not used anymore, but tanh is used in certain cases and ReLU is a very good non-linearity function, which is recently introduced and

is highly used now. It is good because when doing back propagation the gradient does not die but it becomes constant, so it is easier to handle.

### **4.2.1. Steps involved in Neural Network Training**

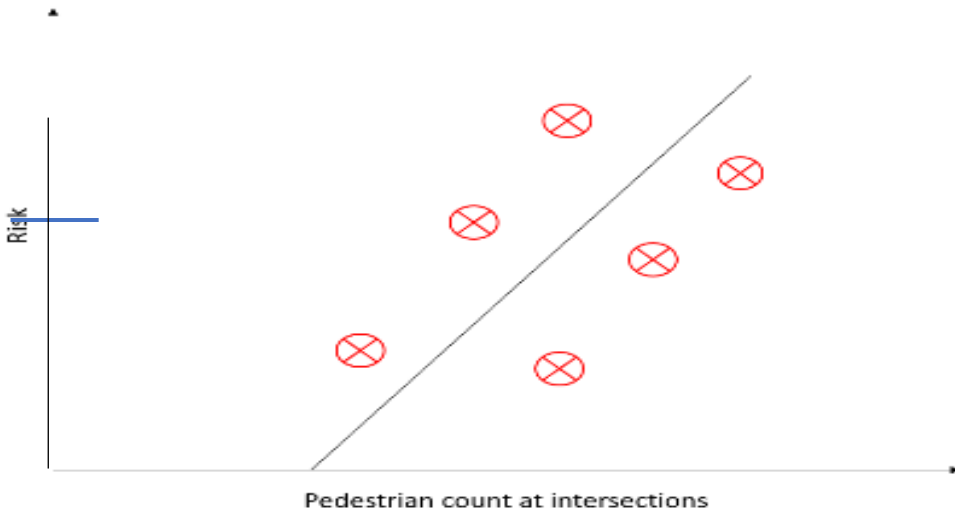
Initialize the weight randomly then do a forward pass, get the results, calculate the loss on those results, the loss here is what the actual output should have been and what the output our model has given, so that difference or loss is calculated and is back propagated through the network. What we mean by back propagation is that we send that loss through the entire network and try to find out how a particular neuron would have the effect on the particular loss in the final output.

### **4.2.2. How a NN works**

Pedestrian is a function of the pedestrian count at the intersection. The linear regression is shown by a straight line in the figure 4.3 and since the pedestrian count cannot be negative the curve bends and becomes zero, as shown by the thick blue line, which is also a function for predicting pedestrian at risk.

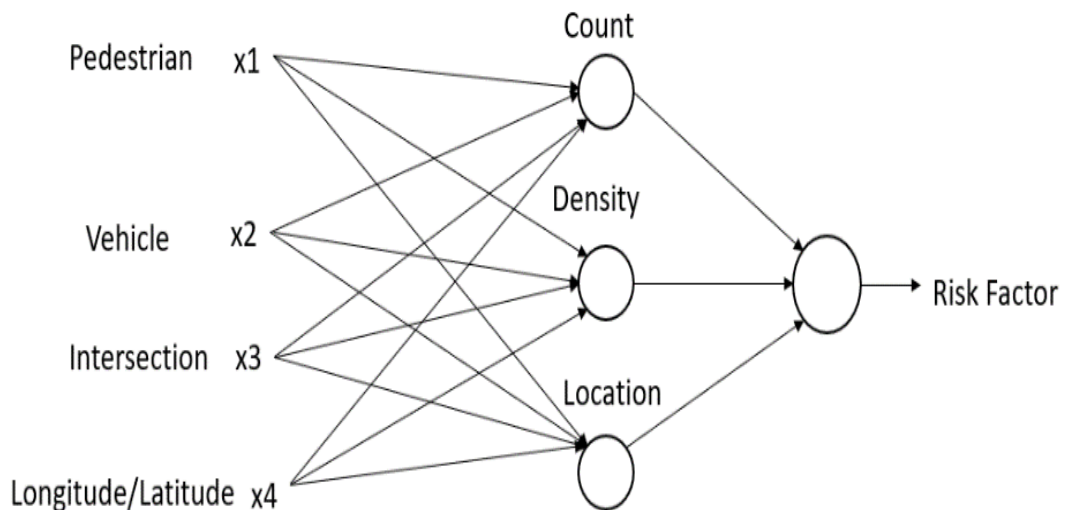
In figure 4.3, a single neuron is used to input the pedestrian count. It computes its linear function and outputs the estimated risk. The function that is formed by the blue line and the straight line is called a ReLu function, which stands for rectified linear unit. So, a single neuron is stacked together with other neurons to form a neural network





**Fig. 4.3. Rectified Linear Unit (Courtesy of [17])**

In figure 4.4, each of the inputs Pedestrian(x1), Vehicle(x2), Intersection name(x3), Longitude/Latitude(x4) help us to achieve the output that is the risk factor. For the neural network, all that is needed is the input and the output, everything else like count, density and location are figured out by itself as they are the hidden units of the neural network



**Fig. 4.4. Neural network, where input is provided and output (risk factor) is achieved**

In supervised learning there is an input  $x$  and an output  $y$ . For our project the aim is to interpret unstructured data.

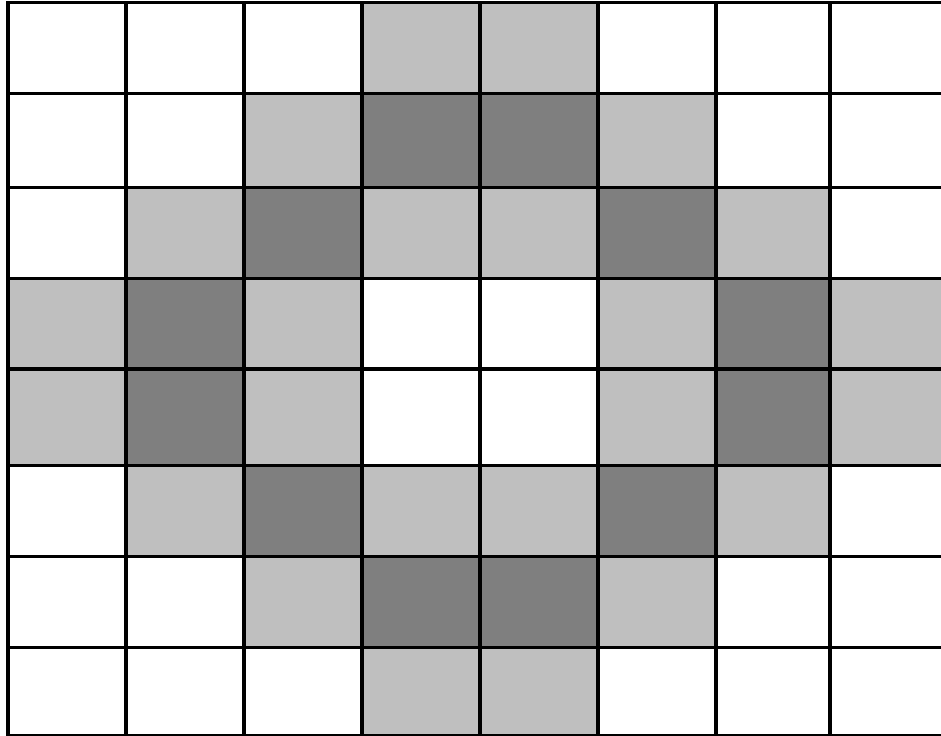
Computer vision is a part of artificial intelligence, it has its own set of Turing test that answers questions related to the image.

Object detection allows us to see the objects' properties and attributes by classifying it. Human brain combines images based on prior knowledge. 25% of a human brain is dedicated to vision. Digital image is the object of computer vision. An optical image can be obtained by propagation of light, depicting the features of the object. Pinhole camera model is supposed to be one of the first optical system. It captures all rays through a single point called the focal point and forms an image on the image plane. Perspective distortion is another optical system in which distortion of real-world objects are introduced, for instance converging vertical lines instead of parallel lines. Dimensionality reduction machine, which also means obtaining 2D images from 3D world, might end up in loss of angles and distances. However we may retain linearity, this principle is also widely used in modern digital camera.

A digital grayscale image is a 2D array as shown in figure 4.5. The meaning of each rectangular grid is that it measures brightness value with one byte.

2D array  $I \in (b_{ij})^{n,m}$ ,

$b_{ij}$  = Brightness value, measured on 2D rectangular grid



**Fig.4.5. 2D array**

To obtain the color of any image, human eyes are capable of [380nm, 780nm] spectrum of visible light.

Image processing is important for human, and image perception for image analysis and recognition. Further discussion of image defects is done later in the chapter. Low contrast, Noisy, Blurry and non- uniform lighting images are observed so it is very difficult to recognize.

Noise corrupts the true signal. Gaussian normal distribution draws intensity of the noise.

$$\text{Image}(i,j) = \text{True}(I,j) + \text{Noise}(I,j)$$

Noise  $(i,j) \sim N(\mu,\sigma)$  simplest way to reduce noise at a static scene is edge computing. A lot of algorithms rely on edge detection.

Consider  $f$  is an image and  $g$  be the kernel. The output generated by convolving of  $f$  with  $g$  is denoted by  $f.g$ .

$$(f \cdot g)_{m,n} = \sum f_{m-k,n-l} g_{k,l}$$

$$(f \cdot g)_{m,n} = \sum f_{m-k,n-l} g_{k,l}$$

The goal of object detection is to detect the exact location of the class in the image. Things are objects of certain size and shape, for instance pedestrians and vehicles. In comparison with image classification, output of the detector is structured. Each object in an image is marked with position of one of the corners, by width and size of the box called bounding box. (x,y,w,h) class label. The metric is IoU or Intersection over union, which is the ratio of the intersection and the union of the predicted box and the ground truth box. If this ratio is greater than the threshold set for the IoU then the predicted detection is very precise. The larger the value over the threshold, the better is the prediction by the model. Moreover, the threshold value determines how precisely the detector should detect the objects. Hence setting a larger threshold value implies that the precision of the detector should be greater. Usually this value is set to 0.5, and the IoU for each detection should be above this value. When the detector generates outputs, it generates detection proposals along with confidence scores for each proposal. These proposals are ranked according to their score, and then the IoU for each proposal is calculated. If the calculated IoU for a detection proposal is greater than the threshold, then it is considered to be a true positive detection, and a false positive detection if the IoU is lower than the threshold. Furthermore, if any ground truth box for the object is not detected, then it is considered to be a false negative/misdetection. These measurements further allow the precision and detection over the entire data set to be calculated. The formulas for precision and recall are as follows:

$$Precision = \frac{Number\ of\ True\ detections}{Number\ of\ detections} \dots\dots\dots 1)$$

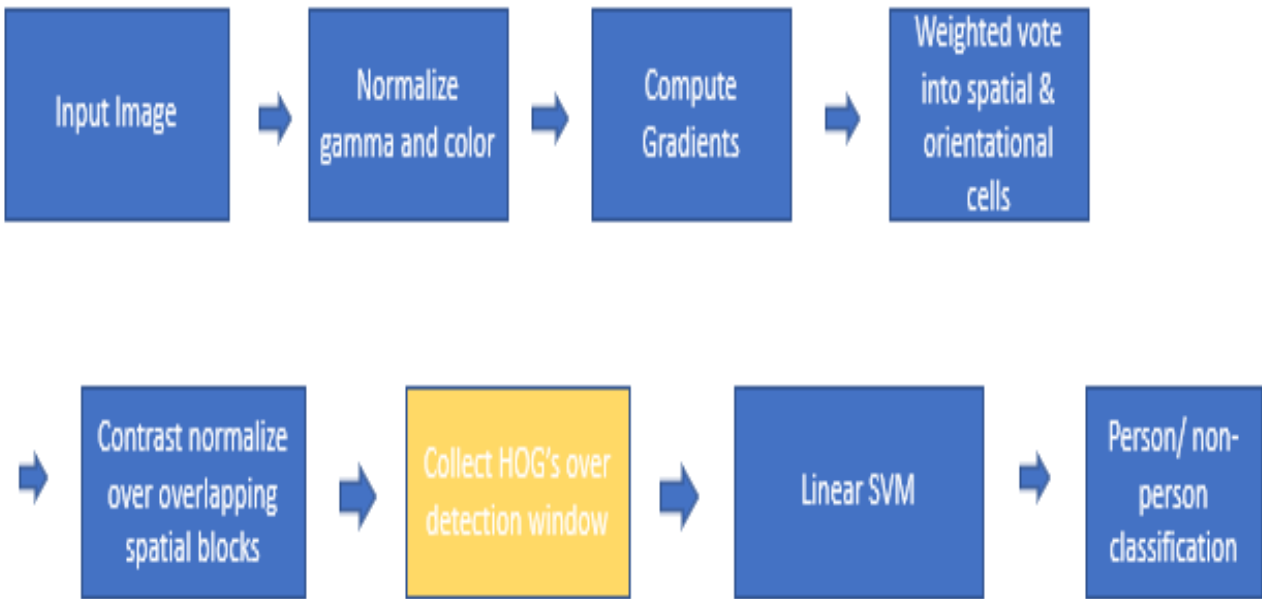
$$Recall = \frac{\text{Number of true detections}}{\text{Number of ground truth objects}} \dots\dots\dots 2)$$

(1) & (2) *Courtesy of [56]*

For object detection it is essential to have a ground truth annotation. Higher requirements on annotation precision are based on higher localization precision that we need to obtain. In this study it has been important to mark each pedestrian from toe to head, to the middle point between the legs, and then bounding boxes width to height aspect ratio is placed on top of this annotation. Similarly, the vehicles were marked using the same approach. There are several algorithms that have been proposed to detect a class of an object, LabelImg was used to label and prepare the test data.

The concept of sliding windows is used to identify the relation between detection and classification. The problem of object detection to image classification is reduced. Classifying images can be accurately done by image classification. In this case we train a pedestrian classifier to say whether this fig. contains pedestrians or not. To identify where exactly the pedestrian is and how many pedestrians are there in the image the main technique is to reduce the problem of detection to image classification, also called Sliding window. We consider a fixed size rectangular window which has been chosen correctly. The pedestrians will occupy most of the window. All possible regions can be seen by scanning the image from left to right, top to bottom with this window. Further we can identify vehicles and name the classifiers pedestrian and vehicles independently. There were instances where one pedestrian overlaps the other, in such a case the need of a powerful classifier to detect partially overlapped images raised. Single objects can be reliably detected by powerful modern detectors. Overlapping and small objects can lead to detector failures sometimes. A sliding window approach is vital for object detection. Various scales and

aspect ratios are maintained by search windows of different size and aspects, or by image scaling. One of the basic object detection method is called Histogram of gradients or HOG detector. It is derived from using image edges as features. They are sensitive to lights and robust features of an image. Information about images is lost in image maps. Dalal and Triggs [86] in 2005 proposed to compute histograms to use them as features of the image instead of edge maps. This method has been applied to pedestrian detection. The first step involves taking a sliding window of a fixed size as an input. Color and normalization to the image is then applied. However, according to Dalal and Triggs [86], this step can be omitted. Histogram normalization made this step unnecessary. The image gradients are then calculated as image convolution filter, and the window is then divided into eight by eight cells of pixels. In each of these cells, histogram of gradient orientation is calculated. The vote for each pixel individually is carried out and weighted by gradient magnitude. Neighboring cells are then combined into blocks. Histogram with full cells in the block are concatenated as each cell is included into several neighboring blocks. Each block can be rectangular or circular. Concatenated histograms in each cell are then normalized as each cell is included into cell blocks. After normalization it produces several image features. Final descriptor contains all the concatenated blocks. The final step could be described as classification of pedestrian and the ground. Dalal and Triggs [86] use linear support vector machine method for classification, this gives us a score which is considered as threshold. If the threshold is positive it is claimed that the pedestrian is identified.



**Fig 4.6. Steps involved in classification of the image. (Courtesy of [28])**

Neural networks have been effectively used as classifiers in sliding window. Strong model CNN classifiers are slow, it is impractical to use them in sliding windows. In order to reach real-time performance, neural networks at the full stages should be very simple and fast. Steps involved in classification of images is shown in figure 4.6. A full stage classifier has one convolutional layer from previous stage classifier, to the outputs of fully convolutional layer of the current state classifier, before final classification of a proposal. There have been great results demonstrated by Convolutional Neural Network for image classification, it can detect presence of a pedestrian from a large image.

### **4.2.3. Tasks Involved in Computer Vision**

There are different tasks that are involved in computer vision, the first is classification, where we take an image of a particular object of a particular class that is contained in that image i.e. if the image contains a pedestrian then the model will predict whether the entire

image contains a pedestrian or not, the pedestrian can be located anywhere in the entire image. Then comes the task of localization, in this task we identify where exactly is the particular object located in the image, in localization we can only find one particular object for a particular image, in object detection we further extend the task of localization. Here we try to identify if there are multiple objects in a particular image, for instance images of pedestrians and vehicles in a particular image bounded by boxes. The final task is segmentation where we try to find out what the probability of each pixel belonging to a particular task is, in this case the object is marked in particular and not the bounding box.

#### **4.2.4. Convolutional Neural Network**

These are a class of neural networks that have been extensively used in visual recognition problems, this was invented by Yann LeCun in 1998. In 2012 Alex Net gave very good results for the image net challenge which beat the previous competitor by large number of percentage points. Hence it can be observed that Convolutional neural networks have taken over the field of computer vision as it can learn directly from the data itself and there is no need of hand engineered features. A typical convolutional neural network consists of an input layer, the filter depth or different kernels from where the ReLU output is taken, which is connected to a fully connected layer. This will produce a kind of probability density about which class the image belongs to. A convolution operation takes a filter or a sliding window that is run through the entire image. For every position our kernel is placed on the image we have an output called a feature map. A feature map takes the output of the entire convolution operation that has been performed over a particular image and the feature map will have outputs that have been passed through a non-linear activation function which is a ReLU non-linearity function. What these feature maps will do is in the lower layer of the network or the initial layers of the network they will learn certain kinds of representations from the



dataset such as edges or small patterns and the deeper we get in the network layer, the more we see the data being created. The deeper neural networks are easily able to learn representations of data for multiple classes. There are different activation maps or the feature maps that are created after a convolution operation, the feature maps are calculated so that they can be passed on to the further layers. It can be imagined as a set of images that have been generated by the model for the next layers itself. So particular convolution layer will create a set of activation maps, for instance if we have two kernels, we will have six activation maps, and the activation maps will also reduce in size unless and until we use the concept of padding.

Region-based convolutional network is a contemporary approach to detect objects. If we have a strong trained CNN classifier, we can apply it on a new data set of a new object detection problem. For this a pre trained CNN classifier can be used as a feature extractor to train and support external machine classifier on these features and then use it in a sliding window framework. Overall, this will be slow as we need to apply CNN to hundreds of thousands of windows.

In a seminal work of Girshik et al, it was proposed to use external object proposal generator to get 2000 objects proposals per image. Features are then extracted with convolutional neural network and then from these proposals and classified with SVM classifier. This method is called region-based convolution network or R-CNN. Selective search is one of the generic object proposal generation methods. The training procedure of the R-CNN consists of three steps. The first step is that CNN is pre-trained for instance on ImageNet for image classification. Second, CNN is fine-tuned for object detection on object detection data set. Third, linear classifier and bounding box regressors are trained on top of CNN features extracted from object proposals. R-CNN outperforms previous object detection methods. Fine tuning of convolutional neural network on turgid data set and application of bounding box regression, both improve the performance significantly. One of

the most important advantages of R-CNN is that it can use any convolutional neural network for feature extraction.

However, basic RCNN method has several problems. First is redundant computations. All features are independently computed even for overlapped proposal regions. Second, need to rescale object proposals to fixed resolution and aspect ratio. Third, the dependency on the external algorithm of hypothesis generation and the last is very complicated training procedure with high file system load. The problem is we need to extract and store CNN features for subsequent SVM training.

Rapid changes have been made to R-CNN, one of this is Fast R-CNN model. A base CNN classifier requires an image with fixed resolution of 224 by 224 pixels as input due to fully convolutional layers. Rescaling to sixth resolution is needed after object proposals are extracted, like scaling changes object appearance and increased variability of images. Any CNN classifier can be adapted to various image resolutions by changing the last pooling layer to the spatial pyramid pooling or SPP layer. The idea of SPP is derived from traditional bag of Spatial Pyramid Pooling. First level of the pyramid is a region of interest itself. The second level divides the region into four cells with two by two grid. The third level divides the region into 16 cells on four by four grid. Average pooling is applied to each cell. Thus, if the last convolutional layer has 256 maps, pooling in each cell produces a vector that is 256 in length. Feature vectors for all cells are concatenated, these are then passed to fully convolutional layer as input. Lastly, we obtain fixed length feature representation for input images having various resolutions. Convolutional layers of base CNN can be applied only once per input image. For each window, Spatial Pyramid Pooling is applied, after this fully convolutional layers of feature extraction are computed.

Girchik proposed that in order to use Region of interest pooling layer, or ROI layer, it's a simplified depiction of SPP layer. In Fast RCNN, there is a region of interest pooling and two other modifications have been introduced as well. First, SoftMax classifier is used instead of SVM classifier. Second, multitask training is used to train classifier and bounding box regressor simultaneously. Therefore, faster RCNN moves us forward. The input image and the set of object proposals are supplied to the neural network. The neural network produces a convolutional feature map, from this the feature vectors are extracted using Region of Interest Pooling Layer. Then, the feature vectors affect into a sequence of fully convolutional layers. The output of fully convolutional layers are branched into K-way softmax, and K by four real valued bounding box coordinates output. Fast R-CNN can be trained to this multitask loss. This multi-task loss is the rate of sum of classification loss, and bounding box regression loss. For the true class  $U$ , log loss is used. The smooth L1 loss is used for the bounding box regression. Because there is no separate SVM training, Fast R-CNN allows end-to-end training, it is much faster, and without intensive write and read to the hard drive it's also empirically demonstrated, the detector precision is improved from multi-task learning. During R-CNN training, 128 Region of Interest are sampled from training set at random for each mini batch. Examples are most likely to come from different images. But for the Fast R-CNN, when we use different images in one batch, the computations are expensive for each window, because in Fast R-CNN convolutional features extracted from the whole image, the receptive field for the Region of Interest Pooling is very large. In the worst case, it can be entire image. So for each example, we need to compute convolutional features for the entire image. If each example comes from a different image, then feature extraction as shown if figure 4.7, for Fast R-CNN is much lower than that of R-CNN. So, the following compromise is made. First, they sample a small number of images, for

example two, many examples from each image, for example 64. In this case, the feature computations which are used for 64 examples in the training of Fast R-CNN become much faster compared to the simple R-CNN. Training and test time for the Fast R-CNN is lower than that of R-CNN and SPP net. Accuracy of Fast R-CNN is also higher. The one last weak point of R-CNN is the dependency on external hypothesis generation method. From the table, you can see that most of the test time is attributed to the Selective Search for the proposal generation.

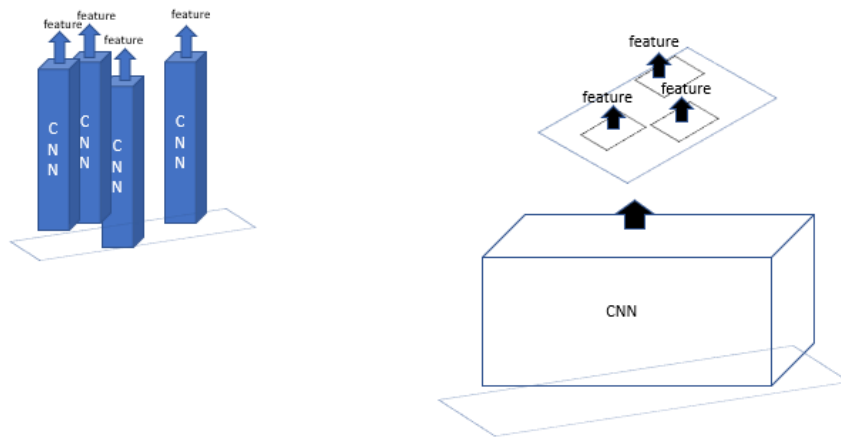


Fig. 4.7 CNN showing feature extraction (Courtesy of [11])

#### 4.2.5. Computer vision and video analysis

Video surveillance cameras are mounted almost everywhere these days. A video is just an older set of frames of the same resolution, usually frame is taken at regular time intervals. When a video process algorithm is constructed, we divide the video into two classes. A video stream is an ongoing video for online processing. In processing video stream, the future frames are unknown. Video sequence is a video of fixed lens, all frames are available at once, and hence the video sequence can be processed as a whole object. A video is a much larger object than an image. A frame width of a

video is usually the line range from 3-5 images per second to 30 or 50 frames per second. The resolution can be up to Full HD or 4K right now. The uncompressed data stream from Full HD video can reach 300 megabytes per second. Thus, super resolution algorithms that can reconstruct higher resolution rely on information that is removed from video during compression. Currently, most video cameras only record video with little or no automatic recognition. The amount of video data is enormous, so it's very difficult to work with such amount without automatic analysis. Most of the video is stored on local drives and is not readily available. Privacy issues also limits video availability, especially when we are working with video surveillance systems. What we receive from the video analysis is that we detect all the interesting objects in the video. In our case we are identifying human pose, there attributes, vehicles and their movements. Some video analysis can process video offline, but for many applications the situational awareness is required. The appearance of the object varies significantly between different viewpoints. For instance, some surveillance cameras are mounted such that they are right above the height of an average human. This allows it to have a higher resolution of them and a clearer quality. On the other hand, some surveillance cameras are mounted on the top of the building to overview the situation and hence each person is seen as a small dot. Currently image recognition and the detection algorithm cannot raise sufficient accuracy and speed simultaneously for both scenarios. Applied video analysis systems are usually obtained only when algorithm is tailored to specific video scenario.

#### **4.2.6. How an object detection algorithm works**

In this research, each object in the image, from a person to vehicle, have been located and identified with a certain level of precision. One of the most basic deep learning approach is Convolutional Neural Networks or CNNs for detecting objects in images.

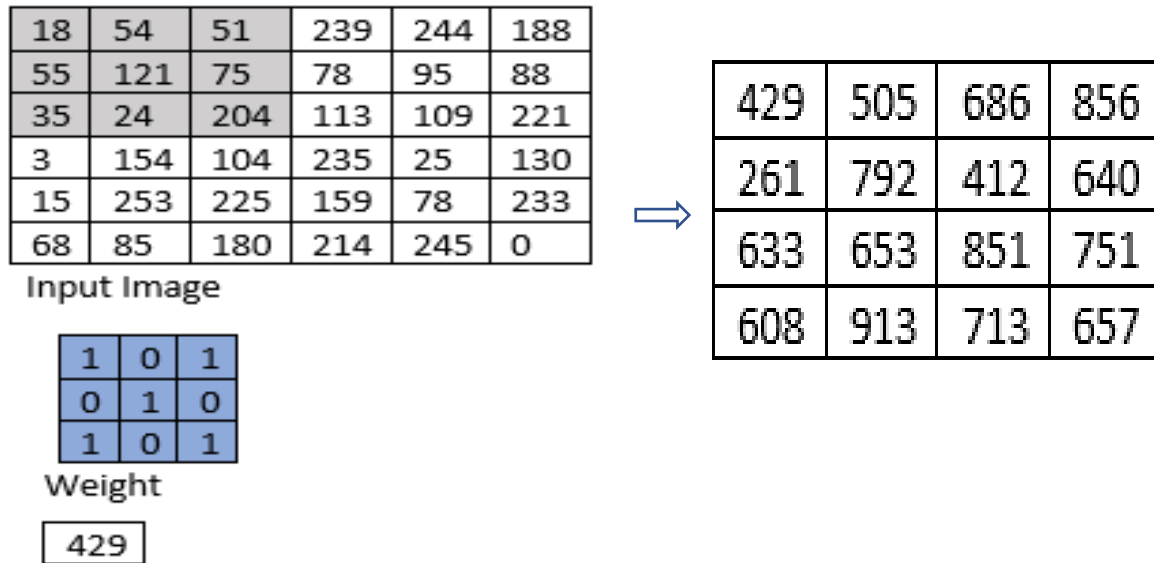
The understanding of how a machine looks at an image is necessary. Every machine is an arrangement of dots (a pixel) arranged in a special order. If the order or the color of the pixel is changed, the image will also change. It basically extracts features from an image by using the spatial arrangement of the images. Any input image can be considered, of which, a weighted matrix is defined, and the input is convoluted to extract specific features from the image without losing the information about its spatial arrangement.

A convolutional neural network basically has three main components:

- i. The convolutional layer
- ii. The pooling layer (which is optional)
- iii. The output layer

#### **4.2.6.1. The convolution layer**

As shown in figure 4.8, for instance, we have an image of size  $6*6$ , a weighted matrix is defined to extract certain features from the images, a  $3*3$  matrix of weight is initialized. This weight now runs across the image such that all the pixels are covered at least once, giving a convolved output. The value 429 above is obtained by adding the values obtained by element wise multiplication of the weight matrix and the highlighted  $3*3$  part of the input image.



**Fig 4.8. Output image from the convolution layer (Courtesy of [11])**

Here, the 6\*6 image is now converted into a 4\*4 image. If the weight matrix is a paint brush painting a wall the brush first paints the wall horizontally and then comes down and paints the next row horizontally. Pixel values are used again when the weight matrix moves along the image. This enables parameter sharing in a convolutional neural network. The weighted matrix behaves like a filter in an image extracting information from the original image matrix. A weight combination might be a particular color, other might be combination of extracting images and another might just blur the unwanted noise. The weights are learnt such that the loss function is minimized similar to a Multilayer perception. Thus, weights are learnt to extract features from the original image which help the network in correct prediction. When we have multiple convolutional layers, the initial layer extracts more generic features, while as the network gets deeper, the features extracted by the weight matrices are more and more suited to the problem at hand.

### 4.2.6.2. Stride and Padding

The weight matrix or the filter move across the entire image moving one pixel at a time. It is a hyper parameter which determines the movement of weight matrix across the image. If the weight matrix moves one pixel at time it is called a stride of 1.

For instance, a stride of 2 would look like as shown in figure 4.9.

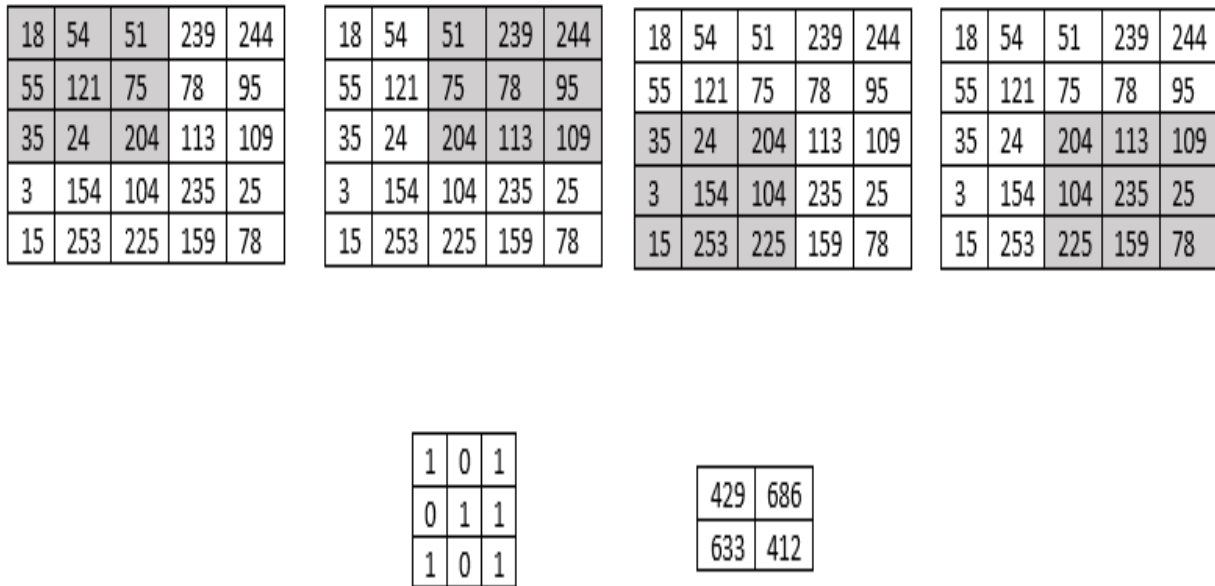


Fig. 4.9. Output image with Stride as 2 (Courtesy of [11])

From the image it is evident that the size of the image keeps on reducing as we increase the stride value. Padding the input image with zeros across, as shown in figure 4.10, solves the problem, and another layer of zeros can be added around the image in case of higher stride values.



0	0	0	0	0	0	0	0
0	18	54	51	239	244	188	0
0	55	121	75	78	95	88	0
0	35	24	204	113	109	221	0
0	3	154	104	235	25	130	0
0	15	253	225	159	78	233	0
0	68	85	180	214	245	0	0
0	0	0	0	0	0	0	0

**Fig. 4.10. Padding (Courtesy of [10])**

The initial shape of the image is retained after we padded the image with a zero. This is known as same padding since the output image has the same size as input.

0	0	0	0	0	0	0	0
0	18	54	51	239	244	188	0
0	55	121	75	78	95	88	0
0	35	24	204	113	109	221	0
0	3	154	104	235	25	130	0
0	15	253	225	159	78	233	0
0	68	85	180	214	245	0	0
0	0	0	0	0	0	0	0

Weight

1	0	1
0	1	1
1	0	1

139	184	250	409	410	283
133	429	506	686	856	441
310	261	792	412	640	341
280	633	653	851	751	317
254	608	913	713	657	503
321	325	592	517	637	78

**Fig. 4.11. Padding to obtain output**

**image of the same size as input image**

**(Courtesy of [11])**

Same padding means that we considered only the valid pixels of the input image. The middle 4\*4 pixels would be the same. Here more information is retained from the borders and we have also preserved the size of the image as shown in figure 4.11.

### 4.2.6.3. The Pooling Layer

When the images are too large, the number of trainable parameters are to be reduced. It is then desired to periodically introduce pooling layers between subsequent convolution layers. Pooling is done for the sole purpose of reducing the spatial size of the image. Pooling is done independently

on each depth dimension; therefore, the depth of the image remains unchanged. The most common form of pooling layer is generally applied in the max pooling as shown in figure 4.12.

429	505	686	856
261	792	412	640
633	653	851	751
608	913	713	657

792	856
913	851

Fig. 4.12. Max pooling (Courtesy of [10])

In this case, the stride is 2, while pooling size is 2 as well. The max operation is applied to each depth dimension of the convolved output. The 4\*4 convolved output is converted to a 2\*2 output after the max pooling operation. The max pooled image retains the information; however, the dimensions of the image might get affected. This helps to reduce the parameters to a great extent.

### Output dimensions:

It is necessary to understand the input and output dimensions after every convolution layer. Three hyperparameters would control the size of the output volume.

- 1) **The number of filters:** the depth of the output volume will be equal to the number of filters applied.
- 2) **Stride:** With a stride of one we move across a single pixel, with higher stride values we move up larger number of pixels and hence produce smaller output volumes.
- 3) **Zero Padding:** It helps to preserve the size of the input image. If a single zero padding is added, a single stride filter movement would retain the size of the original image.

#### 4.2.6.4. The output layer

After multiple layers of convolution and padding, we would need the output for a class. The convolution and pooling layers would only be able to extract features and reduce the number of parameters from the original images. However, to generate the final output, a fully connected layer to generate an output equal to the number of classes is required. Convolutional layers generate 3D activation maps, the output layer has a loss function like categorical cross-entropy, to compute the error in prediction. Once the forward pass is complete the backpropagation begins to update the weight and biases for error and loss reduction. Network showing the entire convolutional layer and working is presented in figure 4.13.

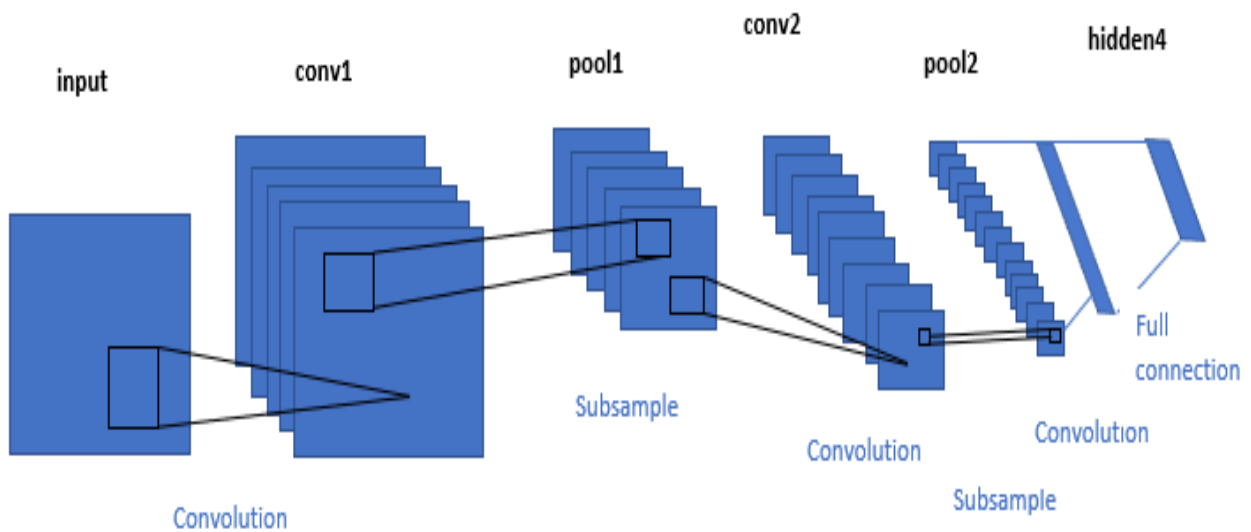


Fig 4.13. Network showing the entire convolutional layer and working (Courtesy of 10)

The entire network looks like this

- An input image is passed to the first convolutional layer. The convoluted output is obtained as an activation map. The filters applied in the convolution layer extract relevant features from the input image to pass further.
- Each filter provides a different feature to aid the correct class prediction. To retain the size of the image, same padding (zero padding) is used, otherwise valid padding is used since it helps to reduce the number of features.
- Pooling layers are then added to further reduce the number of parameters
- Several convolution and pooling layers are added before the prediction is made. Convolutional layer helps in extracting features. Deeper in the network more specific features are extracted as compared to a shallow network where the features extracted are more generic.
- The output layer in a CNN, as mentioned previously, is a fully connected layer, where the input from the other layers is flattened and sent so as to transform the output into the number of classes as desired by the network
- The output is then generated through the output layer and is compared to the output layer for error generation. A loss function is defined in the fully connected output layer to compute the mean square loss. The gradient of error is then calculated.
- The error is then back propagated to update the filter(weights) and bias values.
- One training cycle is completed in a single forward and backward pass.

The image passes to the network, it is then sent through various convolutions and pooling layers. At the end, the image is obtained as an object in the form of object's class. For each input image we get a corresponding class as an output. This technique can also be used to detect various objects

in an image. The problem with using this approach is that objects in the image can have different aspect ratios and spatial locations. For instance, the object might be covering most of the image, while in others the object might only be covering a small percentage of the image. The shapes of the object might also be different for different cases. As a result, many regions resulting in a huge amount of computational time will be required. To solve this problem and to reduce the number of regions, region- based CNN, which selects the regions using a proposal method is used.

## **4.2.7. Region-based convolutional neural network**

### **4.2.7.1. Intuition of RCNN**

Instead of working on a massive number of regions, the RCNN algorithm proposes a bunch of boxes in the image and checks if any of these boxes contain any object. RCNN uses selective search to extract these boxes from an image (these boxes are called regions).

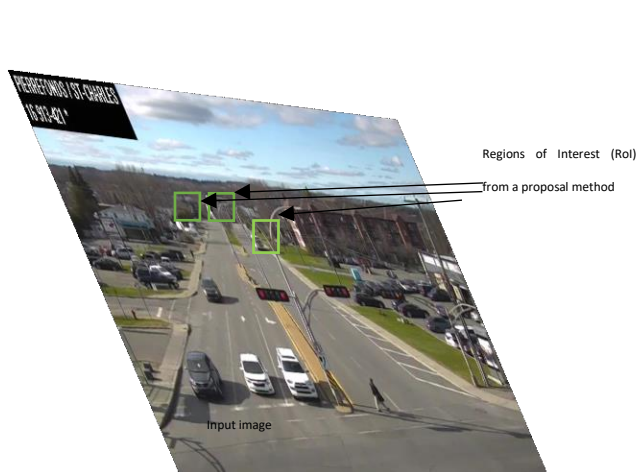
Selective search is used to identify different regions. The four basic regions that form an object are as follows: varying scales, colors, textures and enclosure. Selective search identifies these patterns in the image and based on that, proposes various regions. It first takes the input as an image, then it generates initial sub-segmentations so that multiple regions are generated from the image. The technique further combines the similar regions to form a larger region (based on color similarity, texture similarity, size similarity and shape compatibility)

At the end, these regions then produce the final object locations also called Region of Interest. Following steps are involved in RCNN to detect objects.

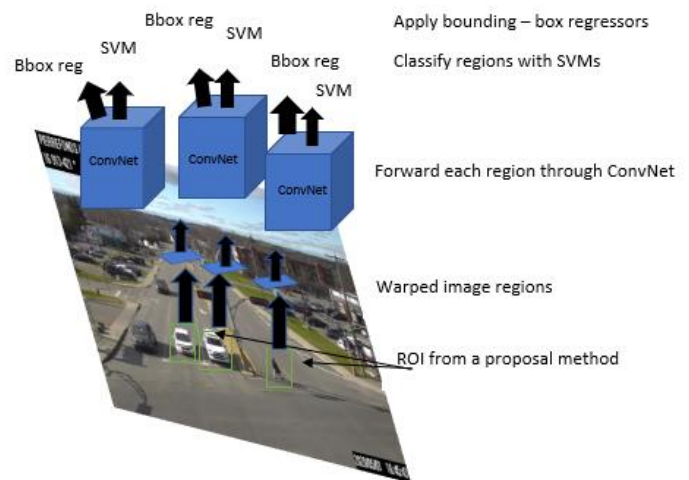
- 1) A pre-trained convolutional neural network is considered.
- 2) The model is retrained, the last layer of the network based on the number of classes that need to be detected are trained.

- 3) Third step is to get the Region of interest for each image. Then, reshaping of all these regions is done to match the CNN input size.
- 4) Once we get the regions, SVM is trained to classify objects and background. For each class, one binary SVM is trained.
- 5) Finally, a linear regression model is generated for tighter bounding boxes for each identified object in the image.

First image is taken as the input as shown in figure 4.14, regions of Interest or (ROI) are obtained using some proposal method. All these regions are then reshaped as per the input of the CNN, and each region is passed in the ConvNet. CNN then extracts features for each region and SVMs are used to divide these regions into different classes. Finally a bounding box regression (Bbox reg) as shown in figure 4.15, is used to predict the bounding box for each identified region. This is how RCNN helps to detect objects



**Fig 4.14. Screen box signifying Region of Interest**



**Fig4.15. Bounding box regressors**

(Courtesy of [11])

While training, we remove the last layers of a fully trained network. An input/output layer is taken for 1000 classes and replace it with 21 classes, where 20 would represent the classes in the PASCAL VOC dataset and the one class would be the background class. Then the warped images are taken and sent through a convolution layer and extract the features for that particular image. Once we have those features, we do SVM classification on top of it and also, we add some kind of a corrective measure to the bounding box coordinate. The images are sent in the form of proposals, so a bounding box regression tries to correct each proposal until it gives a better bounding box for the final image as shown in figure 4.16.

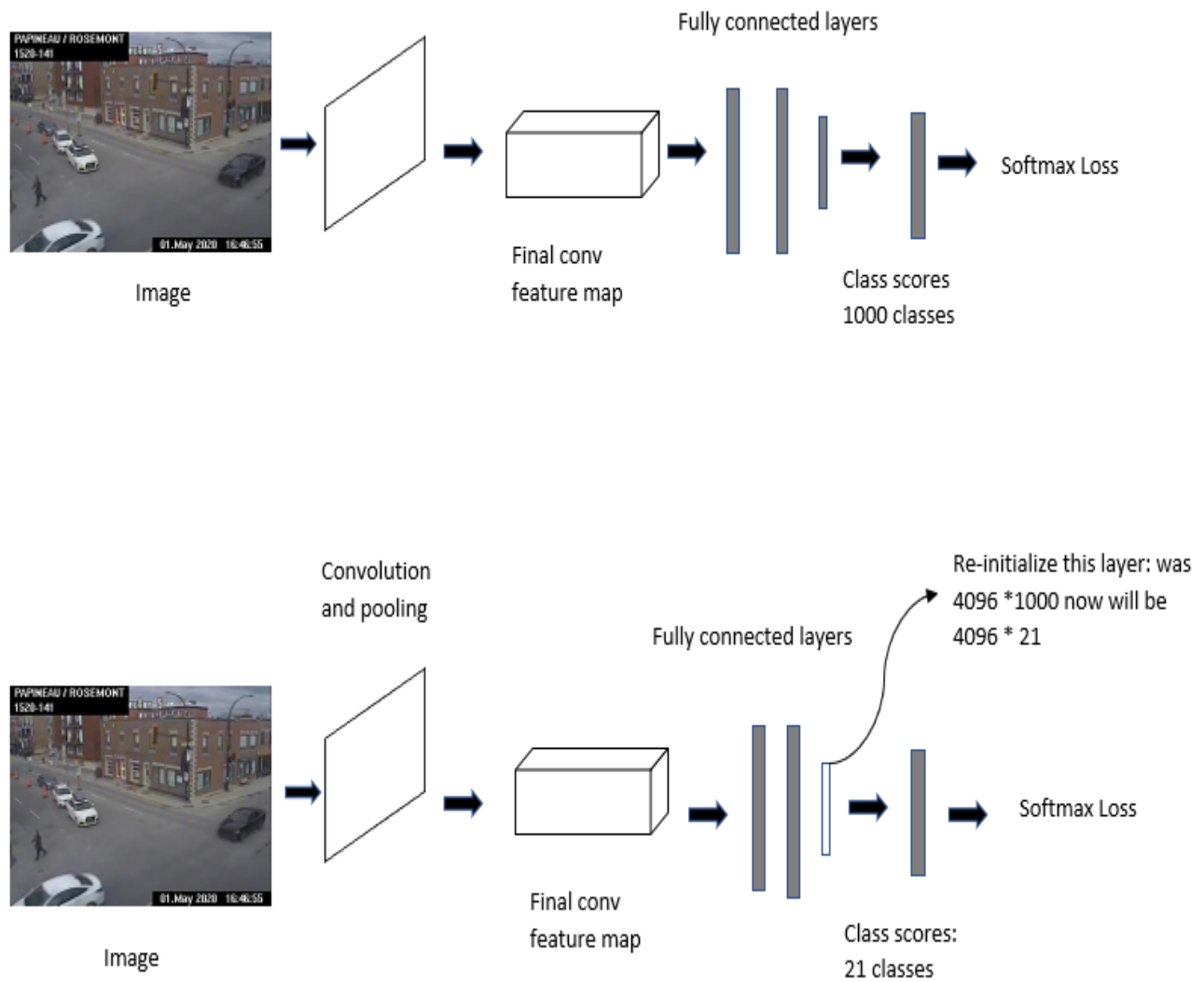


Fig. 4.16. Final Convolutional layer giving SoftMax loss (Courtesy of [11])

#### 4.2.7.2. Problems with RCNN

Although RCNN is helpful for object detection, this technique comes with its own limitations. The RCNN model is quite expensive and slow as it involves extracting 2,000 regions for each image based on selective search, and extracting features using CNN for every image region. If we have  $N$  images, then the number of CNN features will be  $N*2000$ . The entire process of object detection



using RCNN has CNN for feature extraction, Linear SVM classifier for identifying objects, and regression model for tightening the boxes.

All these processes combine to make RCNN very slow. It takes around 40-50 seconds to make predictions for each new image, which essentially makes the model cumbersome and practically impossible to build when faced with a gigantic dataset.

#### **4.2.8. Fast RCNN**

To reduce the computation time a RCNN algorithm typically runs on just one image, instead of running a CNN 2,000 times per image and get all the regions of interest (regions containing some object).

Ross Girshick the author of RCNN, gave the idea of running the CNN just once per image and then finding a way to share that computation across the 2,000 regions. In Fast RCNN, the input image is fed to the CNN, which produces convolutional feature maps. Using these maps, the regions of proposals are extracted, then an RoI pooling layer to reshape the proposed regions into a fix size is used so that it can be fed into a fully connected network. The input image is passed into a ConvNet which in turn generates the Regions of Interest. A RoI pooling layer is applied on all of these regions to reshape them as per the input of ConvNet. Then, each region is passed on to a fully connected network. A SoftMax layer is used on top of the fully connected network to output classes. Along with the SoftMax layer, a linear regression layer is also used parallely to output box coordinates for predicted classes. Hence, instead of using three different models (like in RCNN), Fast RCNN uses a single model which extracts features from the regions, divides them into different classes, and returns the boundary boxes for the identified classes simultaneously.

An input image is taken, the image is then passed to a ConvNet which returns the region of interests accordingly, and then we apply the RoI pooling layer on the extracted regions of interest to make sure all the regions are of the same size. Finally, these regions are passed on to a fully connected network which classifies them, as well as returns the bounding boxes using SoftMax and linear regression layers simultaneously.

This is the process by which Fast RCNN resolves two major issues of RCNN i.e. passing one instead of 2,000 regions per image to the ConvNet and using one instead of three different models for extracting features, classification and generating bounding boxes.

So in this image what we are doing is that we are taking the entire input image, sending it through the network, and getting the convolutional feature map from the convolutional 5<sup>th</sup> layer. These feature maps will be used to generate the regions of interest, so in these regions of interest we calculate the size of the bounding box to the scale of the size of the feature map, so if the bounding box is bigger in the input image, it will be reduced to that particular scale in the feature map. After that we create a smaller region or a warp for the further layers that are fully connected as they will only accept a fixed size input in the proposed regions that have been given from the selective search method, then we create smaller regions that will be fed in to the fully connected layers. Once it is connected to the fully connected layers, we calculate a softmax activation where we try to find out the classes that it will be belonging to for the particular object detected. There is also a linear loss for the bounding box regressions which will try to add that character feature again.

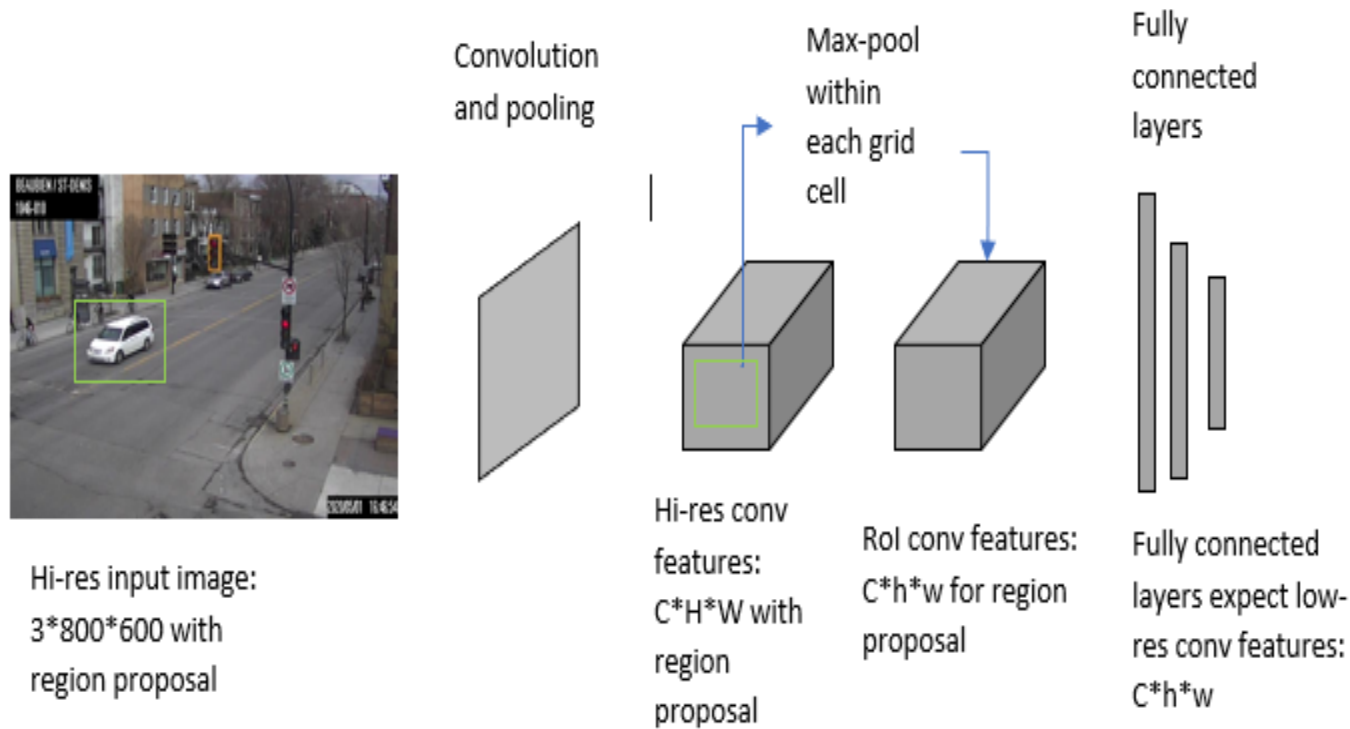


Fig .4.17. Convolution and pooling output (Courtesy of [11])

In the ROI pooling layer as shown in figure 4.17, we calculate the corresponding bounding box for the smaller feature map, and from that particular bounding box of the feature map, we transfer it to a max pool within each grid cell. Essentially, we are trying to reduce the size of different sizes in the feature map region proposed, to a fixed size vector where we have the different activations, which will be sent to the network further to be classified as particular objects.

#### 4.2.8.1. Problems with Fast RCNN

Fast RCNN has certain problem areas. It also uses selective search as a proposal method to find the Regions of interest, which is a slow and time-consuming process. It takes around 2 seconds per image to detect objects, which is much better as compared to RCNN. When large real-life datasets are considered, even a Fast RCNN doesn't look so fast anymore.

The object detection used in this Thesis is Faster RCNN

#### **4.2.9. Faster RCNN**

Faster RCNN is the modified version of Fast RCNN. The major difference between them is that Fast RCNN uses selective search for generating Regions of Interest, while Faster RCNN uses “Region Proposal Network” also known as RPN. RPN takes image feature maps as input and generates a set of object proposals, each with an objectness score as output.

The following steps are followed by a Faster RCNN approach:

An image is taken as the input and passed to the ConvNet which returns the feature map for that image.

Region proposal network is applied on these feature maps. This returns the object proposals along with their objectness score. A RoI pooling layer is applied on these proposals to bring down all the proposals on the same size. At the end, the proposals are passed to a fully connected layer which has a softmax layer and a linear regression layer at its top to classify, and output the bounding boxes for objects.

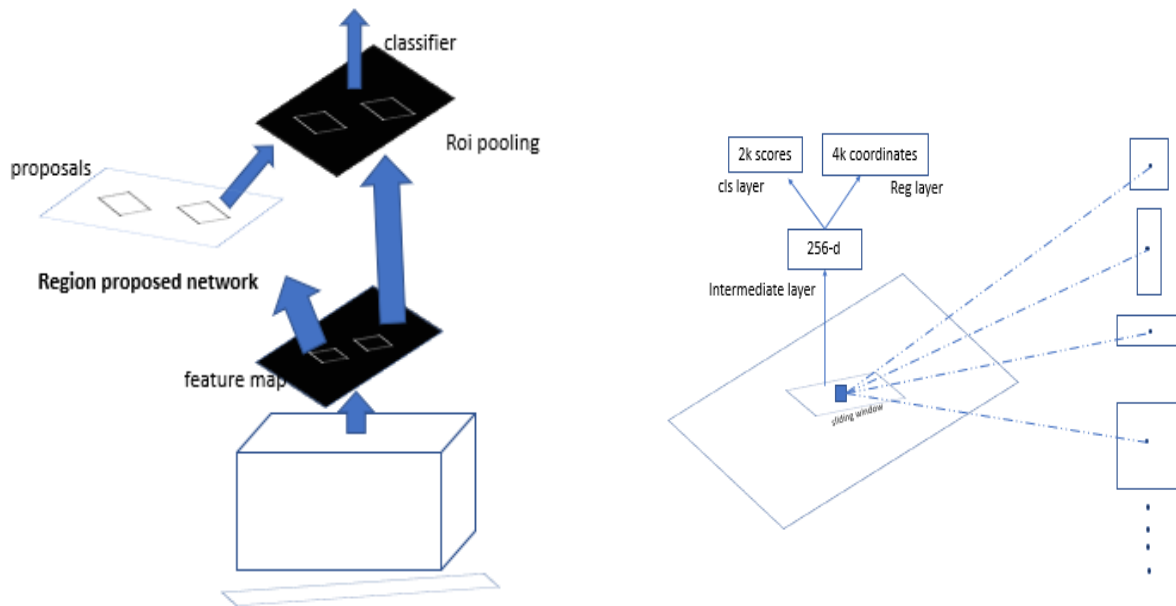


Fig. 4.18. RPN (Courtesy of [11])

Faster RCNN takes the feature maps from CNN and passes them on to the Region Proposal Network. RPN as shown in figure 4.18, uses a sliding window over these feature maps and at each window, it generates  $k$  Anchor boxes of different shapes and sizes. Anchor boxes are fixed sized boundary boxes that are placed throughout the image and have different shapes and sizes. For each anchor, RPN predicts two things:

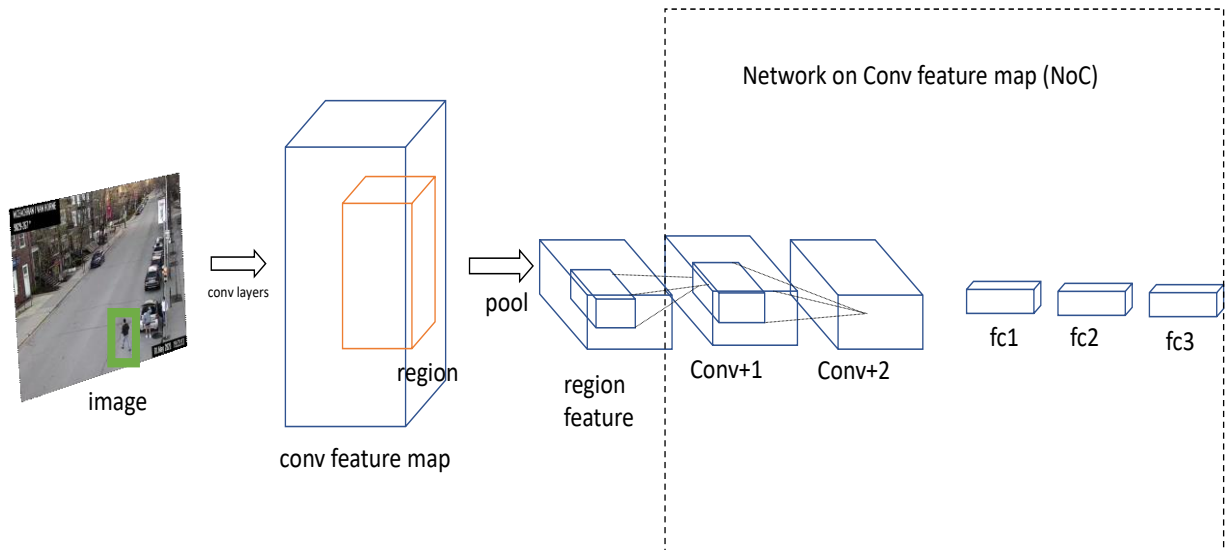
The first is the probability that an anchor is an object (it does not consider which class the object belongs to), second is the bounding box regressor for adjusting the anchors to better fit the object.

So, what RPN does is that on the feature map that have been generated after the convolutional layers we slide a window upon it, then we create  $k$  anchor boxes. Now, anchor boxes are the proposed boxes that are generated by the RPN or the region proposal network.

Bounding boxes of different shapes and sizes which are passed on to the RoI pooling layer are obtained. After the RPN step there are proposals with no classes assigned to them. Each proposal

can be cropped so that each proposal contains an object. This is what the RoI pooling layer does. It extracts fixed size feature maps for each anchor. Further, these feature maps are passed to a fully connected layer which has a SoftMax and linear regression layer. It finally classifies the object and predicts the bounding boxes for the identified objects. There is a classification loss from the region proposed networks which tells if there is an object or not. There is a bounding box proposal loss, and then there is an object loss from the classification itself where we try to determine which class the object belongs to, and then there is another bounding box loss on the top where we try to define where the bounding box have been created by the RPN. Faster R-CNN

Faster R-CNN = Fast R-CNN + Region Proposal Network (RPN), the dependency from external hypothesis generation is removed. Objects are identified in a single pass with a single neural network. RPN serves as a proposal generator. It is similar to Faster RCNN as it is a fully convolutional network that is trained to its multitask class. By decoding a single pixel or one response at a single pixel the object outline can be roughly seen. Much finer localization information has been encoded in the channels of convolutional feature response; this information can be extracted from localization objects with RPN, which is a small network that is slid over convolutional feature map. RPN simultaneously classify corresponding region as object or unknown object in the regress bounding box location.



**Fig. 4.19. Faster RCNN (Courtesy of [11])**

In reference to the image the sliding window provides localization information. Finer localization is provided in reference to the sliding window by box regression.

A set of object proposals is defined at each sliding window. Each such proposal has different size and aspect ratio called anchors. These anchors improve handling of objects of different size and aspect ratio and is a variant of sliding window with different sizes and aspect ratio.

When Intersection over union is marked as qualitative, IoU is larger than 0.7, if IoU is lower than 0.3, anchor is marked as negative sample. Faster R-CNN as shown in figure 4.19, can be trained end to end as one network with four losses: RPN classification loss, RPN regression loss, Fast R-CNN classification loss over classes, Fast R-CNN regression loss to regress the proposal bounding box.

#### **4.2.9.1. Problems with Faster RCNN**

The object detection algorithm discussed use regions to identify the objects. The network does not look at the complete image in one go but focusses on parts of the image sequentially.

This creates two complications:

- 1) The algorithm requires many passes through a single image to extract all the objects.
- 2) As there are different systems working one after the other, the performance of the systems further ahead depends on how the previous systems performed.

Comparison of the characteristics of CNN, RCNN, Fast RCNN, and Faster RCNN is tabulated in table 4.1.

Algorithm	Features	Speedup, Map (VOC 2007) Prediction time/ image with proposals	Limitations
CNN	Divides the image into multiple regions and then classify each region into various classes.	-	Needs a lot of regions to predict accurately and hence high computation time.
RCNN	Uses selective search to generate regions. Extracts around 2000 regions from each image. It is the slowest because independently all the proposed regions are sent to the Convolutional Neural Network and the advantage of RoI pooling layer is not taken.	1x 40-50 seconds 66.0	High computation time as each region is passed to the CNN separately also it uses three different model for making predictions.
Fast RCNN	Each image is passed only once to the CNN and feature maps are extracted. Selective search is used on these maps to generate predictions It combines all the three models used in RCNN together. Here we don't use RPN that is why it is a bit slower because selective search approximately 2 seconds	25x 2 seconds 66.9	Selective search is slow and hence computation time is still high.

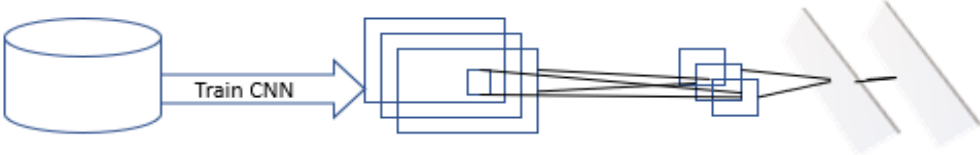


Faster RCNN	Replaces the selective search method with region proposal network which made the algorithm much faster. It is the fastest among the three, this is because it uses RPN and selective search	250x 0.2 seconds 66.9	Object proposal takes time and as there are different systems working one after the other, the performance of systems depends on how the previous system has performed.
-------------	---	-----------------------------	---

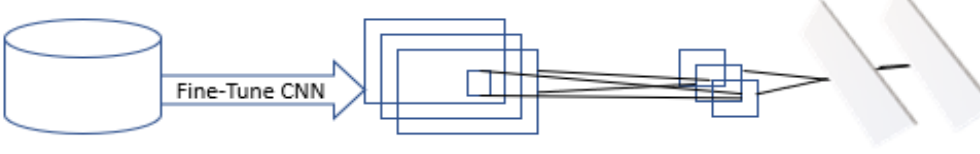
Table 4.1. Comparing the characteristics of CNN, RCNN, Fast RCNN, and Faster RCNN

### 4.2.9.2 RCNN Training

1. Pre-train CNN for image classification



2. Fine-tune CNN object detection



3. Train linear predictor for object detection



Fig. 4.20. RCNN Training (Courtesy of [11])

Understanding the contents of an image as shown in figure 4.20, is one of the core problems for computer vision. Classification of an image attributes can be distributed into two broad categories, namely Binary classification and multiclass classification. For identifying whether the image has a pedestrian or a vehicle individually it needs to answer two different questions.

Is this a pedestrian? If  $y \in \{0,1\}$ , 0- no, 1- yes

Is this a vehicle? If  $y \in \{0,1\}$ , 0- no, 1- yes

On the other hand, Multiclass classification answers that which of the following category does the class belongs to, that is for a class  $D$  for instance the class  $c \in \{1, \dots, S\}$

Machine learning classification is a two-stage pipeline namely: extraction and encoding of meaningful features from the pixels of the images. The next stage involves classification in the feature space that is extracted by the image. The key question that is answered by machine learning algorithm is the key features that we need to extract out of the image. Human brain has a region called fusiform region; human visual cortex has neurons that fire when observing a bar oriented at a specific direction irrespective of the different positions of the bar within the visual field.

### **4.3. Conclusion**

Before delving into the implementation methodology of our proposed system, it was crucial to understand how computer vision works and the various concepts involved in it. We explored the basic concepts behind neural networks. We analyzed the different types of neural networks, their advantages and shortcomings (which was crucial in identifying the best fit for our proposed system), and also learnt how computer vision and image/video analysis takes place. Hence, through this chapter we obtained in-depth knowledge of neural networks and object detection algorithms to

better understand their functioning, and thus apply them in our own proposed system in the following chapters.

# Chapter 5

## Implementation Methodologies

### 5.1. Introduction

This chapter provides a detailed description of how our proposed solution works. The proposed system first involves acquiring pedestrian and vehicle data in order to obtain densities at different intersections during peak and non-peak hours. After the acquisition of the data, we use neural networks to train, test, and detect objects on this data. The objects detection results are then analyzed and visualized for an in-depth understanding of the risks posed to pedestrian safety. The detailed visualization of these results is then followed by the implementation of a cloud-based solution that allows alerts issuance and overall traffic system management for enhancing pedestrian safety. The implementation methodologies are explained stepwise along with explanations of technologies used for each step of the implementation.

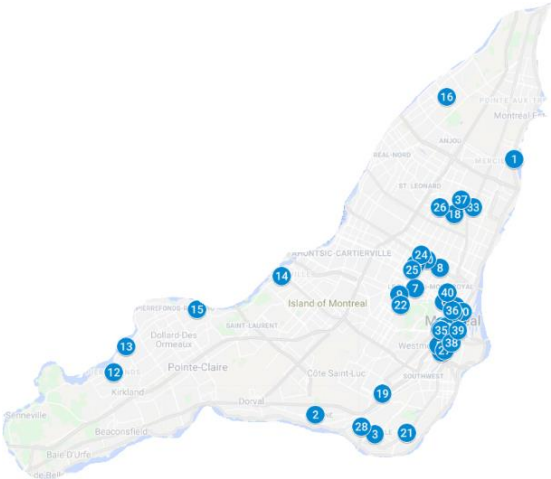
The proposed solution involves the following:

1. A collection of Traffic and Pedestrian data images and videos taken over a whole month (before and after COVID 19) to analyze the pedestrian densities and peak hours.
2. Training Neural Networks to extract pedestrian and vehicle information from the collected data.

3. Visualization and in-depth analysis of the Pedestrian and Traffic data to derive patterns of peak and non-peak hours, high risk intersections, etc.

## 5.2. Step 1: Acquisition of Data

Data collected for some dangerous intersections in Montreal along with their locations for this study is tabulated in table 5.1 as tabulated below.

<p>Through the following figures it can be observed that pedestrians are heavily impacted in terms of accidents monthly.</p>  <p>The month of November is one of the most dangerous for pedestrians. About 25 fatal accidents can be observed in the month of February, June, August, October, November. In the month of October about 200 accidents were observed. Number of Bike accidents were maximum in the month of June going up to 700.S.no</p>	Name of the intersection	Latitude	Longitude
1	81e Avenue et Rue Notre-Dame	45.60437	-73.5126
2	32e Avenue et rue Provost	45.44191	-73.6927
3	90e Avenue et boulevard Newman	45.42918	-73.6389
4	Avenue Atwater et rue Notre-Dame	45.48129	-73.5779
5	Avenue Atwater et rue Saint-Antoine	45.48551	-73.5807

6	Avenue des Pins et Rue Saint-Urbain	45.51381	-73.5762
7	Avenue Dollard et boulevard Newman	45.43081	-73.6347
8	Avenue du Mont-Royal et avenue du parc	45.53461	-73.5731
9	Avenue du Parc et avenue Saint-Viateur	45.5222	-73.6022
10	Avenue Papineau et boulevard Saint-Joseph	45.53521	-73.5794
11	Avenue McEachran et avenue Van horne	45.51835	-73.6169
12	Avenue Papineau et boulevard Rosemont	45.54068	-73.5913
13	Boulevard De Maisonneuve et Boulevard Saint-Laurent	45.51059	-73.5653
14	Boulevard de Pierrefonds et boulevard Saint-Charles	45.46885	-73.8755
15	Boulevard Gouin et boulevard Jacques-Bizard	45.48508	-73.8648
16	Boulevard Gouin et boulevard Laurentien	45.52969	-73.7233
17	Boulevard Gouin et Sortie A-13 (int. Est) boulevard Pitfield	45.50864	-73.7999
18	Boulevard Maurice-Duplessis et boulevard Rodolphe-Forget	45.64379	-73.5732
19	Boulevard Robert-Bourassa et rue Saint-Antoine (inter. ouest)	45.49959	-73.5635
20	Boulevard Robert Bourassa et Rue Sherbrooke	45.50551	-73.5736
21	Boulevard Rosemont et rue Viau	45.56922	-73.5664
22	Boulevard Sainte-Anne-de-Bellevue et rue Saint-Jacques	45.45555	-73.6317
23	Boulevard Saint-Laurent et avenue Viger	45.50753	-73.5587
24	Boulevard Shevchenko et boulevard De La Verendrye	45.43014	-73.6097
25	Chemin de la Cote-Sainte-Catherine et avenue Vincent-d'Indy	45.51174	-73.6155
26	Rue Beaubien et avenue Christophe-Colomb	45.53785	-73.6015
27	Rue Beaubien et avenue Papineau	45.54317	-73.5967
28	Rue Beaubien et rue Saint-Denis	45.53401	-73.605
29	Rue Belanger et rue Viau	45.57351	-73.5799
30	Rue Charlevoix et rue Notre-Dame	45.48264	-73.5757
31	Rue Clement et rue des Oblats	45.43413	-73.6509

32	Rue Crescent et boulevard De Maisonneuve	45.49806	-73.5775
33	Rue De Bleury et Rue Sainte-Catherine	45.50631	-73.5669
34	Rue de la Gauchetiere et rue Peel	45.49777	-73.5687
35	Rue de la Montagne et Rue Sainte-Catherine	45.49789	-73.575
36	Rue Dickson et Rue Sherbrooke	45.57355	-73.549
37	Rue Guy et boulevard De Maisonneuve	45.4957	-73.5794
38	Rue Guy et Rue Sainte-Catherine	45.49506	-73.5779
39	Rue Jeanne-Mance et avenue du President-Kennedy	45.50838	-73.5687
40	Rue Lacordaire et boulevard Rosemont	45.57868	-73.56
41	Rue Notre-Dame et rue des Seigneurs	45.48769	-73.569
42	Rue Notre-Dame et rue Peel	45.49543	-73.563
43	Rue Roy et rue Saint-Denis	45.51925	-73.5726

**Table 5.1. Names of dangerous intersections in Montreal along with their locations for this study**

The city of Montreal has monitoring cameras installed on all major streets of the city. The images from these cameras are constantly uploaded onto their website after short time intervals. In our framework this data is used. The image data obtained from these monitoring cameras is used to train the Convolutional Neural Network to identify pedestrians. To successfully pull the data constantly from this website a python program was developed which saved the images and categorized them based on the Street name, timestamp, and date. A few of the images obtained are shown in the figure 51 below.

The acquired images were used to train the CNN to not only detect pedestrians, but also to detect any distorted images that could then be discarded. The TensorFlow high-level Keras API was used to create the classification model for these images. The first step was to identify and discard distorted images, after which they could be used to train the CNN for pedestrian identification.



**Fig. 5.1. Some sample images of our data set before classification**

These are a few images as shown in figure 5.1, that are captured and collected from the cameras at different traffic signals. There are around 500 cameras in the Montreal area and the majority of them are functional 24/7. There is periodic maintenance which makes the camera unavailable and a message will be shown instead of the footage. The website of Ville Montréal states that these cameras update the images every five minutes, however this happens after every eight minutes. For the training process around 10,000 images were collected.

Each column in table 5.2 shows different type of information that can be deduced from the image.




Name	Description
time	Time of the day
date	Date
location	Location of the Camera
direction	The direction which camera is looking
Cars, trucks, busses	Number of vehicles in the image
pedestrians	Number of pedestrians

**Table 5.2. Information that can be deduced from the images**

Following collected as shown in figure 5.2, is a list of images we expect to get by collecting data from traffic cameras.

Fig. 5.2. Images collected from traffic cameras

Normal Image in day	 <p>GUY / STE-CATHERINE 317-275</p> <p>22.Mar 2020 10:57:25</p> <p>A clear, wide-angle daytime view of a city street intersection. A white car is driving in the center lane, and a red car is visible further down the road. Pedestrians are walking on the sidewalks. The scene is well-lit with shadows cast on the pavement.</p>
Normal Image in night	 <p>PINS / ST-URBAIN 547-013 *</p> <p>2020/05/06 21:05:47</p> <p>A clear, wide-angle nighttime view of a city street intersection. The scene is illuminated by streetlights, creating bright spots and long shadows. A white car is driving in the center lane, and other vehicles are visible in the distance.</p>
Distorted Image	 <p>172.25.160.15 - Camera - 01 871-345</p> <p>The image shows a street scene that has been severely distorted by digital artifacts. The entire image is covered in vertical, wavy lines that obscure the original content. Only faint shapes and colors are visible through the noise.</p>

Blurred image by fog



Blurred image by rain



Image cannot be displayed



### **5.3. Step 2: Training Data**

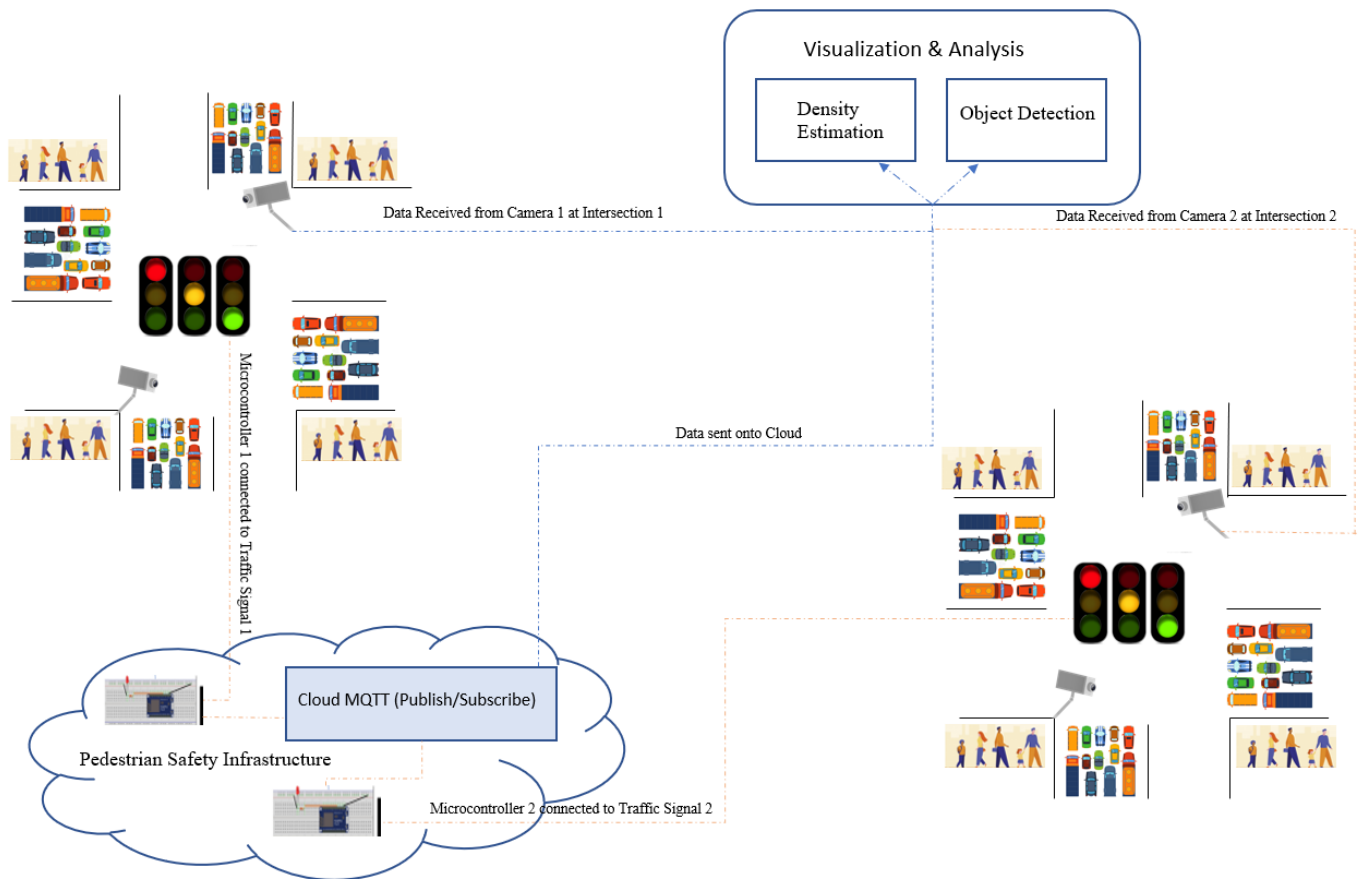
To train and evaluate the model we divided our image data set into Train and Test dataset with a ratio of 90:10 correspondingly. Around 3000 images were pulled from the website and used to train the CNN. The images comprised of different angles, and different roads during different times of the day. The data was divided into training and testing data. 90% of the images were used for training the CNN, and the remaining 10% were used as test to check the classification capabilities of the trained CNN. The CNN was trained solely to identify pedestrians and vehicles. The output image would consist of a box placed around the pedestrians identified. The number of pedestrians and vehicles identified, street name, and the timestamp are then saved in a separate spreadsheet.

This data is then used to draw patterns, pedestrian densities, and any other required data. In order to mark the mentioned elements in the image, we used an open-source application called labelImage [ref]. Label image is available for every OS and helps to mark the features by drawing boxes around each element in the image with a smooth user-friendly UI. This tool will create an XML file alongside the image with the location of each bounding box coordinate. We have to make sure that we completely labeled everything and also labeled them correctly. After creating an XML file containing the coordinates of the bounding boxes for all the train and test set images, we are ready to train a model with Tensor Flow.

### **5.4. Step 3: Connecting with Cloud**

The analyzed data present in the spreadsheet is constantly sent onto the Cloud. This data is then also used to control the traffic signals. The ESP8266 NodeMCU development Boards connected to

the traffic lights can be used to achieve this purpose. The micro-controller will receive messages from the Cloud database via the Cloud MQTT protocol and thus act accordingly. This could involve controlling the traffic signals by increasing or decreasing the timers, flashing the lights on the pedestrian paths when a pedestrian crosses by sensing motion through the PIR motion sensor, or any other required control. In our system prototype, we used the ESP8266 NodeMCU development board to receive the messages from the cloud and turn on/off their attached LEDs thus mimicking a traffic system as shown in the figure 5.3.



**Fig. 5.3. Architecture showing the working of the proposed model**

## 5.5. Outputs and Results

Once the training is done, we can run our model on the collected images from Ville Montreal Cameras. Results are images with bounding boxes with different colors and titles on them marking different types of vehicles alongside the score of the confidence level of the model regarding the detected object in percentage. The Figure 5.4 (a, b, c) shows images that are a few examples of the output result of the image detection model.

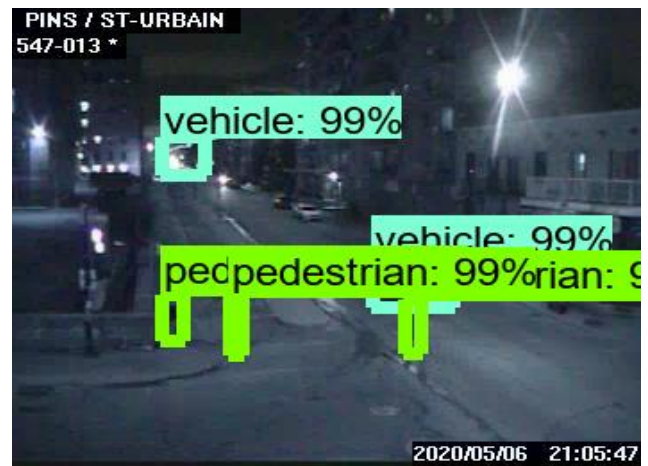
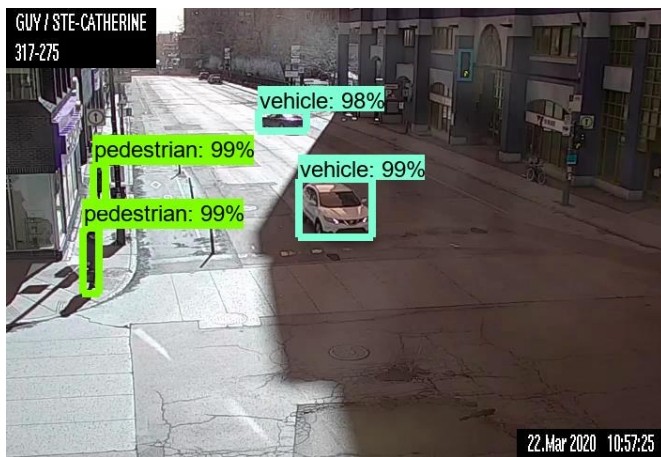


Fig 5.4 (a). Neural network output images of our data set

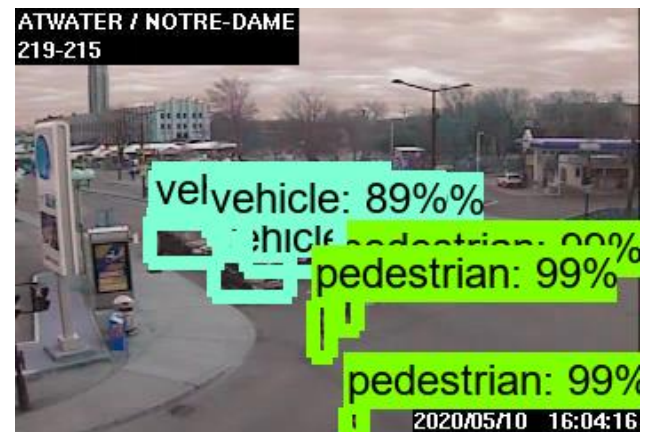
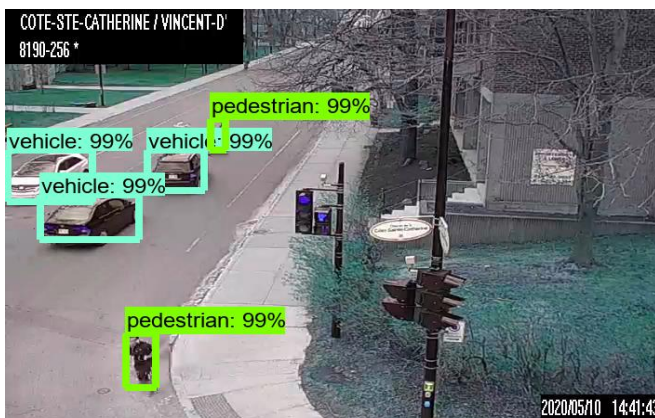


Fig 5.4 (b). Neural network output images of our data set

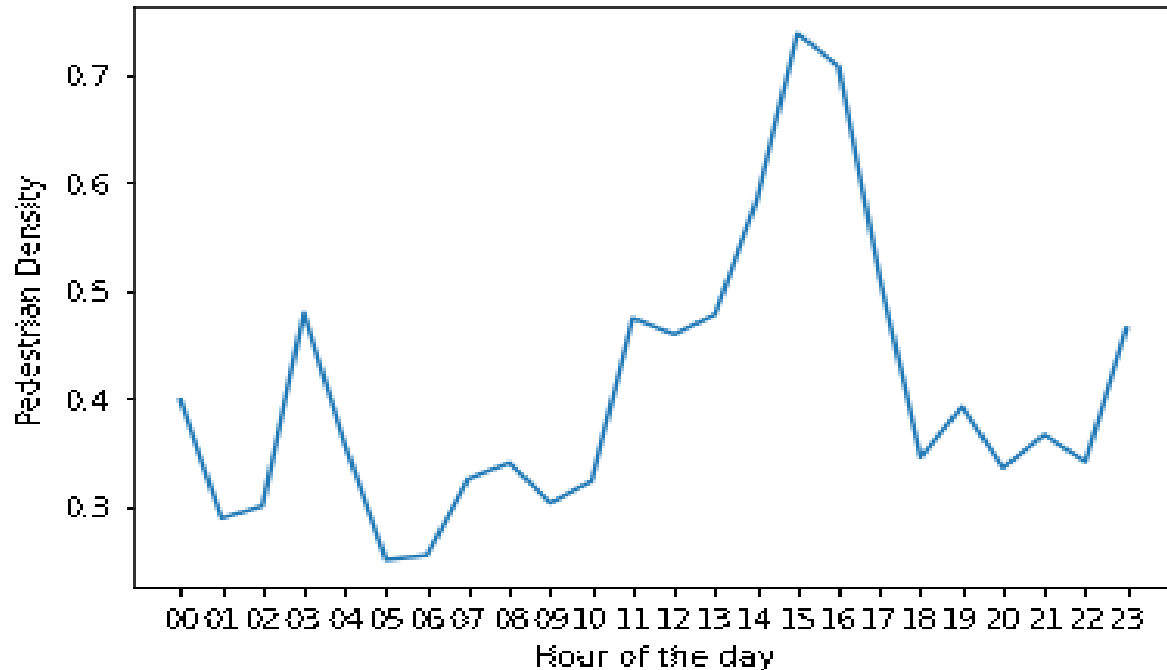


Fig 5.4 (c). Neural network output images of our data set

## 5.6. Visualization and Analysis of Data

### 5.6.1. Pedestrian analysis

Following the recognition of the pedestrians and vehicles at the intersections, an analysis for the number of pedestrians and vehicles on the subsequent intersections was performed. For the intersection: 32e Avenue et rue Provost, this graph plots the average number of pedestrians at a particular hour for all the dates as shown in figure 5.5 of which the data set comprises. It can be seen that the peak hour at which maximum pedestrians are present for this intersection is between 2 pm and 5 pm, and also a considerable pedestrian presence can be observed between 11am-1pm.

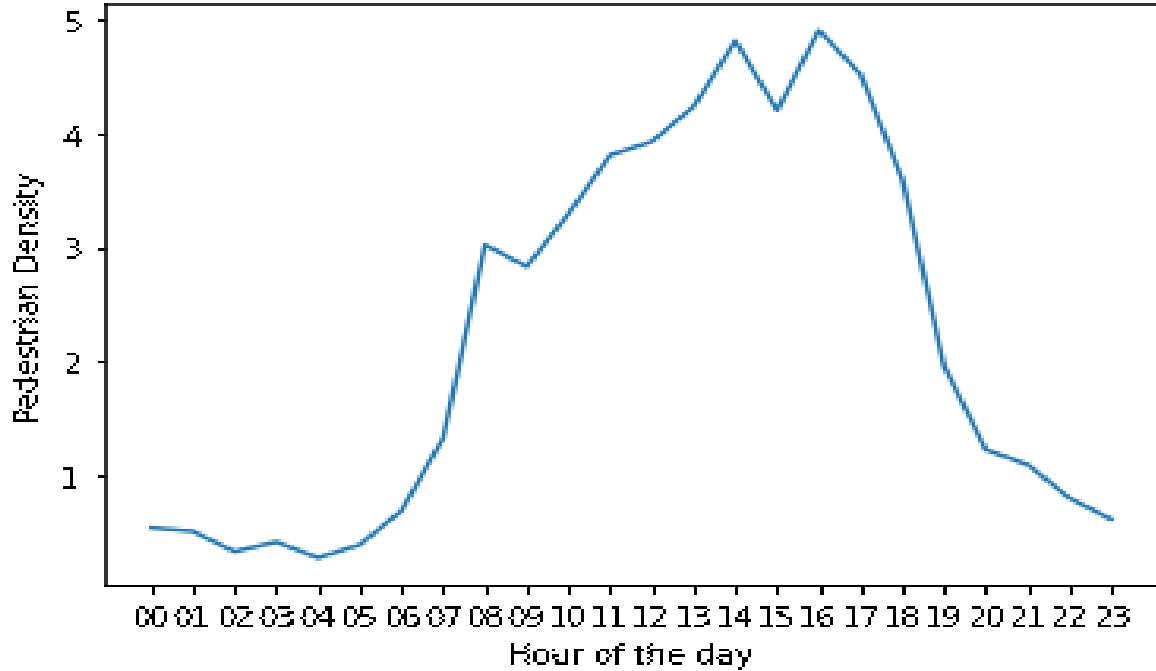


**Fig .5.5 Graph showing average number of pedestrians vs hour of the day for 32e Avenue and rue Provost intersection**

For the street Boulevard Robert-Bourassa et Rue Sherbrooke, the graph in figure 5.6 shows the average pedestrian counts during different times of the day, for all the dates for which the data set is taken.

For this intersection the graph clearly depicts that the peak hours at which pedestrian count is the highest is between 1pm and 6 pm. From 8am onwards the average number of pedestrians start to rise and start declining after 6pm. This intersection thus has a lot of pedestrians throughout the day, and the chances of occurrences of accidents involving pedestrians at this intersection are quite high.





**Fig. 5.6. Graph showing average number of pedestrians vs hour of the day for Boulevard Robert-Bourassa and Rue Sherbrooke intersection**

For the intersection: Rue Guy et Rue Sainte-Catherine as shown in figure 5.7, this graph shows the average number of pedestrians observed throughout the day. Again, it can be easily observed that the peak hours at which maximum pedestrians can be observed lie between 3 pm and 7 pm. The number of pedestrians start rising at 7 am and continue to peak from there on until 6-7pm, after which there is a sharp decline in the pedestrian count up until the late hours of the night.

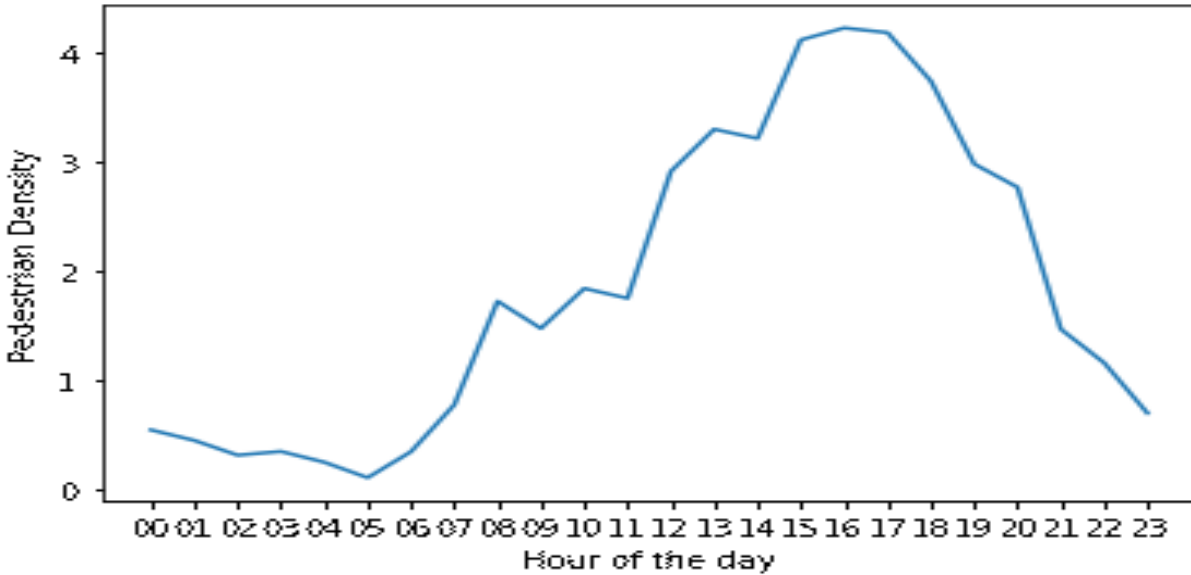


Fig 5.7. Graph showing average number of pedestrians vs hour of the day for Rue Guy and Rue St. Catherine intersection

Similarly, for the intersection Rue Guy et Boulevard de Maisonneuve as shown in figure 5.8, the peak hours at which average pedestrian count is maximum is between 3pm and 6pm. Between 8 am and 6 pm the pedestrian count rises gradually and declines rapidly before and after these timings. The pedestrians are highest at risk during the peak hours (3-6pm).

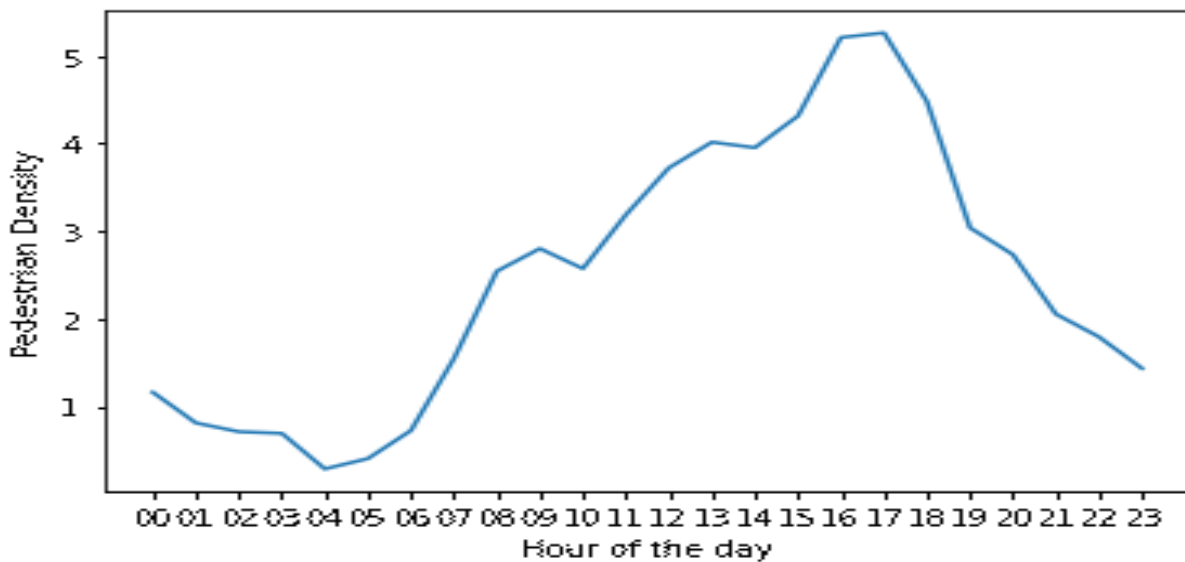
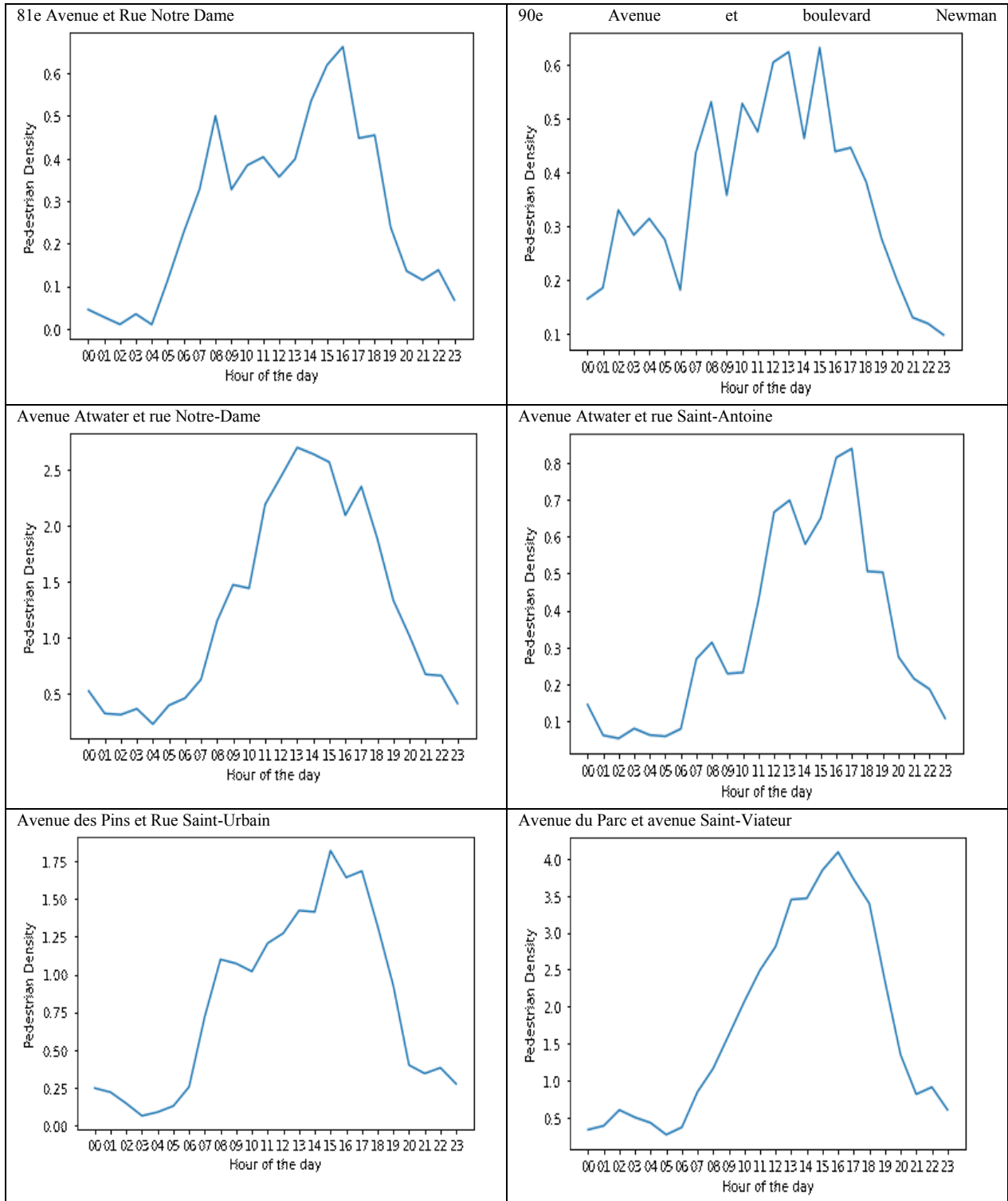
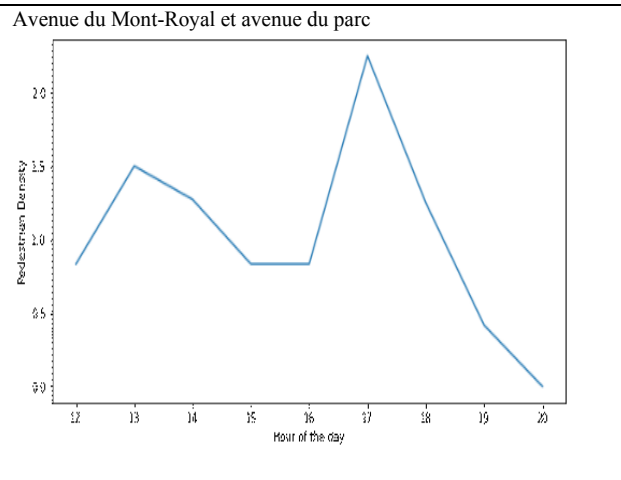
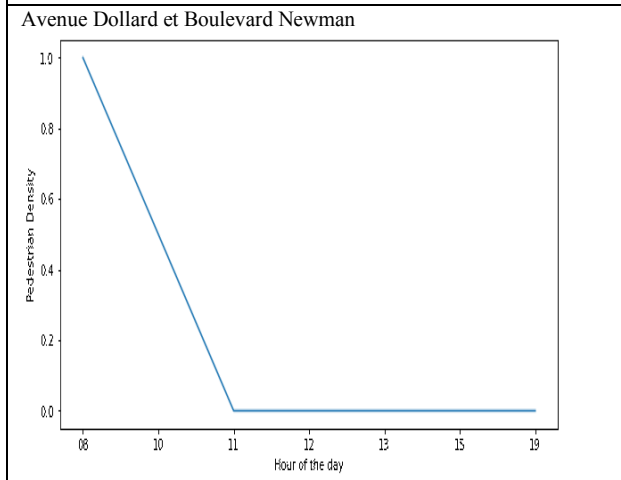
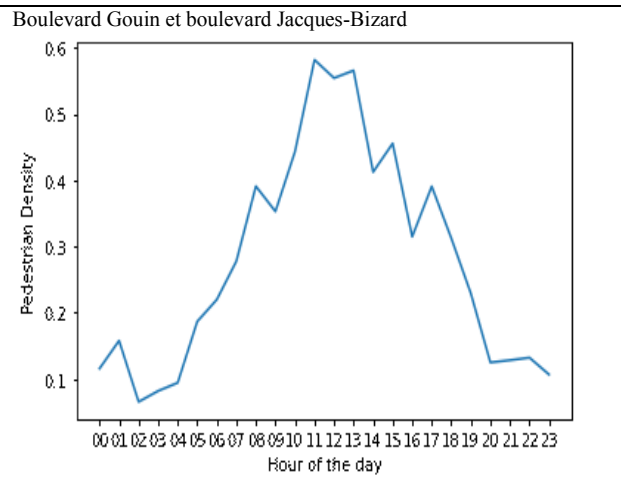
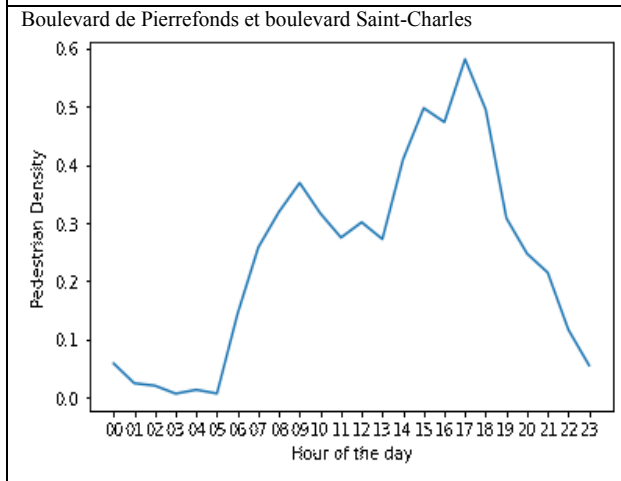
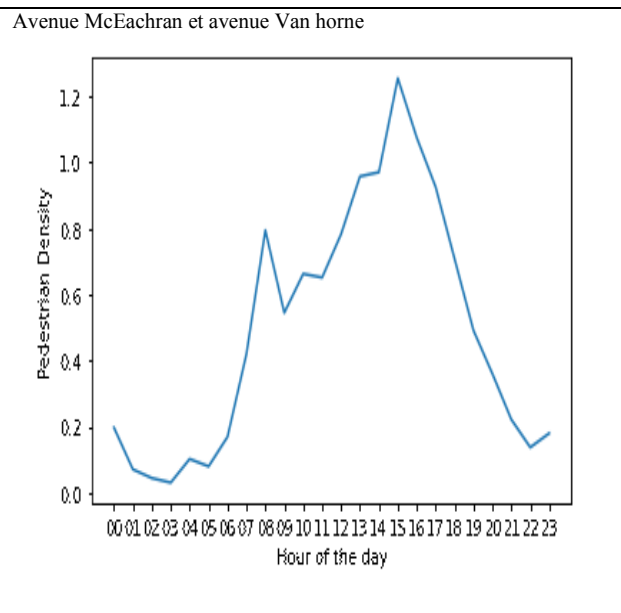
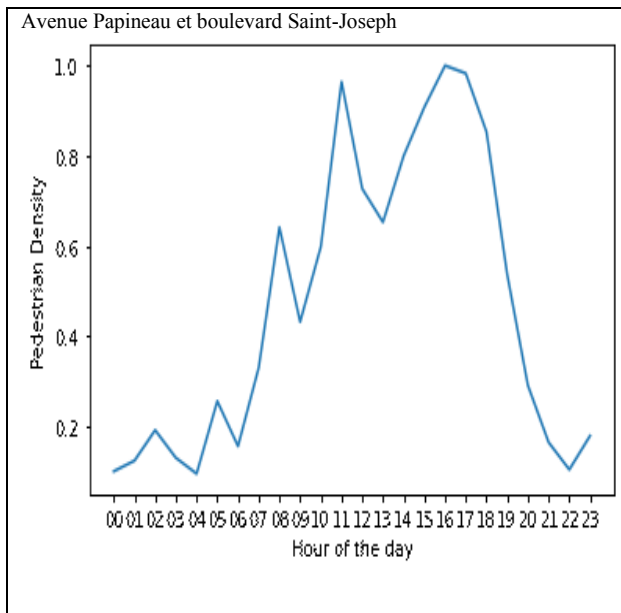


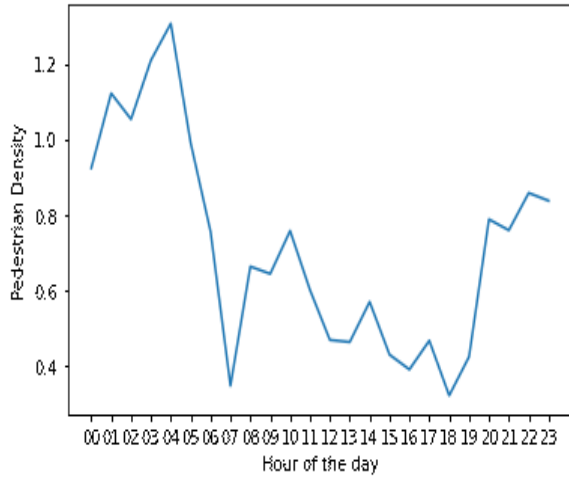
Fig 5.8. Graph showing average number of pedestrian vs hour of the day for Boulevard de Maisonneuve and Rue Guy intersection

Similarly, the average pedestrian count vs hour of the day graphs for the other intersections in the data set are shown in figure 5.9:

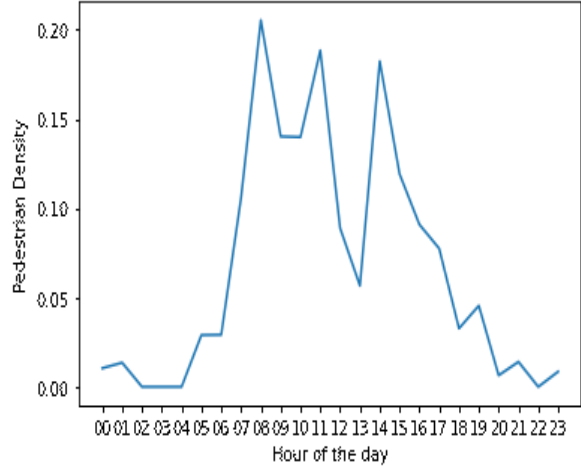




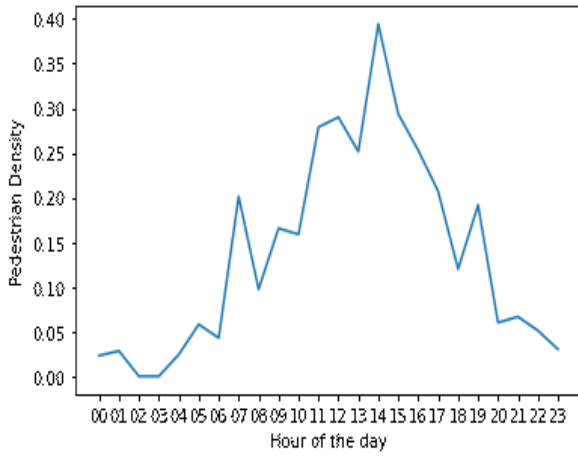
Boulevard Gouin et boulevard Laurentien



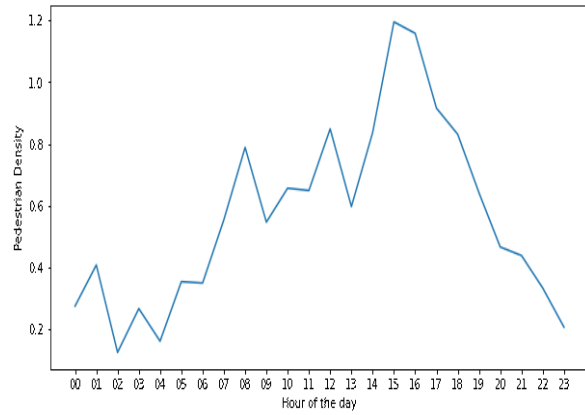
Boulevard Gouin et Sortie A-13 (int. Est) boulevard Pitfield



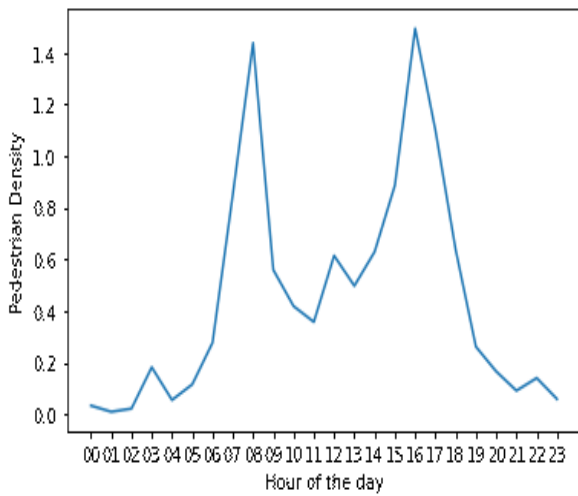
Boulevard Maurice-Duplessis et boulevard Rodolphe-Forget



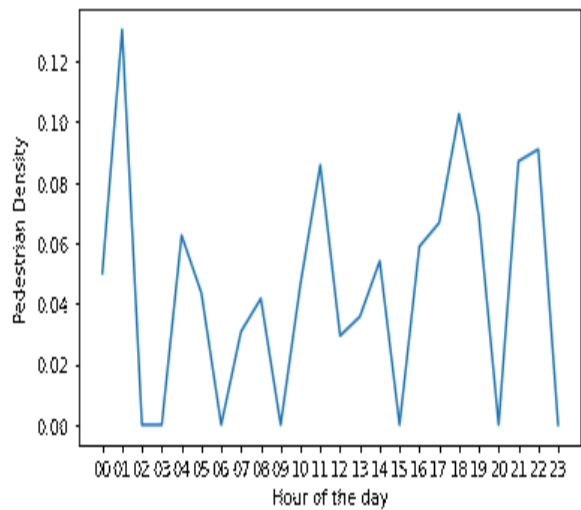
Chemin de la Cote-Saint-Catherine et Avenue Vincent-d'Indy

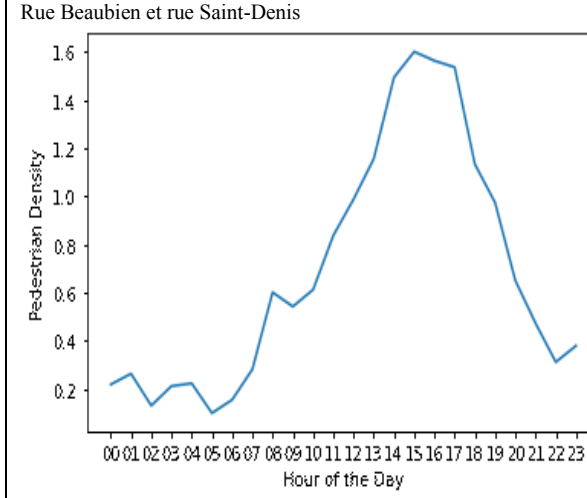
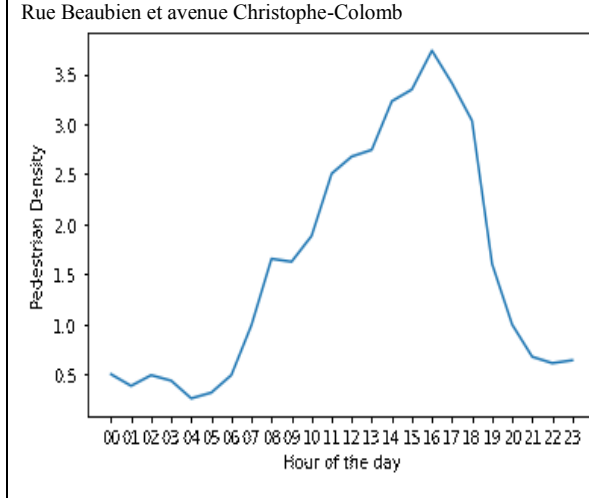
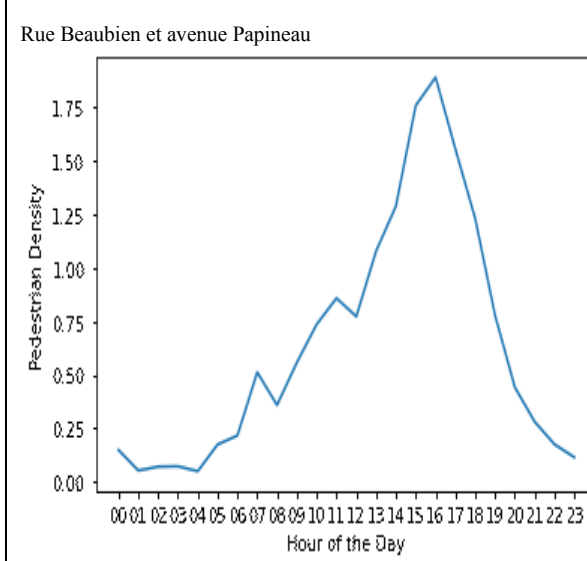
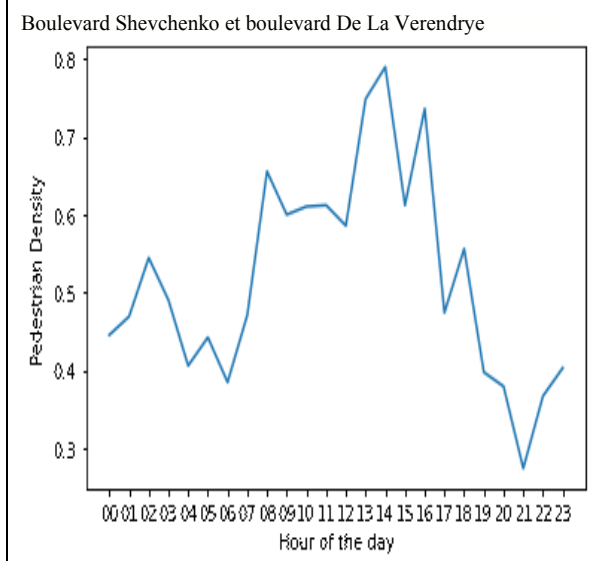
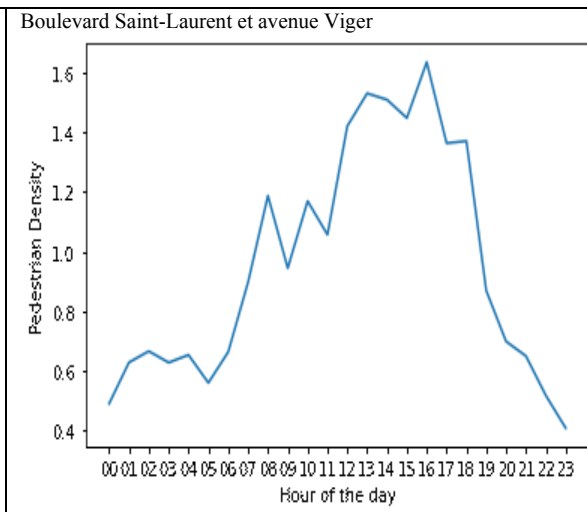
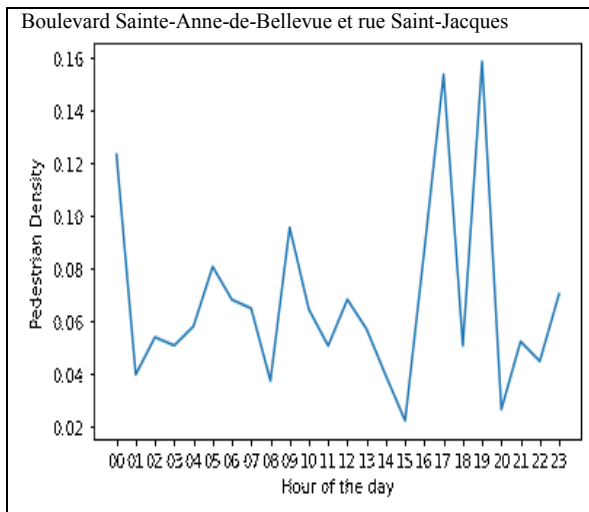


Boulevard Robert-Bourassa et rue Saint-Antoine (inter. ouest)

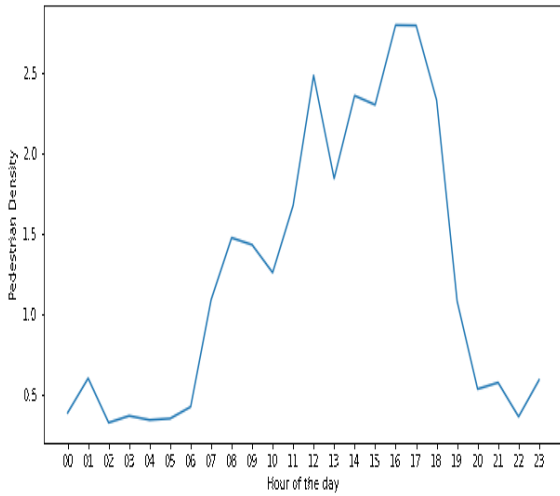


Boulevard Rosemont et rue Viau

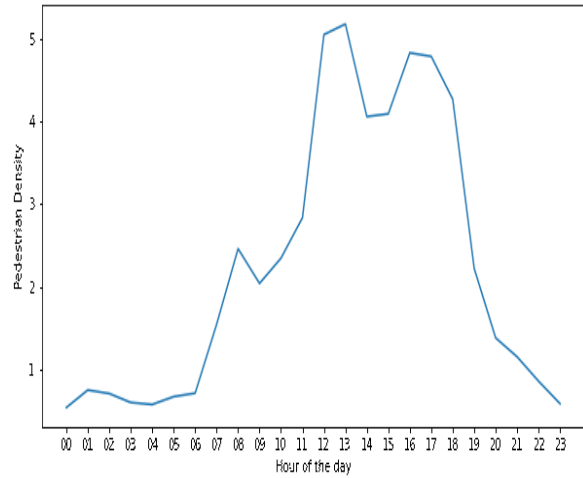




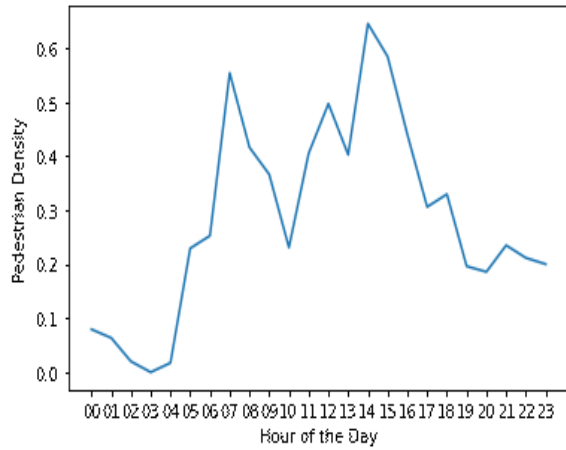
Boulevard de Maisonneuve et boulevard saint Laurent



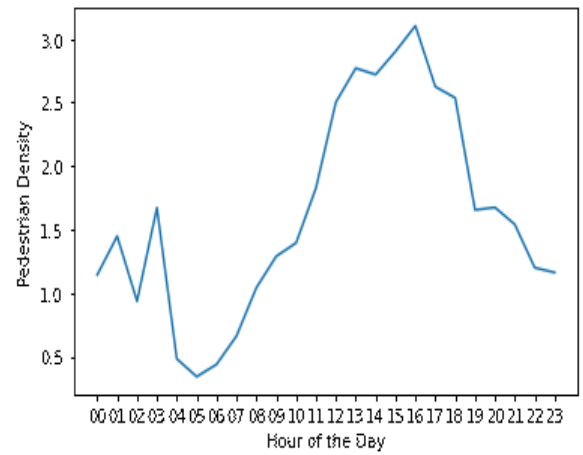
Rue de bleury et rue saint catherine



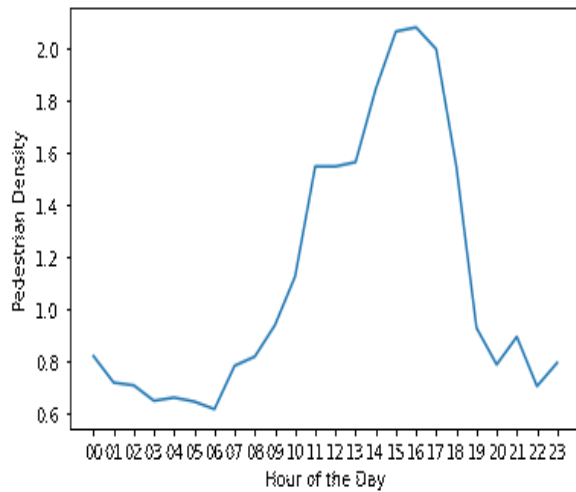
Rue Belanger et rue Viau



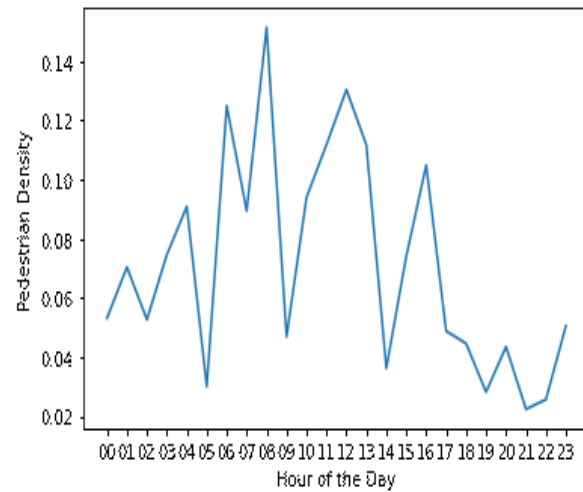
Rue Crescent et boulevard De Maisonneuve

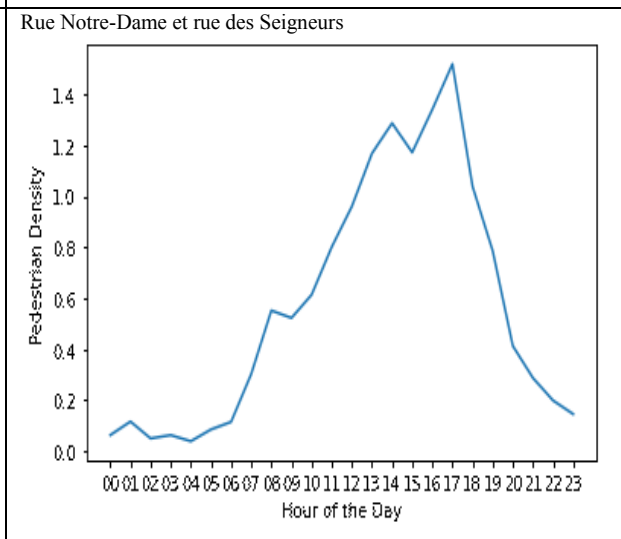
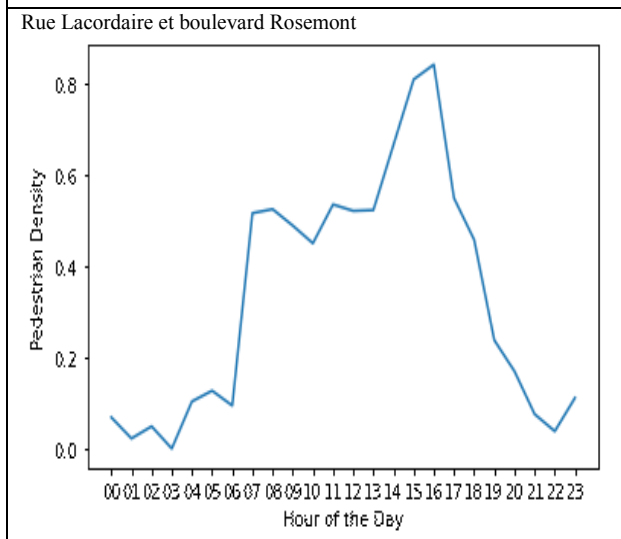
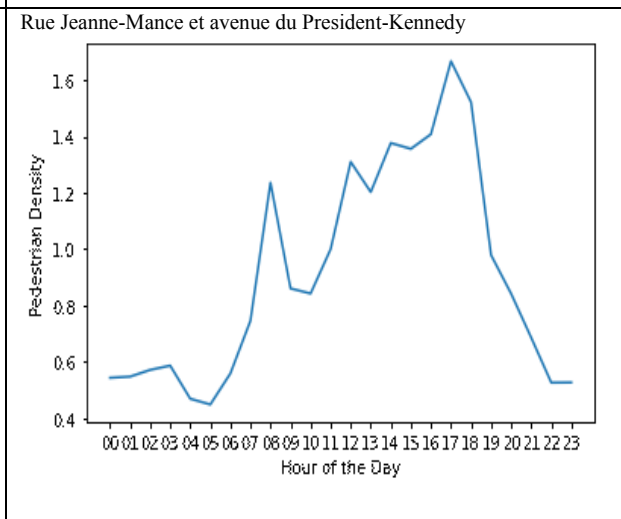
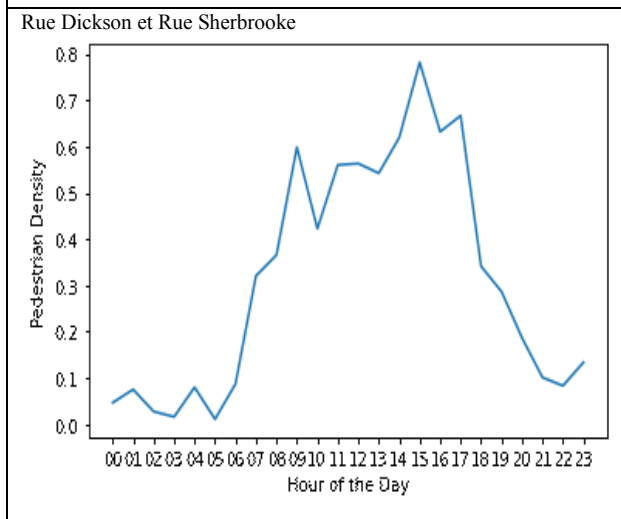
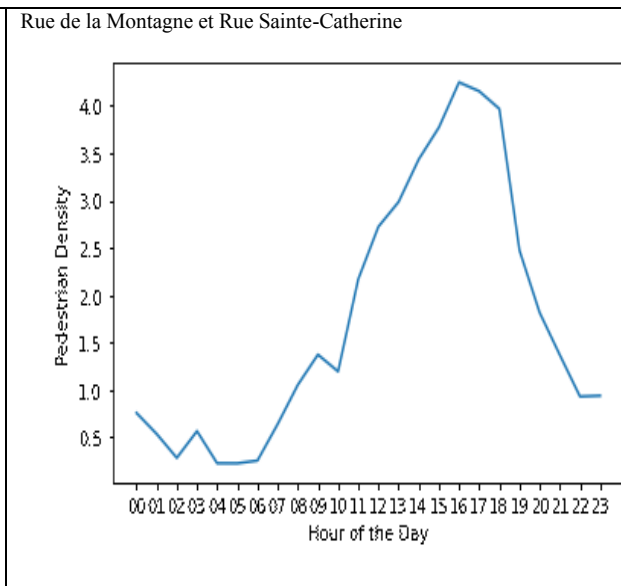
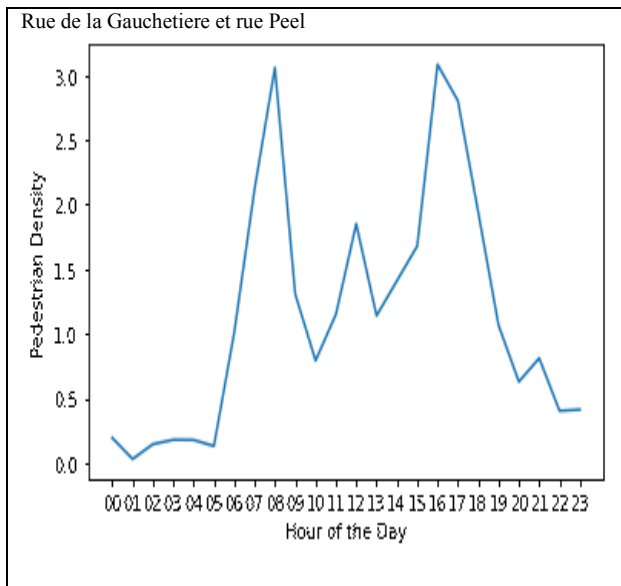


Rue Charlevoix et rue Notre-Dame

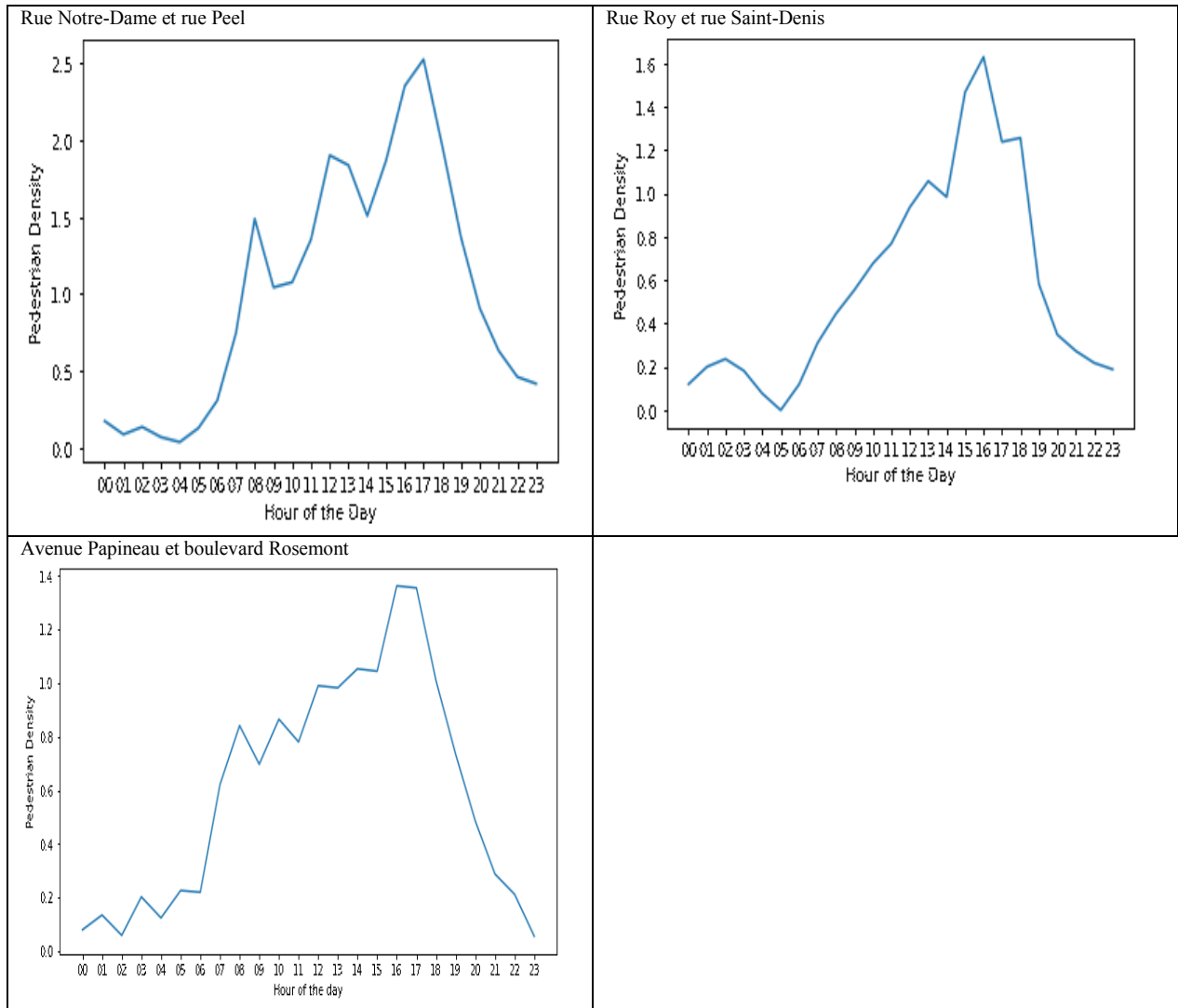


Rue Clement et rue des Oblats









**Fig. 5.9.** Variation of average pedestrian count vs hour of the day

To get a better visualization of the riskiest intersections, i.e. those on which there is maximum pedestrian density and therefore, a need for maximum traffic safety management, another visualization was done, shown in the figure 5.10 below.

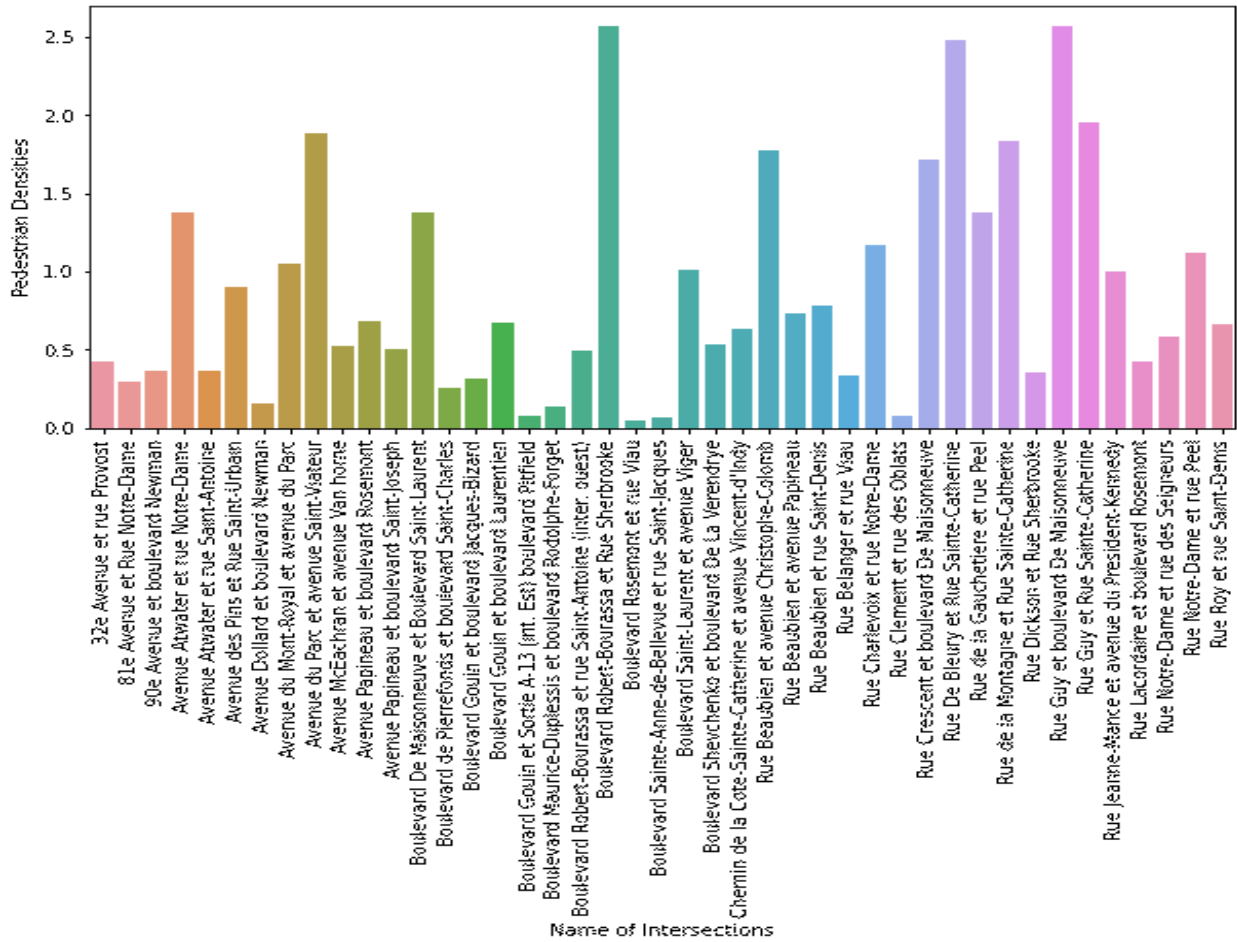


Fig. 5.10. Pedestrian count for each intersection calculated over the entire data set

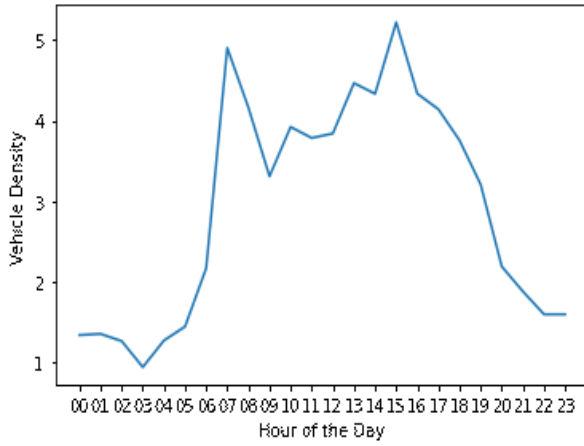
This graph, as shown in figure 5.10, shows the average pedestrian count for each intersection. The graph computes the average pedestrian counts during the 24-hour period for all the days in the dataset and presents them separately for each intersection. Through this graph it is easy to identify the intersections that have the maximum pedestrian average counts, and thus classify them as having the maximum likelihood for accidents involving pedestrians. From the graph it is evident that some of these risky intersections would be: Boulevard Robert-Bourassa et Rue Sherbrooke, Rue Guy et Boulevard de Maisonneuve, Rue De Bleury et Rue Saint Catherine.

### **5.6.2. Vehicle Analysis**

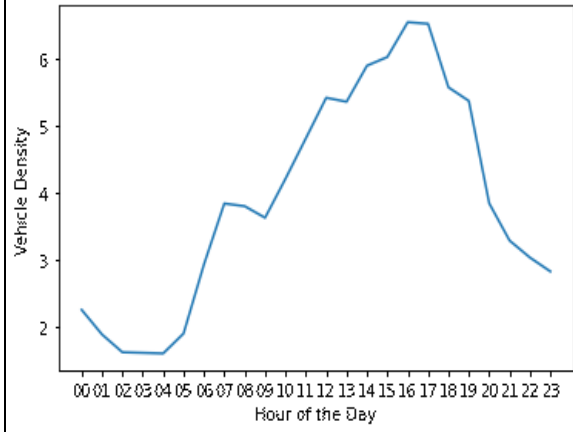
As explained in the previous section, similar graphs are also obtained for the average number of vehicles on the streets, for visualization of vehicle peak/non-peak hours and densities.

The graphs as shown in figure 5.11, shows the average number of vehicles at particular hours for a particular street. These average vehicles counts are computed by taking the entire dataset into consideration, i.e. over a period of 27 days. For instance, the graph for the intersection “32e Avenue et Rue Provost” shows that the peak hours for maximum vehicle counts are approximately 7am – 8am and 2pm – 4 pm. It can be noted that generally 2pm – 5pm are the peak hours for maximum average vehicle count for nearly all the intersections. Thus, it can be noted that these are the hours that would require maximum pedestrian and traffic management for ensuring safety.

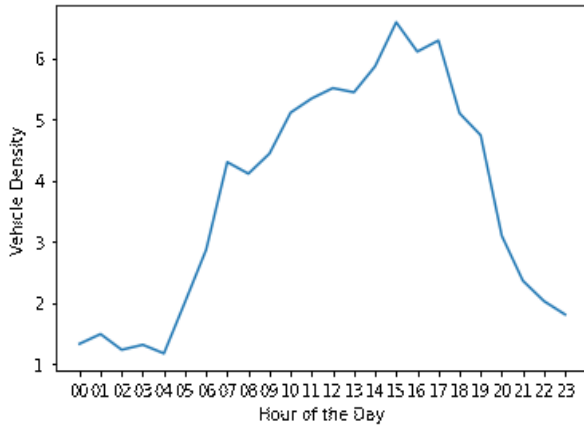
32e Avenue et rue Provost



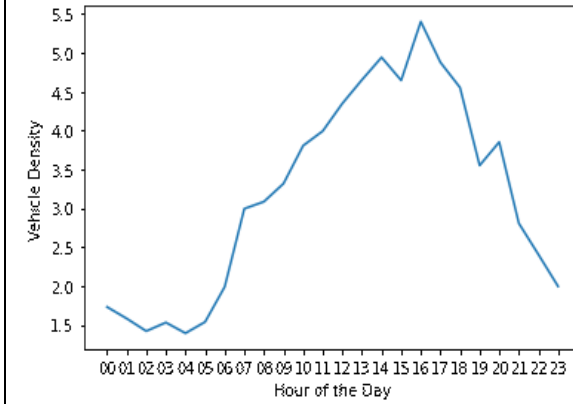
90e Avenue et boulevard Newman



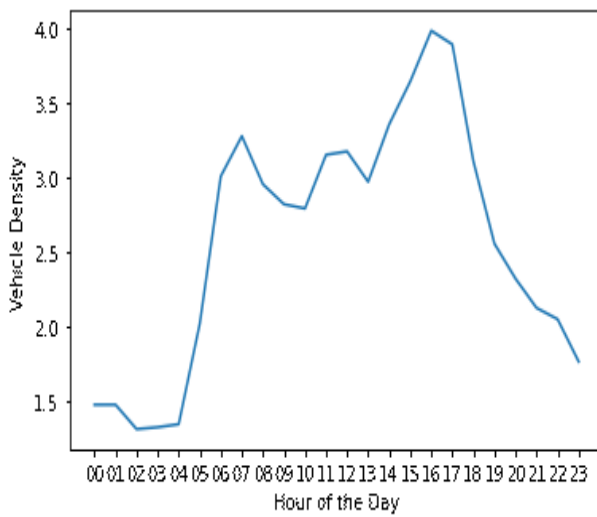
Boulevard Gouin et boulevard Jacques-Bizard



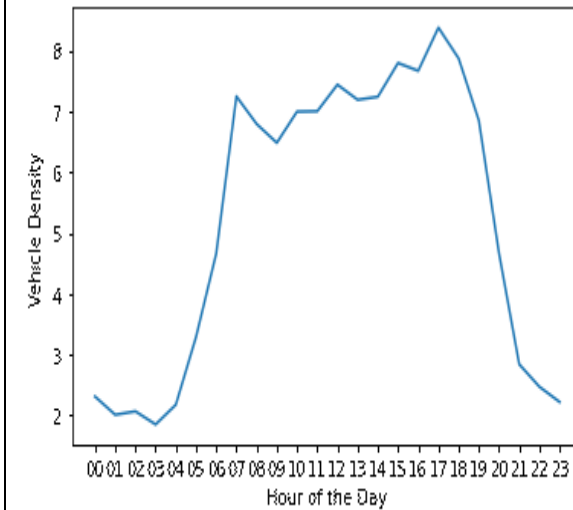
Boulevard Maurice-Duplessis et boulevard Rodolphe-Forget



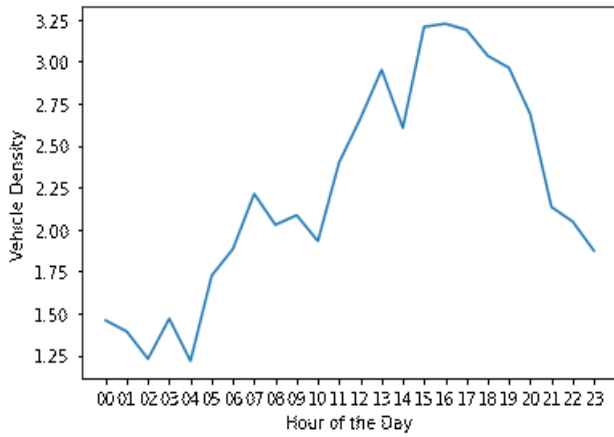
Rue Clement et rue des Oblats



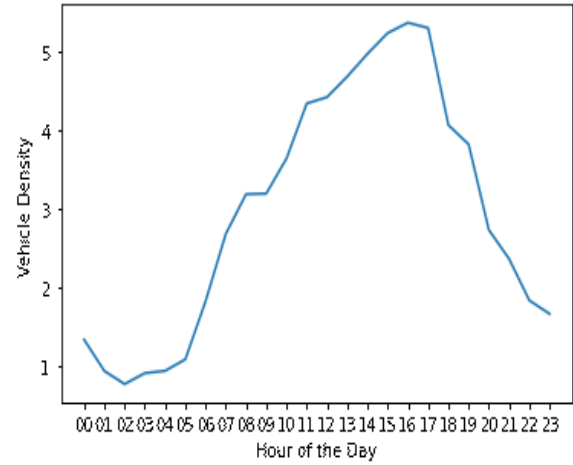
Boulevard de Pierrefonds et boulevard Saint-Charles



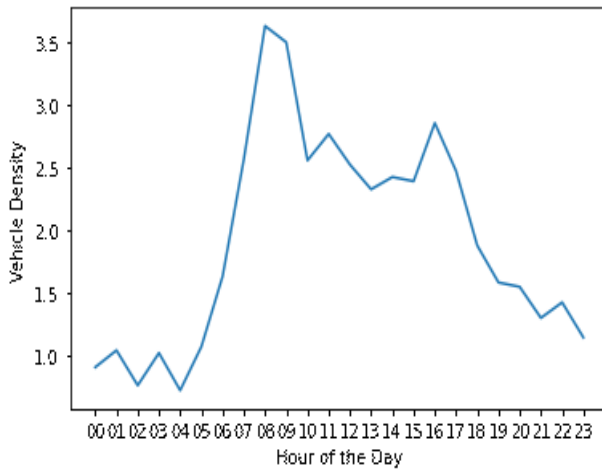
Boulevard Gouin et Sortie A-13 (int. Est) boulevard Pitfield



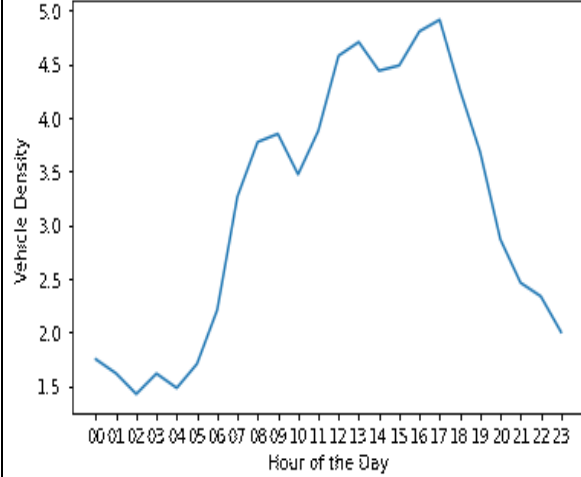
Boulevard Shevchenko et boulevard De La Verendrye



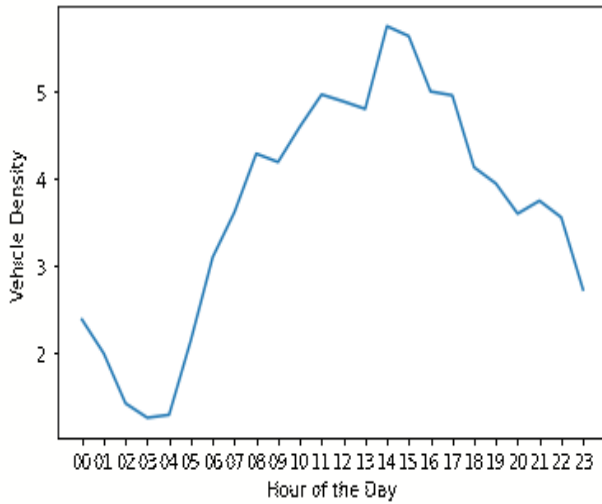
Rue Jeanne-Mance et avenue du President-Kennedy



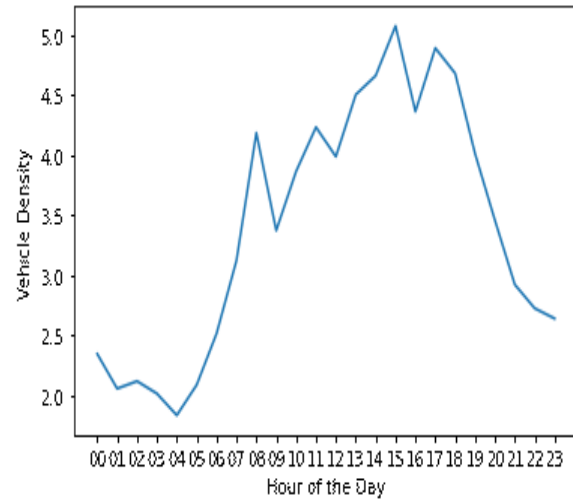
Avenue des Pins et Rue Saint-Urbain



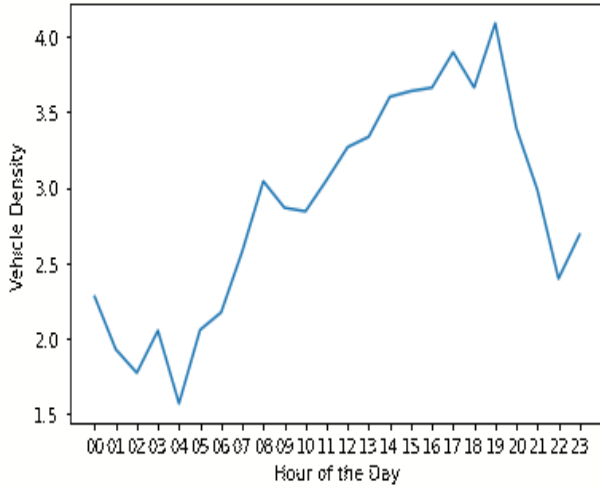
Avenue du Parc et avenue Saint-Viateur



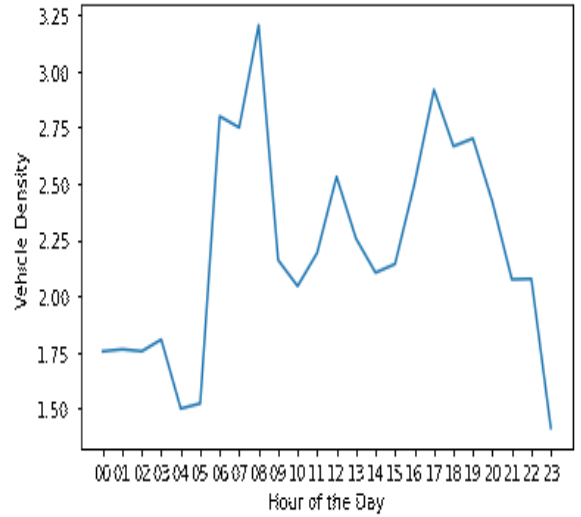
Avenue McEachran et avenue Van home



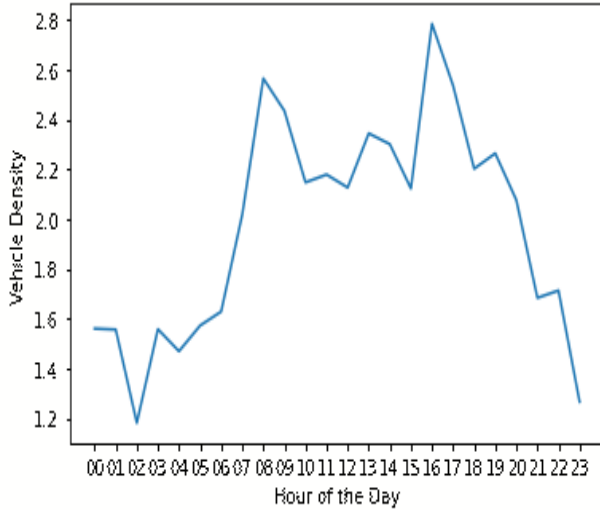
Avenue Papineau et boulevard Saint-Joseph



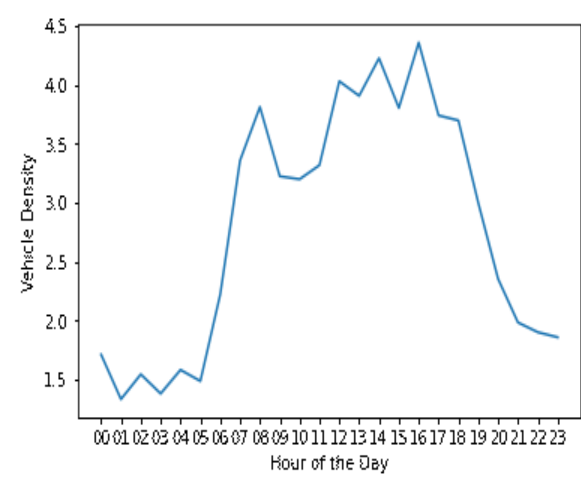
Boulevard De Maisonneuve et Boulevard Saint-Laurent



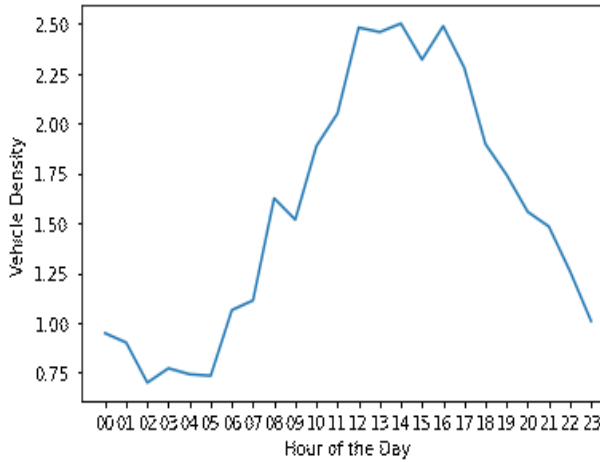
Boulevard Robert-Bourassa et Rue Sherbrooke



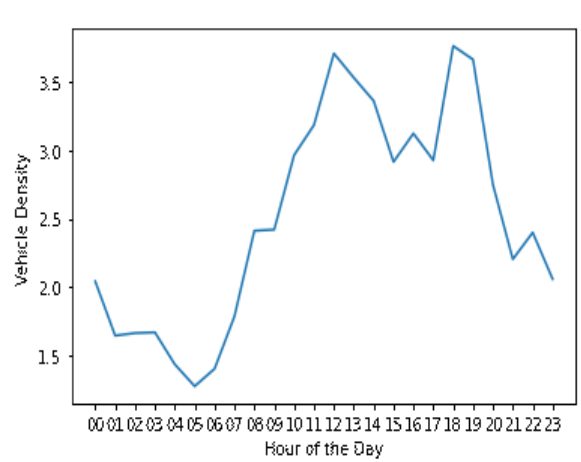
Rue Beaubien et rue Saint-Denis

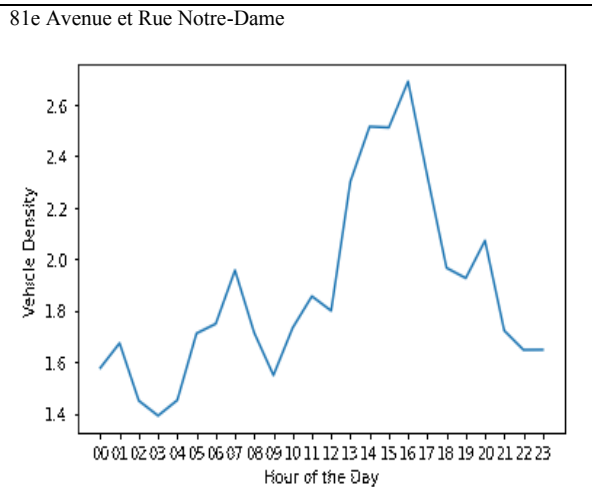
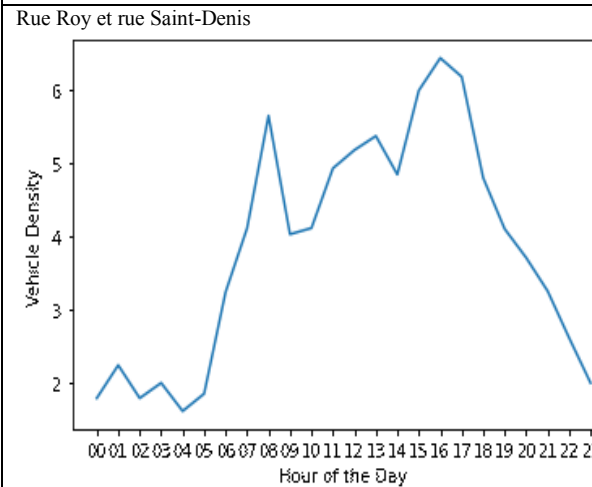
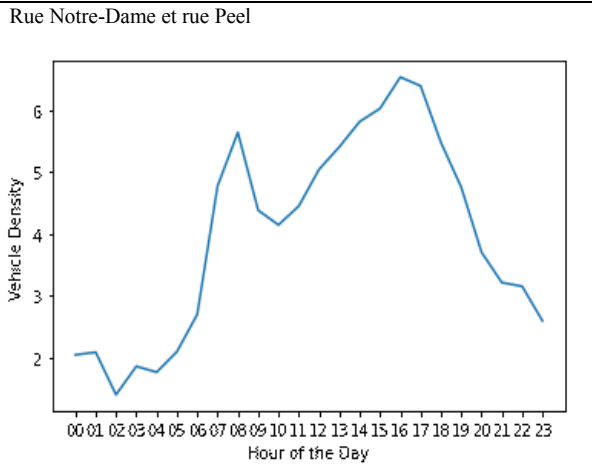
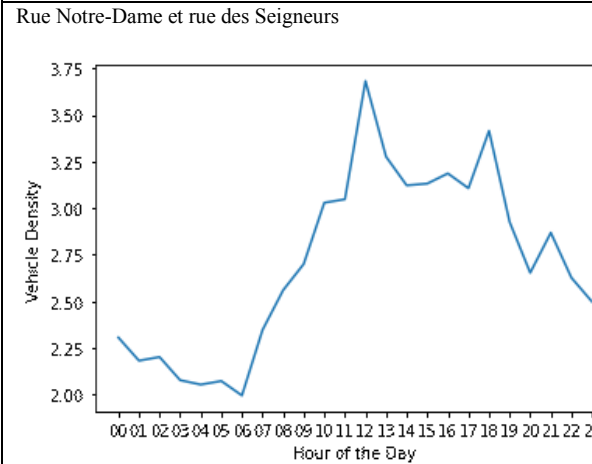
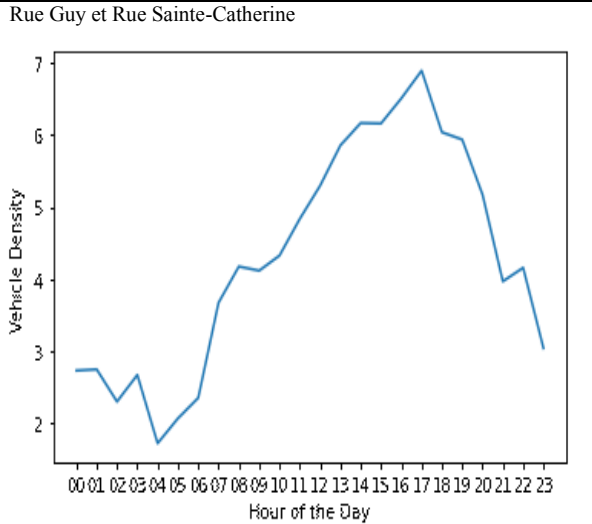
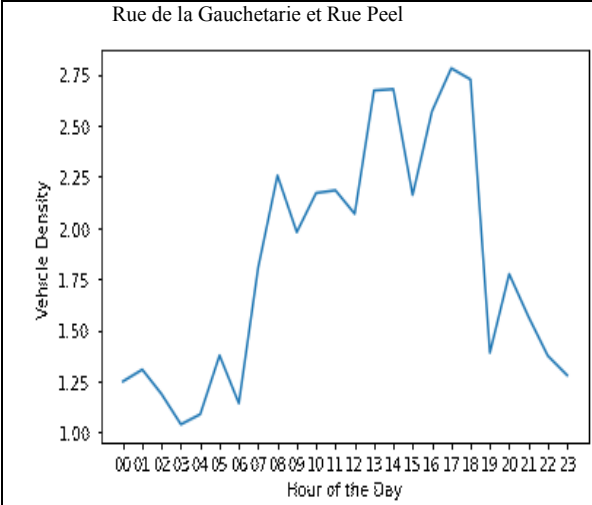


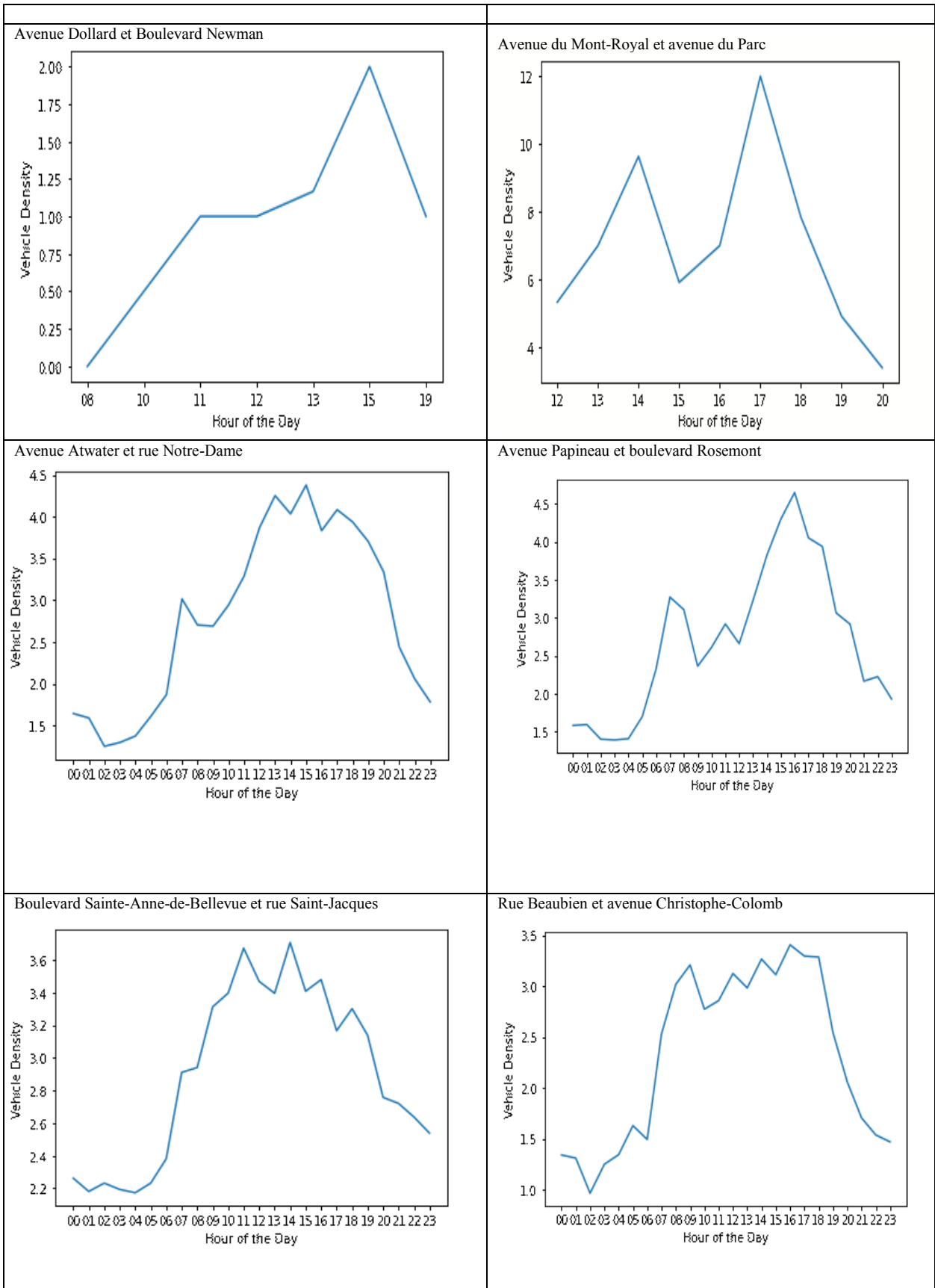
Rue Charlevoix et rue Notre-Dame



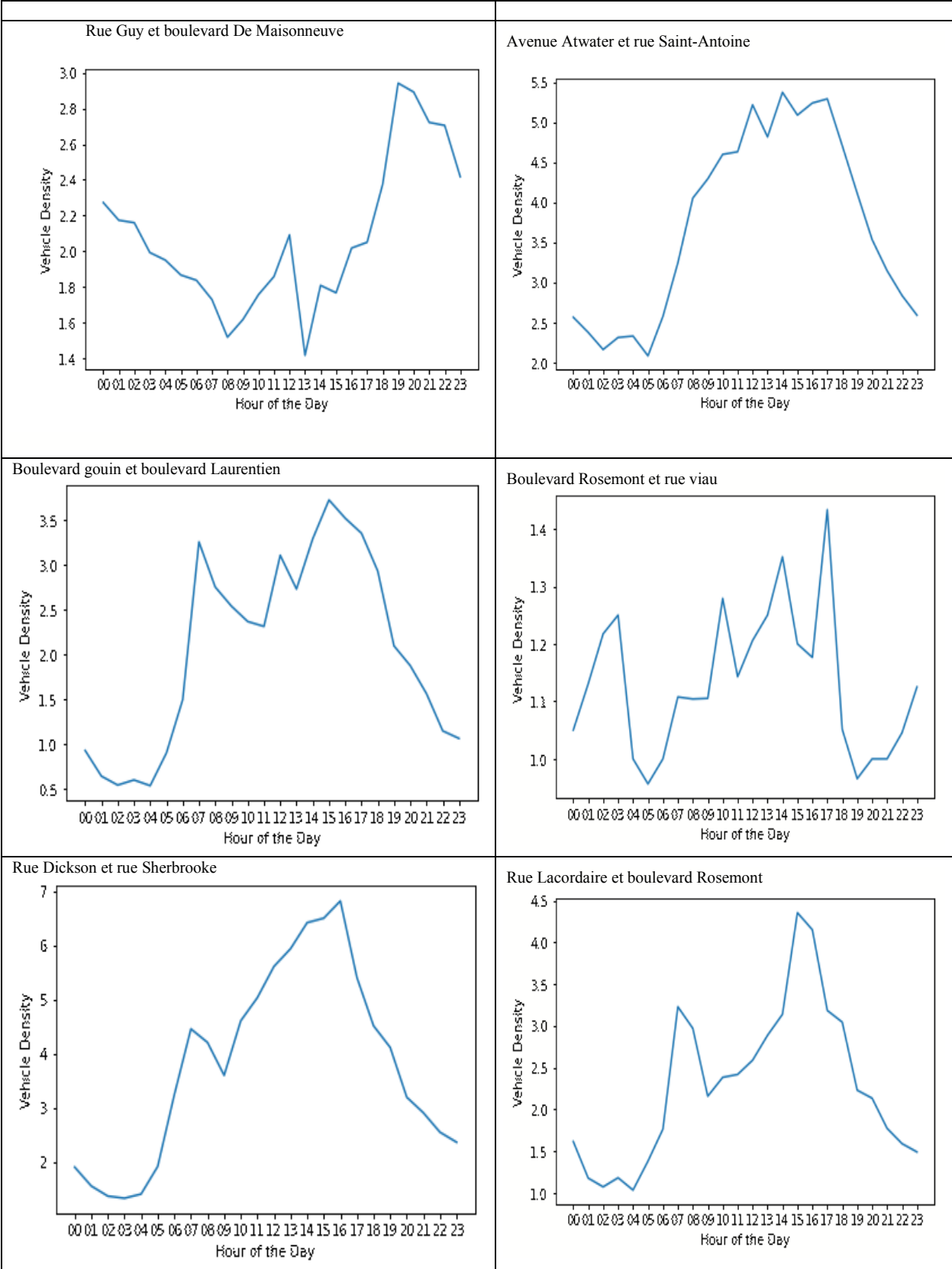
Rue Crescent et boulevard De Maisonneuve

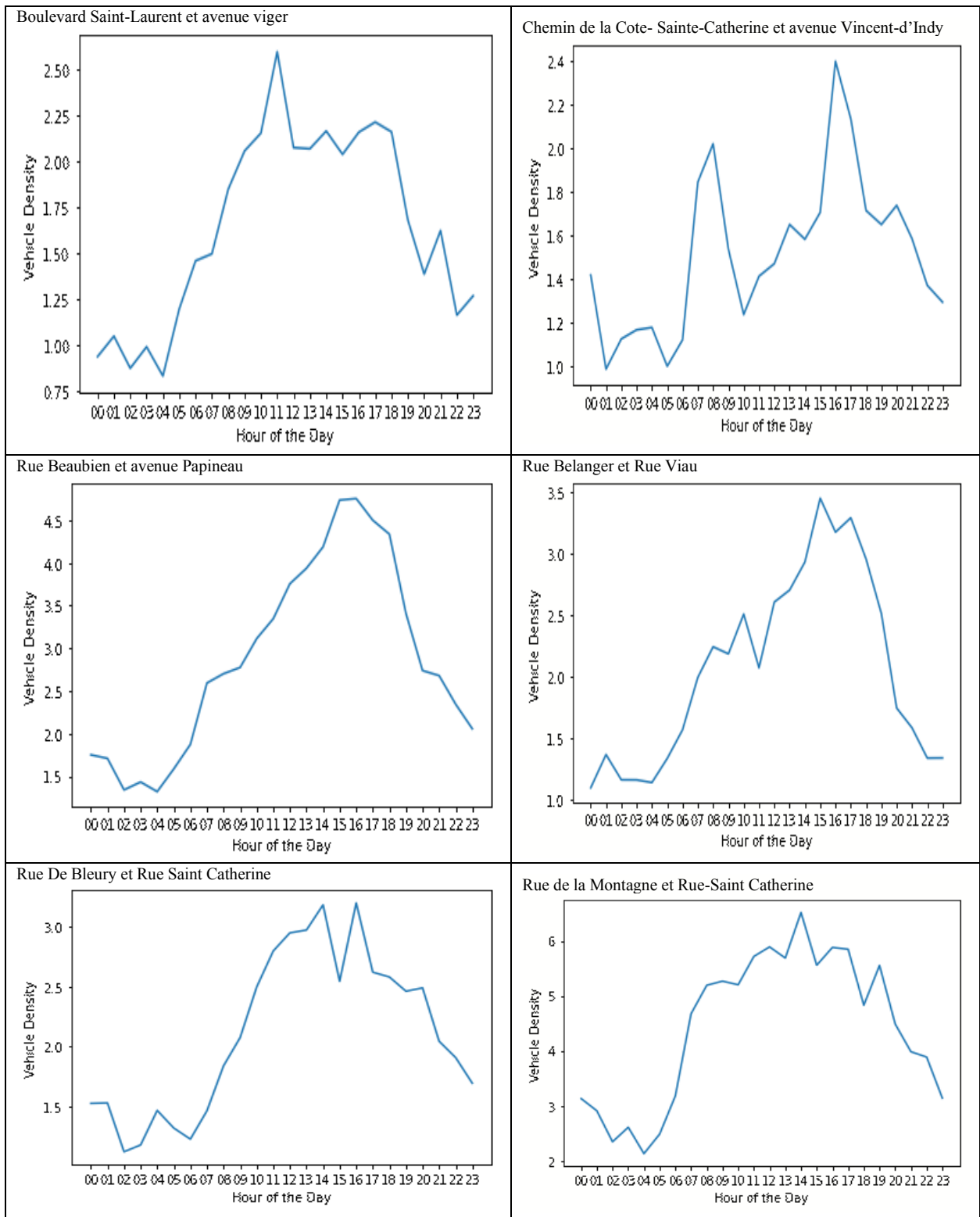


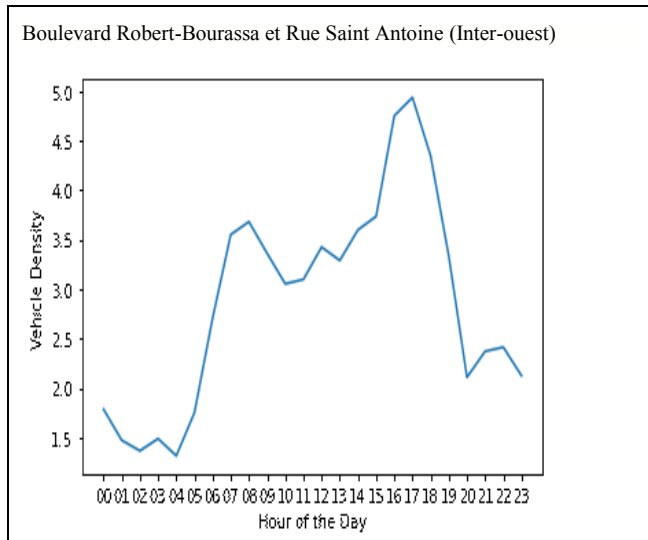












**Fig. 5.11 Variation of vehicle density with hour of day**

To better understand the risks associated with these intersections, it is essential to compare the average vehicle counts for each intersection in order to determine the ones that pose most risk to pedestrians and are most susceptible to encountering accidents. The graph as shown in figure 5.12, illustrates the average vehicle counts for each intersection throughout the 27-day period in the data set for all the peak and non-peak hours (i.e. all 24 hours). From the graph it is evident that some of the intersections with maximum average vehicle densities are: Avenue du Mont Royal et Avenue Du Park, Rue Guy et Rue Saint Catherine, Rue de la Montagne et Rue Saint Catherine, Boulevard de Pierrefonds et Boulevard Saint Charles. These are the intersections that would require maximum management for ensuring that accident counts remain low and that overall traffic safety is enhanced.

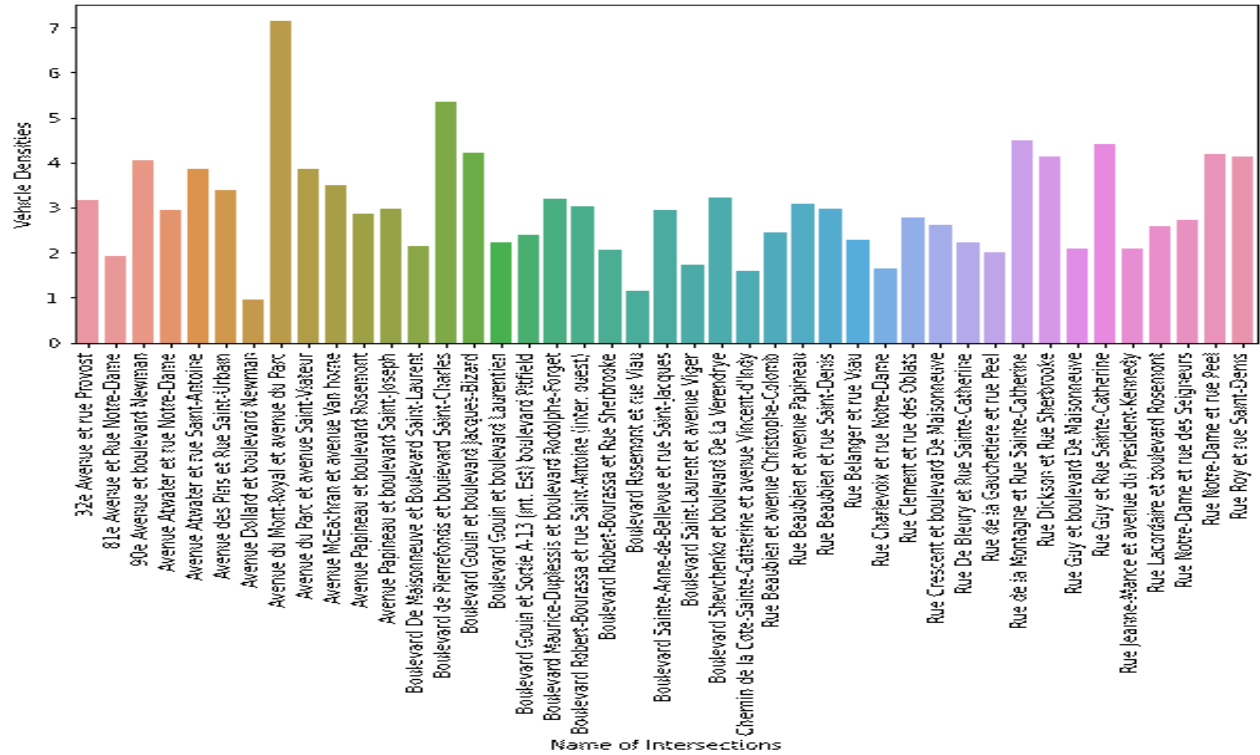


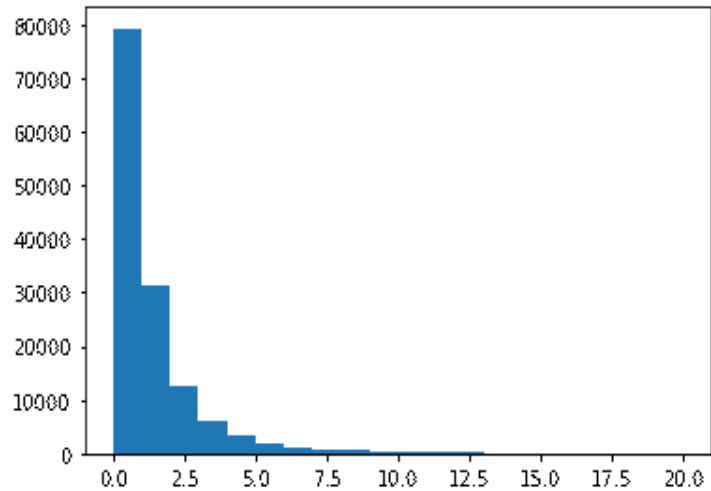
Fig. 5.12 Vehicle count for each intersection calculated over the entire data set

## 5.7. Zero Inflated Poisson Distribution

In this thesis the data set consists of the count of number of pedestrians and vehicles at an intersection during different times of the day. There are several instances when there are 0 or no pedestrians or vehicles at the intersection. Hence the zero valued counts are relatively high especially during the nighttime when the rush hour decreases. The zero inflated poisson

regression model, often referred to as ZIP model, will be used to structure the pedestrian and vehicle count.

Zero	
pedestrian	
0	79223
1	31319
2	12579
3	6002
4	3352
5	1892
6	1096
7	691
8	523
9	372
10	277
11	204
12	166
13	99
14	85
15	54
16	34
17	43
18	27
19	18
20	5



**Fig. 5.13. Figures to depict the zero counts contained in the Data set**

As per the data set obtained, we can observe that 79,223 rows have pedestrian count equal to 0 (nearly 57% of the rows in our data set). Hence, it is essential to apply the zero inflation poisson distribution model on it.

The structure of ZIP model can be explained as, for instance we have a data set containing  $n$  samples with  $p$  regression variable per sample. Thus, the matrix of size  $(n \times p)$  represents the

regression variables X and each row  $x_i$  in the matrix X represents a vector whose size (1xp) corresponds to the dependent variable value  $y_i$ :

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

A data set (y,X) in matrix notation

On the basis of our assumption that y is a Poisson distributed random variable, a Poisson distribution model can be derived. This model consists of:

- 1) A Poisson Probability Mass Function (PMF) represented as  $p(y_i=k)$ , this is used to calculate the probability of observing k events in a unit interval given a mean rate of  $\lambda$  events / unit time.
- 2) To express the mean rate  $\lambda$  as a function of the regression variables X a link function is required.

Here,  $P(y_i = k) = \frac{e^{-\lambda_i} * \lambda_i^k}{k!}$

Probability of observing k events in the  $i^{\text{th}}$  sample

$y_i$  = observed count for the  $i^{\text{th}}$  row in the data set

$x_i$  = regression variables in the  $i^{\text{th}}$  row

$\lambda_i$  = Event rate corresponding to the  $i^{\text{th}}$  sample

Where  $\lambda_i = e^{x_i \beta}$ , here  $x_i$  = Regression variables in the  $i^{\text{th}}$  row

$\beta$  is the vector regression coefficients

The Zero Inflated Poisson Distribution model works with the idea that there is a second underlying process that determines whether a count is zero or non-zero.

In the case when the count is considered non-zero, regular Poisson process determines the actual non-zero value based on Poisson process's PMF.

Thus, a ZIP regression model has three parts:

- 1) A PMF  $P(y_i=0)$  calculates probability of observing a zero count
- 2) A second PMF  $P(y_i=k)$  calculates the probability of observing  $k$  events, given  $k>0$
- 3) A link function that is used to express  $\lambda$  or the mean rate as a function of regression variables  $X$ .

Probability Mass Function the Zero-Inflated Poisson regression model

$$P(y_i = 0) = \phi_i + (1 - \phi_i) * e^{-\lambda_i}$$

When  $y_i = 1, 2, 3, \dots$

$$P(y_i = k) = (1 - \phi_i) \frac{e^{-\lambda_i} * \lambda_i^k}{k!}$$

Where  $\lambda_i = e^{x_i \beta}$

Here  $\phi_i$  = Proportion of zeroes for the  $i^{\text{th}}$  row in the data set.

The random variable  $y_i$  is the random variable that denotes the observed count corresponding to the regression variables row  $x_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{ip}]$ .

$\phi_i$  is a measure of the proportion of excess zeroes corresponding to the  $i^{\text{th}}$  row ( $y_i, x_i$ ) in the data set; in this case 110402 observations were taken. Here, independent variables are name and time and dependent variables are pedestrians. They have the same combination of regression variable values  $x_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{ip}]$ . Since  $y_i$  is a random variable, a different value of  $y_i$  can be observed in 110402 values. For instance, out of 1000  $y_i$  values, 874 are zero values, out of these 874 zero values, the regular Poisson distribution assumed for

$y_i$  will be able to explain only up to 7 zero values. Hence, remaining 867 zero values are excess zero observations. So, for the  $i$ th row in your data set,  $\phi_{-i} = 867/1000=0.867$ .

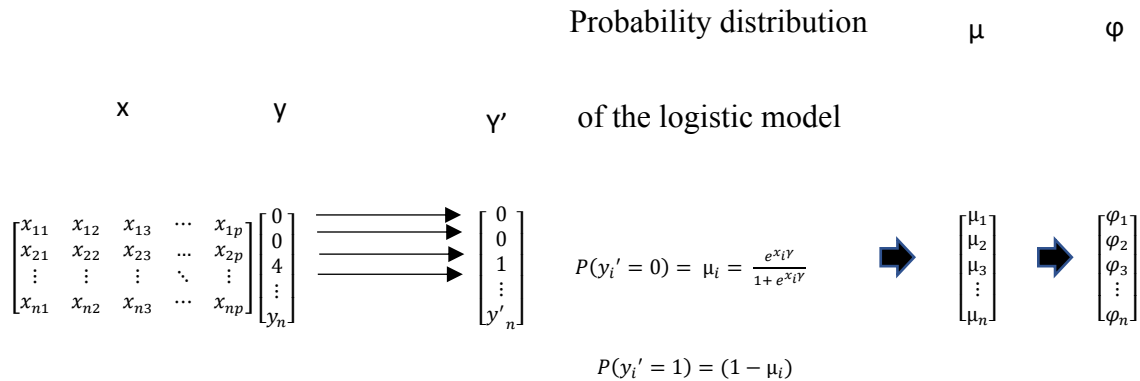
When the data set does not have any excess zeroes in the dependent variable, the value of  $\phi$  works out to be zero and the PMF of the ZIP model reduces to the PMF of the standard Poisson model (this can be verified by setting  $\phi$  to 0 in the ZIP model's PMF).

To estimate  $\phi_{-i}$  is by setting it to the following ratio:

$$\phi_{-i} = \frac{\text{Number of zero valued samples}}{\text{Total number of samples}}$$

Another realistic way of calculating  $\phi_{-i}$  is by estimating it as a function of regression variables  $X$ . This is achieved by transforming the  $y$  variable to a binary 0/1 random variable  $y'$  ( $y_{\text{prime}}$ ) which takes the value 0 if the underlying  $y$  is 0 and 1 in all other cases. Then a Logistic regression model on the  $y'$  is fitted. Then the training of Logistic regression model on the data set  $[X, y']$  is made and it gives a vector of fitted probabilities  $\mu_{\text{fitted}} = [\mu_1, \mu_2, \mu_3, \dots, \mu_n]$ , this is what a Logistic regression model does. After getting  $\mu_{\text{fitted}}$  vector, it is set to vector  $\phi$ . Thus,  $[\phi_1 = \mu_1, \phi_2 = \mu_2, \phi_3 = \mu_3, \dots, \phi_n = \mu_n]$ .

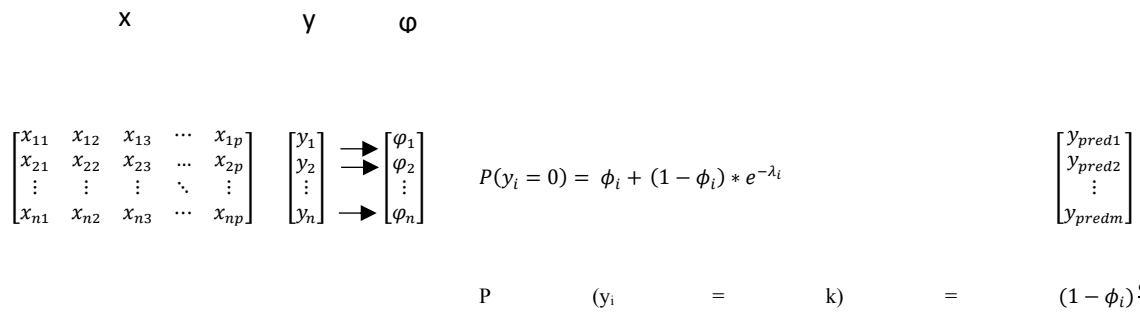




Data Set

1. Transform y to a binary y'
2. Train a logistic regression model on (y', X) to get vector of fitted probabilities μ
3. Set φ to μ

Probability distribution of the ZIP model



Data Set

1. Calculate the φ vector
2. Use this 2-part probability distribution to train the ZIP model on the (y,X) data set
3. Use the trained model to make predictions on the test data

## 5.7.1. Zero Inflated Poisson Distribution Results

Figure 5.14 shows zero Inflated Poisson Distribution Results

ZeroInflatedPoisson Regression Results						
=====						
Dep. Variable:	pedestrian	No. Observations:	74915			
Model:	ZeroInflatedPoisson	Df Residuals:	74872			
Method:	MLE	Df Model:	42			
Date:	Mon, 20 Jul 2020	Pseudo R-squ.:	0.1317			
Time:	16:21:05	Log-likelihood:	-98021.			
converged:	False	LL-Null:	-1.1289e+05			
Covariance Type:	nonrobust	LLR p-value:	0.000			
=====						
		coef	std err	z	P> z	[0.025 0.975]
-----						
inflate Intercept		-0.4725	0.089	-5.296	0.000	-0.647 -0.298
inflate_name[T.81e Avenue et Rue Notre-Dame]		0.6465	0.132	4.891	0.000	0.387 0.906
inflate_name[T.90e Avenue et boulevard Newman]		0.0832	0.122	0.679	0.497	-0.157 0.323
inflate_name[T.Avenue Atwater et rue Notre-Dame]		-0.0115	0.105	-0.109	0.913	-0.218 0.195
inflate_name[T.Avenue Atwater et rue Saint-Antoine]		0.5280	0.122	4.316	0.000	0.288 0.768
inflate_name[T.Avenue Dollard et boulevard Newman]		0.1106	1.218	0.091	0.928	-2.278 2.499
inflate_name[T.Avenue McEachran et avenue Van horne]		0.2185	0.118	1.858	0.063	-0.012 0.449
inflate_name[T.Avenue Papineau et boulevard Rosemont]		0.1445	0.139	1.044	0.297	-0.127 0.416
inflate_name[T.Avenue Papineau et boulevard Saint-Joseph]		0.0476	0.117	0.407	0.684	-0.181 0.277
inflate_name[T.Avenue des Pins et Rue Saint-Urbain]		-0.0627	0.113	-0.557	0.578	-0.283 0.158
inflate_name[T.Avenue du Mont-Royal et avenue du Parc]		0.0786	0.322	0.244	0.807	-0.552 0.709
inflate_name[T.Avenue du Parc et avenue Saint-Viateur]		-0.4470	0.105	-4.256	0.000	-0.653 -0.241
inflate_name[T.Boulevard De Maisonneuve et Boulevard Saint-Laurent]		-0.4278	0.102	-4.202	0.000	-0.627 -0.228
inflate_name[T.Boulevard Gouin et Sortie A-13 (int. Est) boulevard Pitfield]		1.1219	0.160	7.016	0.000	0.809 1.435
inflate_name[T.Boulevard Gouin et boulevard Jacques-Bizard]		0.4885	0.132	3.712	0.000	0.231 0.746
inflate_name[T.Boulevard Gouin et boulevard Laurentien]		-0.7025	0.125	-5.631	0.000	-0.947 -0.458
inflate_name[T.Boulevard Maurice-Duplessis et boulevard Rodolphe-Forget]		1.0484	0.132	7.946	0.000	0.790 1.307
inflate_name[T.Boulevard Robert-Bourassa et Rue Sherbrooke]		-0.5034	0.104	-4.834	0.000	-0.708 -0.299
inflate_name[T.Boulevard Robert-Bourassa et rue Saint-Antoine (inter. ouest)]		0.4956	0.109	4.562	0.000	0.283 0.709
inflate_name[T.Boulevard Rosemont et rue Viau]		0.6883	0.231	2.979	0.003	0.235 1.141
inflate_name[T.Boulevard Saint-Laurent et avenue Viger]		-0.8422	0.123	-6.842	0.000	-1.083 -0.601
inflate_name[T.Boulevard Sainte-Anne-de-Bellevue et rue Saint-Jacques]		1.1408	0.157	7.284	0.000	0.834 1.448
inflate_name[T.Boulevard Shevchenko et boulevard De La Verendrye]		0.0335	0.126	0.266	0.791	-0.214 0.281
inflate_name[T.Boulevard de Pierrefonds et boulevard Saint-Charles]		0.6656	0.122	5.464	0.000	0.427 0.904
inflate_name[T.Chemin de la Cote-Sainte-Catherine et avenue Vincent-d'Indy]		-0.0774	0.125	-0.619	0.536	-0.322 0.167
inflate_name[T.Rue Beaubien et avenue Christophe-Colomb]		-0.4928	0.107	-4.603	0.000	-0.703 -0.283
inflate_name[T.Rue Beaubien et avenue Papineau]		0.1106	0.111	0.998	0.318	-0.107 0.328
inflate_name[T.Rue Beaubien et rue Saint-Denis]		0.1129	0.112	1.012	0.312	-0.106 0.332
inflate_name[T.Rue Belanger et rue Viau]		0.4144	0.130	3.178	0.001	0.159 0.670
inflate_name[T.Rue Charlevoix et rue Notre-Dame]		-0.9375	0.121	-7.752	0.000	-1.175 -0.700
inflate_name[T.Rue Clement et rue des Oblats]		0.9779	0.149	6.573	0.000	0.686 1.269
inflate_name[T.Rue Crescent et boulevard De Maisonneuve]		-0.4220	0.104	-4.042	0.000	-0.627 -0.217
inflate_name[T.Rue De Bleury et Rue Sainte-Catherine]		-0.7817	0.107	-7.333	0.000	-0.991 -0.573
inflate_name[T.Rue Dickson et Rue Sherbrooke]		0.3745	0.125	2.990	0.003	0.129 0.620
inflate_name[T.Rue Guy et Rue Sainte-Catherine]		0.0436	0.100	0.434	0.664	-0.153 0.240
inflate_name[T.Rue Guy et boulevard De Maisonneuve]		-0.9794	0.107	-9.157	0.000	-1.189 -0.770
inflate_name[T.Rue Jeanne-Mance et avenue du President-Kennedy]		-0.4757	0.118	-4.045	0.000	-0.706 -0.245
inflate_name[T.Rue Lacordaire et boulevard Rosemont]		0.1085	0.128	0.849	0.396	-0.142 0.359
inflate_name[T.Rue Notre-Dame et rue Peel]		-0.0004	0.104	-0.004	0.997	-0.205 0.204

inflate_name[T.Rue Lacordaire et boulevard Rosemont]	0.1085	0.128	0.849	0.396	-0.142	0.359
inflate_name[T.Rue Notre-Dame et rue Peel]	-0.0004	0.104	-0.004	0.997	-0.205	0.204
inflate_name[T.Rue Notre-Dame et rue des Seigneurs]	0.4001	0.111	3.593	0.000	0.182	0.618
inflate_name[T.Rue Roy et rue Saint-Denis]	0.0906	0.112	0.811	0.417	-0.128	0.309
inflate_name[T.Rue de la Gauchetiere et rue Peel]	-0.0532	0.105	-0.508	0.611	-0.258	0.152
inflate_name[T.Rue de la Montagne et Rue Sainte-Catherine]	-0.1176	0.101	-1.165	0.244	-0.315	0.080
Intercept	0.0672	0.039	1.735	0.083	-0.009	0.143
name[T.81e Avenue et Rue Notre-Dame]	-0.6676	0.070	-9.548	0.000	-0.805	-0.531
name[T.90e Avenue et boulevard Newman]	-0.5086	0.055	-9.254	0.000	-0.616	-0.401
name[T.Avenue Atwater et rue Notre-Dame]	0.6964	0.045	15.484	0.000	0.608	0.785
name[T.Avenue Atwater et rue Saint-Antoine]	-0.6141	0.061	-9.996	0.000	-0.735	-0.494
name[T.Avenue Dollard et boulevard Newman]	-0.0193	0.564	-0.034	0.973	-1.125	1.086
name[T.Avenue McEachran et avenue Van horne]	-0.1316	0.054	-2.421	0.015	-0.238	-0.025
name[T.Avenue Papineau et boulevard Rosemont]	-0.1361	0.064	-2.128	0.033	-0.261	-0.011
name[T.Avenue Papineau et boulevard Saint-Joseph]	-0.0117	0.052	-0.226	0.821	-0.113	0.090
name[T.Avenue des Pins et Rue Saint-Urbain]	0.3569	0.049	7.324	0.000	0.261	0.452
name[T.Avenue du Mont-Royal et avenue du Parc]	0.1045	0.148	0.707	0.480	-0.185	0.394
name[T.Avenue du Parc et avenue Saint-Viateur]	0.9151	0.043	21.511	0.000	0.832	0.999
name[T.Boulevard De Maisonneuve et Boulevard Saint-Laurent]	0.6974	0.042	16.545	0.000	0.615	0.780
name[T.Boulevard Gouin et Sortie A-13 (int. Est) boulevard Pitfield]	-1.2835	0.098	-13.070	0.000	-1.476	-1.091
name[T.Boulevard Gouin et boulevard Jacques-Bizard]	-0.9031	0.067	-13.398	0.000	-1.035	-0.771
name[T.Boulevard Gouin et boulevard Laurentien]	0.0786	0.047	1.672	0.095	-0.014	0.171
name[T.Boulevard Maurice-Duplessis et boulevard Rodolphe-Forget]	-1.1631	0.077	-15.031	0.000	-1.315	-1.011
name[T.Boulevard Robert-Bourassa et Rue Sherbrooke]	1.4477	0.041	35.364	0.000	1.367	1.528
name[T.Boulevard Robert-Bourassa et rue Saint-Antoine (inter. ouest)]	0.2594	0.053	4.884	0.000	0.155	0.363
name[T.Boulevard Rosemont et rue Viau]	-0.7252	0.123	-5.917	0.000	-0.965	-0.485
name[T.Boulevard Saint-Laurent et avenue Viger]	0.1685	0.046	3.684	0.000	0.079	0.258
name[T.Boulevard Sainte-Anne-de-Bellevue et rue Saint-Jacques]	-1.3094	0.097	-13.552	0.000	-1.499	-1.120
name[T.Boulevard Shevchenko et boulevard De La Verendrye]	-0.4786	0.056	-8.513	0.000	-0.589	-0.368
name[T.Boulevard de Pierrefonds et boulevard Saint-Charles]	-0.7285	0.063	-11.517	0.000	-0.852	-0.605
name[T.Chemin de la Cote-Sainte-Catherine et avenue Vincent-d'Indy]	-0.1441	0.054	-2.679	0.007	-0.250	-0.039
name[T.Rue Beaubien et avenue Christophe-Colomb]	0.8443	0.043	19.625	0.000	0.760	0.929
name[T.Rue Beaubien et avenue Papineau]	0.2571	0.050	5.162	0.000	0.159	0.355
name[T.Rue Beaubien et rue Saint-Denis]	0.1949	0.050	3.881	0.000	0.096	0.293
name[T.Rue Belanger et rue Viau]	-0.6339	0.065	-9.803	0.000	-0.761	-0.507
name[T.Rue Charlevoix et rue Notre-Dame]	0.2562	0.045	5.731	0.000	0.169	0.344
name[T.Rue Clement et rue des Oblats]	-1.2523	0.088	-14.220	0.000	-1.425	-1.080
name[T.Rue Crescent et boulevard De Maisonneuve]	0.9786	0.042	23.122	0.000	0.896	1.062
name[T.Rue De Bleury et Rue Sainte-Catherine]	1.1628	0.041	28.104	0.000	1.082	1.244
name[T.Rue Dickson et Rue Sherbrooke]	-0.6479	0.061	-10.631	0.000	-0.767	-0.528
name[T.Rue Guy et Rue Sainte-Catherine]	1.2405	0.042	29.804	0.000	1.159	1.322
name[T.Rue Guy et boulevard De Maisonneuve]	1.2232	0.041	29.901	0.000	1.143	1.303
name[T.Rue Jeanne-Mance et avenue du President-Kennedy]	0.2986	0.047	6.384	0.000	0.207	0.390
name[T.Rue Lacordaire et boulevard Rosemont]	-0.3811	0.058	-6.573	0.000	-0.495	-0.267
name[T.Rue Notre-Dame et rue Peel]	0.6700	0.045	14.970	0.000	0.582	0.758
name[T.Rue Notre-Dame et rue des Seigneurs]	-0.0531	0.053	-0.995	0.320	-0.158	0.052
name[T.Rue Roy et rue Saint-Denis]	0.2151	0.050	4.299	0.000	0.117	0.313
name[T.Rue de la Gauchetiere et rue Peel]	0.9169	0.044	20.936	0.000	0.831	1.003
name[T.Rue de la Montagne et Rue Sainte-Catherine]	1.2023	0.042	28.912	0.000	1.121	1.284

Fig. 5.14. Zero Inflated Poisson distribution Results

It can be seen in the above results, that for certain intersections, the p-value in the rows `inflate_name[intersection name]` is  $<0.05$  (which is the 95% confidence interval value). This indicated that these are the intersections that the nested Logistic Regression model used for estimating the probability of there being pedestrians on the road or not. As it can be seen from the results, intersections such as 81e Avenue et Rue Notre Dame, 90e Avenue et Boulevard Newman, Avenue Atwater et Rue St Antoine, Boulevard de Maisonneuve et Boulevard St Laurent, Boulevard Robert-Bourassa et Rue Sherbrooke, Boulevard Robert-Bourassa et Rue Saint Antoine, Avenue du Parc et Avenue Saint-Viateur, Rue Crescent et Boulevard de Maisonneuve, etc. all have statistically significant coefficients of regression (p values lower than 0.05). Hence, these were statistically most significant to the Logistic Regression model. In fact, these are the intersections that had significantly larger pedestrian densities throughout the data set especially during peak hours, and hence remained useful to the model.

The rows under the 'Intercept' row contain information about the intersections which were used by the model to calculate the pedestrian density given that pedestrian density is greater than 0. The graph depicts that nearly all the intersections were statistically significant for this observation (since their p values are less than 0.05 for 95% confidence interval). However, there are very few intersections for which this value is greater than 0.05. These intersections are: Avenue Dollard et Boulevard Newman, Avenue Papineau et Boulevard Rosemont, Avenue Papineau et Boulevard Saint-Joseph, Avenue du Mont-Royal et Avenue Du Parc, Boulevard Gouin et Boulevard Laurentien, Rue Notre-Dame et Rue des Seigneurs. These intersections did not play a significant role in the Poisson distribution model where the pedestrian density was  $>0$ . For instance, in our data set, only 20 well detected images of the intersection Avenue Dollard et Boulevard Newman existed, since the rest of the time the IP

Cameras at this intersection had not been functioning. Thus, this street could not play a significant role in pedestrian density prediction, as is indicated through the model results. The remaining intersections also had a similar scenario, which justifies their results obtained in this model.

## **5.8. Heat Map**

A heat map is a representation in the form of color coding to visualize datasets assisting data viewers making it easier to know which areas should get most attention. It makes the data set easy to assimilate and make decisions from. However, there are several ways of generating heat maps, one of them being google analytics. Since we are already using cloud services, restricting security issues of public data or exploitation, it is convenient to generate heat map using the CSV as shown in the figure.

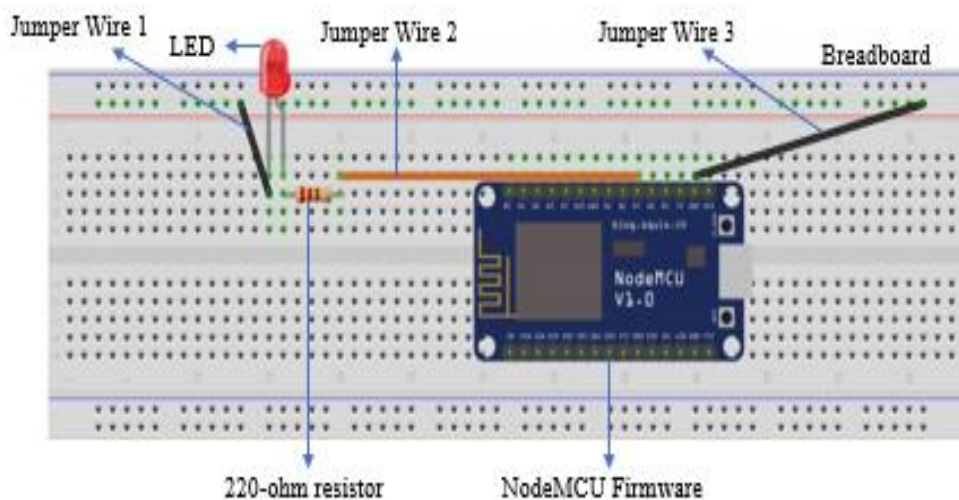
The heat map as shown in figure 5.15, shows the segregation of data based on number of pedestrians and vehicles at the intersection at a given time. The red marking shows that 0 pedestrians are observed at a time, yellow shows an increase by 1 or 2 pedestrian presence at the intersection, light green shows more pedestrians are observed and dark green signifies maximum number of pedestrians at any given time at an intersection. This heat map allows us to predict during what times we can expect an increase in vehicles as well as pedestrians. Hence, this heat map will be essential for the applications to operate and allow the facilities to identify the timings and locations when providing solutions using the prototype.

Number of Pedestrian	Number of Vehicles	Name of Intersection	Time	Date	Number of Pedestrian	Number of Vehicles	Name of Intersection	Time	Date
0	1	Boulevard De Maisonneuve et boulevard Saint	03:08	5/5/2020	2	6	90e Avenue et boulevard Newman	17:56	6/5/2020
1	1	Boulevard Shevchenko et boulevard De La Ver	03:08	5/5/2020	0	3	Boulevard Gouin et boulevard Jacques-Bizard	17:56	6/5/2020
0	1	Rue Beaubien et rue Saint-Denis	03:08	5/5/2020	2	2	Boulevard Maurice-Duplessis et boulevard Ro	17:56	6/5/2020
1	7	Rue de la Montagne et Rue Sainte-Catherine	03:08	5/5/2020	0	3	Boulevard Sainte-Anne-de-Bellevue et rue Sair	17:56	6/5/2020
0	3	Rue Guy et Rue Sainte-Catherine	03:08	5/5/2020	4	3	Rue Beaubien et avenue Christophe-Colomb	17:56	6/5/2020
1	4	Avenue Papineau et boulevard Saint-Joseph	03:08	5/5/2020	0	2	Rue Beaubien et avenue Papineau	17:56	6/5/2020
0	1	Boulevard Robert-Bourassa et rue Saint-Antoir	03:08	5/5/2020	0	3	Rue Clement et rue des Oblats	17:56	6/5/2020
0	1	Rue Belanger et rue Viau	03:08	5/5/2020	0	3	Rue Lacordaire et boulevard Rosemont	17:56	6/5/2020
1	0	Rue Charlevoix et rue Notre-Dame	03:08	5/5/2020	1	10	Boulevard de Pierrefonds et boulevard Saint-C	18:04	6/5/2020
0	1	Rue Notre-Dame et rue des Seigneurs	03:08	5/5/2020	6	1	Rue Guy et boulevard De Maisonneuve	18:04	6/5/2020
0	1	32e Avenue et rue Provost	03:08	5/5/2020	0	1	81e Avenue et Rue Notre-Dame	18:04	6/5/2020
0	2	Boulevard Maurice-Duplessis et boulevard Ro	03:08	5/5/2020	1	1	Avenue Atwater et rue Saint-Antoine	18:04	6/5/2020
0	4	Boulevard Sainte-Anne-de-Bellevue et rue Sair	03:08	5/5/2020	3	6	Avenue du Parc et avenue Saint-Viateur	18:04	6/5/2020
0	1	Boulevard de Pierrefonds et boulevard Saint-C	03:16	5/5/2020	0	8	Avenue McEachran et avenue Van horn	18:04	6/5/2020
0	2	Rue Guy et boulevard De Maisonneuve	03:16	5/5/2020	3	2	Boulevard De Maisonneuve et boulevard Saint	18:04	6/5/2020
0	2	Avenue McEachran et avenue Van horn	03:16	5/5/2020	0	2	Boulevard Gouin et Sortie A-13 (int. Est) boule	18:04	6/5/2020
0	2	Boulevard Saint-Laurent et avenue Viger	03:16	5/5/2020	0	1	Boulevard Robert-Bourassa et Rue Sherbrooke	18:04	6/5/2020
1	0	Boulevard Shevchenko et boulevard De La Ver	03:16	5/5/2020	1	3	Boulevard Saint-Laurent et avenue Viger	18:04	6/5/2020
0	1	Rue Beaubien et rue Saint-Denis	03:16	5/5/2020	2	2	Boulevard Shevchenko et boulevard De La Ver	18:04	6/5/2020
0	1	Rue Crescent et boulevard De Maisonneuve	03:16	5/5/2020	8	2	Rue Beaubien et rue Saint-Denis	18:04	6/5/2020
1	4	Rue de la Montagne et Rue Sainte-Catherine	03:16	5/5/2020	1	1	Rue Crescent et boulevard De Maisonneuve	18:04	6/5/2020
1	2	Rue Guy et Rue Sainte-Catherine	03:16	5/5/2020	5	0	Rue De Bleury et Rue Sainte-Catherine	18:04	6/5/2020
0	2	90e Avenue et boulevard Newman	03:16	5/5/2020	2	1	Rue de la Montagne et Rue Sainte-Catherine	18:04	6/5/2020
0	1	Avenue Atwater et rue Notre-Dame	03:16	5/5/2020	5	5	Rue Guy et Rue Sainte-Catherine	18:04	6/5/2020
0	1	Avenue des Pins et Rue Saint-Urbain	03:16	5/5/2020	4	2	Rue Jeanne-Mance et avenue du President-Ke	18:04	6/5/2020
0	3	Avenue Papineau et boulevard Saint-Joseph	03:16	5/5/2020	2	2	Avenue Atwater et rue Notre-Dame	18:04	6/5/2020
0	1	Boulevard Robert-Bourassa et rue Saint-Antoir	03:16	5/5/2020	1	6	Avenue des Pins et Rue Saint-Urbain	18:04	6/5/2020
0	1	Rue Beaubien et avenue Papineau	03:16	5/5/2020	0	2	Avenue Papineau et boulevard Rosemont	18:04	6/5/2020
0	1	Rue Notre-Dame et rue des Seigneurs	03:16	5/5/2020	0	5	Avenue Papineau et boulevard Saint-Joseph	18:04	6/5/2020
0	2	Boulevard Maurice-Duplessis et boulevard Ro	03:16	5/5/2020	2	2	Boulevard Robert-Bourassa et rue Saint-Antoir	18:04	6/5/2020
0	3	Boulevard Sainte-Anne-de-Bellevue et rue Sair	03:16	5/5/2020	0	2	Rue Belanger et rue Viau	18:04	6/5/2020
0	1	Boulevard de Pierrefonds et boulevard Saint-C	03:24	5/5/2020	1	3	Rue Charlevoix et rue Notre-Dame	18:04	6/5/2020
1	1	Rue Guy et boulevard De Maisonneuve	03:24	5/5/2020	1	3	Rue de la Gauchetiere et rue Peel	18:04	6/5/2020
0	4	Avenue McEachran et avenue Van horn	03:24	5/5/2020	0	1	Rue Dickson et Rue Sherbrooke	18:04	6/5/2020
0	1	Boulevard Gouin et boulevard Laurentien	03:24	5/5/2020	0	5	Rue Lacordaire et boulevard Rosemont	18:04	6/5/2020
1	0	Boulevard Shevchenko et boulevard De La Ver	03:24	5/5/2020	0	4	Rue Notre-Dame et rue des Seigneurs	18:04	6/5/2020

Fig. 5.15. Heat Map showing low and high pedestrian counts as color codes

## The Circuit

Figure 5.16 shows circuit with the microcontroller, LED, Male to female jumper wires, 3.3V resistor and a breadboard



**Fig. 5.16. :** Circuit with the microcontroller, LED, Male to female jumper wires, 3.3V resistor and a breadboard

The circuit consists of ESP8266 development board, LED, 220-ohm resistor and a few jumper wires. In figure 5.16, the LED module is on the right side. The LED has a long and a short leg. The longer one is the positive of LED and the shorter one is the negative of LED. Two pins of the development board are being used. The jumper wire 1 is connected to the negative of the breadboard as the negative is available in the entire row of the breadboard. The second pin which is being used is D7 which is the GPIO pin. The longer leg of the LED is on the right side and on the positive, and the shorter leg is negative and on the left-hand side. The positive of the LED is connected to the 220-ohm resistor and from the resistor a wire is connected to

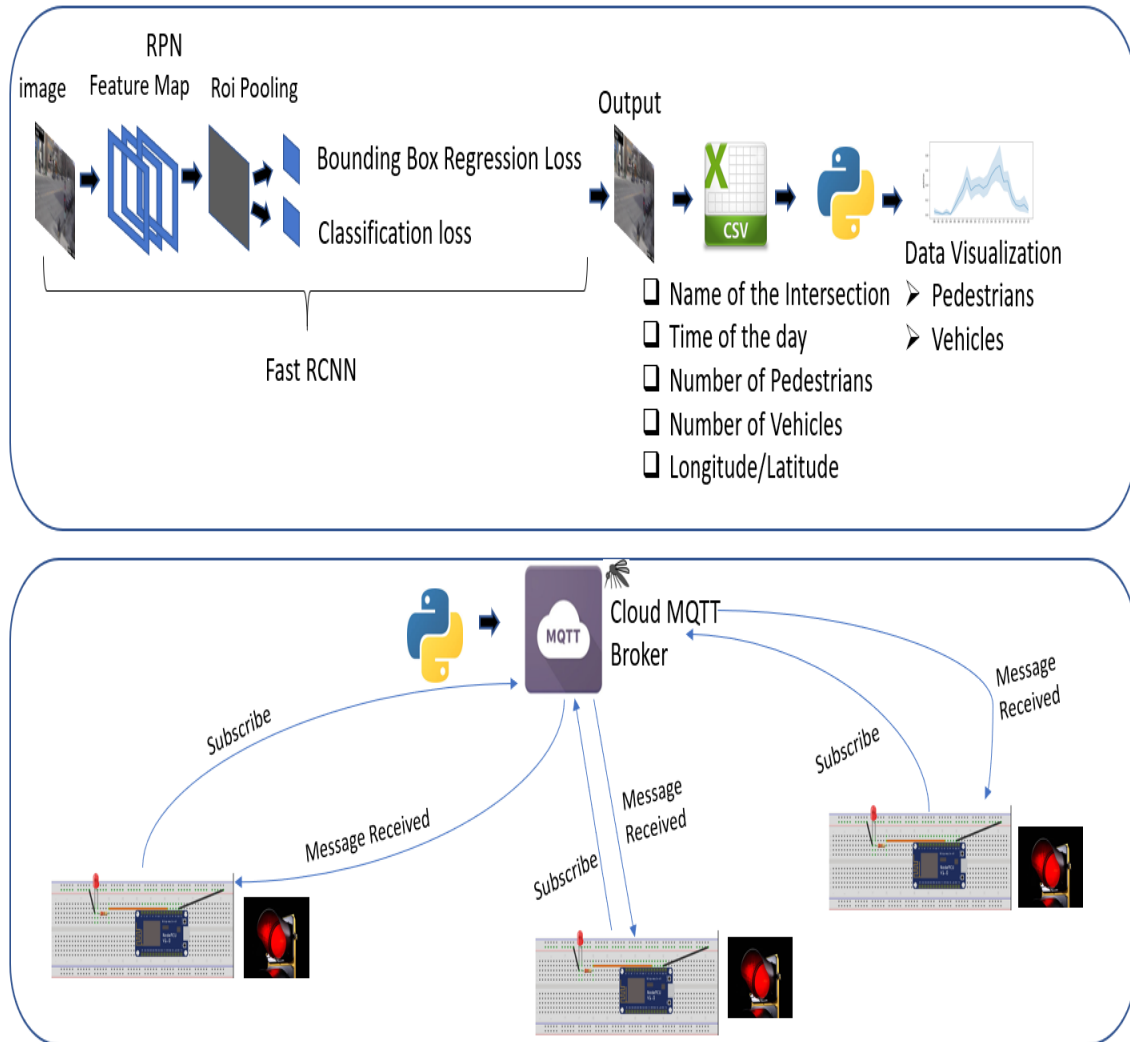
D7 which is GPIO pin. The negative end of the LED is connected to the negative of the breadboard.

To execute our program using Lua script an IDE for ESP8266 was used, for this purpose JDK 7 was installed as a prerequisite. ESPlorer was installed. The COM port is required to set as soon as it connects to the computer. The Baud rate is set to 115200. Then the NodeMCU is flashed. After this the commands were inputted into the ESPlorer and transferred to the microcontroller

## **5.9. Architecture & Outputs**

Figure 5.17 shows schematic diagram of prototype architecture.

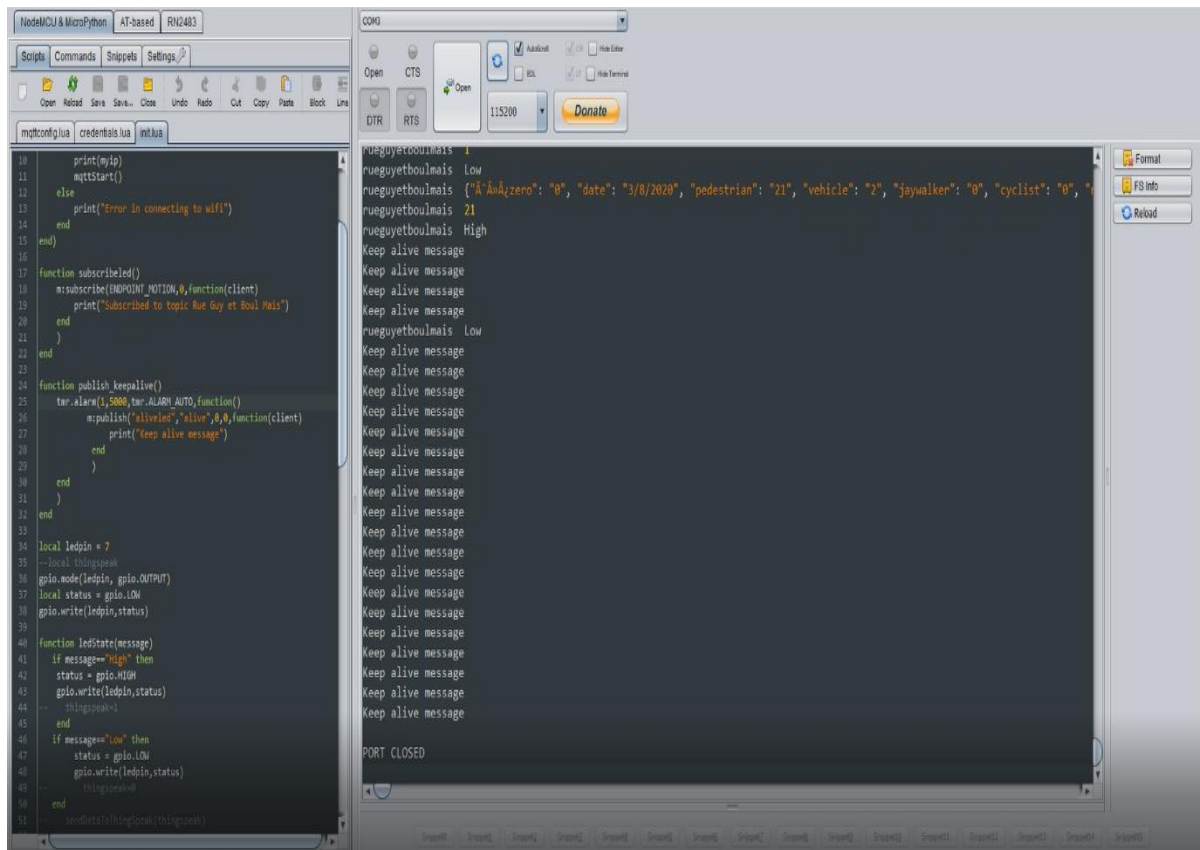




**Fig. 5.17. Prototype Architecture**

This includes how the data received after analysis and visualization is sent on to the cloud. An application in python was written to send this data to the specific cloud MQTT instance from where it will be possible to control the subscribed microcontrollers. Initially, the data that was collected was sent via a python application onto cloud MQTT (Define). An instance was created on cloud MQTT, and ESP8266, a microcontroller, was programmed to be subscribed to our topic onto this instance. Then, on the cloud MQTT, multiple topics were created for each intersection to control devices associated with that intersection. For instance, in this research two intersections were

considered for the prototype, Rue Guy et Boulevard De Maisonneuve and 32e Avenue et Rue Provost. 2 ESP8266 chips were taken, one was subscribed to the topic of Rue Guy et boulevard De Maisonneuve and the other one to the topic 32e Avenue et Rue Provost. Once the messages were received on the cloud MQTT instance, on these particular topics, the ESP8266 microcontrollers were programmed, such that only the microcontroller subscribed to that topic could receive messages for that intersection of which that particular topic received the message. So for instance when the message would arise on the Rue Guy and Boulevard de Maisonneuve topic, only the microcontroller that is subscribed to and corresponds to this intersection would accordingly be controlled. Pedestrian values were sent onto these topics and the threshold values were set to 5 meaning that if the pedestrian count is more than 5, then the microcontroller should send command to the corresponding LED to turn on. This mimics the traffic safety system where if the pedestrian count  $> 5$  then the microcontroller should turn on and corresponding LED should turn on. Since it was impossible to control the traffic lights, the prototype was implemented using microcontroller and LED. Every time the pedestrian threshold went over 5, the microcontroller subscribed to that topic and would turn the corresponding LED on. A “High Volume” message would be received at the microcontroller.



**Fig. 5.18. Esplora IDE**

The NodeMCU firmware is first installed on the ESP8266 chip.

This figure 5.18 shows the Esplora IDE which was used for programming the ESP8266 microcontroller in the prototype. The code written in Lua is pushed onto the microcontroller. The code written in Lua involves subscribing to the topic of the particular intersection for which this microcontroller would be responsible in the prototype. Once the microcontroller is subscribed to the MQTT instance and the particular topic, it starts receiving the messages published onto that topic by the python application.

The figure 5.19 shows the MQTT instance websocket showing the messages received on each topic of this instance.

## Received messages



Topic	Message
32eAvenueetrueProvost	{\"@zero\": \"0\", \"date\": \"3/8/2020\", \"pedestrian\": \"0\", \"vehicle\": \"2\", \"jaywalker\": \"0\", \"cyclist\": \"0\", \"name\": \"32e Avenue et rue Provost\", \"time\": \"1:00\"}
32eAvenueetrueProvost	0
32eAvenueetrueProvost	Low
32eAvenueetrueProvost	{\"@zero\": \"0\", \"date\": \"3/8/2020\", \"pedestrian\": \"0\", \"vehicle\": \"2\", \"jaywalker\": \"0\", \"cyclist\": \"0\", \"name\": \"32e Avenue et rue Provost\", \"time\": \"1:05\"}
32eAvenueetrueProvost	0
32eAvenueetrueProvost	Low
32eAvenueetrueProvost	{\"@zero\": \"0\", \"date\": \"3/8/2020\", \"pedestrian\": \"0\", \"vehicle\": \"2\", \"jaywalker\": \"0\", \"cyclist\": \"0\", \"name\": \"32e Avenue et rue Provost\", \"time\": \"1:10\"}
32eAvenueetrueProvost	0
32eAvenueetrueProvost	Low
32eAvenueetrueProvost	{\"@zero\": \"0\", \"date\": \"3/8/2020\", \"pedestrian\": \"0\", \"vehicle\": \"2\", \"jaywalker\": \"0\", \"cyclist\": \"0\", \"name\": \"32e Avenue et rue Provost\", \"time\": \"1:15\"}
32eAvenueetrueProvost	0
32eAvenueetrueProvost	Low
rueguyetboulmais	{\"@zero\": \"0\", \"date\": \"3/8/2020\", \"pedestrian\": \"4\", \"vehicle\": \"1\", \"jaywalker\": \"0\", \"cyclist\": \"0\", \"name\": \"Rue Guy et boulevard De Maisonneuve\", \"time\": \"1:00\"}
rueguyetboulmais	4
rueguyetboulmais	Low
rueguyetboulmais	{\"@zero\": \"0\", \"date\": \"3/8/2020\", \"pedestrian\": \"4\", \"vehicle\": \"1\", \"jaywalker\": \"0\", \"cyclist\": \"0\", \"name\": \"Rue Guy et boulevard De Maisonneuve\", \"time\": \"1:05\"}
rueguyetboulmais	4
rueguyetboulmais	Low
rueguyetboulmais	{\"@zero\": \"0\", \"date\": \"3/8/2020\", \"pedestrian\": \"1\", \"vehicle\": \"2\", \"jaywalker\": \"0\", \"cyclist\": \"0\", \"name\": \"Rue Guy et boulevard De

32eAvenueetRueProvost	0
32eAvenueetRueProvost	Low
32eAvenueetRueProvost	{ "i": "zero", "date": "3/8/2020", "pedestrian": "0", "vehicle": "2", "jaywalker": "0", "cyclist": "0", "name": "32e Avenue et rue Provost", "time": "1:15" }
32eAvenueetRueProvost	0
32eAvenueetRueProvost	Low
rueguyetboulmais	{ "i": "zero", "date": "3/8/2020", "pedestrian": "4", "vehicle": "1", "jaywalker": "0", "cyclist": "0", "name": "Rue Guy et boulevard De Maisonneuve", "time": "1:00" }
rueguyetboulmais	4
rueguyetboulmais	Low
rueguyetboulmais	{ "i": "zero", "date": "3/8/2020", "pedestrian": "4", "vehicle": "1", "jaywalker": "0", "cyclist": "0", "name": "Rue Guy et boulevard De Maisonneuve", "time": "1:05" }
rueguyetboulmais	4
rueguyetboulmais	Low
rueguyetboulmais	{ "i": "zero", "date": "3/8/2020", "pedestrian": "1", "vehicle": "2", "jaywalker": "0", "cyclist": "0", "name": "Rue Guy et boulevard De Maisonneuve", "time": "1:10" }
rueguyetboulmais	1
rueguyetboulmais	Low
rueguyetboulmais	{ "i": "zero", "date": "3/8/2020", "pedestrian": "10", "vehicle": "2", "jaywalker": "0", "cyclist": "0", "name": "Rue Guy et boulevard De Maisonneuve", "time": "1:15" }
rueguyetboulmais	10
rueguyetboulmais	High

## CloudMQTT

**Fig. 5.17. Cloud MQTT instance receiving messages on different intersection topics**

For example, the topic 32eAvenueetRueProvost shows the messages published onto the topic for the intersection 32e Avenue et Rue Provost. The microcontroller subscribed to this topic would immediately receive this message and act accordingly. Since the messages received on this topic are ‘Low’, it indicates that the pedestrian count is below the threshold, and thus the microcontroller

would turn the LED in its circuit 'Off'. The figure 5.17 shows the message received as 'High'. During such a circumstance, the microcontroller would receive the 'High' message and turn the LED on, thus mimicking a traffic light system, which can be extremely useful in ensuring pedestrian safety.

The messages received on the Cloud MQTT instance suffered minimal to no delay, and the attached microcontroller was also able to control the LED with minimal/no delay. This is because the CloudMQTT platform is extremely efficient and the ESP8266 microcontroller is also extremely robust, efficient, and reliable. The only minimal delays were due to the internet speed of the Wi-Fi to which the microcontroller was connected. However, high speed internet connections showed delays of the order of few milliseconds.

## **5.10. Conclusion**

This chapter focusses on a step-by-step implementation of the proposed solution in this thesis. First, image and video data are collected, which is the most crucial step to the solution. Object detection techniques using TensorFlow are applied to this data and the obtained results are visualized and analyzed in detail in order to identify the various risks posed to pedestrians. This visualized and analyzed data is then utilized in the proposed cloud-based system that allows for enhancement of pedestrian safety by controlling the traffic management system in an effective manner. The implementation of the proposed cloud-based prototype is explored in-depth and its results are analyzed.

# Chapter 6

## Conclusion and Future Scope

In this chapter, initially, the contributions of this thesis are summarized. This is followed by a focus on the possible future scope and research direction of this thesis.

### 6.1. Contributions Summary

Pedestrian safety is an increasing concern all over the World, especially in areas which are densely populated. This work analyses many patterns and risks to pedestrian safety that have recently been observed, and the solution proposed aims to handle these pressing concerns as well. Although many systems exist to enhance pedestrian safety, the system proposed in this work provides a solution that is not only low-cost and effective but is extremely reliable during emergencies. When implemented on a higher scale, this system can prove to be a boon to the traffic management system and prevent accidents that otherwise could not have been prevented.

In the early days-controlled signals were used after which the traffic signals came into use. Over the period these traffic signals are centrally controlled by systems that are coordinated in real time to changing traffic patterns. Now, to understand the traffic patterns video analytic techniques to recognize the traffic patterns play a major role across cities. Studies have shown that 40% of the accidents or collisions take place at intersections when drivers often try to beat a red light or disobey traffic rules due to poorly timed and absent traffic signals at the intersections.

This work presents an efficient solution that can aid in enhancing pedestrian safety in a cost effective, and reliable way. The issue of pedestrian safety is of utmost importance as the traditional

systems for pedestrian safety pose many limitations and disadvantages that render them relatively ineffective. The technique proposed in this paper aims to eliminate those limitations and provide a reliable and extremely efficient system. This camera data collected for each street by Montreal's government is used to train a neural network using object detection and Tensorflow, which then identifies pedestrian patterns that are observed during each time of the day. The neural network is trained to identify the number of pedestrians in the streets, peak hours at which pedestrian density is maximum, streets which have most pedestrian density throughout the day, and various other patterns. This data can then control the Pedestrian and traffic lights, issue warnings onto the pedestrians' smartphones in the vicinity of the risk area etc. This proposed solution has been compared and contrasted with several other solutions that currently exist. Its efficiency, reliability, and cost-effectiveness make it a better alternative than most other works.

## **6.2. Future Scope and Research Direction**

The work also possesses great future scope. In figure 6.1 the future scope of the proposed system is presented. This technique can also provide information and alerts to emergency services and other higher authorities if further developed as shown in the figure below. This not only makes it ideal for pedestrian safety, but also aids in developing Smart Cities. Video analysis of data from the cameras placed at intersections will give real-time information to further control the devices. The wifi enabled module of the ESP8266 is also continuously developed to establish connection with 4G LTE.



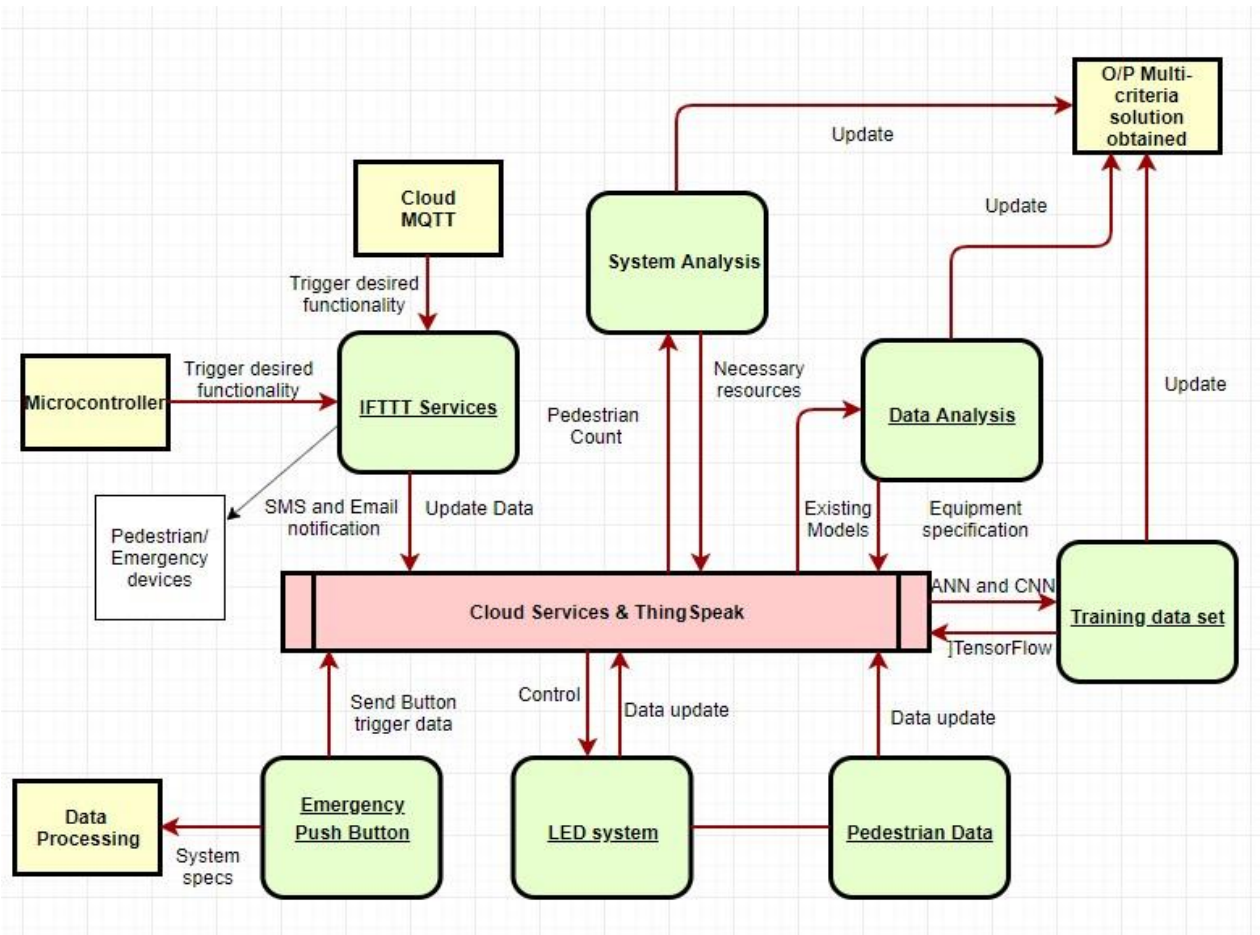


Fig. 6.1. Future scope of the the proposed system

# Bibliography

1. Abhishek.P, Md. Adil, Md. Javed H. Taiyabashadaka Rafa Pedestrian-safer IoT-based Smart Crossing System with Object Tracking. (2020). *International Journal of Recent Technology And Engineering*, 9(1), 1948-1953. doi: 10.35940/ijrte.a2710.059120
2. A. Kumari and B. Satish, "Automated traffic control for pedestrian safety - IEEE Conference Publication", *Ieeexplore.ieee.org*, 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/7975589>. [Accessed: 10-Jun-2019].
3. Accident data map. Retrieved 7 January 2020, from [http://www.montrealgazette.com/news/road-safety/map/map\\_page.html](http://www.montrealgazette.com/news/road-safety/map/map_page.html)
4. Ali Tayarani, Z. (2020). Artificial Neural Networks Analysis Used to Evaluate the Molecular Interactions between Selected Drugs and Human Cyclooxygenase2 Receptor. Retrieved 28 August 2020, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3909632/>
5. Allhoff, F., & Henschke, A. (2018). The Internet of Things: Foundational ethical issues. *Internet Of Things*, 1-2, 55-66. doi: 10.1016/j.iot.2018.08.005
6. Al-Turjman, F., & Malekloo, A. (2019). Smart parking in IoT-enabled cities: A survey. *Sustainable Cities And Society*, 49, 101608. doi: 10.1016/j.scs.2019.101608
7. Andročec, D., & Vrček, N. (2018). Machine Learning for the Internet of Things Security: A Systematic Review. Proceedings Of The 13Th International Conference On Software Technologies. doi: 10.5220/0006841205630570
8. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 27872805. doi: 10.1016/j.comnet.2010.05.010
9. Bagheri, M., Siekkinen, M., & Nurminen, J. (2016). Cloud-Based Pedestrian Road-Safety with Situation-Adaptive Energy-Efficient Communication. *IEEE Intelligent Transportation Systems Magazine*, 8(3), 45-62. doi: 10.1109/imits.2016.2573338
10. Brownlee, J. (2020). A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. Retrieved 26 August 2020, from [https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/#:~:text=For%20example%2C%20a%20pooling%20layer,%C3%973%20\(9%20pixels\).&text=Maximum%20Pooling%20\(or%20Max%20Pooling.patch%20of%20the%20feature%20map.work](https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/#:~:text=For%20example%2C%20a%20pooling%20layer,%C3%973%20(9%20pixels).&text=Maximum%20Pooling%20(or%20Max%20Pooling.patch%20of%20the%20feature%20map.work).

11. (2020). Retrieved 9 May 2020, from <https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529>
12. Bellavista, P., Berrocal, J., Corradi, A., Das, S., Foschini, L., & Zanni, A. (2019). A survey on fog computing for the Internet of Things. *Pervasive And Mobile Computing*, 52, 71-99. doi: 10.1016/j.pmcj.2018.12.007
13. Cai, H., Xu, B., Jiang, L., & Vasilakos, A. (2016). IoT-based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges. *IEEE Internet Of Things Journal*, 1-1. doi: 10.1109/jiot.2016.2619369
14. Cao, Y., Jiang, T., & Han, Z. (2016). A Survey of Emerging M2M Systems: Context, Task, and Objective. *IEEE Internet Of Things Journal*, 3(6), 1246-1258. doi: 10.1109/jiot.2016.2582540
15. Chiang, M., & Zhang, T. (2016). Fog and IoT: An Overview of Research Opportunities. *IEEE Internet Of Things Journal*, 3(6), 854-864. doi: 10.1109/jiot.2016.2584538
16. Cui, L., Yang, S., Chen, F., Ming, Z., Lu, N., & Qin, J. (2018). A survey on application of machine learning for Internet of Things. *International Journal of Machine Learning And Cybernetics*, 9(8), 1399-1417. doi: 10.1007/s13042-018-0834-5
17. CS231n Convolutional Neural Networks for Visual Recognition. (2020). Retrieved 28 August 2020, from <https://cs231n.github.io/neural-networks-1/>
18. Desai, A., & Jhaveri, R. (2019). The Role of Machine Learning in Internet-of-Things (IoT) Research: A Review. Retrieved 22 September 2019, from <https://www.semanticscholar.org/paper/The-Role-of-Machine-Learning-in-Internet-of-Things-Desai-Jhaveri/69c13f18d2052687bdb42a7ca88d11e243fa110a>
19. Donahue, J., Hendricks, L., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., & Darrell, T. (2019). Long-term Recurrent Convolutional Networks for Visual Recognition and Description. Retrieved 30 August 2019, from <https://arxiv.org/abs/1411.4389v4>
20. Environment - Level Crossing Model. Retrieved 22 September 2019, from [https://www.researchgate.net/publication/259864678\\_Multi-agent\\_Approach\\_to\\_Traffic\\_Simulation\\_in\\_NetLogo\\_Environment\\_-\\_Level\\_Crossing\\_Model](https://www.researchgate.net/publication/259864678_Multi-agent_Approach_to_Traffic_Simulation_in_NetLogo_Environment_-_Level_Crossing_Model)
21. Ertam, F., & Aydin, G. (2017). Data classification with deep learning using Tensorflow. 2017 International Conference On Computer Science And Engineering (UBMK). doi: 10.1109/ubmk.2017.8093521
22. Estimation of intersection traffic density on decentralized architectures with deep networks - IEEE Conference Publication. (2020). Retrieved 3 July 2020, from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8090799>

23. G. Pau, T. Campisi, A. Canale, A. Severino, M. Collotta and G. Tesoriere, "Smart Pedestrian Crossing Management at Traffic Light Junctions through a Fuzzy-Based Approach", *Semanticscholar.org*, 2019. [Online]. Available: <https://www.semanticscholar.org/paper/Smart-Pedestrian-Crossing-Management-at-Traffic-a-PauCampisi/ba898fa87af917316e7884e798971611bef69f0d>. [Accessed: 10- Jun-2019].
24. 50 идей для благоустройства города своими руками. Часть 1 - Kislrod.io. (2020). Retrieved 28 August 2020, from <https://kislrod.io/prosto-o-slozhnom/50-idej-dla-blagoustrojstva-goroda-svoimi-rukami-chast-1/>
25. Giovanni Pau, Tiziana C., Antonino C., Alessandro S., Smart Pedestrian Crossing Management at Traffic Light Junctions through a Fuzzy-Based Approach. (2018). *Future Internet*, 10(2), 15. doi: 10.3390/fi10020015
26. González García, C., Meana-Llorián, D., Pelayo G-Bustelo, B., Cueva Lovelle, J., & Garcia-Fernandez, N. (2017). Midgar: Detection of people through computer vision in the Internet of Things scenarios to improve the security in Smart Cities, Smart Towns, and Smart Homes. *Future Generation Computer Systems*, 76, 301-313. doi: 10.1016/j.future.2016.12.033
27. Govindarajulu, P., & Ezhumalai, P. (2019). In-Vehicle Intelligent Transport System for Preventing Road Accidents Using Internet of Things. Retrieved 22 August 2019, from <https://www.semanticscholar.org/paper/In-Vehicle>
28. Histogram of Oriented Gradients — Image Processing and Computer Vision 2.0 documentation. (2020). Retrieved 28 June 2020, from <https://staff.fnwi.uva.nl/r.vandenboomgaard/IPC20172018/20172018/LabExercises/HOG.html>
29. Hannah, Charlotte & Spasić Spasić, Irena & Corcoran, Pdraig. (2018). *Modelling Pedestrian Safety with respect to Road Traffic Crashes by Estimating the Safety of Paths*. (2019). Retrieved 22 August 2019, from <https://www.fhwa.dot.gov/publications/research/safety/99090/99090.pdf>
30. Hussein, M., Sayed, T., Reyad, P., & Kim, L. (2015). Automated Pedestrian Safety Analysis at a Signalized Intersection in New York City: Automated Data Extraction for Safety Diagnosis and Behavioral Study. *Transportation Research Record: Journal of The Transportation Research Board*, 2519(1), 17-27. doi: 10.3141/2519-03
31. Informatics, S. (2019). Applied Soft Computing. Retrieved 22 September 2019, from <https://www.journals.elsevier.com/applied-soft-computing/former-call-for-papers/special-issue-on-applying-machine-learning-systems-for-iot>
32. Intelligent-Transport-System-for-Road-of-GovindarajuluEzhumalai/eed17950db77feb7d45780a8323830a16bc7eb5a Taha, A.

- (2018). An IoT Architecture for Assessing Road Safety in Smart Cities. *Wireless Communications And Mobile Computing*, 2018, 1-11. doi: 10.1155/2018/8214989
33. Janota, A., Rastocny, K., & Zahradník, J. (2005). Multi-agent Approach to Traffic Simulation in NetLogo
34. Javaid, S., Sufian, A., Pervaiz, S., & Tanveer, M. (2019). Smart traffic management system using Internet of Things - IEEE Conference Publication. Retrieved 22 August 2019, from <https://ieeexplore.ieee.org/document/8323770>
35. K. An, S. Lee, Y. Jeong and D. Seo, 2019. [Online]. Available: [https://www.researchgate.net/publication/321948233\\_Pedestrian-Safe\\_Smart\\_Crossing\\_System\\_Based\\_on\\_IoT\\_with\\_Object\\_Tracking](https://www.researchgate.net/publication/321948233_Pedestrian-Safe_Smart_Crossing_System_Based_on_IoT_with_Object_Tracking). [Accessed: 10- Jun- 2019].
36. Kouicem, D., Bouabdallah, A., & Lakhlef, H. (2018). Internet of things security: A top-down survey. *Computer Networks*, 141, 199-221. doi: 10.1016/j.comnet.2018.03.012
37. la..., "6 aménagements pour accroître la mobilité et la sécurité des piétons | 100°", 100°, 2019. [Online]. Available: <https://centdegres.ca/magazine/amenagement/6-amenagements-pour-accroitre-la-mobilite-et-lasecurite-des-pietons/>. [Accessed: 10- Jun- 2019].
38. Lacinák, M., & Ristvej, J. (2017). Smart City, Safety and Security. *Procedia Engineering*, 192, 522-527. doi: 10.1016/j.proeng.2017.06.090
39. Lanke, N., & Koul, S. (2013). Smart Traffic Management System. *International Journal Of Computer Applications*, 75(7), 19-22. doi: 10.5120/13123-0473
40. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. doi: 10.1038/nature14539
41. Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., & Hays, J. et al. (2019). Microsoft COCO: Common Objects in Context. Retrieved 22 September 2019, from <https://arxiv.org/pdf/1405.0312.pdf>
42. Liu, T., & Stathaki, T. (2018). Faster R-CNN for Robust Pedestrian Detection Using Semantic Segmentation Network. *Frontiers In Neurorobotics*, 12. doi: 10.3389/fnbot.2018.00064
43. Ma, Y., Lu, S., & Zhang, Y. (2020). Analysis on Illegal Crossing Behavior of Pedestrians at Signalized Intersections Based on Bayesian Network. *Journal Of Advanced Transportation*, 2020, 1-14. doi: 10.1155/2020/2675197

44. Mahdavinejad, M., Rezvan, M., Barekatin, M., Adibi, P., Barnaghi, P., & Sheth, A. (2018). Machine learning for internet of things data analysis: a survey. *Digital Communications And Networks*, 4(3), 161-175. doi: 10.1016/j.dcan.2017.10.002
45. Mandhare, P., Kharat, V., & Patil, C. (2018). Intelligent Road Traffic Control System for Traffic Congestion A Perspective. *International Journal Of Computer Sciences And Engineering*, 6(7), 908-915. doi: 10.26438/ijcse/v6i7.908915
46. Meetiayagoda, L. (2018). Pedestrian safety in Kandy Heritage City, Sri Lanka: Lessons from World Heritage Cities. *Sustainable Cities And Society*, 38, 301-308. doi: 10.1016/j.scs.2018.01.017
47. Moh, M., & Raju, R. (2018). Machine Learning Techniques for Security of Internet of Things (IoT) and Fog Computing Systems. 2018 International Conference On High Performance Computing & Simulation (HPCS). doi: 10.1109/hpcs.2018.00116
48. Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. *2015 IEEE Conference On Computer Vision And Pattern Recognition (CVPR)*. doi: 10.1109/cvpr.2015.7299101
49. Ngu, A., Gutierrez, M., Metsis, V., Nepal, S., & Sheng, M. (2016). IoT Middleware: A Survey on Issues and Enabling technologies. *IEEE Internet Of Things Journal*, 1-1. doi: 10.1109/jiot.2016.2615180
50. Noh, B., No, W., Lee, J., & Lee, D. (2020). Vision-Based Potential Pedestrian Risk Analysis on Unsignalized Crosswalk Using Data Mining Techniques. *Applied Sciences*, 10(3), 1057. doi: 10.3390/app10031057
51. Nord, J., Koohang, A., & Paliszkievicz, J. (2019). The Internet of Things: Review and theoretical framework. *Expert Systems With Applications*, 133, 97-108. doi: 10.1016/j.eswa.2019.05.014
52. NumPy. (2020). Retrieved 7 August 2020, from <https://numpy.org/#:~:text=NumPy%20supports%20a%20wide%20range,GPU%2C%20a%20sparse%20array%20libraries.&text=NumPy's%20high%20level%20syntax%20makes,any%20background%20or%20experience%20level.>
53. Omar, H. (2015). Intelligent Traffic Information System Based on Integration of Internet of Things and Agent Technology. *International Journal Of Advanced Computer Science And Applications*, 6(2). doi: 10.14569/ijacsa.2015.060206
54. Pimpinella, A., Redondi, A., & Cesana, M. (2019). Walk this way! An IoT-based urban routing system for smart cities. *Computer Networks*, 162, 106857. doi: 10.1016/j.comnet.2019.07.013

55. Pop, D., Rogozan, A., Chatelain, C., Nashashibi, F., & Bensrhair, A. (2019). Multi-Task Deep Learning for Pedestrian Detection, Action Recognition and Time to Cross Prediction. *IEEE Access*, 7, 149318-149327. doi: 10.1109/access.2019.2944792
56. Precision and Recall - Custom Plugin - Supervisely. (2020). Retrieved 2 December 2019, from <https://supervise.ly/explore/plugins/precision-and-recall-75278/overview>
57. Padding and Stride — Dive into Deep Learning 0.14.3 documentation. (2020). Retrieved 26 August 2020, from [https://d2l.ai/chapter\\_convolutional-neural-networks/padding-and-strides.html](https://d2l.ai/chapter_convolutional-neural-networks/padding-and-strides.html)
58. Python Libraries for Data Science You Should Know – Dataquest. (2020). Retrieved 28 August 2020, from <https://www.dataquest.io/blog/15-python-libraries-for-data-science/>
59. Qu, H., Wang, M., Zhang, C., & Wei, Y. (2018). A Study on Faster R-CNN-Based Subway Pedestrian Detection with ACE Enhancement. *Algorithms*, 11(12), 192. doi: 10.3390/a11120192
60. R-CNN for Object detection. (2019). Retrieved 7 August 2020, from <https://towardsdatascience.com/r-cnn-for-object-detection-a-technical-summary-9e7bfa8a557c>
61. Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 39(6), 1137-1149. doi: 10.1109/tpami.2016.2577031
62. Repko, J., & DeBroux, S. (2012). (PDF) Smart Cities Literature Review and Analysis. Retrieved 7 August 2020, from [https://www.researchgate.net/publication/236685572\\_Smart\\_Cities\\_Literature\\_Review\\_and\\_Analysis](https://www.researchgate.net/publication/236685572_Smart_Cities_Literature_Review_and_Analysis)
63. Retrieved July 14,2020 from <https://saaq.gouv.qc.ca/fileadmin/documents/publications/detailed-profile-facts-statistics-pedestrians.pdf>
64. Rizwan, P., Suresh, K., & Babu, M. (2016). Real-time smart traffic management system for smart cities by using Internet of Things and big data. 2016 International Conference On Emerging Technological Trends (ICETT). doi: 10.1109/icett.2016.7873660
65. S Arnold, L., & Ragland, D. (2009). a) Pilot Model for Estimating Pedestrian Intersection Crossing Volumes. *Transportation Research Record: Journal Of The Transportation Research Board*, 2140(1), 13-26. doi: 10.3141/2140-02  
b) Diogenes, M., Greene-Roesel, R., Arnold, L., & Ragland, D. (2007). Pedestrian Counting Methods at Intersections. *Transportation Research Record: Journal Of The Transportation Research Board*, 2002(1), 26-30. doi: 10.3141/2002-04
66. Shanzhi Chen, Hui Xu, Dake Liu, Bo Hu, & Hucheng Wang. (2014). A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective. *IEEE Internet Of Things Journal*, 1(4), 349-359. doi: 10.1109/jiot.2014.2337336

67. Sharma, V., & Tiwari, R. (2019). Retrieved 30 August 2019, from <https://pdfs.semanticscholar.org/d778/c2d305badd69ea1ad69edee9aef1a3e80507.pdf>.
68. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet Of Things Journal*, 3(5), 637-646. doi: 10.1109/jiot.2016.2579198
69. Sinha, A., Shrivastava, G., & Kumar, P. (2019). Architecting user-centric internet of things for smart agriculture. *Sustainable Computing: Informatics And Systems*, 23, 88-102. doi: 10.1016/j.suscom.2019.07.001
70. Snippets: Deep Networks for Video Classification. Retrieved 30 August 2019, from <https://arxiv.org/abs/1503.08909v2>.
71. Tang, T., & Ho, A. (2019). A path-dependence perspective on the adoption of Internet of Things: Evidence from early adopters of smart and connected sensors in the United States. *Government Information Quarterly*, 36(2), 321-332. doi: 10.1016/j.giq.2018.09.010
72. Thibaud, M., Chi, H., Zhou, W., & Piramuthu, S. (2018). Internet of Things (IoT) in high-risk Environment, Health and Safety (EHS) industries: A comprehensive review. *Decision Support Systems*, 108, 79-95. doi: 10.1016/j.dss.2018.02.005
73. Tiwari, G., Bangdiwala, S., Saraswat, A., & Gaurav, S. (2007). Survival analysis: Pedestrian risk exposure at signalized intersections. *Transportation Research Part F: Traffic Psychology And Behaviour*, 10(2), 77-89. doi: 10.1016/j.trf.2006.06.002
74. Turner, S., Fitzpatrick, K., Brewer, M., & Park, E. (2006). Motorist Yielding to Pedestrians at Unsignalized Intersections. *Transportation Research Record: Journal Of The Transportation Research Board*, 1982(1), 1-12. doi: 10.1177/0361198106198200102
75. Temporary Traffic Calming Curbs. (2020). Retrieved 8 August 2020, from <https://www.calgary.ca/transportation/roads/traffic/traffic-safety-programs/temporary-traffic-calming-curbs.html>
76. Understanding Smart Cities: An Integrative Framework. (2020). Retrieved 14 July 2020, from [https://www.researchgate.net/publication/254051893\\_Understanding\\_Smart\\_Cities\\_An\\_Integrative\\_Framework](https://www.researchgate.net/publication/254051893_Understanding_Smart_Cities_An_Integrative_Framework)
77. Widyantara, I., & Sastra, N. (2015). Internet of Things for Intelligent Traffic Monitoring System: A Case Study in Denpasar. *International Journal Of Computer Trends And Technology*, 30(3), 169-173. doi: 10.14445/22312803/ijctt-v30p130
78. Wu, Q., Ding, G., Xu, Y., Feng, S., Du, Z., Wang, J., & Long, K. (2014). Cognitive Internet of Things: A New Paradigm Beyond Connection. *IEEE Internet Of Things Journal*, 1(2), 129-143. doi: 10.1109/jiot.2014.2311513



79. Xu, W., Ruiz-Juri, N., Huang, R., Duthie, J., & Clary, J. (2018). Automated pedestrian safety analysis using data from traffic monitoring cameras. *Proceedings Of The 1St ACM/EIGSCC Symposium On Smart Cities And Communities - SCC '18*. doi: 10.1145/3236461.3241972
80. Xu, Y. (2019). Recent Machine Learning Applications to Internet of Things (IoT). Retrieved 22 September 2019, from [https://www.cse.wustl.edu/~jain/cse570-15/ftp/iot\\_ml/](https://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_ml/)
81. Yang, Y., Wu, L., Yin, G., Li, L., & Zhao, H. (2017). A Survey on Security and Privacy Issues in Internet-ofThings. *IEEE Internet Of Things Journal*, 4(5), 1250-1258. doi: 10.1109/jiot.2017.2694844
82. Zantalis, F., Koulouras, G., Karabetsos, S., & Kandris, D. (2019). A Review of Machine Learning and IoT in Smart Transportation. *Future Internet*, 11(4), 94. doi: 10.3390/fi11040094
83. Zhao, H., Sha, J., Zhao, Y., Xi, J., Cui, J., Zha, H., & Shibasaki, R. (2012). Detection and Tracking of Moving Objects at Intersections Using a Network of Laser Scanners. *IEEE Transactions On Intelligent Transportation Systems*, 13(2), 655-670. doi: 10.1109/tits.2011.2175218
84. (2020). Retrieved 18 August 2020, from <https://saaq.gouv.qc.ca/fileadmin/documents/publications/detailed-profile-facts-statistics-pedestrians.pdf>
85. Zhu, J., Chen, S., Tu, W., & Sun, K. (2019). Tracking and Simulating Pedestrian Movements at Intersections Using Unmanned Aerial Vehicles. *Remote Sensing*, 11(8), 925. doi: 10.3390/rs11080925
86. Dalal, N., & Triggs, B. Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference On Computer Vision And Pattern Recognition (CVPR'05)*. doi: 10.1109/cvpr.2005.177