Deep Learning-based Internet of Vehicles Framework for Real-Time Traffic Rules Violation Detection

Report I, literature review & system design.

A report submitted to the

Faculty of Engineering

In partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer engineering

Prepared by:

Abdelrahman Elsayed 51610251

Ahmed Omar 51610305

Mohamed Ammar 51610268

Mohamed Fawzy 51610135

Mohamed Mosa 51610310

Supervised by

Dr.Islam Tharwat

Ahram Canadian University

2020/2021

Table of Contents

1-INTRODUCTION

In urban cities, we notice that drivers respect traffic rules in the presence of security controls, while in most of the way they do not respect traffic rules. For example, in places where radars are located, we find drivers slowing down to the permissible limit. At the checkpoint, we find drivers wearing belts, some of them quitting smoking and others putting the phone from their hands until they pass through the barrier.

Imagine if there was a system that solves all these problems and has the ability to add all traffic violations in the future.

1.1-Problem Statement

Traffic and transportation have increased even in developing countries, which led to many difficulties in capturing traffic rules violations, especially in urban cities. Covering every street with human forces or law authorities is not the best solution anymore nor fixed CCTV surveillance systems all over a big city such as Cairo, Egypt. The use of CCTV in a traditional way is not efficient except after the traffic violation takes place. It also needs to visit the site violation and manually checks the information from the camera then needs to view the full length of the video to detect the traffic violation. We will discuss an automatic traffic violation detection system to solve this problem effectively. Through cameras with a system designed by us to monitor some violations now (to monitor all anomaly and violations in the future). These cameras are mobile to cover all roads and are installed in vehicles whose movement is periodic, such as public transport buses.

Fig 1.1 Traffic jam

1.2-Motivation

Over time, traffic congestion and accident rates are increasing which cause large problems, so we need to detect violation and reduce traffic violations, to do that we need to track every violation that happens and put a penalty to prevent people from making any violation. The current method of recording violations will not be able to capture every traffic rule violation, especially in urban cities. Because the traditional method of detecting violators depends on the observation of traffic officers and with the number of vehicles on the road is growing exponentially, which leads to heavy traffic on the road and therefore a lot of traffic rules are broken, making it difficult for the human eye to notice all traffic violations. This is about the violations that police officers detect also in other types of violations like speeding out of limit, speaking in phone and other violations which in the traditional method of detecting it depends on camera like radar, Car Electronic System, ..etc these method detects violation in a fixed point in the road and doesn't cover all the road because it depends on fixed camera and violator always obey traffic rule in that point and after passing that point return again to violate rule so in our project we try to solve this problem by using a flexible camera which means the cameras are installed in the vehicles themselves than you can cover all the road and can detect violation in any point in the road ,we could be identifying violators automatically and sending their details to traffic officers. This was done by using machine learning algorithms and Deep Learning Convolutional Neural Network (CNN) method to capture, analyze and detect the type of violation caused by a vehicle which makes it easy to implement this project. So, our systems are predicted to reduce traffic violations, accident rates, and increase safety and security.

The Central Bureau of Statistics announced that the number of car accidents on the roads increased during 2019, reaching 9,992 accidents, compared to 8,480 accidents in 2018, an increase of 17.8%. In a report on the accident rate, the agency said that the number of deaths resulting from car accidents rose to 3,484 dead in 2019, an increase of 12.9%, noting that the rate of car accidents for housing was one accident per 10,000 people, while the rate for vehicles was 0.9 accidents per 1000 vehicles. The death rate is 3.6 deaths per 100,000 people, 30.3 deaths per 100,000 vehicles in 2019, and the car accident rate increased to 27.4 accidents per day in 2019. The device attributed the main cause of car accidents to the human element, which amounted to 79.7%, followed by technical defects in the vehicle, 13.5% of the total causes of road accidents in 2019. Now our main goal is to reduce the percentage of traffic accidents, especially those which are due to the human element, as after using our project, this percentage will decrease significantly and noticeably, as it covers the entire road, and therefore the driver will not be able to avoid any radars or not fasten the seat belt except in the places of the radars only etc. The total area of Egypt is two million square kilometres, mostly desert, of which only 5.5% is inhabited. The cultivated area is 8.6 million feddans, which represents 3% of the total area of Egypt. According to the Egyptian Central Agency for Mobilization and Statistics, in a statistic at the beginning of 2021, the current population of Egypt is approximately 101 million. The entire population of Egypt is concentrated in the cultivated 3%, which causes massive overcrowding, resulting in traffic congestion and traffic accidents, all this in addition to a large number of cars and the lack of use of public transportation, which also leads to traffic congestion. Traffic congestion leads to drivers not paying attention to the laws of the road, so they only avoid the radars and break all the rules on the roads where there are no radars. If this project is implemented, every driver will have to observe the laws of the road in all the roads, and he will not be able to break any rule and thus we have reduced the traffic congestion rate significantly.

1.3-Scope definition

Computer Vision is not science fiction anymore. It is applicable and made available for everyone. Convolutional neural networks are showing great results in detecting anomalies in pre-trained models with great accuracy. This helps in facilitating road safety surveillance cameras. Detecting traffic rules violations to provide complete safety from accidents and to force violators to the law. Various techniques are adopted to apply security and safety on highways and main roads such as accurate Computer-vision based fixed radars along main highways and also using lidars in security patrol cars and many other approaches to ensure traffic rules are kept committed without human interaction and many other intelligent video surveillance (IVS) approaches. Time by time, surveillance applications, traffic rules violation detection and Machine learning based surveillance procedures are being improved with enhanced functions and accuracy.

We have limitation: working on open source dataset and result accuracy depend on CPU capacity.

1.4-Project Description

This section describes the traffic rule violation detection management framework of the project for Real-Time Traffic Rules Violation Detection. The system provides accurate violation detection using convolution Neural Network (CNN) approaches. Cameras connected to public transport buses will cover the area that is in front of the vehicle in stand-by mode waiting a violator shows up driving in the opposite direction or any violation the driver doing it. The system uses the Convolutional neural networks approach to detect a car's frontal side which is not supposed to be seen in the front of our vehicle. The system then captures the frame and extracts the car licence plate and uploads them through 3G/4G-LTE module to General Directorate of Traffic servers so they are able to view the picture with licence plate and take a decision.

2-BACKGROUND

2.1-What is Artificial Intelligence?

Originally coined in the 1950s, the term “artificial intelligence” initially began as the simple theory of human intelligence being exhibited by machines [1]. In 1976, Jerrold S. Maxmen foretold that artificial intelligence (AI) would bring about the “post-physician era” in the twenty-first century [2]. But achieving an artificially intelligent being was not so simple. After several reports criticizing progress in AI, government funding and interest in the field dropped off – a period from 1974–80 that became known as the "AI winter." The field later revived in the 1980s when the British government started funding it again in part to compete with efforts by the Japanese. The field experienced another major winter from 1987 to 1993, coinciding with the collapse of the market for some of the early general-purpose computers, and reduced government funding. But research began to pick up again after that, and in 1997, IBM's Deep Blue became the first computer to beat a chess champion when it defeated Russian grandmaster Garry Kasparov.

Fig 2.1 Garry Kasparov vs IBM’s computer

In today’s era of rapid technological advancement and exponential increases in extremely large data sets (“big data”), AI has transitioned from mere theory to tangible application on an unprecedented scale [3]. From evaluating extraordinarily large data sets in near real-time, autonomous driving cars and stream history-influenced video viewing recommendations, to online purchase recommendations, advertisements, and fraud detection (Amazon, Seattle, WA, USA), AI has become fundamentally ingrained within many facets of society and often functions invisibly in the background of our personal electronic devices. And in 2011, the computer giant's question-answering system Watson won the quiz show "Jeopardy!" by beating reigning champions Brad Rutter and Ken Jennings. Artificial Intelligence can do many things, but in business, it becomes powerful when applied to data analytics. AI is extremely powerful and fast at searching for patterns in data, finding relationships, and testing the assumptions when new datasets are given. For the most part, AI doesn't need any human interaction. With AI, any business can analyze large complex datasets, find patterns and relationships, and make predictions to improve their business. AI is a broad field, and as that field has progressed, several subsets have emerged that describe applications of AI. Understanding the utilization of Machine Learning, Neural Networks, and Deep Learning individually will help you understand the big picture.

Fig 2.2 History of Artificial Intelligence

Considered a subset of AI, machine learning (ML) exhibits the experiential "learning" associated with human intelligence, while also having the capacity to learn and improve its analyses through the use of computational algorithms [1, 3]. These algorithms use large sets of data inputs and outputs to recognize patterns and effectively "learn" in order to train the machine to make autonomous recommendations or decisions. After sufficient repetitions and modification of the algorithm, the machine becomes able to take an input and to predict an output [1, 3]. Outputs are then compared with a set of known outcomes in order to judge the accuracy of the algorithm, which is then iteratively adjusted to perfect the ability to predict further outcomes [4].

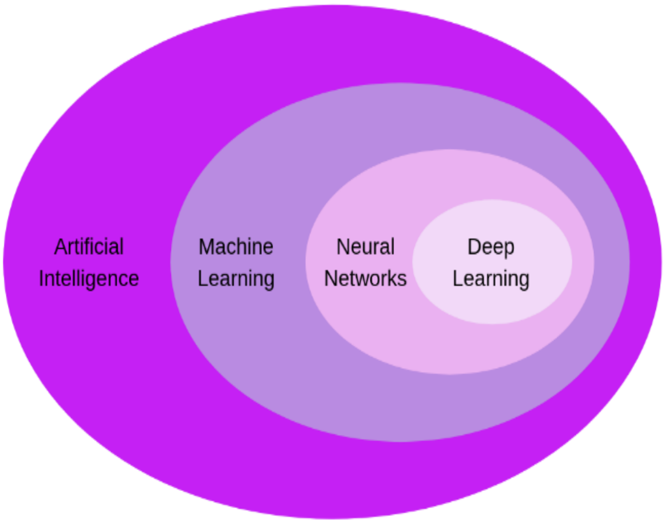
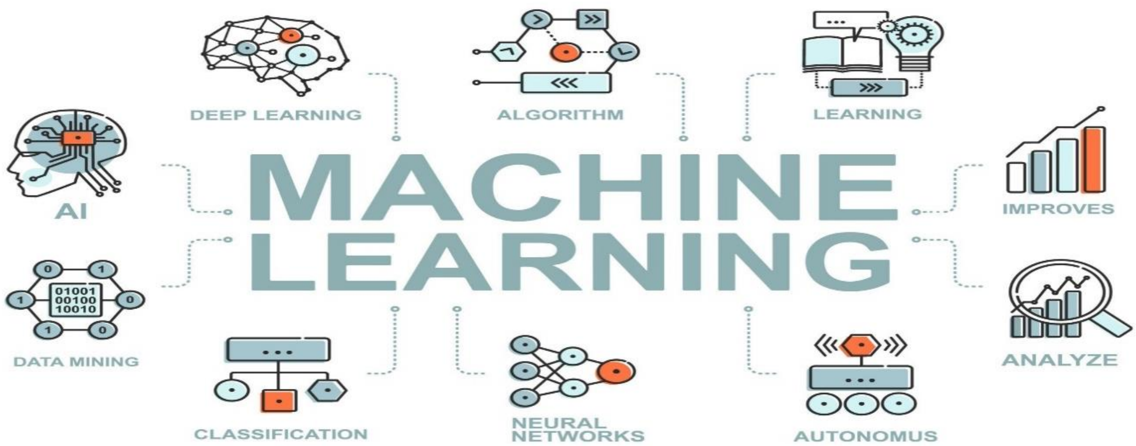


Fig. 2.3 Subsets of AI

2.1.1-**Machine learning:**



Machine learning (ML) has recently received considerable attention for its ability to accurately predict a wide variety of complex phenomena. However, there is a growing realization that, in addition to predictions, ML models are capable of producing knowledge about domain relationships contained in data, often referred to as interpretations. These interpretations have found uses in their own right, e.g, medicine, policymaking, and science, as well as in auditing the predictions themselves in response to issues such as regulatory pressure and fairness. In these domains, interpretations have been shown to help with evaluating a learned model, providing information to repair a model (if needed), and building trust with domain experts. In the absence of a well-formed definition of interpretability, a broad range of methods with a correspondingly broad range of outputs (e.g., visualizations, natural language, mathematical equations) have been labeled as interpretation. This has led to considerable confusion about the notion of interpretability. In particular, it is unclear what it means to interpret something, what common threads exist among disparate methods, and how to select an interpretation method for a particular problem/ audience [5]. Broadly, there are 3 types of Machine Learning Algorithms Supervised Learning, Unsupervised Learning, Reinforcement Learning.

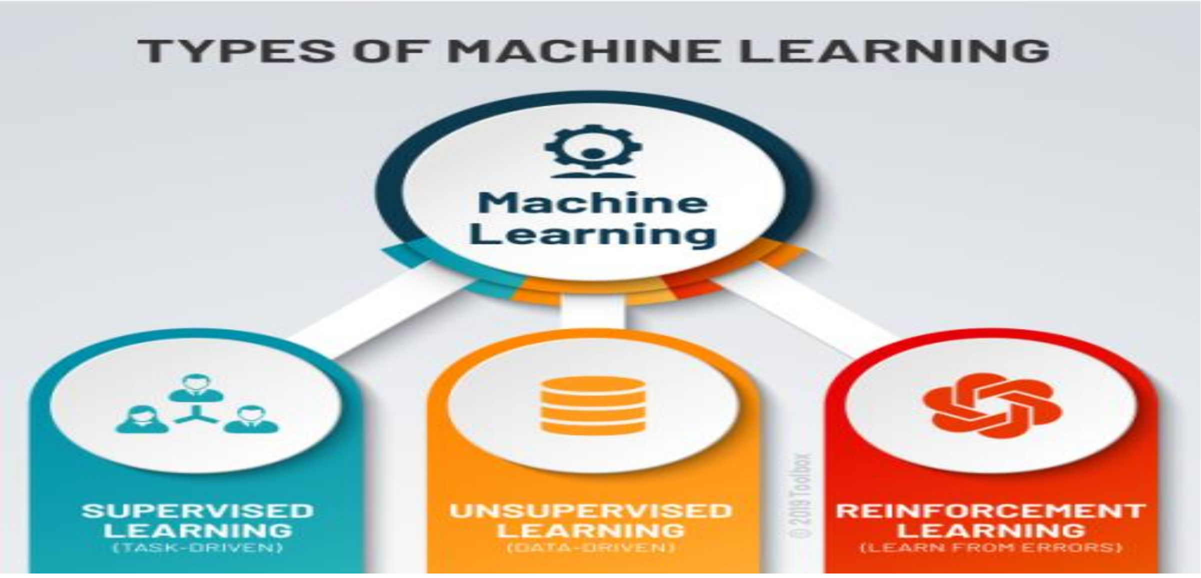


Fig 2.4 Types of ML

Supervised Learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately, which occurs as part of the cross validation process. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox. Supervised learning uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized. Supervised learning can be separated into two types of problems when data mining classification and regression: Classification uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labeled or defined. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbor, and random forest. Regression is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as for sales revenue for a given business. Linear regression, logistical regression, and polynomial regression are popular regression algorithms [6].

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition. Unsupervised learning models are utilized for three main tasks clustering, association, and dimensionality reduction. Machine learning techniques have become a common method to improve a product user experience and to test systems for quality assurance. Unsupervised learning provides an exploratory path to view data, allowing businesses to identify patterns in large volumes of data more quickly when compared to manual observation. Some of the most common real-world applications of unsupervised learning are: News Sections, Computer vision, Medical imaging, Anomaly detection, Customer personas, Recommendation Engines [7].

Unsupervised machine learning and supervised machine learning are frequently discussed together. Unlike supervised learning, unsupervised learning uses unlabeled data. From that data, it discovers patterns that help solve for clustering or association problems. This is particularly useful when subject matter experts are unsure of common properties within a data set. Common clustering algorithms are hierarchical, k-means, and Gaussian mixture models. Semi-supervised learning occurs when only part of the given input data has been labeled. Unsupervised and semi-supervised learning can be more appealing alternatives as it can be time-consuming and costly to rely on domain expertise to label data appropriately for supervised learning [6, 7].

Reinforcement machine learning is a behavioral machine learning model that is similar to supervised learning, but the algorithm isn’t trained using sample data. This model learns as it goes by using trial and error. A sequence of successful outcomes will be reinforced to develop the best recommendation or policy for a given problem. The IBM Watson® system that won the Jeopardy! challenge in 2011 makes a good example. The system used reinforcement learning to decide whether to attempt an answer (or question, as it were), which square to select on the board, and how much to wager especially on daily doubles [8].

There are some examples about applications works by ML:

Fig 2.5 Handwriting Recognition

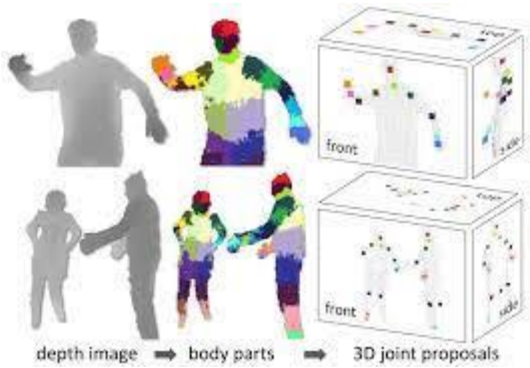


Fig 2.6 Kinect using in video game

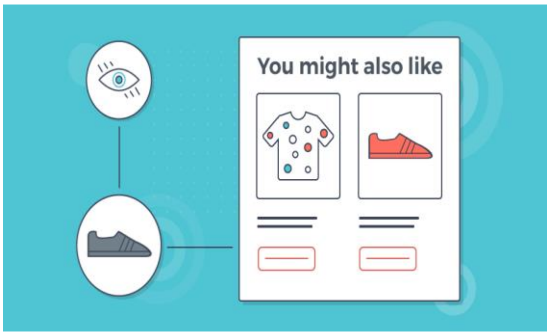


Fig 2.7 Product Recommendation

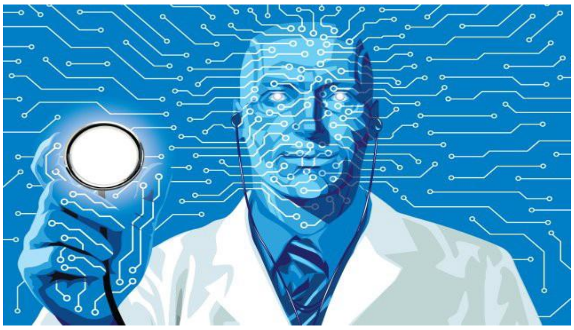


Fig 2.8 Medical Diagnosing

2.1.2-What is Neural Networks?

Neural networks reflect the behavior of the human brain, allowing computer programs to recognize patterns and solve common problems in the fields of AI, machine learning, and deep learning. Also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their

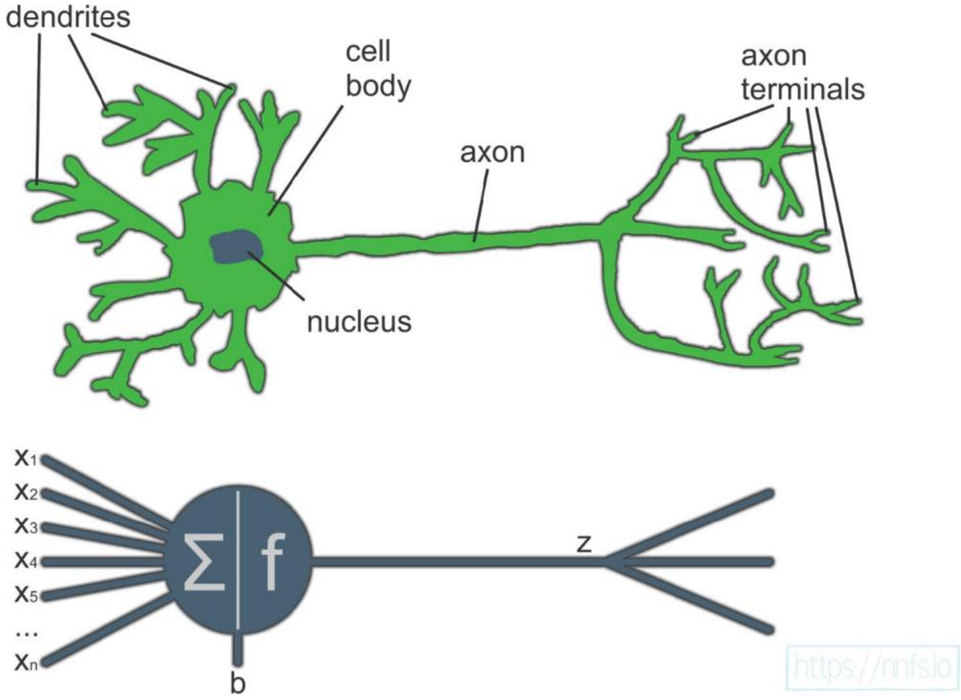


Fig 2.9 Neural network and similar between neurons

name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another. [10]

Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. [10]

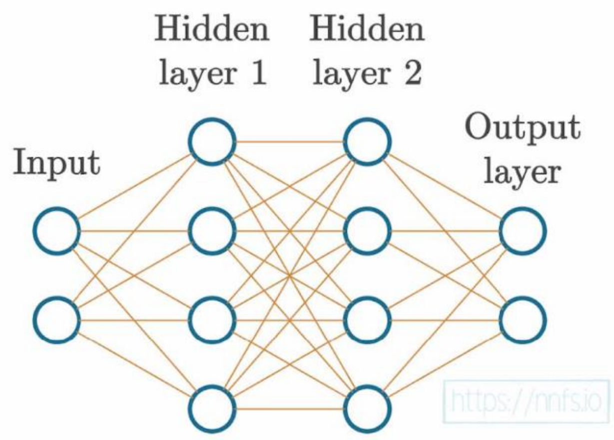


Figure 2.10 basic neural network has two hidden layer

A single neuron by itself is relatively useless, but, when combined with hundreds or thousands (or many more) of other neurons, the interconnectivity produces relationships and results that frequently outperform any other machine learning methods. [11]

The perceptron is the oldest neural network, created by Frank Rosenblatt in 1958. It has a single neuron and is the simplest form of a neural network:

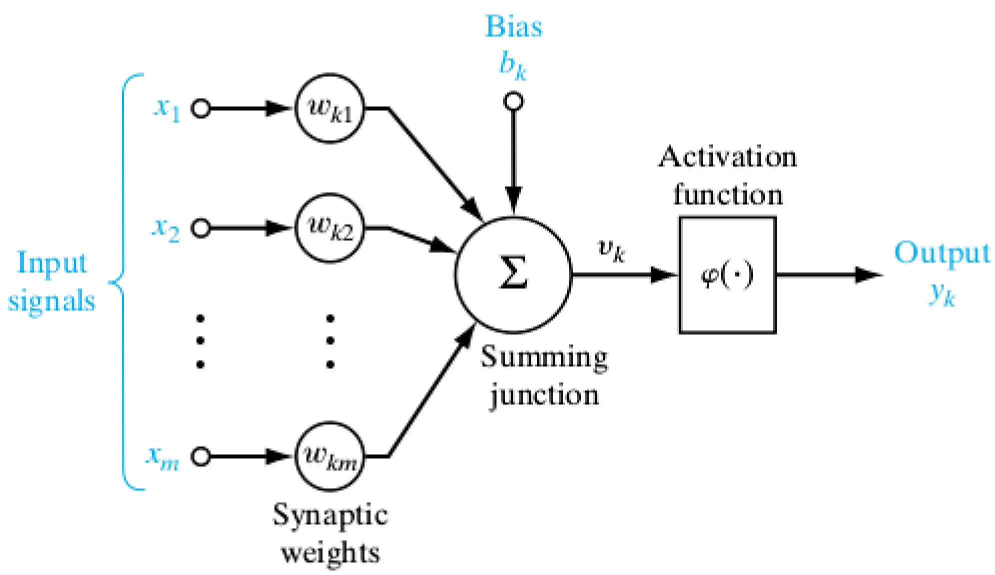


Figure 2.11 single neuron

The history of neural networks is longer than most people think. While the idea of “a machine that thinks” can be traced to the Ancient Greeks, we’ll focus on the key events that led to the evolution of thinking around neural networks, which has ebbed and flowed in popularity over the years:

1943: Warren S. McCulloch and Walter Pitts published “A logical calculus of the ideas immanent in nervous activity” This research sought to understand how the human brain could produce complex patterns through connected brain cells, or neurons. One of the main ideas that came out of this work was the comparison of neurons with a binary threshold to Boolean logic (i.e., 0/1 or true/false statements).

1958: Frank Rosenblatt is credited with the development of the perceptron, documented in his research, “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain”. He takes McCulloch and Pitt’s work a step further by introducing weights to the equation. Leveraging an IBM 704, Rosenblatt was able to get a computer to learn how to distinguish cards marked on the left vs. cards marked on the right.

1974: While numerous researchers contributed to the idea of backpropagation, Paul Werbos was the first person in the US to note its application within neural networks within his PhD thesis.

1989: Yann LeCun published a paper illustrating how the use of constraints in backpropagation and its integration into the neural network architecture can be used to train algorithms. This research successfully leveraged a neural network to recognize hand-written zip code digits provided by the U.S. Postal Service.

Neural networks were conceived in the 1940s, but figuring out how to train them remained a mystery for 20 years. The concept of backpropagation (explained later) came in the 1960s, but neural networks still did not receive much attention until they started winning competitions in 2010. Since then, neural networks have been on a meteoric rise due to their sometimes seemingly magical ability to solve problems previously deemed unsolvable, such as image captioning, language translation, audio and video synthesis, and more. [10]

Currently, neural networks are the primary solution to most competitions and challenging technological problems like self-driving cars, calculating risk, detecting fraud, and early cancer detection, to name a few.

Deep Learning and neural networks tend to be used interchangeably in conversation, which can be confusing. As a result, it’s worth noting that the “deep” in deep learning is just referring to the depth of layers in a neural network. A neural network that consists of more than three layers—which would be inclusive of the inputs and the output—can be considered a deep learning algorithm. A neural network that only has two or three layers is just a basic neural network.

Perhaps the easiest way to think about artificial intelligence, machine learning, neural networks, and deep learning is to think of them like Russian nesting dolls. Each is essentially a component of the prior term.

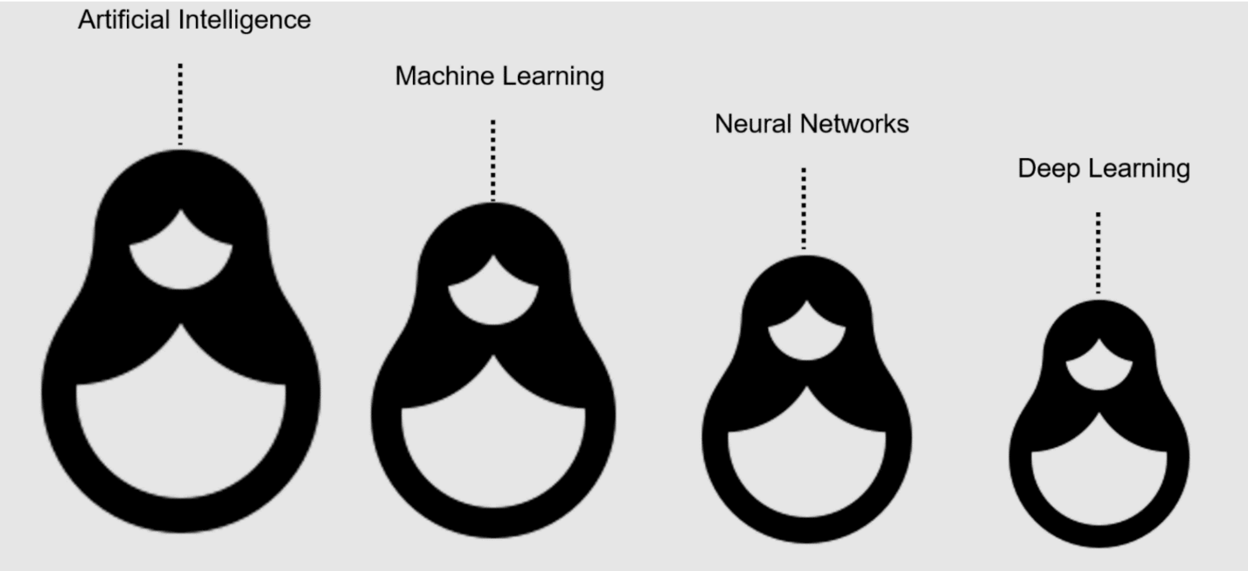


Fig 2.12 AI and SUBFIELDS

That is, machine learning is a subfield of artificial intelligence. Deep learning is a subfield of machine learning, and neural networks make up the backbone of deep learning algorithms. In fact, it is the number of node layers, or depth, of neural networks that distinguishes a single neural network from a deep learning algorithm, which must have more than three.

While it was implied within the explanation of neural networks, it's worth noting more explicitly. The "deep" in deep learning is referring to the depth of layers in a neural network. A neural network that consists of more than three layers—which would be inclusive of the inputs and the output—can be considered a deep learning algorithm. This is generally represented using the following diagram:

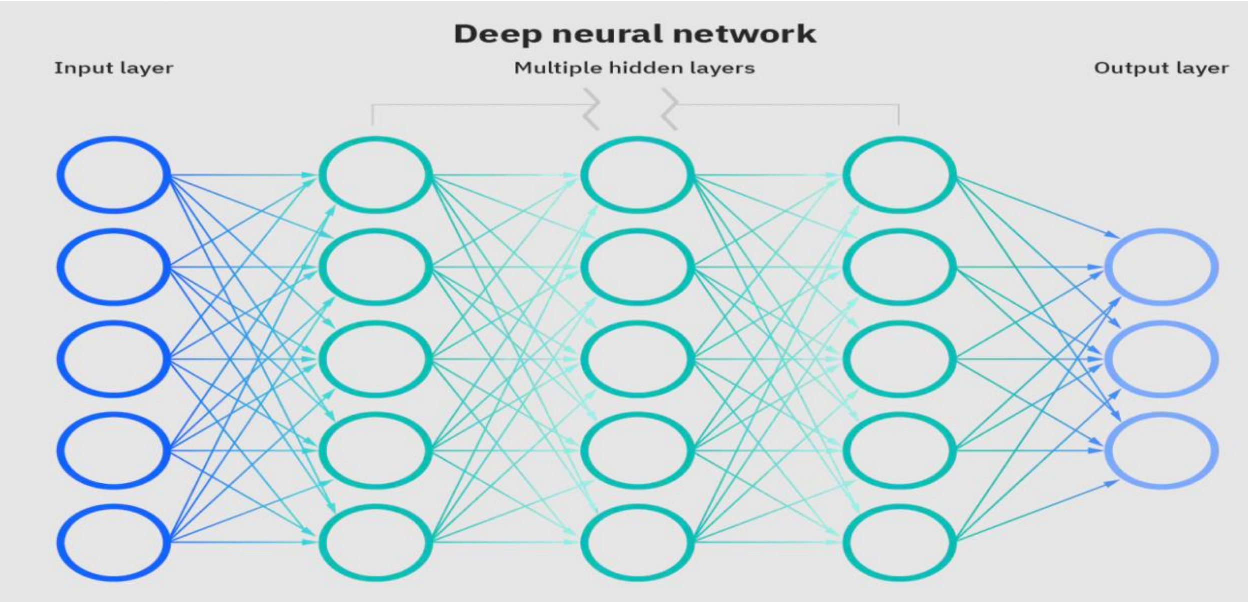


Fig 2.13 Deep Neural Network

Most deep neural networks are feed-forward, meaning they flow in one direction only from input to output. However, you can also train your model through backpropagation; that is, move in opposite direction from output to input. Backpropagation allows us to calculate and attribute the error associated with each neuron, allowing us to adjust and fit the algorithm appropriately.

2.1.3-Deep learning:

Deep learning attempts to mimic the human brain-albeit far from matching its ability-enabling systems to cluster data and make predictions with incredible accuracy.

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behaviour of the human brain-albeit far from matching its ability-allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

How deep learning works? Deep learning neural networks, or artificial neural networks, attempts to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data.

Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called visible layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.

Another process called backpropagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and backpropagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate.

The above describes the simplest type of deep neural network in the simplest terms. However, deep learning algorithms are incredibly complex, and there are different types of neural networks to address specific problems or datasets. For example,

- Convolutional neural networks (CNNs), used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition. In 2015, a CNN bested a human in an object recognition challenge for the first time.
- Recurrent neural network (RNNs) are typically used in natural language and speech recognition applications as it leverages sequential or times series data.

CNN's were first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. It was mostly used in the postal sectors to read zip codes, pin codes, etc. The important thing to remember about any deep learning model is that it requires a large amount of data to train and also requires a lot of computing resources. This was a major drawback for CNNs at that period and hence CNNs were only limited to the postal sectors and it failed to enter the world of machine learning.

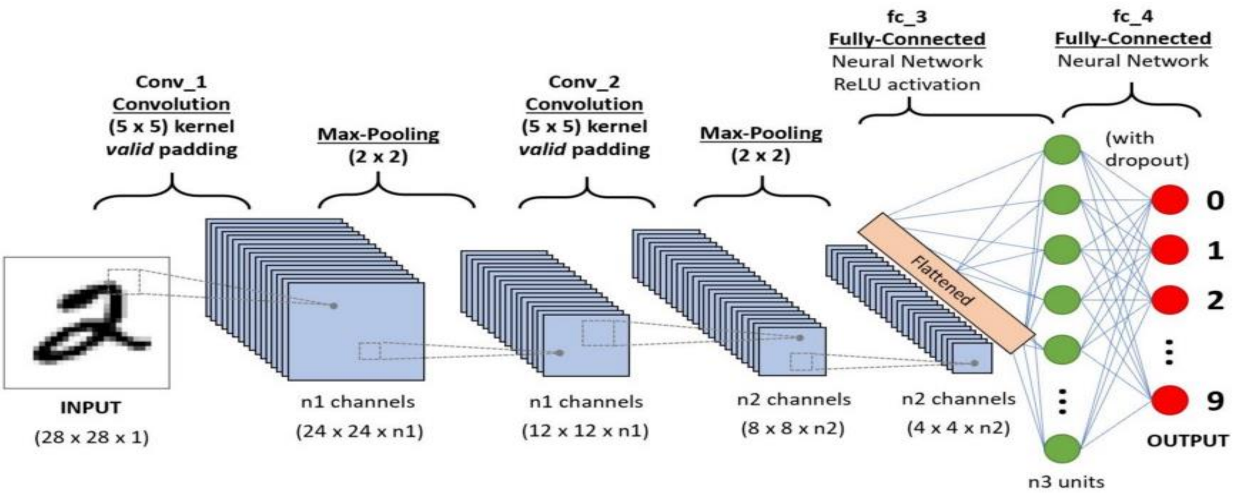


Fig 2.14 Structure of CNN

In 2012 Alex Krizhevsky realized that it was time to bring back the branch of deep learning that uses multi-layered neural networks. The availability of large sets of data, to be more specific ImageNet datasets with millions of labelled images and an abundance of computing resources enabled researchers to revive CNNs.

In deep learning, **a convolutional neural network (CNN/ConvNet)** is a class of deep neural networks, most commonly applied to analyse visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

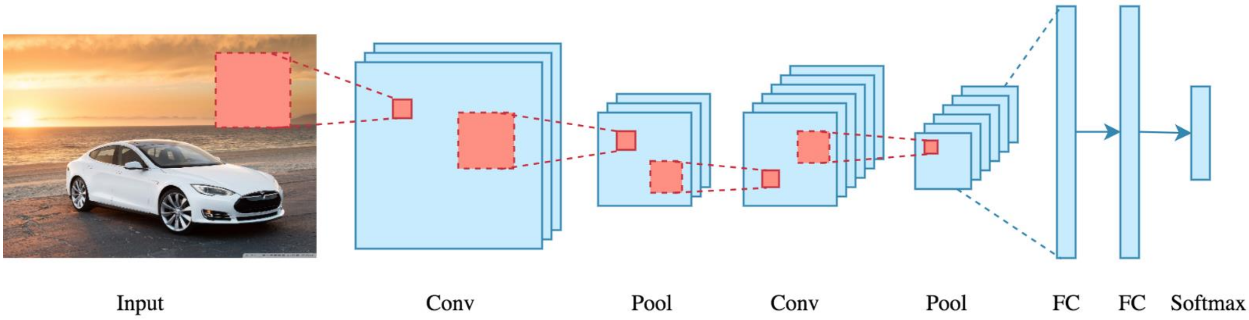


Fig 2.15 How it CNN works

But we don’t really need to go behind the mathematics part to understand what a CNN is or how it works and we will talk about in details in the technical part.

Bottom line is that the role of the ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction. [12]

A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (NLP), speech recognition, and image captioning; they are incorporated into popular applications such as Siri, voice search, and Google Translate. Like feedforward and convolutional neural networks (CNNs), recurrent neural networks utilize training data to learn. They are distinguished by their “memory” as they take information from prior inputs to influence the current input and output. While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of recurrent neural networks depend on the prior elements within the sequence. While future events would also be helpful in determining the output of a given sequence, unidirectional recurrent neural networks cannot account for these events in their predictions.

Recurrent Neural Network vs. Feedforward Neural Network

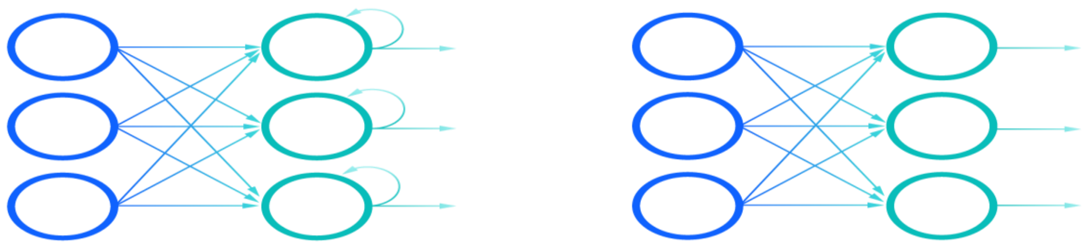


Fig 2.16 Recurrent Neural Network vs feedforward Neural Network

Let’s take an idiom, such as “feeling under the weather”, which is commonly used when someone is ill, to aid us in the explanation of RNNs. In order for the idiom to make sense, it needs to be expressed in that specific order. As a result, recurrent networks need to account for the position of each word in the idiom and they use that information to predict the next word in the sequence.

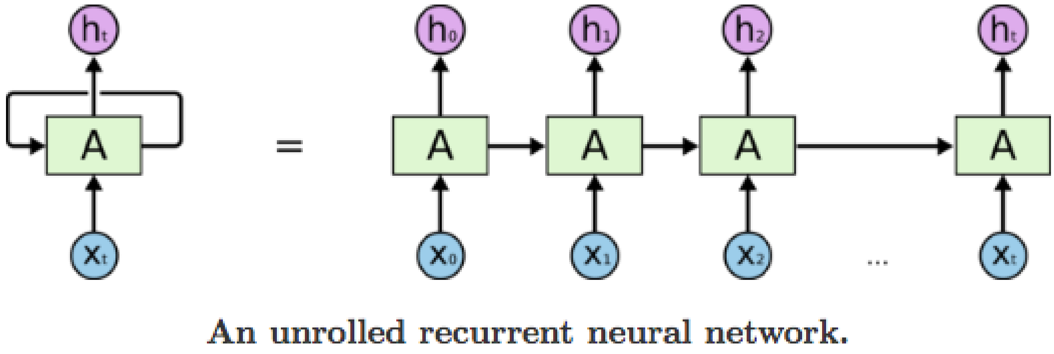
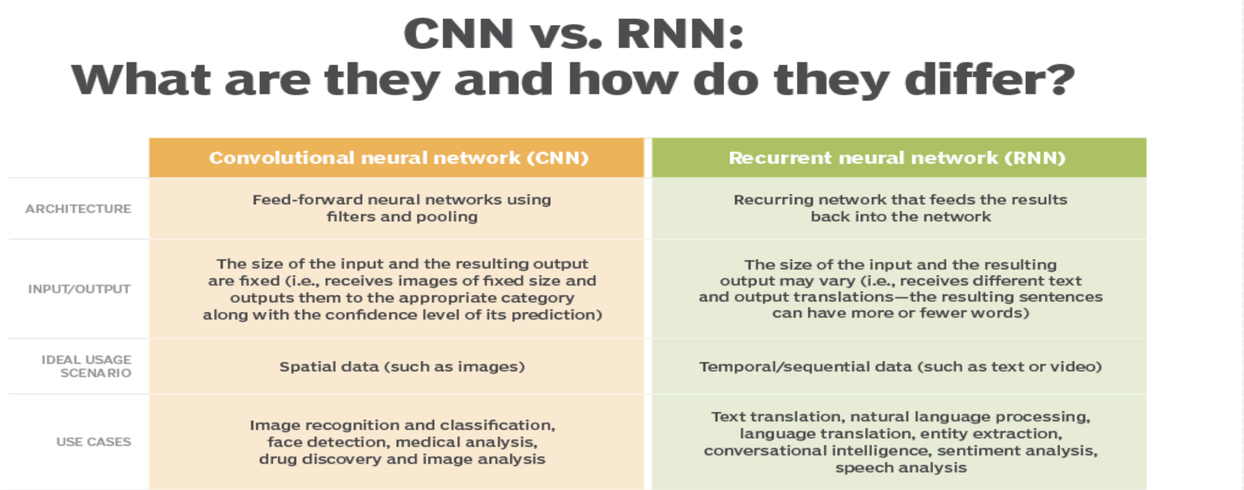


Fig 2.17 Explain of RNN

Recurrent neural networks leverage backpropagation through time (BPTT) algorithm to determine the gradients, which is slightly different from traditional backpropagation as it is specific to sequence data. The principles of BPTT are the same as traditional backpropagation, where the model trains itself by calculating errors from its output layer to its input layer. These calculations allow us to adjust and fit the parameters of the model appropriately. BPTT differs from the traditional approach in that BPTT sums errors at each time step whereas feedforward networks do not need to sum errors as they do not share parameters across each layer. [13]



CNN vs. RNN:
What are they and how do they differ?

	Convolutional neural network (CNN)	Recurrent neural network (RNN)
ARCHITECTURE	Feed-forward neural networks using filters and pooling	Recurring network that feeds the results back into the network
INPUT/OUTPUT	The size of the input and the resulting output are fixed (i.e., receives images of fixed size and outputs them to the appropriate category along with the confidence level of its prediction)	The size of the input and the resulting output may vary (i.e., receives different text and output translations—the resulting sentences can have more or fewer words)
IDEAL USAGE SCENARIO	Spatial data (such as images)	Temporal/sequential data (such as text or video)
USE CASES	Image recognition and classification, face detection, medical analysis, drug discovery and image analysis	Text translation, natural language processing, language translation, entity extraction, conversational intelligence, sentiment analysis, speech analysis

Fig 2.18 the difference between CNN and RNN

2.2-DATASET

2.2.1-What is dataset?

Oxford Dictionary defines a **dataset** as “a collection of data that is treated as a single unit by a computer”. It contains a lot of separate pieces of data but can be used to train an algorithm with the goal of finding predictable patterns inside the whole dataset. This means that the data collected should be made uniform and understandable for a machine that doesn't see data the same way as humans do

Your business has always been based on data. Factors such as what the customer bought, the popularity of the products, seasonality of the customer flow have always been important in business making. However, with the advent of machine learning, now it's important to collect this data into datasets. Sufficient volumes of data allow you to analyze the trends and hidden patterns and make decisions based on the dataset you've built. However, while it may look rather simple, working with data is more complicated and might take most of your time since it requires, first of all, proper treatment of the data you have, from the purposes of using a dataset to the preparation of the raw data for it to be actually usable. For this reason, it's important to understand what a dataset in machine learning is, how to collect the data, and what features a proper dataset has.

2.2.2-dataset type and shape

As shown in figure 2.19 Dataset usually presented in tabular form. Each single column represents a particular variable it called **feature** it is a component of an observation and is also called an attribute of a data instance. Some features may be inputs to a model (the predictors) and others may be outputs or the features to be predicted **(Target Attribute)**. Each row corresponds to a given member of the dataset in question. Every single row

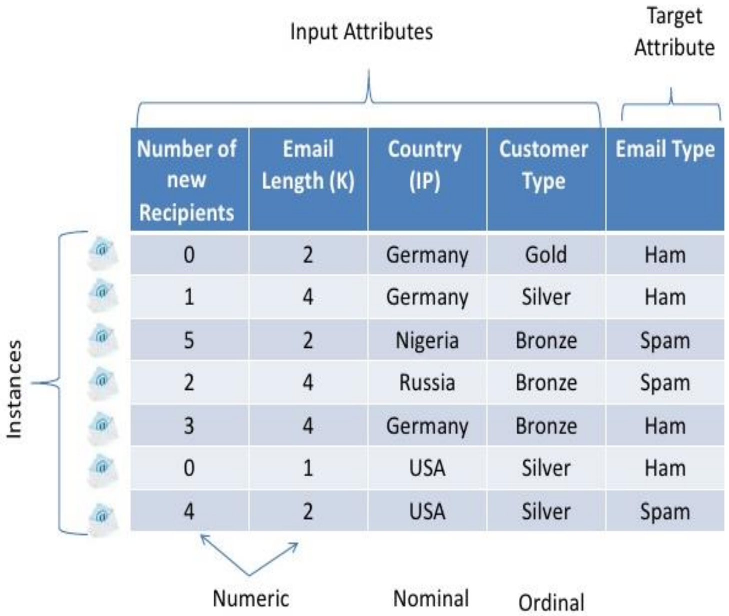


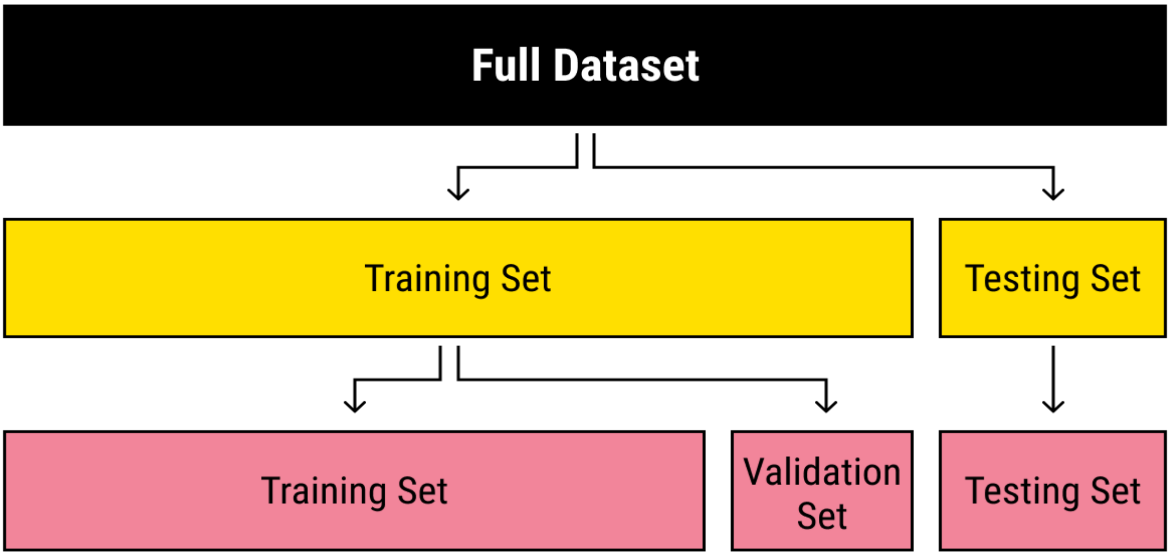
Fig 2.19 dataset shape

of data is called an **instance** it is an observation from the domain. It lists values for each of the variables (feature), such as height and weight of an object. Each value is known as a datum. The dataset may comprise data for one or more members, corresponding to the number of rows.

This data can be image data, Text data, Sound data and any other type of data, these can be text files, .csv files, looking at nested data structures in JSON and XML files and data repositories.

2.2.3-how machine learning deal with dataset?

Usually, a dataset is used not only for training purposes. A single training dataset that has already been processed is usually split into several parts, which is needed to check how well the training of the model went. For this purpose, a testing dataset is usually separated from the data. Next, a validation dataset, while not strictly crucial, is quite helpful to avoid training your algorithm on the same type of data and making biased predictions.



.

Fig 2.20 splitting data

1. Training dataset

This is perhaps the most important among the datasets for machine learning. It is fed to a machine learning algorithm to create a model. The algorithm looks for data patterns to identify input variables. This will help it to reach its ultimate goal or the desired output. The output of this data set is a machine learning model that you can use for predicting results. About 60% of the data set is taken up by a training data set.

2. Validation dataset

A validation data set is used at the validation stage, while creating a machine learning project. This stage comes right after training. This data set is important for evaluating the machine learning model. Machine learning engineers use this set to tweak and adjust the hyperparameters of the model. These hyperparameters are parameters that have values set before the program starts learning.

Their values cannot be estimated from the data. For example, hyperparameters can include the depth of a tree or a number of undetected layers in a neural network.

3. Test dataset

The test datasets for machine learning are used for understanding how the machine learning model will work in the future. Using this data set, you will be able to understand how accurate your data model is. In simple terms, this data set will tell you how much your data model has learned from the training set. These sets take up 20% of the data. The set will contain input variables along with verified outputs. However, in machine learning projects, we generally do not use a training data set in the testing stage. This is because the algorithm will be aware of the expected output, as it has learned from this data set previously.

After the testing phase, the data model is usually not adjusted anymore. This is because further adjustment can lead to overfitting. Overfitting occurs when a data model is trained with too much data. In this case, the model starts learning from the inaccurate data entries in the given data set. As a result, it does not work properly on new data sets. It is like trying to fit into oversized jeans when you can't! , But for the machine learning model to work successfully, you need to provide it with a good data set. Without datasets for machine learning, the algorithm will not be able to learn and solve the problems. For example, when you do not have the right books and resources, you cannot ace the test you want to.

2.2.4-Building Proper Dataset for a Machine Learning Project

Raw data is a good place to start but you obviously cannot just shove it into a machine learning algorithm and hope it offers you valuable insights into your customers' behaviors. There are quite a few steps you need to take before your dataset becomes usable.

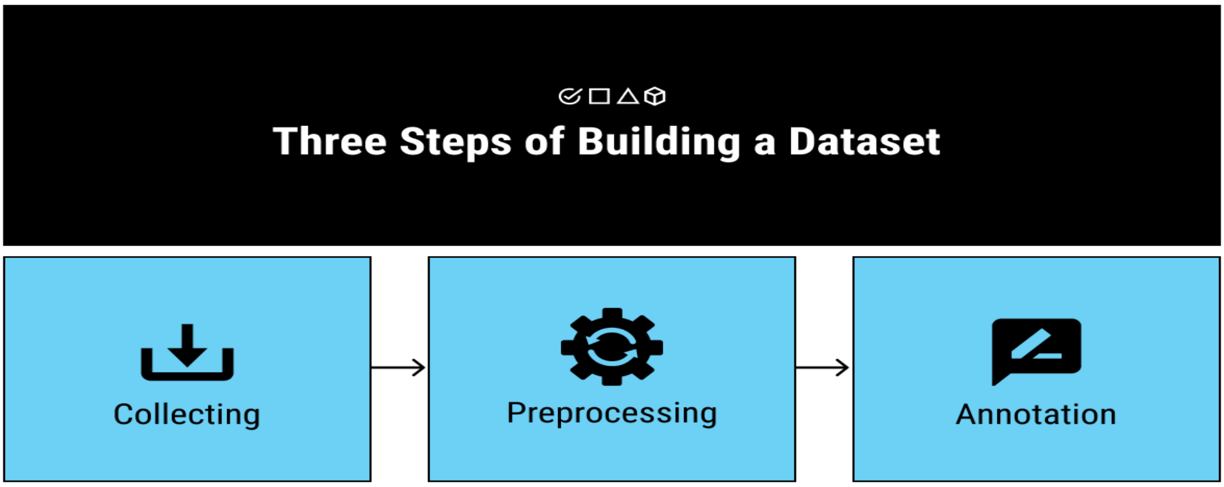


Fig 2.21 Three steps of data processing in machine learning

1-**Collect**: The first step is to collect all the relevant data that you may need for your machine learning model. The amount of data will depend upon the complexity of the machine learning project. A simple project will require less data than a complicated one. So, you need to determine all that you actually need to solve the problem at hand. Data can be collected easily by answering the following questions:

- What type of data is available to you for the project?
- What data is not available that you need for the project? – This may include certain databases or data stored in cloud systems. You may need to derive this data.
- What data can you remove from the existing data? This means clearing out the unwanted data that is irrelevant to your project.

When you have the answers to all these questions, you can start deciding on the sources you'll be using to collect the data. Usually, there are three types of sources you can choose from: the freely available open-source datasets, the Internet, and the generators of artificial data. Each of these sources has its pros and cons and should be used for specific cases .We'll talk about this step in more detail in the next section (2.2.5).

2- **Preprocess**: There's a principle in data science that every experienced professional adheres to. Start by answering this question: has the dataset you're using been used before? If not, assume this dataset is flawed. If yes, there's still a high probability you'll need to re-appropriate the set to fit your specific goals. After covering the sources.

The preprocessing method is converting raw datasets into meaningful sets that are usable. The process consist of the three steps below:

- Formatting
  The raw data that you have collected many not be in a format that is suitable for your machine learning model. It may be in a JSON file or a relational database. You need to convert this data into a text file or a .csv file as per your convenience.

- Cleaning
  This is the process where you fix and remove missing and unwanted data from your data set. These instances of data may not help to solve the problem. Additionally, there may be sensitive information within some of the attributes that you may need to hide or remove completely. This makes your datasets for machine learning more meaningful.

- Sampling
  You may have collected a lot more data than you actually need for the project. Large data sets consume a lot of memory space. They also cause longer runtimes and much more computation when fed to a machine learning algorithm. To avoid these problems, you have to make smaller samples of the selected data that your model can use easily. This process is called sampling.

3-**Annotate**: After you've ensured your data is clean and relevant, you also need to make sure it's understandable for a computer to process. Machines do not understand the data the same way as humans do (they aren't able to assign the same meaning to the images or words as we). This step is where a lot of businesses often decide to outsource since keeping a trained annotation professional is not always viable

2.2.5-The sources of the dataset collection

There are quite a few different sources to get the training data sets from, and the choice of these sources depends on your goals, the requirements of your machine learning project, as well as your budget, time, and personnel restrictions.

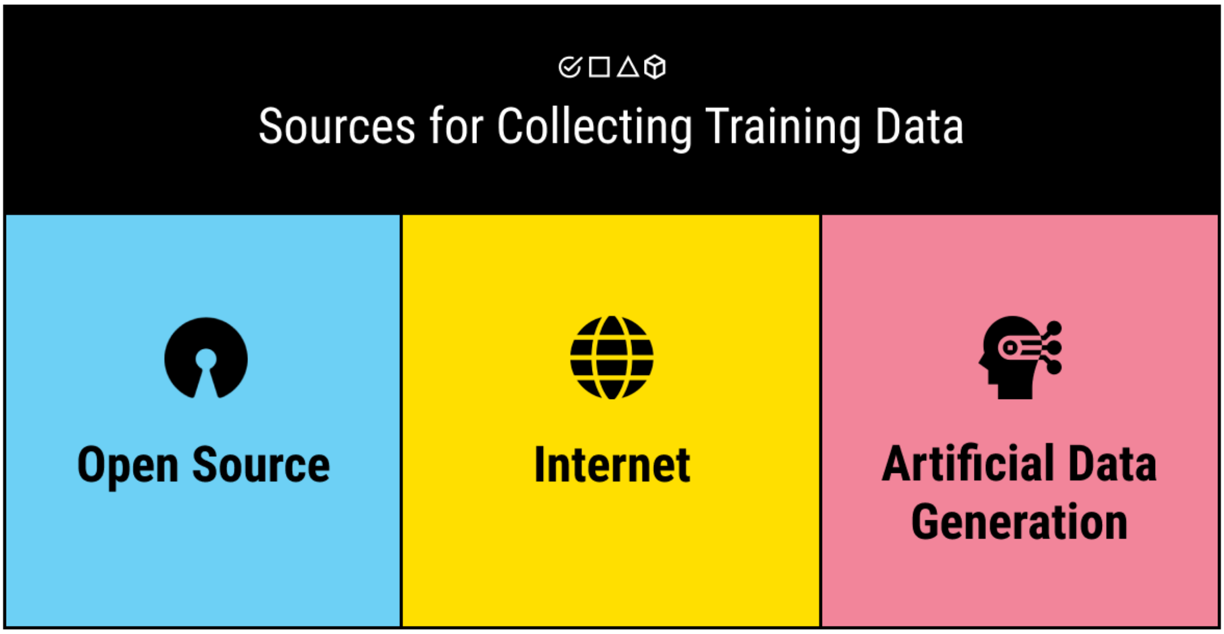


Fig 2.22 Collecting training data: open-source, web and IoT, ML

1. **Open-source training data sets**. This might be an acceptable solution if you're very lucky or otherwise for smaller businesses and start-ups that don't have enough free resources to spend on data collection and labeling. The great benefit of this option is that it's free and it's already collected. But there's a catch (isn't there always?): such data sets were not initially tailored for your algorithm's specific purposes but for some other project. What this means for you is that you'll need to tweak and probably re-annotate the data set to fit your training needs.

2. **Web and IoT**. This is a very common way of collecting training data sets that most middle-sized machine learning companies use. This means that you use the Internet to collect the pieces of data. Alternatively, sensors, cameras, and other smart devices may provide you with the raw data that you will later need to annotate by hand. This way of gathering a training data set is much more tailored to your project because you're the one collecting and annotating the data. On the downside, it requires a lot of time and

resources, not to mention the specialists you know how to clean, standardize, anonymize, and label the data.

3. **Artificial training data sets**. This is the way that starts to gain traction in recent years. What it basically means is that you first create an ML model that will generate your data. This is a great way if you need large volumes of unique data to train your algorithm. It is sparing in terms of financial and personnel resources as it only needs to spend them on designing the model to create your data. Still, this method of collecting training data requires a lot of computational power, which is not usually in free access for small and middle-sized businesses. Besides, if you need truly vast amounts of data, it will take some time to generate a voluminous high-quality training data set.

2.3-What are traffic rules?

Traffic laws are the laws which govern traffic and regulate vehicles, while rules of the road are both the laws and the informal rules that may have developed over time to facilitate the orderly and timely flow of traffic. [17] Organized traffic generally has well-established priorities, lanes, right-of-way, and traffic control at intersections.

Traffic is formally organized in many jurisdictions, with marked lanes, junctions, intersections, interchanges, traffic signals, or signs. Traffic is often classified by type: heavy motor vehicle (e.g., car, truck), other vehicle (e.g., moped, bicycle), and pedestrian. Different classes may share speed limits and easement, or may be segregated. Some jurisdictions may have very detailed and complex rules of the road while others rely more on drivers' common sense and willingness to cooperate.

Organization typically produces a better combination of travel safety and efficiency. Events which disrupt the flow and may cause traffic to degenerate into a disorganized mess include road construction, collisions, and debris in the roadway. On particularly busy freeways, a minor disruption may persist in a phenomenon known as traffic waves. A complete breakdown of organization may result in traffic congestion and gridlock. Simulations of organized traffic frequently involve queuing theory, stochastic processes and equations of mathematical physics applied to traffic flow.

The word traffic originally meant "trade" (as it still does) and comes from the Old Italian verb trafficare and noun traffico. The origin of the Italian words is unclear. Suggestions include

Catalan trafegar "decant",[18] an assumed Vulgar Latin verb transfricare 'rub across',[19] an assumed Vulgar Latin combination of trans- and facere 'make or do',[19][20] Arabic tafriq 'distribution',[19] and Arabic taraffaqa, which can mean 'seek profit'.[20] Broadly, the term covers many kinds of traffic including network traffic, air traffic, marine traffic and rail traffic, but it is often used narrowly to mean only road traffic.

2.3.1-History of traffic rules

When cars became popular, local governments established traffic laws to limit collisions with horse-drawn wagons and ensure safety. The mandatory registration of automobiles was one of the first traffic regulations in the United States. New York became the role model in 1901 by being the first state to require that automobile owners register their vehicles. By 1920, license plates were mandatory in all states. It took longer for the states to require a driver’s license. In 1935, there were just 39 states that issued the licenses and only a few tested applicants. Before the 1930s, most drivers received their training from automobile salesmen, nonprofit organizations such as the YMCA, family members and friends. Soon, however, driver's education was provided in the high schools. [21]

It did not take cars long to clog the streets and cities to begin setting speed limits, installing traffic lights, designing one-way streets, and adding parking meters. Yet, it did take drivers longer to start obeying these laws. The book "Rules of the Road" was written by William P. Eno in 1903. Eno, "the father of traffic safety," introduced many road regulations, such as the need for slow traffic to remain to the right and cars to pass only on the left, as well as one-way streets, crosswalks for pedestrians, stop signs and safety islands. He believed that stoplights would never work, and police would always be necessary at intersections. Ironically, he had his chauffeur drive him around. This is the History of Traffic.

2.3.2-Obeying traffic rules

Traffic laws now make up a major part of most state regulations. Their main purpose is to improve unsafe driving and to provide education to bad drivers. Research shows that most people will obey the traffic laws, even when they hit a red light at 3 a.m. and there is not another car in sight. There is, however, a group of people who constantly get caught for ignoring the laws. Studies show that more people follow the laws when they think that there is a good chance of being caught and less adhere to them when they believe they can get away with it.

2.3.2.1-MOVING AND NON-MOVING VIOLATIONS

Since the beginning of traffic laws, drivers have received tickets for "strict liability" offenses. In other words, the person can be found guilty of breaking the law, without any criminal intent to do so. The police only need to prove that the person did not follow the law. Examples of these moving and non-moving strict-liability traffic offenses are: not properly using turn signals, going through a stop sign without making a full stop, driving a car with only one working headlight, speeding, driving too close to another car, not putting enough money in a parking meter and parking in a loading zone. In most cases, except for instances such as driving under the influence and a hit and run, the driver does not have to go to criminal court. Fines can be large, however, for offenses such as speeding, and a driver may lose his or her license after accumulating several points for traffic violations.

2.3.3-The importance of traffic rules

Setting a set of rules to maintain traffic flow and to keep it smooth without anomalies where everyone can take his part from the road without interference. to keep it efficient, where we ensure that every road user cuts the shortest mileage to the required destination with least cost of time, fuel, and road usage. And to keep it safe by lesser randomness in driving and vehicles directions of movement, also by adding limitations for speed and globalize set of rules which in turn aims for a descent organized environment for driving and transportation.

Just as everything and every institution require a set of I rules, traffic also needs rules in order to remain orderly I and disciplined. The question that next arises in our minds is that, what is the necessity of rules?

It is necessary to have rules everywhere in order to make the functioning smooth and efficient. If there were to be no rules then, it would be a picture of total chaos and confusion. Rules regulate the work and help it move along the desired path.

Thus, in order to have a smooth movement of traffic on the roads, the traffic rules are made by the traffic police. These rules are meant to be followed to the last word by each and every individual moving on the roads, and becoming a part of the traffic.

It is necessary to have rules for the road, but it is still more important for all of us to follow the set of rules. Once an individual is on the road, it is absolutely compulsory for him/her to follow the rules, and that also explicitly.

2.3.4-Why do we Follow rules?

Fig 2.23 Traffic Jam

We have just got to follow rules because, without following them there will be absolute chaos and confusion on the road, and no one will be able to move about. This chaos would lead not only to delays in movements but would also lead to struggles and even accidents.

When, for example we are supposed to cross the road from the zebra crossing, we must make sure that we do so, for, if we cross from elsewhere, there is a chance that we meet with an accident. If we jump a red light we are putting ourselves to danger and are inviting trouble with the possibility of an accident.

Thus, rules must be followed for maintaining discipline on the roads, and above all for our own safety. It is in our own interest that, when on the road, we follow the road traffic rules to the last word. The rules are there to keep us safe, and following them is in our own interest.

When we break the rules we are inviting trouble to ourselves and doing no harm to any one else. The traffic rules in Egypt are as strict as they are anywhere else in the world. However, the difference between the rules outside Egypt and the rules in Egypt is basically just one. That is the rules in foreign countries are followed, and here in Egypt they are broken day in and day out.

Besides having more or less the same rules, over here we Egyptians have an instinct to break all laws and rules. That instinct is so very vivid on the Egyptian roads. Traffic rules are stringent but, when they are not followed, what is the use. Besides, another major difference between foreign countries and Egyptians in this matter is, when a person in a foreign land breaks any traffic rule, he is punished according to the set rules. Here in Egypt, the punishment is also there but, no one gets punished for flouting the road rules.

When rules are there and are broken with impunity and no one can do anything about it, there is utter confusion. That is the exact situation in Egypt. The roads of at least the Metros look like racing tracks, with cars and other vehicles just rushing to where – God alone knows.

There is a speed limit for different categories of vehicles but, who follows the norms? Thus, in Egypt, rules are the same and as stringent as elsewhere but there, in other countries the rules are strictly followed but in Egypt they are strictly broken.

This is the basic difference in the traffic rules and the following of them. A road for instance in Delhi looks a motley picture of confusion and chaos, and this, because everyone is making the road a racing track instead of using it as a road.

Traffic in other countries appears to be more disciplined because these rules are followed and here in Egypt they are not followed. This is exactly why rules are made, and they are made to be followed, and not to be broken as in Egypt.

2.3.5-History of catching violators and appliance of law

In a popular article written by Joseph Stromberg on Nov 4, 2015, under title “The forgotten history of how automakers invented the crime of jaywalking.” On Vox blog newspaper.

He says: A hundred years ago, if you were a pedestrian, crossing the street was simple: You walked across it. Today, if there's traffic in the area and you want to follow the law, you need to find a crosswalk. And if there is a traffic light, you need to wait for it to change to green.

IN THE 1920S, AUTO GROUPS REDEFINED WHO OWNED THE CITY STREET

Fail to do so, and you are committing a crime: jaywalking. In some cities — Los Angeles, for instance — police ticket tens of thousands of pedestrians annually for jaywalking, with fines of up to $250. To most people, this seems part of the basic nature of roads. But it's actually the result of an aggressive, forgotten 1920s campaign led by auto groups and manufacturers that redefined who owned the city streets.

"In the early days of the automobile, it was drivers' job to avoid you, not your job to avoid them," says Peter Norton, a historian at the University of Virginia and author of Fighting Traffic: The Dawn of the Motor Age in the American City. "But under the new model, streets became a place for cars — and as a pedestrian, it's your fault if you get hit."

One of the keys to this shift was the creation of the crime of jaywalking. Here's a history of how that happened.

It's strange to imagine now, but prior to the 1920s, city streets looked dramatically different than they do today. They were considered to be a public space: a place for pedestrians, pushcart vendors, horse-drawn vehicles, streetcars, and children at play.

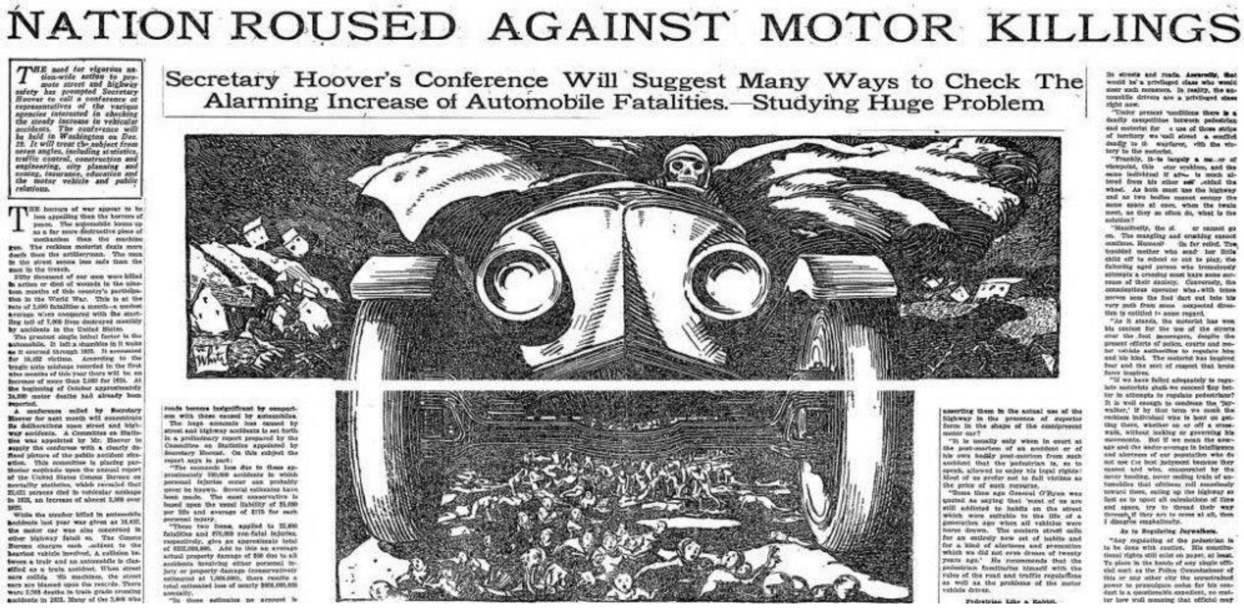
"Pedestrians were walking in the streets anywhere they wanted, whenever they wanted, usually without looking," Norton says. During the 1910s there were few crosswalks painted on the street, and they were generally ignored by pedestrians.

As cars began to spread widely during the 1920s, the consequence of this was predictable: death. Over the first few decades of the century, the number of people killed by cars skyrocketed.

Those killed were mostly pedestrians, not drivers, and they were disproportionately the elderly and children, who had previously had free rein to play in the streets.

The public response to these deaths, by and large, was outrage. Automobiles were often seen as frivolous playthings, akin to the way we think of yachts today (they were often called "pleasure cars"). And on the streets, they were considered violent intruders.

Cities erected prominent memorials for children killed in traffic accidents, and newspapers covered traffic deaths in detail, usually blaming drivers. They also published cartoons demonizing cars, often associating them with the Grim Reaper[22].



NATION ROUSED AGAINST MOTOR KILLINGS

Secretary Hoover's Conference Will Suggest Many Ways to Check The Alarming Increase of Automobile Fatalities.—Studying Huge Problem

THE need for vigorous nation-wide action to promote street and highway safety has prompted Secretary Hoover to call a conference of representatives of the various agencies interested in checking the steady increase in vehicular accidents. The conference will be held in Washington on Dec. 15. It will treat the subject from seven angles, including statistics, traffic control, construction and engineering, city planning and zoning, insurance, education and the motor vehicle and public relations.

THE horrors of war appear to be less appalling than the horrors of peace. The automobile looms up as a far more destructive piece of mechanism than the machine gun. The reckless motorist deals more death than the artilleryman. The man in the street seems less safe than the man in the trench.

Before formal traffic laws were put in place, judges typically ruled that in any collision, the larger vehicle — that is, the car — was to blame. In most pedestrian deaths, drivers were charged with manslaughter regardless of the circumstances of the accident.

How cars took over the roads

Manhattan 1925

In 1925 Midtown Manhattan, pedestrians compete for space with increasing automobile traffic. (Edwin Levick)

As deaths mounted, anti-car activists sought to slow them down. In 1920, Illustrated World wrote, "Every car should be equipped with a device that would hold the speed down to whatever number of miles stipulated for the city in which its owner lived."

The turning point came in 1923, says Norton, when 42,000 Cincinnati residents signed a petition for a ballot initiative that would require all cars to have a governor limiting them to 25 miles per hour. Local auto dealers were terrified, and sprang into action, sending letters to every car owner in the city and taking out advertisements against the measure.

Cincinnati ad

A 1923 ad in the Cincinnati Post, taken out by a coalition of auto dealers. (Cincinnati Post)

The measure failed. It also galvanized auto groups nationwide, showing them that if they weren't proactive, the potential for automobile sales could be minimized.

In response, automakers, dealers, and enthusiast groups worked to legally redefine the street — so that pedestrians, rather than cars, would be restricted.

I’m a gay ex-NFL player. I can’t wait until players like Carl Nassib don’t need to “come out.”

"THIS IS THE TRAFFIC LAW THAT WE'RE STILL LIVING WITH TODAY"

The idea that pedestrians shouldn't be permitted to walk wherever they liked had been present as far back as 1912, when Kansas City passed the first ordinance requiring them to cross streets at crosswalks. But in the mid-20s, auto groups took up the campaign with vigor, passing laws all over the country.

Most notably, auto industry groups took control of a series of meetings convened by Herbert Hoover (then secretary of commerce) to create a model traffic law that could be used by cities across the country. Due to their influence, the product of those meetings — the 1928 Model Municipal Traffic Ordinance — was largely based off traffic law in Los Angeles, which had enacted strict pedestrian controls in 1925.

"The crucial thing it said was that pedestrians would cross only at crosswalks, and only at right angles," Norton says. "Essentially, this is the traffic law that we're still living with today."

The shaming of “jaywalking”[23]

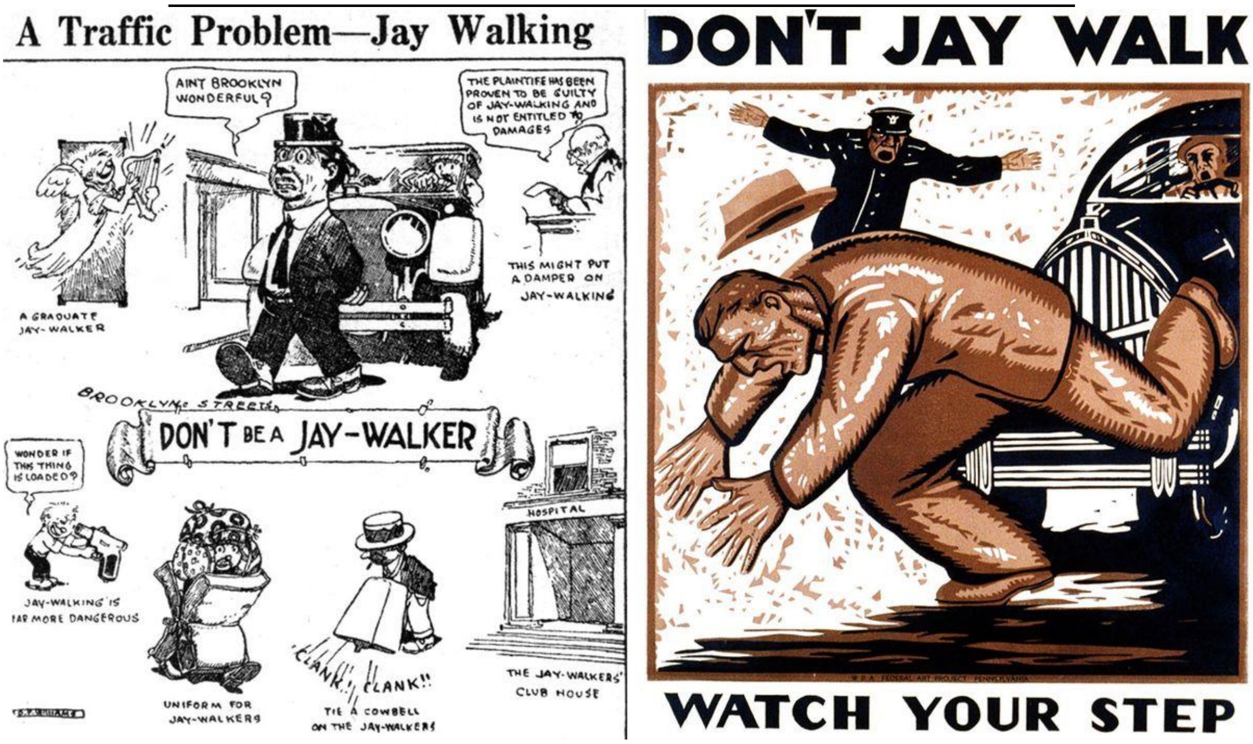


Fig 2.24 Jaywalking Posters

jaywalking posters

Government safety posters ridicule jaywalking in the 1920s and '30s. (National Safety Council/Library of Congress). Even while passing these laws, however, auto industry groups faced a problem: In Kansas City and elsewhere, no one had followed the rules, and they were rarely enforced by police or judges. To solve it, the industry took up several strategies.

One was an attempt to shape news coverage of car accidents. The National Automobile Chamber of Commerce, an industry group, established a free wire service for newspapers: Reporters could send in the basic details of a traffic accident and would get in return a complete article to print the next day. These articles, printed widely, shifted the blame for accidents to pedestrians — signaling that following these new laws was important.

Similarly, AAA began sponsoring school safety campaigns and poster contests, crafted around the importance of staying out of the street. Some of the campaigns also ridiculed kids who didn't follow the rules — in 1925, for instance, hundreds of Detroit school children watched the "trial" of a 12-year-old who'd crossed a street unsafely, and, as Norton writes, a jury of his peers sentenced him to clean chalkboards for a week.

This was also part of the final strategy: shame. In getting pedestrians to follow traffic laws, "the ridicule of their fellow citizens is far more effective than any other means which might be adopted," said E.B. Lefferts, the head of the Automobile Club of Southern California in the 1920s. Norton likens the resulting campaign to the anti-drug messaging of the '80s and '90s, in which drug use was portrayed as not only dangerous but stupid.

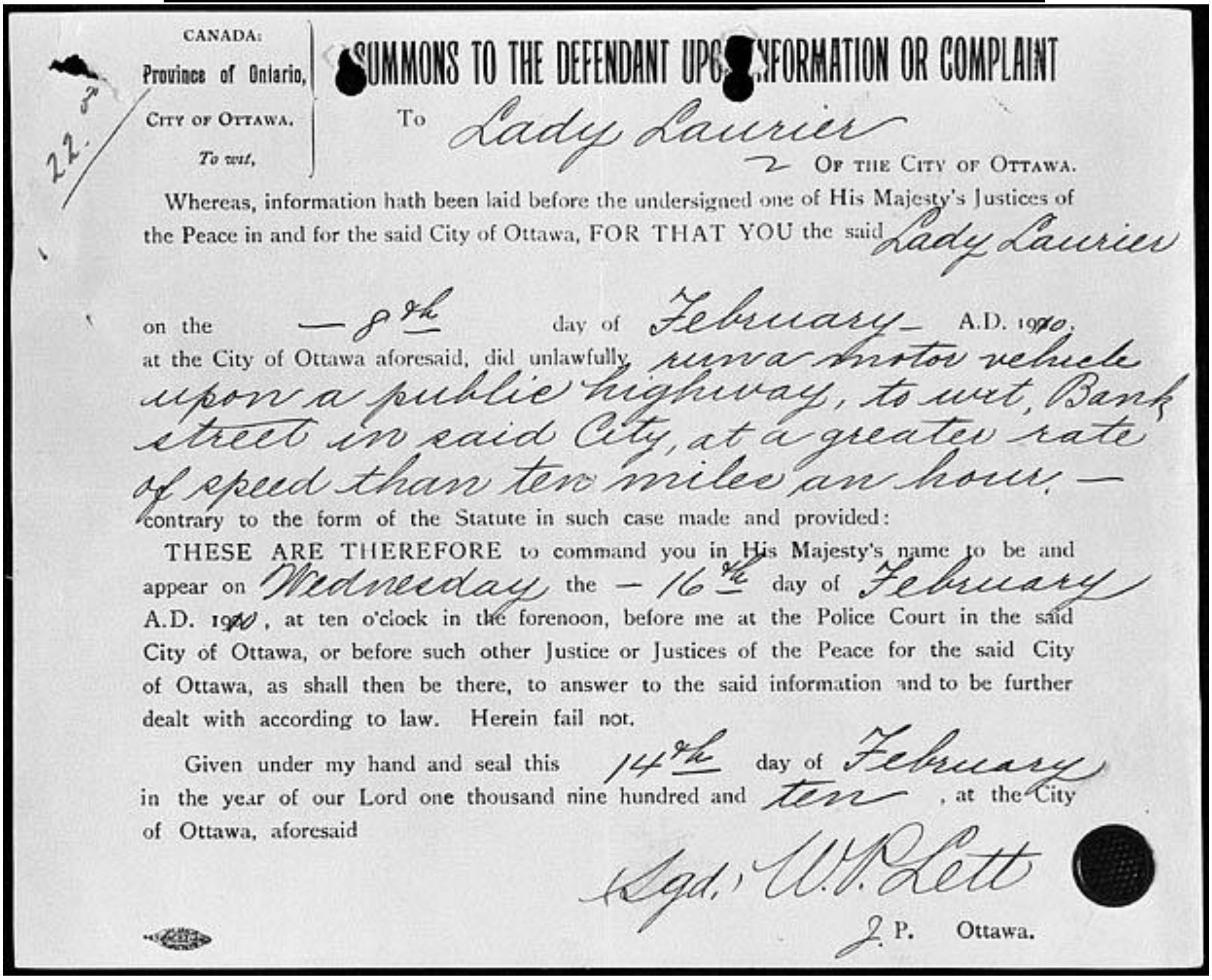
2.3.6-The era of traffic tickets

A traffic ticket is a notice issued by a law enforcement official to a motorist or other road user, indicating that the user has violated traffic laws. Traffic tickets generally come in two forms, citing a moving violation, such as exceeding the speed limit, or a non-moving violation, such as a parking violation, with the ticket also being referred to as a parking citation, or parking ticket.

In some jurisdictions, a traffic ticket constitutes a notice that a penalty, such as a fine or deduction of points, has been or will be assessed against the driver or owner of a vehicle; failure to pay generally leads to prosecution or to civil recovery proceedings for the fine. In others, the ticket constitutes only a citation and summons to appear at traffic court, with a determination of guilt to be made only in court.

There are many competing claims as to the first speeding ticket ever issued depending whether the claim goes by the first traffic violation or the first paper ticket ever issued. Great Britain may have the earliest claim with the first person to be convicted of speeding, Walter Arnold of East Peckham, Kent, who on 28 January 1896 was fined for speeding at 8 mph (13 km/h) in a 2 mph (3.2 km/h) zone. He was fined 1 shilling plus costs.[24] A New York City cab driver named Jacob German was arrested for speeding on May 20, 1899 for driving 12 miles per hour on Lexington Avenue in Manhattan. In Dayton, Ohio, police issued a paper ticket to Harry Myers for going twelve miles per hour on West Third Street in 1904.[25]

Another early speeding ticket was issued in 1910 to Lady Laurier, the wife of Wilfrid Laurier, Prime Minister of Canada, in Ottawa, Ontario, Canada, for exceeding the 10 miles per hour speed limit.[26]



CANADA:
Province of Ontario, | SUMMONS TO THE DEFENDANT UPON INFORMATION OR COMPLAINT
CITY OF OTTAWA.
To wit,

22. 3

To Lady Laurier
Of the City of Ottawa.

Whereas, information hath been laid before the undersigned one of His Majesty's Justices of the Peace in and for the said City of Ottawa, FOR THAT YOU the said Lady Laurier on the 8th day of February A.D. 1910, at the City of Ottawa aforesaid, did unlawfully run a motor vehicle upon a public highway, to wit, Bank street in said City, at a greater rate of speed than ten miles an hour, — contrary to the form of the Statute in such case made and provided:

THESE ARE THEREFORE to command you in His Majesty's name to be and appear on Wednesday the 16th day of February A.D. 1910, at ten o'clock in the forenoon, before me at the Police Court in the said City of Ottawa, or before such other Justice or Justices of the Peace for the said City of Ottawa, as shall then be there, to answer to the said information and to be further dealt with according to law. Herein fail not.

Given under my hand and seal this 14th day of February in the year of our Lord one thousand nine hundred and ten, at the City of Ottawa, aforesaid

Sgd. W. P. Lett
J. P. Ottawa.

Fig 2.25 Ticket in 1910

2.3.7-History of traffic radars

Under a title “Who Made That Traffic Radar?” posted on New York times by Pagan Kennedy Aug. 30, 2013 [27]

John L. Barker Sr. — then an engineer at Automatic Signal Company — had been working on traffic lights in the 1930s, but during World War II, he dedicated himself to military research. “He pointed the machine out a window, and he tested it on cars that drove through an intersection.” Barker was about 8 years old at the time and admits that his memories are hazy. “I have no idea why I was even involved,” he says.

When the war ended, Barker Sr. experimented with a new, peacetime application for the technology. He would pack the radar equipment in the trunk of his car and play cop on the Merritt Parkway. “He would pull off the road and open the trunk so that the equipment faced traffic,” his son says. At the time, police officers had no precise way to clock a car; Barker knew that his device could change the rules of the road.

In 1947, the town of Glastonbury, Conn., deployed Barker’s machine on Route 2, creating what was perhaps the world’s first speed trap. “This is the latest scientific method,” a police captain named Ralph Buckley told a reporter in 1949. “It removes the possibility of human error.” And he added, “Any speeder who gets caught will have to argue with a little black box.”

Argue they did. In 1955, a Connecticut woman contested her speeding ticket in court, claiming that the radar must have been broken on the day it registered her blowing through a 25-mile-per-hour zone. Her lawyer further protested about a “lack of fair play” on the part of the police for pointing this high-powered, wartime technology at drivers. Barker was called in as an expert witness. He insisted his radar machine didn’t lie. (According to his son, he frequently appeared in court to defend his invention.

By the late ’70s, every New York City Highway Patrol car carried a radar gun. “Sometimes a citizens’-band operator will spot a radar unit and warn everybody that ‘Smokey’ is taking pictures,” one of the state troopers told The Times in 1977.

Today the police use digital radar guns that are far more accurate than the old analog radar. But the most important part of the speed detector may have nothing to do with its hardware. As John Barker Sr. himself pointed out in a 1948 issue of The Traffic Quarterly, drivers change their behavior when they think they are being watched. As soon as Connecticut State Police first parked their radar truck beside the highway, the traffic slowed down — even though the troopers were handing out “courtesy cards,” not speeding tickets.

2.3.8-High speed ticketing and speeding detection systems using AI

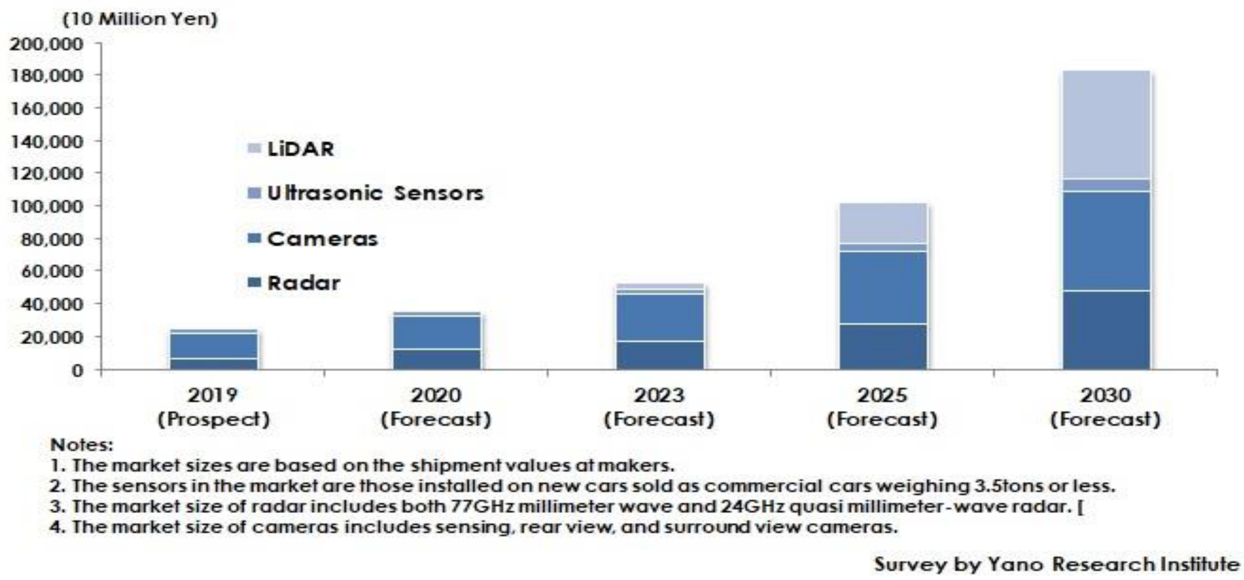


Fig 2.26 Total market size of the sensors for ADAS

Above figure [28] The total market size of the sensors for ADAS and autonomous driving systems in China is projected to achieve 252,940 million yen in 2019. With regard to the market sizes by sensor type, radar reached 68,040 million yen; cameras achieved 156,100 million yen; and ultrasonic sensors 28,800 million yen. (Both 77GHz millimeter wave and 24GHz quasi millimeter-wave radars are included in the market size of radar-type sensors, while sensing, rear view, and surround view cameras are included in camera-type sensors.)

The demand for radars is increasing, not only that for 77GHz long range radar but also short range radar installed on the both rear sides of the vehicle (both 77GHz and 24GHz are used for short range radar). Short range radar is used for BSD (blind-spot detection system) and DOW (door open warning) systems. DOW systems are highly demanded in China, because the traffic accidents caused by motorbikes closing in from the rear without being noticed occur from time to time. The number of cameras used in ADAS for forward sensing, rear-view and surround view has been expanding. Those made by local Chinese makers are also increasing in number, primarily used as rear-view cameras. Supersonic sensors are increasingly used for collision warning as well as for parking support systems. Four to twelve supersonic sensors are installed per vehicle (private or commercial vehicle) depending on the use.

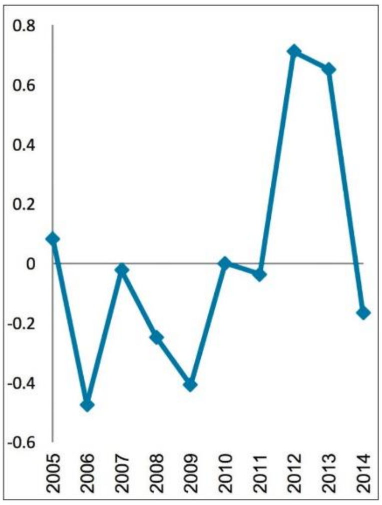
2.3.9-How new smart violation detection systems affected traffic

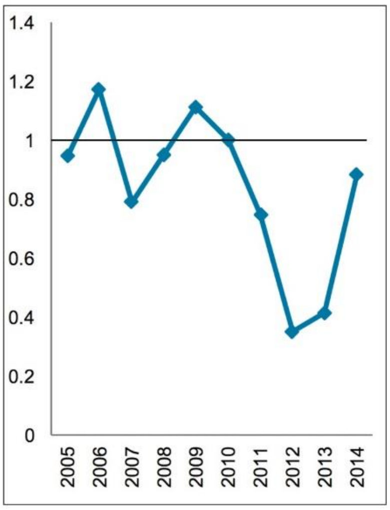
In an article “Outcomes of road traffic injuries before and after the implementation of a camera ticketing system: a retrospective study from a large trauma center in Saudi Arabia” [29]

Their objective was to examine injury severity and associated mortality at a large trauma center before and after the implementation of the ticketing system.

The study included all trauma registry patients seen in the emergency department for a crash-related injury. All health outcomes improved in the period following implementation of the ticketing system. Following implementation, ISS scores decreased (-3.1, 95% CI -4.6, -1.6) and GCS increased (0.47, 95% CI 0.08, 0.87) after adjusting for other covariates. The odds of death were 46% lower following implementation than before implementation. When the data were log-transformed to account for skewed data distributions, the results remained statistically significant.

Figure 2.27 Figure 2.28 Figure 2.29





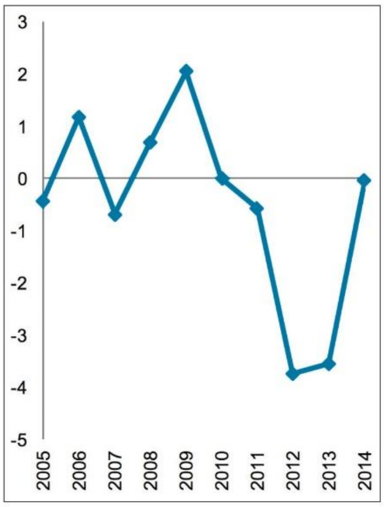


Fig 2.27 Glasgow coma scale scores relative to the year 2010.

Fig 2.28 Odds ratio of death relative to the year 2010.

Fig 2.29 Injury severity scale scores relative to the year 2010.

This study suggests positive health implications following the implementation of the camera ticketing system. Further investment in public health interventions is warranted to reduce preventable RTIs.

2.4-Internet of Vehicle (IoV)

2.4.1-Background and concept

The new era of the Internet of Things is driving the evolution of conventional vehicular ad-hoc networks (VANETs) into the Internet of Vehicles (IoV). IoV refers to the real-time data interaction between vehicles and roads, vehicles and the vehicle owners, vehicles and a centralized server, vehicles and vehicles, as well as vehicles and cities, using mobile-communication technology, vehicle navigation systems, smart-terminal devices, and information platforms to enable information exchange/interaction and a driving–instruction–controlling network system. IoV enables the gathering and sharing of information regarding vehicles, roads, and their surroundings. Moreover, it features the processing, computing, sharing, and secure release of information onto information platforms, including Internet systems. Based on such information, information platforms can effectively guide and supervise vehicles, and provide abundant multimedia and mobile Internet application services. IoV is an integrated network for supporting intelligent traffic management, intelligent dynamic information services, and intelligent vehicle control, representing a typical application of IoT technology in intelligent transportation systems (ITS). The concept of IoV has been recognized by more and more people in recent years, and it is currently in a stage of evolving from concept to reality. ITS in Europe and Japan have adopted certain forms of IoV technology. In New Delhi, all 55,000 licensed rickshaws have been fitted with GPS devices so that drivers can be held accountable for their questionable route selection. China’s Ministry of Transport had ordered that GPS systems be installed and connected on all long-haul buses and hazmat vehicles by the end of 2011, to ensure good driving habits and reduce the risk of accidents and traffic jams. The Brazilian government has set a goal for all cars in circulation to be fitted with electronic ID chips from its National Automated Vehicle Identification System (Siniav). IoV is a complex integrated network system, which connects different people within vehicles, different vehicles, and different environmental entities within cities. With the rapid development of computation and communications technologies, IoV promises huge commercial interest and research value.

2.4.2-Type of Communication

2.4.2.1-Communication between the vehicles and the vehicle owners

2.4.2.2-Communication between vehicles.

2.4.2.3-Communication between vehicles and a centralized server.

2.4.2.4-Communication between server and third parties like police patrol, ambulance, fire-engine, etc.

2.4.2.1-Communication between the vehicles and the vehicle owners

Communication between vehicles and the Vehicle Owners Few attributes of the vehicle like the vehicle speed and fuel level are directly reported to the users in the vehicles, only when the vehicle is in use. However, to enable the user to receive active updates even when the vehicle is not being used and when the user is away from the vehicle, an onboard processor is useful.

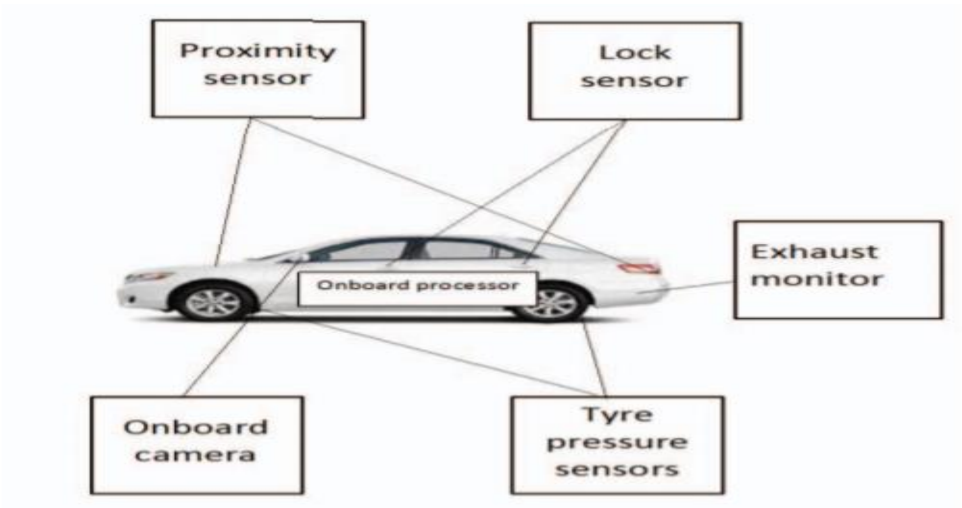


Fig 2.30 Vehicle to its Owner Communication

The active updates mentioned may involve.

a. security alert about the vehicle,

b. damage alert about the vehicle

c. the attributes like proximity, tire pressure and vehicle lock

The sensors and the onboard processors in the vehicle are shown in Figure 1.

2.4.2.2-Communication between vehicles

Communication between vehicles involves the sharing of these data:

1. Proximity between the vehicles
2. Monitoring of the immediate surroundings of a vehicle through onboard cameras.
3. Speed of vehicles within a particular radius of the vehicle under consideration.
4. Tyre burst related accidental information.

When a vehicle is on the road or even when a vehicle is parked, its proximity to other vehicles in its immediate vicinity can prove to be crucial in avoiding accidents and damage to the vehicle.

Being able to know the speed of the vehicles surrounding a particular vehicle can help in issuing a warning to the nearby vehicles on the road about a fast-approaching vehicle. Thus, the vehicle which receives the warning message will alert the driver regarding the problem next to him [5].

2.4.2.3-Communication between vehicles and a centralized server

The data monitored from the vehicle is relayed to the nearest communications node via an onboard computer. The node in-turn communicates the data via a satellite to the communications node of the server which monitors breaches. The server stores the data in the database and analyzes the data for the breach. It then provides a suitable solution to the vehicle through the same channel from which it received the messages which is shown in Figure 2.31.

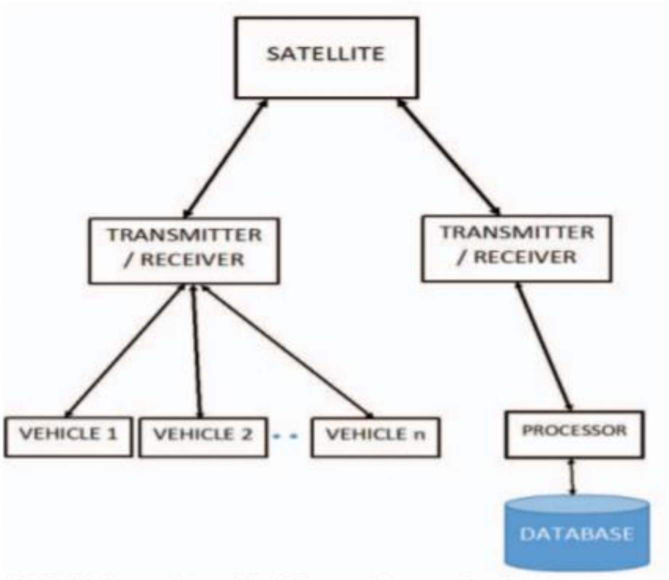


Fig 2.31 Vehicles to Centralized Server Communication

To monitor the metrics of the vehicles, a number of sensors are deployed on each vehicle. IoV make these sensors work in unison to be able to derive reasonable inferences from the data generated. It is not uncommon for automobiles to have sensors in-built. However, with the vast amount of data that needs to be analyzed, sensors are to be standardized to have effective results. The metrics of the automobile that need to be monitored are:

- Tire pressure
- Fuel level
- Speed! velocity reading
- Exhaust gases' contents
- Vehicle lock

Sensors are fixed at their respective positions to continuously monitor the data being generated. All these localized sensors are to be set with threshold values. When breached, all the data regarding the breach are to be sent to a processing and communications device on-board. The processor will basically be a minicomputer on board, powerful enough to handle the processing of the incoming data and the communication modules implemented on board. Use of a raspberry pi processor on board has been tested and proven to be successful and can be extended to vehicle management as well. It offers some significant advantages in terms of power consumption and speed of processing, and it is used as a communications device.

2.4.2.4-Communication between Server and Third Parties

This mode of communication occurs between the server and the third parties including:

- Emergency response like ambulance, fire-engine
- Pollution control
- Police patrol

Data deemed to be of primary concern are the data regarding vehicular collision, temperature spikes, theft etc. When these data are reported from the onboard processor on the vehicle, to the server, they are forwarded to the respective third parties. These third parties then correspond directly with the vehicle under consideration and take measures to provide necessary assistance. Deciding when the third parties need to be triggered is up to the centralized server. Information regarding how to classify the incoming messages is of primal importance and is to be pre-fed into the server's decision-making algorithm.

3- RELATED WORK AND SIMILAR SYSTEM

3.1 Related Work

Convolution neural network has long been used in the field of digital image processing and object detection, and has achieved great success Overviews of existing solutions in digital image processing object detection field that utilize convolutional neural networks (CNN) are reviewed for the related literature. JIUDONG YANG and JIANPING LI [30] give a clearer explanation of the structure of CNN. At the same time, give a brief summary of current CNN research in image processing and natural language processing.

In object detection, previous studies tested accuracy of their own algorithm and validated it by comparing the results against the results of methods from earlier studies with the same parameters. The study of Vibha et al. [31] yielded the most accurate method in detecting moving vehicles with a percentage of 100%.

In another study researcher focused on determining vehicles physical features with image processing and deep image classification [32].

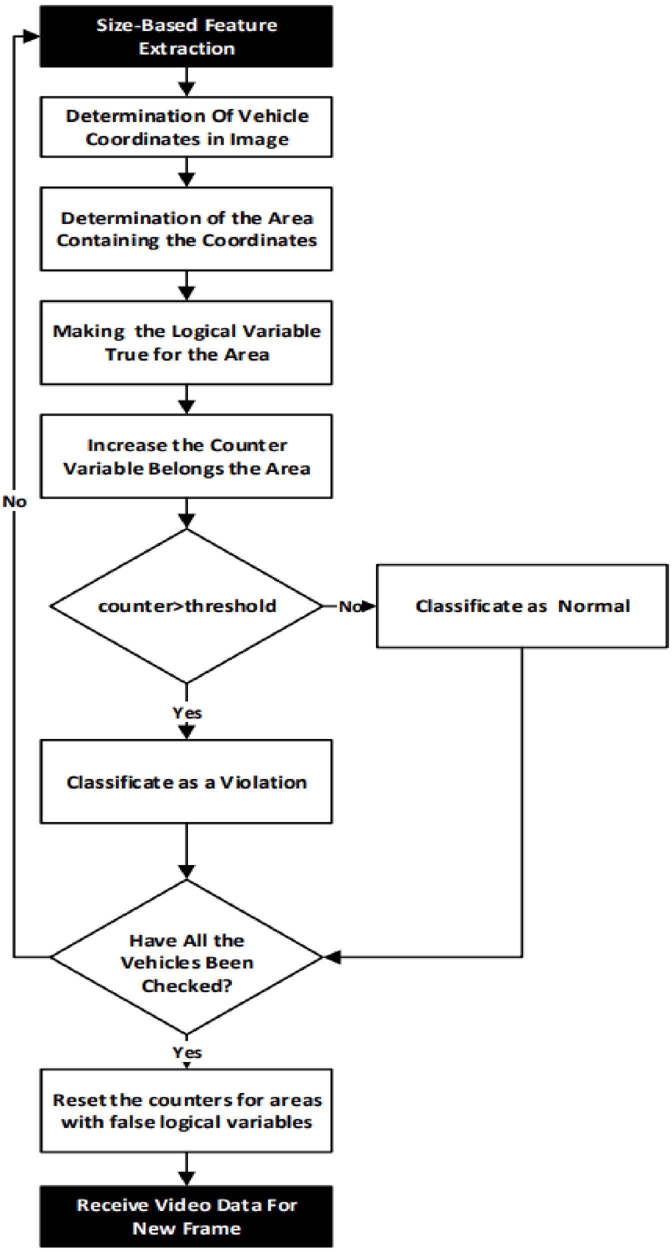


Fig 3.1 Block diagram of literature (35) study.

The study tested their own methods using a video feed containing eleven actual moving vehicles and yielded a result of detecting and tracking all of the vehicles present.

Over the past few decades many researchers and developers have developed many algorithms for detecting traffic rule violations, accident detection and vehicle classifications. Several previous studies used estimation of speed in order to predict if the vehicle is a red-light runner (RLR). This is done by measuring the rate of deceleration of the vehicle would be enough for it to stop before the violation line [33] [34].

One of the study in the literature focused on solution of the parking problems [35]. It focused on the parking time violation detection [35]. Researchers made some pre-processing operations. Then they made background subtraction and detection of contours. Finally, they found vehicle size as pixel and area as pixelsquare for size based feature extraction. Researchers used for vehicle detection with this size based feature extraction. Then they used the fps ratio and frame count for calculation of parking time. The block diagram of mentioned proposed method is given in Figure3.1 .

The research of Wang et al. used the same approaches in detecting the speed of a vehicle [36]. A pixel to distance conversion was applied to all of the frames. After mapping the frames with the conversion, a kinematics equation for the calculation of speed was be used with the factors gathered from the video.

3.2-Similar System in The Traffic Violation Detection

This is some applications used in recent days to reduce traffic violation and accident like: Car Electronic System, Radars like Saher in Saudia arabia and Fixed CCTV but it contains some drawbacks we will review it later.

Fig 3.2 Car Electronic System

3.2.1-Let’s talk now about Car Electronic System. In this system, we will see a small data sticker stuck on the windshield of the vehicle which will be readable through special infrared devices that will be installed at traffic lights as well as carried by traffic officials in the street or at a checkpoint. This sticker comes equipped with a SIM card, and by reading your sticker, officials can quickly get your vehicle’s data. It is a method of protecting the vehicles from theft, as they feature the vehicle owner’s full information. The e-stickers are also important in terms of whether the vehicles are being searched for in relation to criminal activities. This provides accurate information that can be used by the concerned authorities to survey areas of traffic density, and issue reports and statistics to manage and regulate traffic. Also this sticker aids integration with other security systems to identify the vehicles and any expired licences. It also allows for traffic rules to be applied and violations to be electronically recorded in a unified manner for all citizens. The sticker aims to facilitate the movement of citizens and their use of roads, as well as easing the payment of various kinds of fees due, including road fees, parking fees, and the like, without having to stop. A text message is sent after each violation, indicating the fees required from the vehicle's driver [20].

The sticker is a code number for each car by means of a device. Those violations are sent to the operating rooms without the intervention of the traffic man, and a fine for the violation will be signed according to its severity, and the rooms will be equipped to receive all the data of the violating cars while on the roads. Also, a sticker will be identified for each vehicle, and traffic lights will be provided with infrared reading devices, which read the code, and the information will be sent to the computer at the General Traffic Department. offending points. The new traffic law will require all motorists to issue an electronic sticker for each vehicle, and the electronic sticker must always be valid for use according to its purpose, and the vehicle may not be driven without this sticker, hidden or tampered with because the violator will expose himself to the legal issue and double the fine for the violation. This electronic chip includes all the data related to the vehicle, which is "the year of manufacture – the brand – the model – the shape – the color – the numbers of the chassis and the motor" and other data related to its plates, which are the "number – type of license – the license unit", and the data of its owner, which is The name, address, national number, telephone number and personal e-mail "if any", in return for the costs of issuing one poster at a value of 50 pounds at Egypt, Abdelrahman sayed et al [21].

Let's talk about some of the problems of the electronic label: the driver of the vehicle can hide it in the places where it is read. Any attempt to remove the e-sticker will damage it immediately. His equation for calculating the required speed is inaccurate. Also the e-sticker doesn't cover all of the roads just at traffic light, This gives drivers an opportunity to cheat on most of the road and avoid traffic lights as much as possible. We can say that these defects are because the product is new and may improve in its future plans, but defects remain at the present time and we are trying our best to make these defects advantages.

3.2.2-In 2009, the Kingdom of Saudi Arabia launched a traffic control system called **SAHER**, which in Arabic means, "watchful", and stands for the "Automated Traffic Violations Administering and Monitoring" program. The purpose of SAHER was to minimize accidents and maximize overall traffic efficiency throughout the kingdom. It includes the deployment of an intelligent transportation system, using the latest technology in traffic enforcement, traffic management sub-systems, and services to enhance safety on roadway networks. SAHER accomplishes this by optimizing the transportation infrastructure, attempting to decrease fatality rate, and improving road congestion. The system uses a digital camera network connected to and monitored by, the National Information Center of the Ministry of Interior (NIC). SAHER uses the radar system only. This system, called Multanova, is the same name of the company that produces this system as well. The Multanova Company was established in 1952 in Switzerland and is known in the traffic monitoring industry as one of the first companies to supply radar speed monitoring systems with photographic recording. Multanova offers solutions for red light and speed monitoring in road traffic (Multanova.ch) [22].

The Saher system has reduced the rate of accidents and deaths among drivers, by limiting exceeding the speed limit on the roads and adhering to traffic signals. The system helped prevent the recurrence of tragic accidents among citizens at a reasonable rate. Like any new system, Saher has some positive and negative advantages, including: Obstructing rescue vehicles, such as firefighting as some drivers deliberately make way for them. The lack of the proposed speed plates in some streets sufficiently. This causes vehicles to stop suddenly when drivers see Saher cameras without prior warning [23]. Also Saher doesn't see all the road where the drivers trying to avoid the places that saher in it as much as possible.

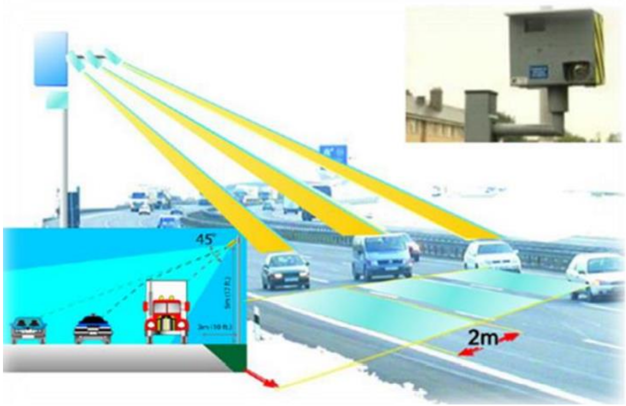


Fig 3.3 SAHER radar on a high-way

3.2.3-**A traffic enforcement camera** (also red light camera, road safety camera, road rule camera, photo radar, photo enforcement, speed camera, Gatso, safety camera, bus lane camera, flash for cash, Safe-T-Cam, depending on use) is a camera which may be mounted beside or over a road or installed in an enforcement vehicle (should be fixed) to detect motoring offenses, including speeding, vehicles going through a red traffic light, vehicles going through a toll booth without paying, unauthorized use of a bus lane, or for recording vehicles inside a congestion charge area. It may be linked to an automated ticketing system. Also this system generally has many problems, as the camera does not perform more than one function, unlike what is in our system, and each camera covers a specific place such as the radar (Saher), and some of them may not work except under specific wether conditions.

3.2.4-Feature of our system:

As we see this 3 examples with details and abstract about drawbacks to every one of them. We can say all of this drawbacks are advantages in our system, with camera can cover all the roads, so the drivers now can’t cheat in all of the roads and he will respect all the traffic rules. Also, there is nothing in his car that can cover it like an electronic sticker Since he can't cover the license plate, and the license plate is also always visible, if any appearance of concealment occurs to it, he will also be monitored as a violation.

4- SYSTEM OVERVIEW AND PLANNING

4.1 Project Planning:

- We start to make Systematic Review about our topic on the most popular library like IEEE, ScienceDirect, ACM, Springer and MDPI.

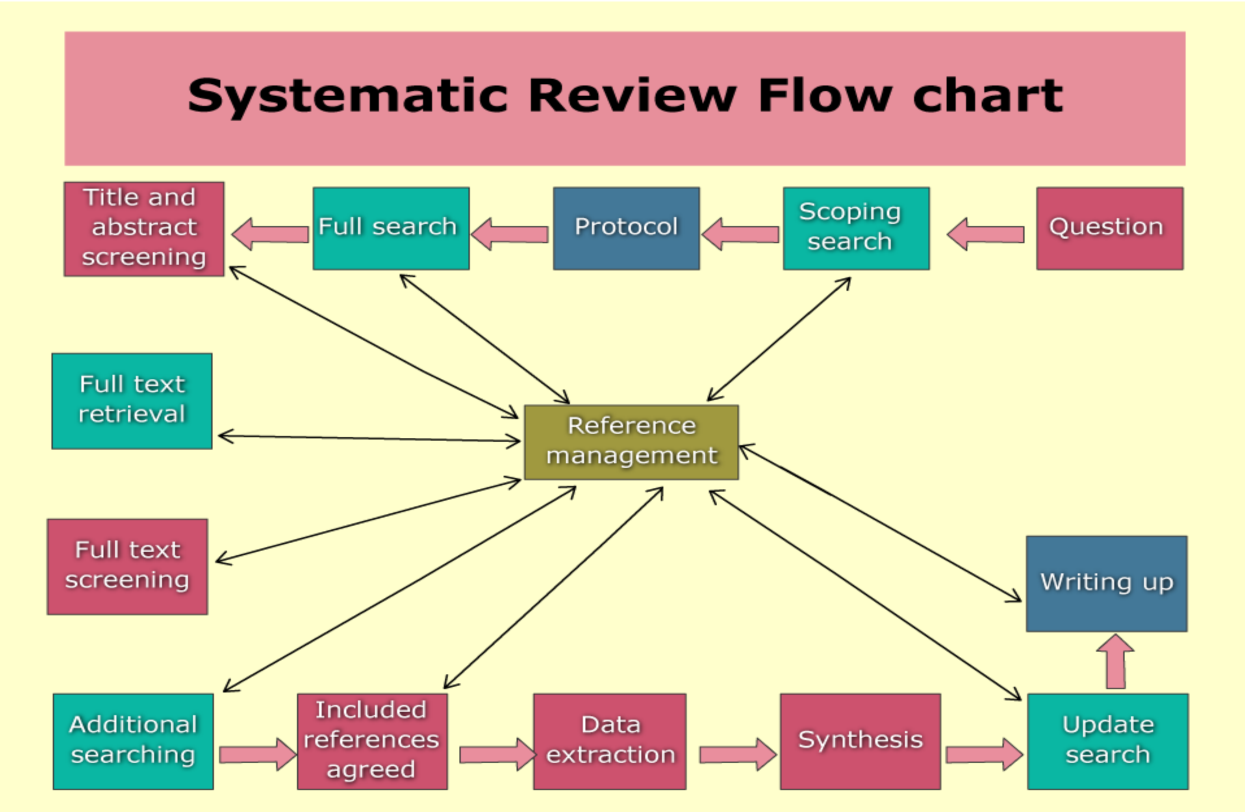


Figure 4.1 Systematic Review

As shown as in figure 4.1 we start by define clearly the question about our topic to be formulated this question like what are the current trends or research direction in the real time traffic anomaly detection, what are the open research areas in this domain and …etc, and then determines scope of search by using Search String " Document Title: "Anomaly Detection", All Metadata: vehicle and ("machine learning" OR "deep leaning")". Protocol consist of search all reliable studies addressing our question with our scope in different sources by that we make full search, then we Select the studies by means of clear inclusion and execution criteria, and evaluate the quality of these studies, after that we Extract data from each study and display them clearly, then Evaluate heterogeneity among the studies

- After finish Systematic Review we started writing proposal of our project which contain a brief description of our project and that identifies the subject of a forthcoming research project, outlines a research strategy, design of system, and provides a tentative list of references.
- After that we started to implement our design, in implementation we started to try different algorithms of CNN and then determine most suitable algorithm for our project or the algorithm that gives most accurate result with our project actually in object detection.
- After that we started to use that algorithm to detect speed limit violation
- tried to detect reverse direction violation
- Then detect number in License Plate to determine the identity of the violator to record violation of that violator.
- Then start to predict collision

4.2 Project Management Activities:

- Determine project objectives.

  The goal of the project is to automate the traffic rule violation detection system and make it easy for the traffic police department to monitor the traffic and take action against the violated vehicle owner in a fast and efficient way. Detecting and tracking the vehicle and their activities accurately is the main priority of the system.

- Planning the project activities according to project objectives.

  Due to our objective we need to build model can make object detection and apply That model in three use case speed limit violation, Reverse direction violation and collision prediction after that start to install hardware and complement possible adjustments.

- Assessing and mitigating risk.

  Cameras can be stolen from a car or can be damaged due to external factor, and we will try to mitigate that by increase number of camera in different car to reduce the effect.

  Any false detection in system can cause problem, and we will try to mitigate that by try to increase accuracy of system

- Estimating resources and budget.

- 4 cameras carry a picture that carries with it resolution FHD, their price is 8000 pounds, in order to take pictures to be processed.

- 2 CPUs priced at 8000 pounds are used in image processing in vehicles.

- 1 storage unit of 850 pounds used to store data for the idea that the driver has problems.

- 1 4G / 3G / 2G network communication unit, which works with a GPS chip, at a price of 1000 pounds, used to send data to the server.

- Reserving a server to receive the data equipped with an image processing unit to confirm the violation for a period of 6 months to produce 6300 pounds (1050 pounds per month ($ 60)), In order for the data sent from the car to be processed to the server.

- Organizing the project activities.

Project activities	Due Date
Systematic Review delivery	1/12/2020
Proposal delivery	15/12/2020
Building Algorithm and core programming delivery	14/02/2021
Apply Algorithm on first use case (speed limit violation)	01/03/2021
Apply Algorithm on second use case (Reverse direction violation)	14/03/2021
Apply Algorithm on third use case (collision prediction)	29/03/2021
Use Algorithm to detect License Plate to catch the violator	15/04/2021
Hardware installation	20/05/2021
Finalization, complement and possible adjustments	29/05/2021
Documentation delivery	4/07/2021

- Reporting progress.

- Analyzing the results based on the facts achieved.

- Forecasting future trends in the project

5-SYSTEM DESIGN

System consist of two main part Edge Node processing and Server.

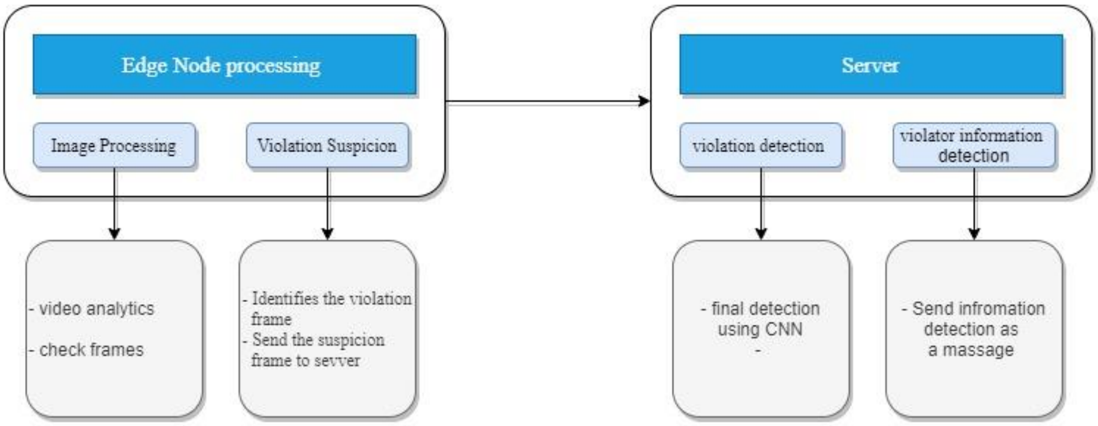


Fig 5.1

5.1 HARDWARE ARCHITCTURE

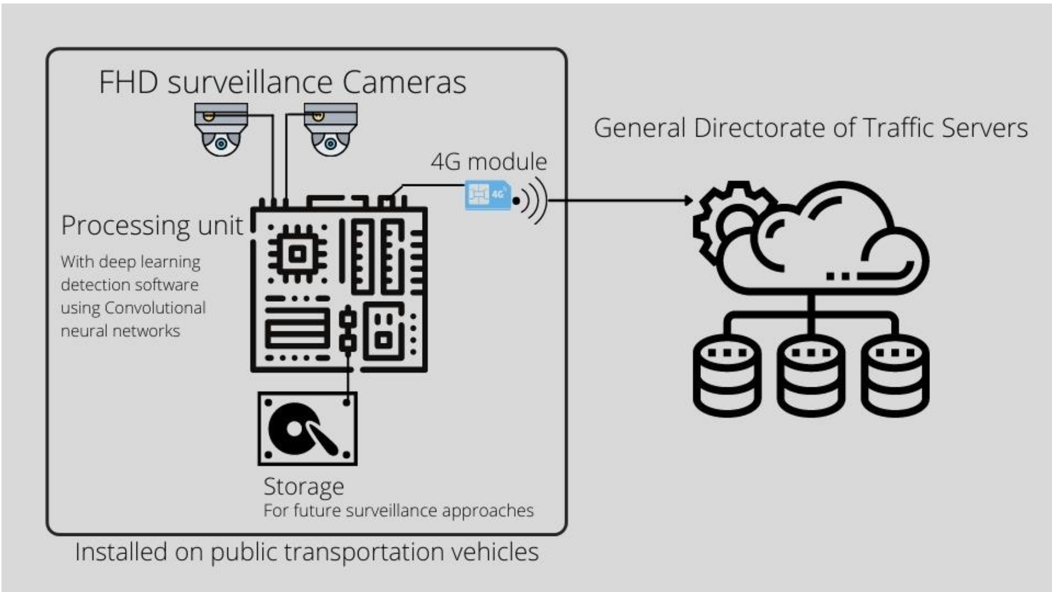


Fig 5.2

Our proposed hardware is composed of:

1- central processing unit for image processing
2- Full HD surveillance cameras set.

3- 4G module for connectivity between our proposed system and servers of general directorate of traffic.

4- Hard disk drive for surveillance purposes.

5.2 PROJECT ARCHITCTURE

Figure 5.3 shows one function of our proposed system which is speed violation detection.

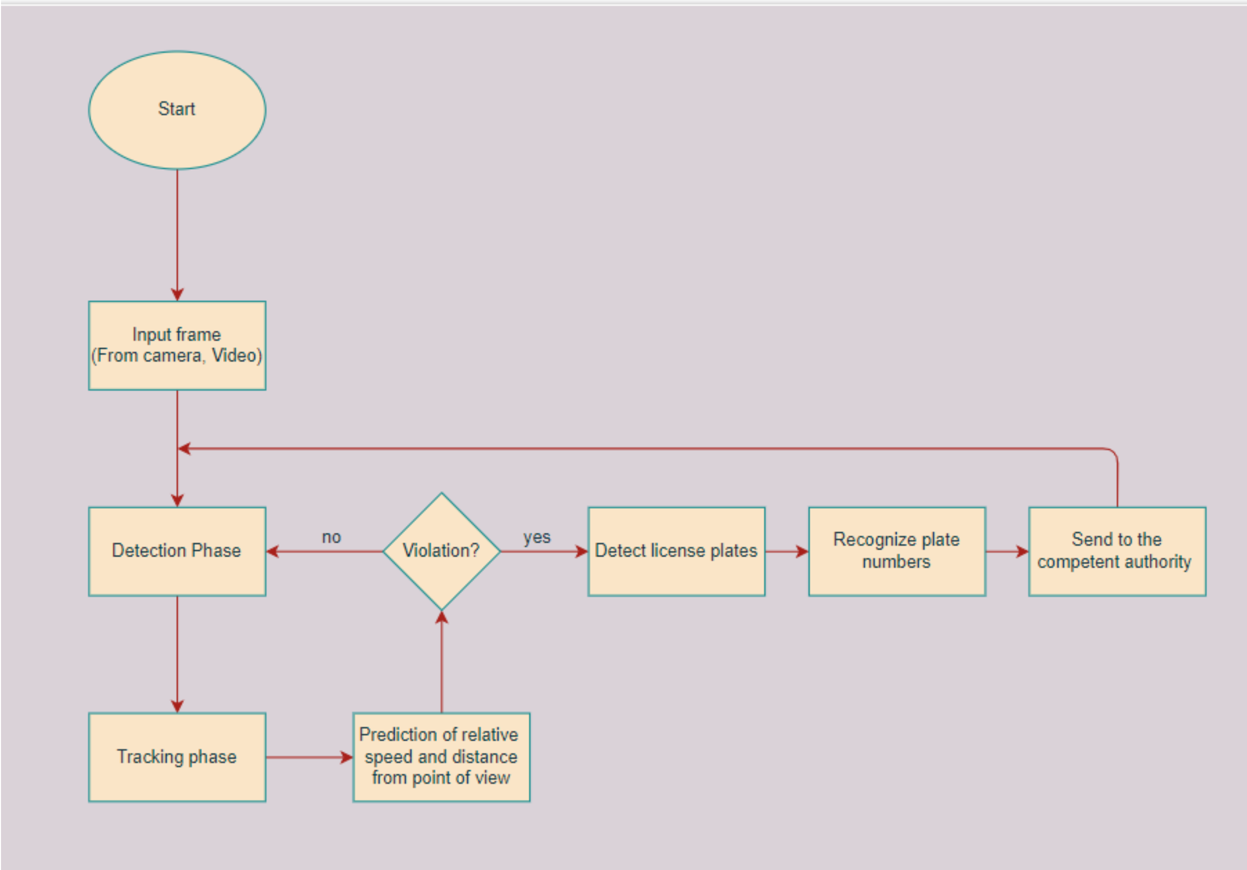


Fig 5.3

Our system takes an input from live camera stream or video stream. Then detection phase starts which every frame is being searched for an object to be recognized. It then starts tracking this object and estimates its relative speed (depending on speed of camera) and uses our proposed speed estimation method. If the object exceeds the speed limit, after a small delay to ensure confidence, it captures a shot of the violator then starts searching for license plates in this shot.

After finding a license plate it recognizes the numbers and characters and send them to general directorate of traffic.

Figure 5.4 shows another function of our proposed system which is opposite direction drivers or reckless drivers detection.

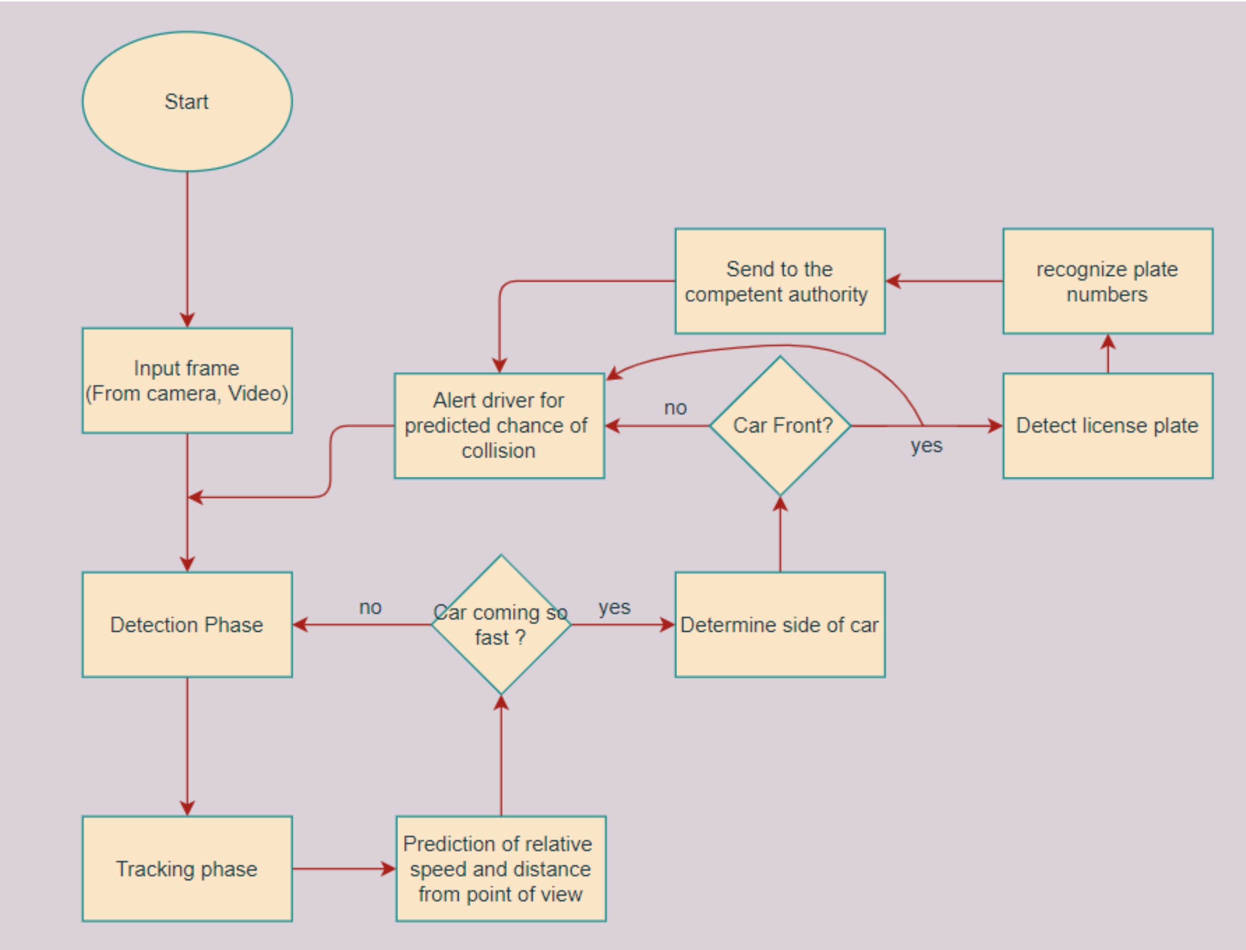


Fig 5.4

Our system takes an input from live camera stream or video stream. Then detection phase starts which every frame is being searched for an object to be recognized. It then starts tracking this object. If it detects that the object is coming fast and safe distance is also decreasing fast, it instantly takes a shot of the object and determines if it is an opposite direction violator or just a car that is hardly breaking. In case of an opposite direction violator, it also searches for a license plate to send it to general directorate of traffic in parallel with alerting the driver to act.

5.3 SOFTWARE ARCHITCTURE

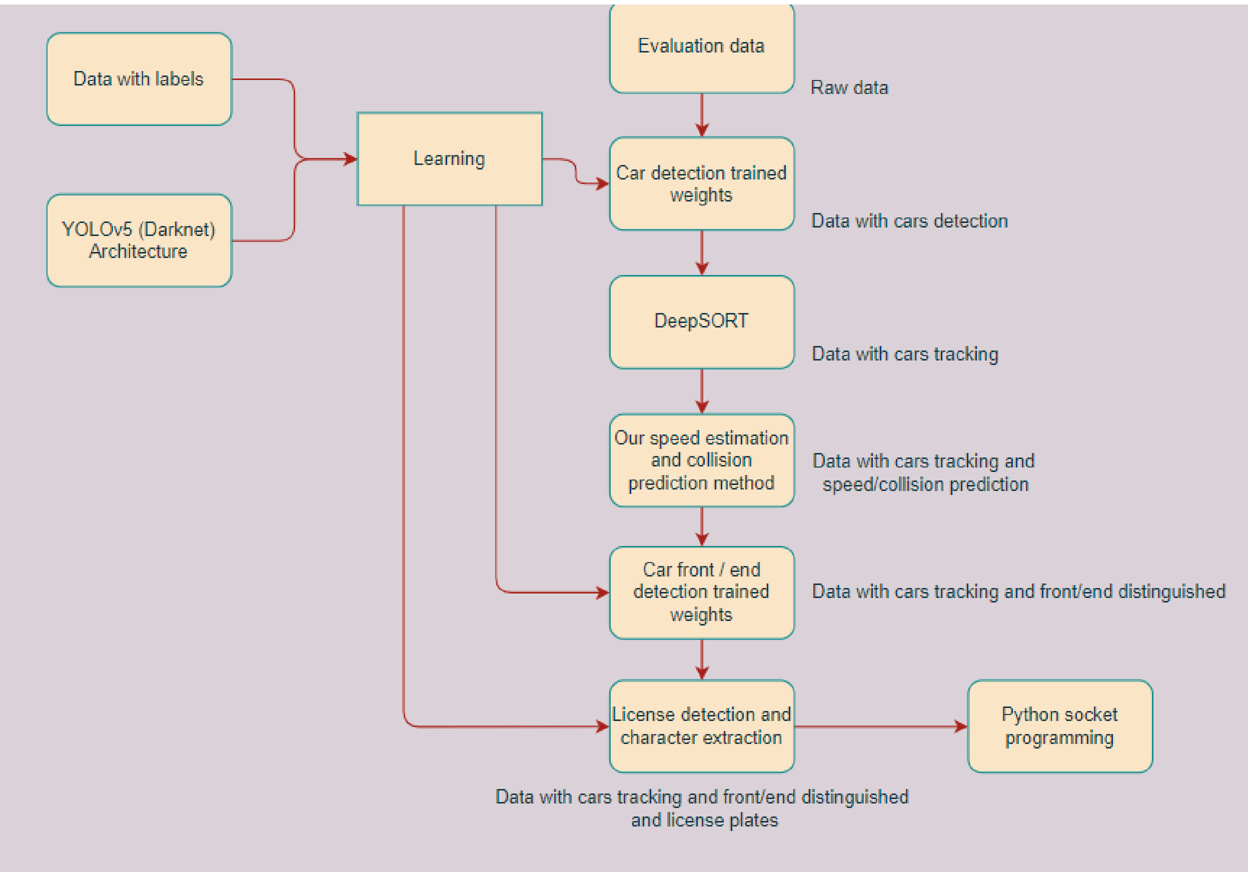


Fig 5.5

In figure 5.5 , We showed our proposed model's software architecture which starts with labeled dataset and a structure of a learning algorithm. Then a process of learning starts that gives weights as result. Then we take raw evaluation data as an input to our weights to start detection. After detection, the data is passed to DeepSORT for tracking. Then under certain circumstances and situations such as detection of violation or prediction of a collision, it starts to recognize car front or end then capture their license numbers.

5.4-Identify the sequence diagram

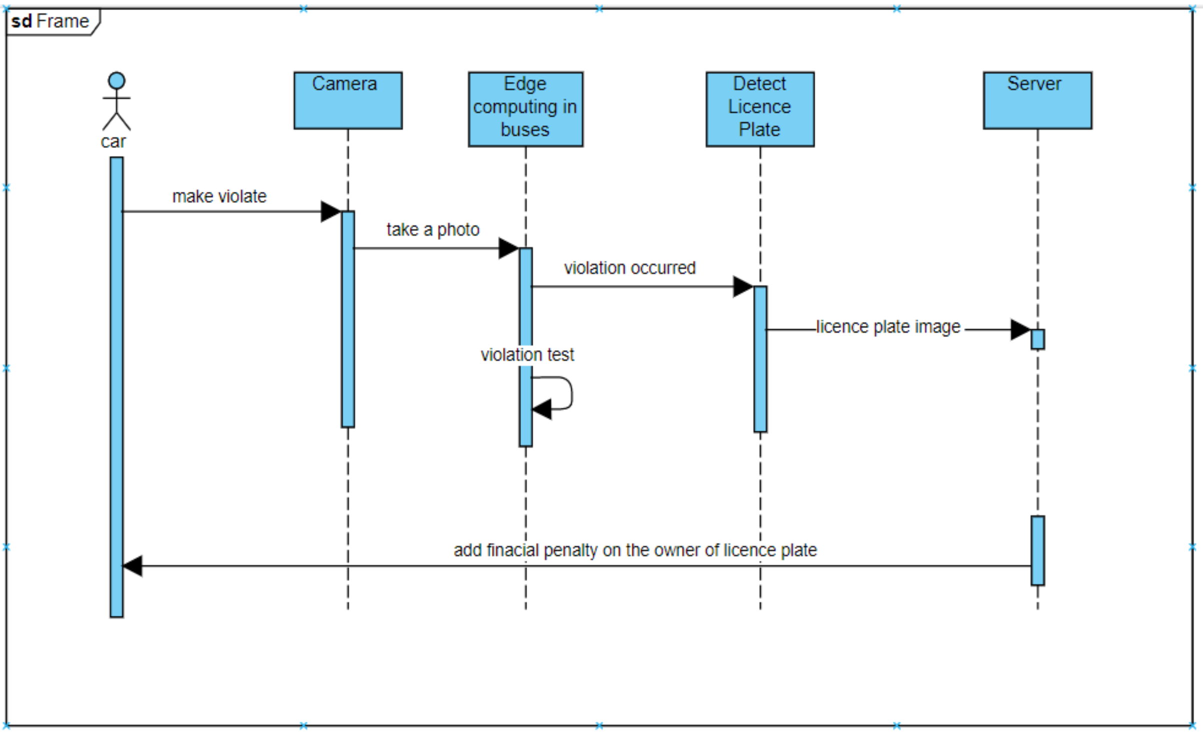


Fig 5.6 sequence diagram

6-TECHNICAL OVERVIEW

The goal of the project is to automate the traffic rule violation detection system and make it easy for the traffic police department to monitor the traffic and act against the violated vehicle owner in a fast and efficient way. Detecting and tracking the vehicle and their activities accurately is the main priority of the system.

6.1-Project Overview

In this project, we have fixed two cameras CCTV in public transport vehicles, when the vehicle works these cameras turn on, these cameras work by an algorithm called YOLOv3. This algorithm detects everything that appears in the image and identifies it. When one of the cameras detects a car with suspicion of violation, it sends these suspicious frames to a special server through a GSM chip equipped with a GPS, and through the server, it is determined whether this car is offensive or not, in case the vehicle violated, the server sends the violation to the competent authorities.

Fig 6.1 project overview

6.2-What is object detection?

Object detection, a subset of computer vision, is an automated method for locating interesting objects in an image with respect to the background. For example, Figure 1 shows two images with objects in the foreground. There is a bird in the left image, while there is a dog and a person in the right image.

Solving the object detection problem means placing a tight bounding box around these objects and associating the correct object category with each bounding box. Like other computer vision tasks, deep learning is the state-of-art method to perform object detection.

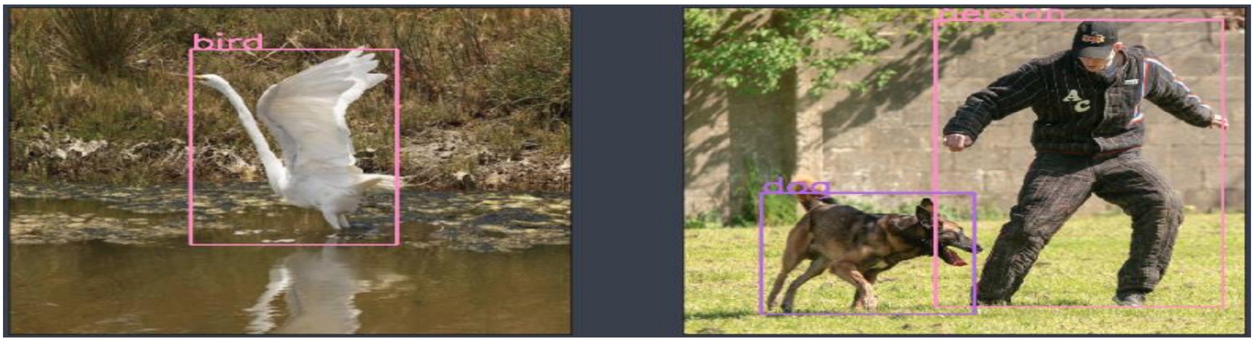
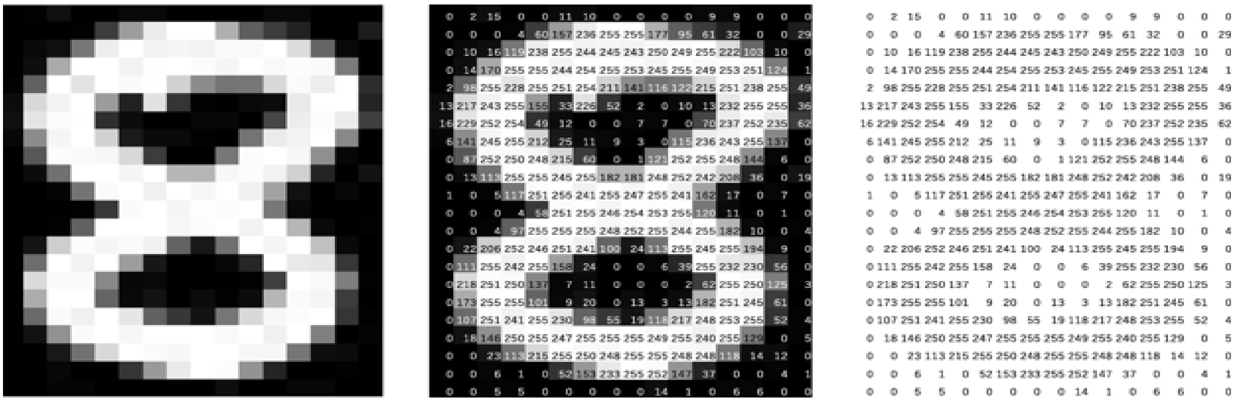


Fig 6.2 object detection

6.2.1-Computer vision is a field of artificial intelligence that trains computers to interpret and understand the visual world. Using digital images from cameras and videos and deep learning models, machines can accurately identify and classify objects and then react to what they "see." [38]



A computer sees an image as an array of numbers. The matrix on the right contains numbers between 0 and 255, each of which corresponds to the pixel brightness in the left image. Both are overlaid in the middle image.

6.2.2-How object detection works?

A key issue for object detection is that the number of objects in the foreground can vary across images. But to understand how object detection works, let's first consider restricting the object detection problem by assuming that there is only one object per image. If there is only one object per image, finding a bounding box and categorizing the object can be solved in a straightforward manner. The bounding box consists of four numbers, so learning the bounding box location can naturally be modeled as a regression problem. From there, categorizing the object is a classification problem.

The convolutional neural network (CNN) shown in Figure 6.3 provides a solution to the regression and classification problems for our restricted object detection problem. Like other traditional computer vision tasks such as image recognition, key points detection, and semantic segmentation, our restricted object detection problem deals with a fixed number of targets. These targets can be fit, by modeling the targets as a fixed number of classification or regression problems.

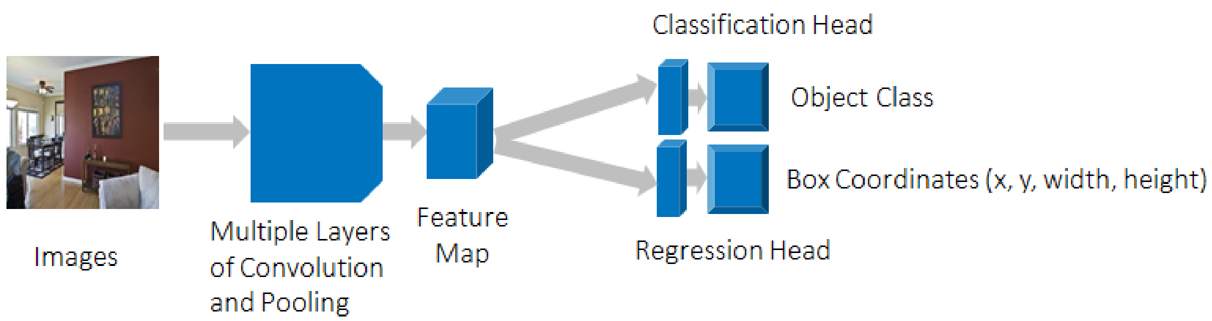


Fig 6.3 Network architecture to detect one object in the image

As already noted, true object detection must be able to deal with N objects, where N varies from image to image. Unfortunately, the CNN shown in Figure 2 cannot solve this more general problem. It may be possible to use a variant of the CNN by hypothesizing many rectangle box locations and sizes and simply use the CNN for object classification. We often refer to these rectangle boxes as windows. To be comprehensive, the window hypotheses must cover all possible locations and sizes in the image. For each window size and location, a decision can be made on whether there is an object present, and if so, the category for the object.

Figure 6.4. shows some of the possible windows when realizing object detection by this approach. Since there is a discrete number of pixels in the image, the total number of windows is a huge but countable number. However, this approach is computationally impractical given the huge number of windows to consider. [37]

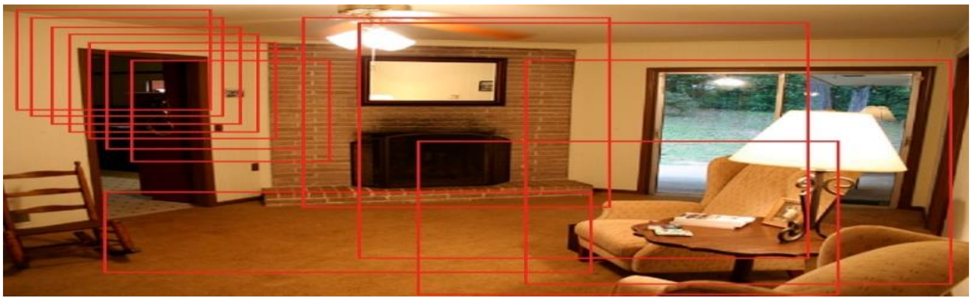


Fig 6.4 possible windows object detection

There are two approaches to find this subset of windows, which lead to two different categories of object detection algorithms.

1. The first algorithm category is to do region proposal first. This means regions highly likely to contain an object are selected either with traditional computer vision techniques (like selective search), or by using a deep learning-based region proposal network (RPN). This category includes algorithms like Faster R-CNN, R_FCN and FPN-FRCN.
2. The second algorithm category only looks for objects at fixed locations with fixed sizes. These locations and sizes are strategically selected so that most scenarios are covered. These algorithms usually separate the original images into fixed size grid regions. For each region, these algorithms try to predict a fixed number of objects of certain, pre-determined shapes and sizes. Algorithms belonging to this category are called single-stage methods. Examples of such methods include YOLO, SSD and RetinaNet.

6.2.3-YOLO for object detection

YOLO (You Only Look Once) is the representative algorithm in single-stage object detection method. The steps it follows to detect objects are represented in Figure 6.5 and in the list below:

Fig 6.5 object detection by YOLO

We will talk about it in details in other part.

6.3-The convolutional neural network (CNN):

The essence of CNN is a multi-layer perceptron. The key to its success lies in the local connection and the way of sharing weights. CNN is a kind of neural network, its weight sharing network structure makes it more similar to the biological neural network, reduces the complexity of the network model and the number of weights [39].

CNN Structure:

1. A convolutional layer is a set of parallel features that are composed by sliding different convolution kernels. In addition, at each sliding position, an operation of a product corresponding to the product and summation is performed between the convolution kernel and the input image to project the information in the receptive field onto an element in the feature map. Convolution kernel's size is much smaller than the input image. The size of the convolution (feature map) can be calculated by Equation (1):

$$Dim(H_1, W_1, D_1) = ((H + 2Zp - k_1)/Z_S + 1, (W + 2Zp - k_2)/Z_S + 1), K_D \quad (1)$$

Convolution layer has several main features. The first one is local perception. The reason for the use of local perception is that most people think that the parts in the picture that are close to each other are more relevant and the parts that are far away are less relevant. Local perception greatly reduces the number of weights, and does not duplicate convolution on the same sliding window. As well as parameter sharing. Before introducing the parameter sharing, we should know that the weight of the convolution kernel is obtained by learning, and the weight of the convolution kernel in the convolution process will not change. This is the idea of parameter sharing.

CNN structure as shown in Fig.5 blew:

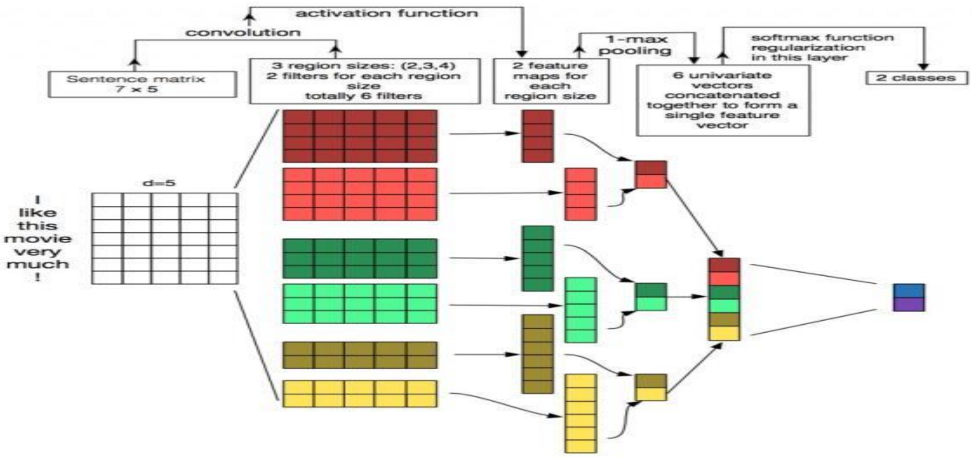


Fig 6.6 CNN Structure

2. Activation function, not to deactivate what, but how to activate the characteristics of neurons through the function to retain and map the features. The activation function defines the output of a given neuron after a given set of inputs. We pass the weighted sum of linear network input values to the activation function for nonlinear conversion. The role of this step is to preserve the features and remove some redundancy in the data, which is the key to the neural network's ability to solve nonlinear problems. Several common activation functions as shown in Fig.2 below:

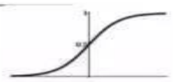
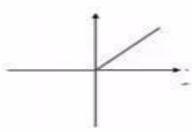
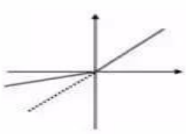
Name	Functions	Derivatives	Figure
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))^2$	
tanh	$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$	
ReLU	$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0. \end{cases}$	$f'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0. \end{cases}$	
Leaky ReLU	$f(x) = \begin{cases} 0.01x & \text{if } x < 0 \\ x & \text{if } x \geq 0. \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0. \end{cases}$	
Softmax	$f(x) = \frac{e^x}{\sum_1^i e^x}$	$f'(x) = \frac{e^x}{\sum_1^i e^x} - \frac{(e^x)^2}{(\sum_1^i e^x)^2}$	

Fig 6.7 Common Activation Function

3. A pooling layer refers to the down sampled layer that combines the output of a cluster of front neurons with the underlying individual neurons. The dimensions of the pooling layer can be calculated by Equation (2):

$$Dim(H_2, W_2, D_2) = ((H_1 - k) / Z_S + 1, (W_1 - k) / Z_S + 1), D_n \quad (2)$$

4. The fully connected dense layer acts as a classifier. If operations such as convolutional layer, pooling layer, and activation function layer map the raw data to hidden feature space, the fully connected layer serves to map the learned features into the sample markup space.

5. The loss function is used to estimate the degree of inconsistency between the predicted value and the true value of your model. It is a non-negative real value function, usually expressed as L (Y, f (x)). The standard form of the loss function:

$$L(Y, f(x)) = | Y - f(x) | \quad (3)$$

The smaller the loss function, the better the model's robustness.

6. The feed forward propagation of a convolutional network can be mathematically interpreted as multiplying the input value by the random initialized weight, then adding an initial offset term to each neuron, and finally summing all the products of all the neurons Fed into the activation function, activating the function to non-linearly convert the input value and output the activation result. The feedforward neural network has the advantages of simple structure and wide application, and can approximate any continuous function and square integral function with arbitrary precision. Moreover, any finite training sample set can be accurately implemented. The structure of feed forward propagation is shown in Fig.7.

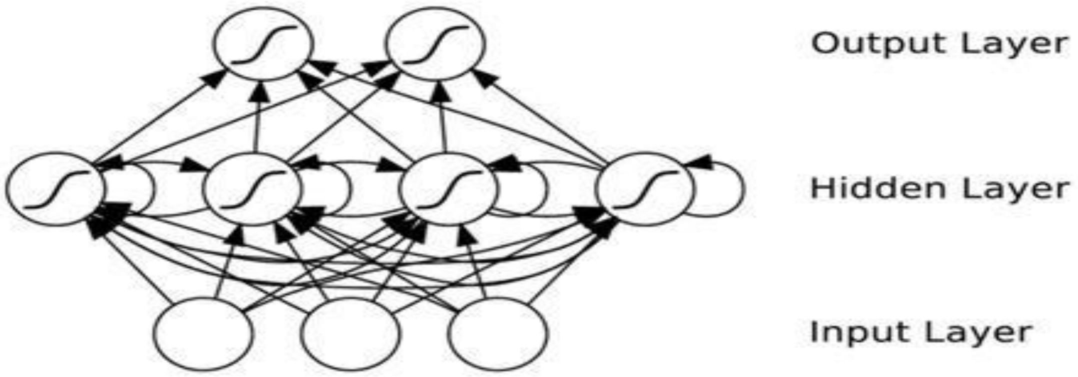


Fig 6.8 Structure of feed forward propagation

7. The back propagation algorithm of neural network is based on the steepest descent gradient. The goal of back propagation is to update the weights of the connections so that the actual output of each neuron is closer to the expected output, reducing errors in each neuron and the entire network.

The structure of back propagation is shown in Fig.6.9

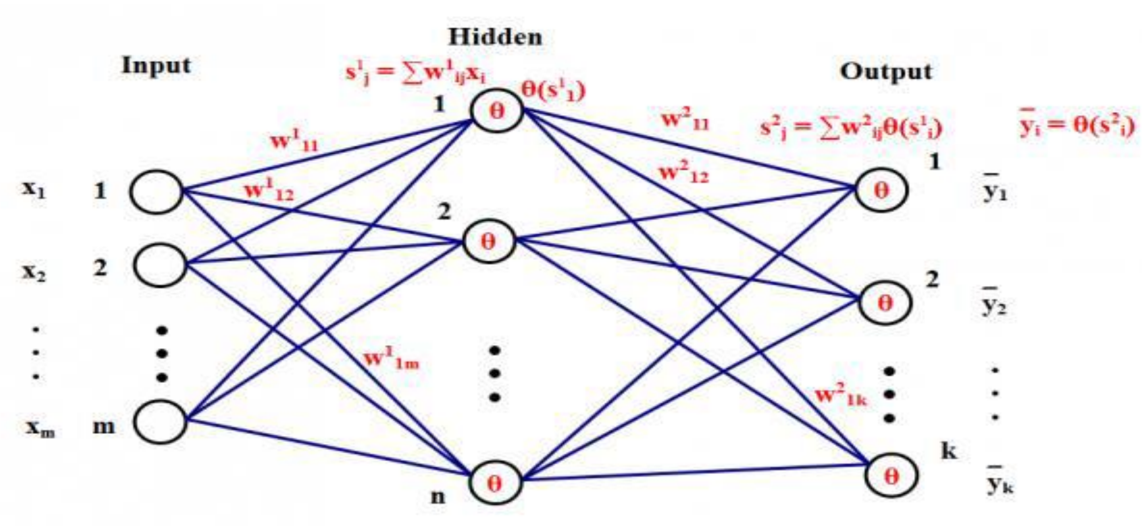


Fig 6.9 Structure of back propagation

The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. For example, given many pictures of cats and dogs, it can learn the key features for each class by itself.

6.4-Let’s talk about YOLO (You Only Look Once):

6.4.1-Introduction

People glancing at an image, can instantly recognize what the objects are and where they are located within the image. The ability to detect objects fast combined with the knowledge of a person helps to make an accurate judgment about the nature of the object. A system that simulates the ability of the human visual system to detect objects is something that scientists are researching on. Fast and accurate are the two prerequisites for which an object detection algorithm is examined. Object detection is one of the classical problems in computer vision. It not only classifies the object in image but also localizes that object. In previous decades, the methods used to address this problem consisted of two stages: (1) extract different areas in the image using sliding windows of different sizes and (2) apply the classification problem to determine what class the objects belong to. These approaches have the disadvantage of demanding a large amount of computation and being broken down into multiple stages. That makes the system difficult to be optimized in terms of speed [41].

For a long time, DPMv5(Deformable parts model) was like a gold standard in object detection taking 14 seconds to process a single image with an mAP of 33.7. This is extremely far from being real-time [42].

Then in 2014, R-CNN came out offering a huge boost in the accuracies (66 mAP) but longer processing times of about 20 seconds. Imagine these times in an application like a Self-driving car. The car would be pretty much blinded looking at each frame for 20 seconds. A lot of work after that focused on making R-CNN faster [42].

That lead to Fast R-CNN with improvement in accuracy (70 mAP) and also processing times (2 s/image). Even though these metrics significantly improve from previous algorithms, they are still slower in being real-time. This led to Faster R-CNN which was concurrently developed with YOLO, which bought detection speed to near real-time with 73.2mAP and 140ms/image which is 7 frames per second [42].

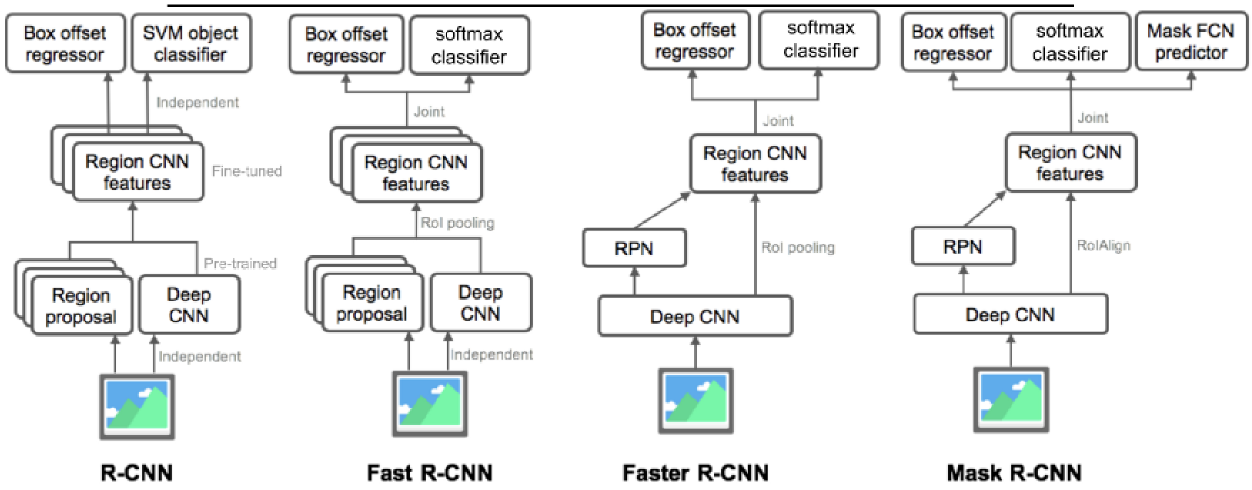


Fig 6.10 pic credits: https://lilianweng.github.io/

Then came YOLO in 2015 with huge improvements in speed making it real-time. The metrics for YOLO were 63.4mAP for accuracies and 22ms/image which is 45 frames per second. That's when the whole YOLO buzz started [42].

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
YOLO	63.4	45 FPS	22 ms/img

Fig 6.11

6.4.2-History of YOLO

6.4.2.1-YOLO v1

YOLO v1 was introduced in 2016 by Joseph Redmon et al with a research paper called "You Only Look Once: Unified, Real-Time Object Detection". This was the initial paper by Redmon that revolutionized the industry and changed the Real-Time Object detection methods totally. It can detect the objects with a speed of 45fps(frames per second), another

YOLO v1 type, Fast YOLOv1 was able to achieve 155fps with little less accuracy. It used the Darknet framework that was trained on the ImageNet-1000 dataset. But YOLOv1 has many limitations like it can't detect the objects properly when the objects are small and it also can't generalize the objects if the image is of different dimensions [43].

6.4.2.2-YOLOv2(YOLO9000)

The second version of YOLOv2 was released in 2017 by Ali Farhadi and Joseph Redmon. This time Joseph collaborated with Ali for major bug fixes and accuracy increment. The research they published was "YOLO9000: Better, Faster, Stronger." The name of the second version of YOLO was *YOLO9000*. The major competitor of YOLO9000 was Faster R-CNN, which was also an object detection algorithm that uses Region Proposal Network & (SSD)Single-shot Multbox Detector to identify the multiple objects from an image [43].

Some of the features of YOLOv2 are [43]:

- YOLOv2 added Batch Normalization as an improvement that normalizes the input layer of the image by altering the activation functions.
- Higher-resolution input: input size has been increased from 224*224 to 448*448.
- Anchor boxes.
- Multi-Scale training.
- Darknet 19 architecture with 19 convolution layers and 5 Max Pooling layers.

6.4.2.3-YOLOv3

After one year, on March 25, Joseph Redmon and Ali Farhadi came up with another version of YOLO and a research paper called: "YOLOv3: An Incremental improvement". YOLOv3 runs with 22ms at 28.2 mAP with great accuracy. It is three times faster than the previous SSD and four times faster than RetinaNet. YOLOv3 followed the methodology of the previous YOLOv2 version: YOLO9000. In this approach, Redmond uses Darknet 53 architecture, which was a significantly improved version and had 53 convolution layers [43].

Some of the new, improved features in YOLOv3 was:

- Class Predictions.
- Feature Pyramid Networks (FPN).
- Darknet 53 architecture.

6.4.2.4-YOLOv4

As Redmond was not currently working on the CV for a long time, a new team of three developers released YOLOv4. It was released by Alexey Bochoknovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Alexey is the one who developed the Windows version of YOLO back in the days [43].

Some of the new features of YOLOv4 is [43]:

- Anyone with a 1080 Ti or 2080 ti GPU can run the YOLOv4 model easily.
- YOLOv4 includes CBN (Cross-iteration batch normalization) and PAN (Pan aggregation network) methods.
- Weighted-Residual-Connections (WRC).
- Cross-Stage-Partial connections (CSP), a new backbone to enhance CNN (convolution neural network)
- Self-adversarial-training(SAT): A new data augmentation technique
- DropBlock regularization.

6.4.2.5-YOLOv5

A month after YOLOv4 was released, researcher Glenn and his team published a new version of the YOLO family, called YOLOv5 (Jocher, 2020). Glenn Jocher is a researcher and CEO of Ultralystics LLC. YOLO models were developed on a custom framework Darknet which is written mainly in C by Alexey Bochkovsky. Ultralystic is the company that converts previous versions of YOLO on one of the most famous frameworks in the field of deep learning, PyTorch which is written in the Python language [41]. Despite being released a month after YOLOv4, the start of research for YOLOv4 and YOLOv5 was quite close (March – April 2020). For avoiding collision, Glenn decided to name his version of YOLO, YOLOv5. Thus, basically, both researchers applied the state-of-the-art innovations in the field of computer vision at that time. That makes the architecture of YOLOv4 and YOLOv5 very similar and it makes many people dissatisfied with the name YOLOv5 (5th generation of YOLO) when it does not contain multiple outstanding improvements compared to the previous version YOLOv4. Besides, Glenn did not publish any paper for YOLOv5. However, YOLOv5 possessed the advantages in engineering. YOLOv5 is written in Python programming language instead of C as in previous versions. That makes installation and integration on IoT devices easier. In addition, the PyTorch community is also larger than the Darknet community, which means that PyTorch will receive more contributions and growth potential in the future. Due to being written in 2 different languages on 2 different frameworks, comparing the performance between YOLOv4 and YOLOv5 is difficult to be accurate. But after a while, YOLOv5 has proved higher performance than YOLOv4 under certain circumstances and partly gained confidence in the computer vision community besides YOLOv4 [41].

6.5-DEEP SORT

Deep SORT is an algorithm commonly used in Multi-Object Tracking, and it is a Detection Based Tracking method. This algorithm industry has a lot of attention.

6.5.1-The main steps of MOT

In "DEEP LEARNING IN VIDEO MULTI-OBJECT TRACKING: A SURVEY", a review of multi-object tracking based on deep learning, four main steps in the MOT problem are described:

The original frame of the given video.

Run target detectors such as Faster R-CNN, YOLOv3, SSD, etc. to detect and obtain target detection frames.

Cut out all the corresponding targets in the target frame and perform feature extraction (including apparent features or motion features).

Calculate the similarity to calculate the matching degree between the two frames before and after the target (the distance between the front and back belonging to the same target is relatively small, and the distance between different targets is relatively large)

Data association, assign target ID to each object.

The above are the four core steps, the core of which is detection. As mentioned in the abstract of the SORT paper, just changing to a better detector can improve target tracking performance by 18.9%.

6.5.2-SORT

The predecessor of Deep SORT algorithm is SORT, the full name is Simple Online and Realtime Tracking. To briefly introduce, the biggest feature of SORT is the target detection method based on Faster R-CNN, and the use of Kalman filter algorithm + Hungarian algorithm, which greatly improves the speed of multi-target tracking, while achieving the accuracy of SOTA.

This algorithm is indeed an algorithm that is widely used in practical applications. The core is two algorithms: Kalman filter with Hungary algorithm。

Kalman filter algorithm is divided into two processes, forecast and update. The algorithm defines the motion state of the target as 8 normally distributed vectors.

Prediction: When the target moves, the target frame and speed of the previous frame are used to predict the target frame position and speed of the current frame.

Update: The predicted value and the observed value, the two normal distribution states are linearly weighted to get the current state of the system prediction.

**Hungarian algorithm: **Solved is an allocation problem. In the main step of MOT, the similarity is calculated, and the similarity matrix of the two frames before and after is obtained. The Hungarian algorithm solves the real matching goal of the two frames before and after solving this similarity matrix. This part of the sklearn library has a corresponding function linear_assignment to solve.

SORT algorithm In the middle, the similarity matrix is constructed through two IOU frames before and after, so the SORT calculation speed is very fast.

The figure below is a flowchart of SORT core algorithm:

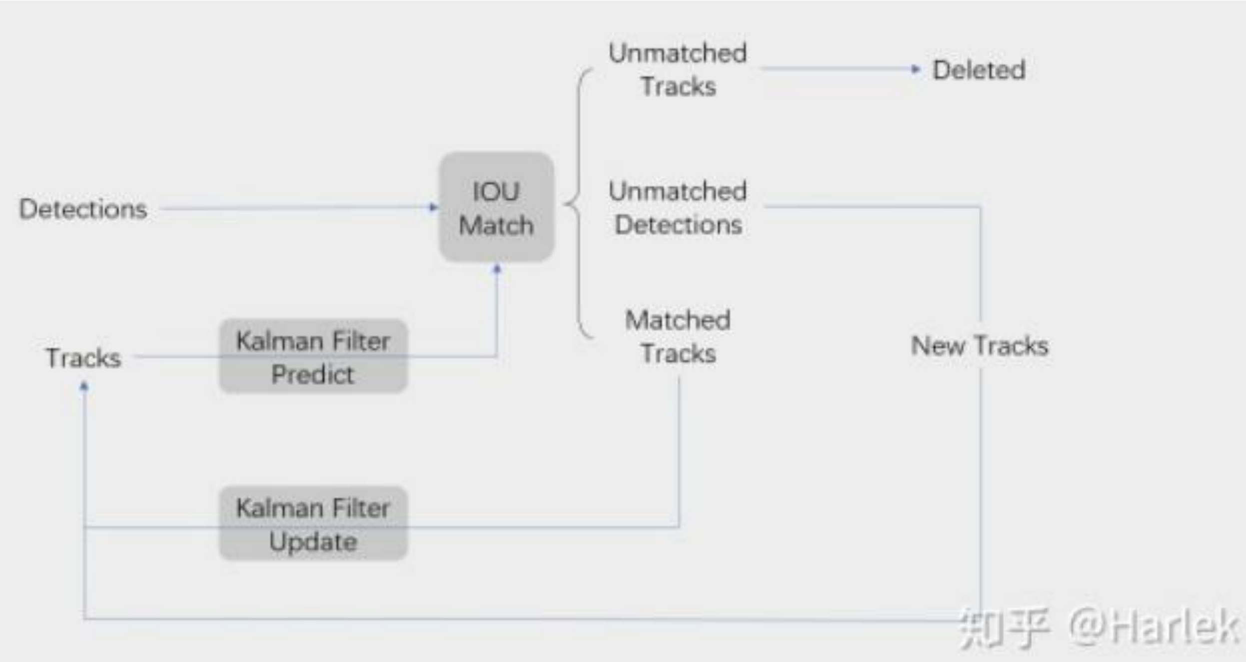


Fig 6.12 flowchart of SORT core algorithm

Detection is the target frame obtained by the target detector; Tracks It is a trajectory. The core is the matching process and the Kalman filter prediction and update process.

The process is as follows: The target detector obtains the target frame Detections, while the Kalman filter predicts the Tracks of the current frame, and then performs IOU matching on the Detections and Tracks. The result is divided into:

Unmatched Tracks, this part is considered a mismatch, Detection and Track cannot be matched, if the mismatch continues T_{lost} TlostSecond, the target ID will be deleted from the picture.

Unmatched Detections, this part shows that no track can match Detection, so a new track must be allocated for this detection.

Matched Track, this part shows that the match is obtained.

Kalman filter can be based on Tracks status prediction The target frame state of the next frame.

Kalman filterUpdateIt is to update the status of all tracks on the observed value (Track on the match) and estimated value.

6.5.3-Deep SORT

The biggest feature in DeepSort is to joinAppearance information, Borrowing the ReID domain model to extract features, reducing the number of ID switches. The overall flow chart is as follows:

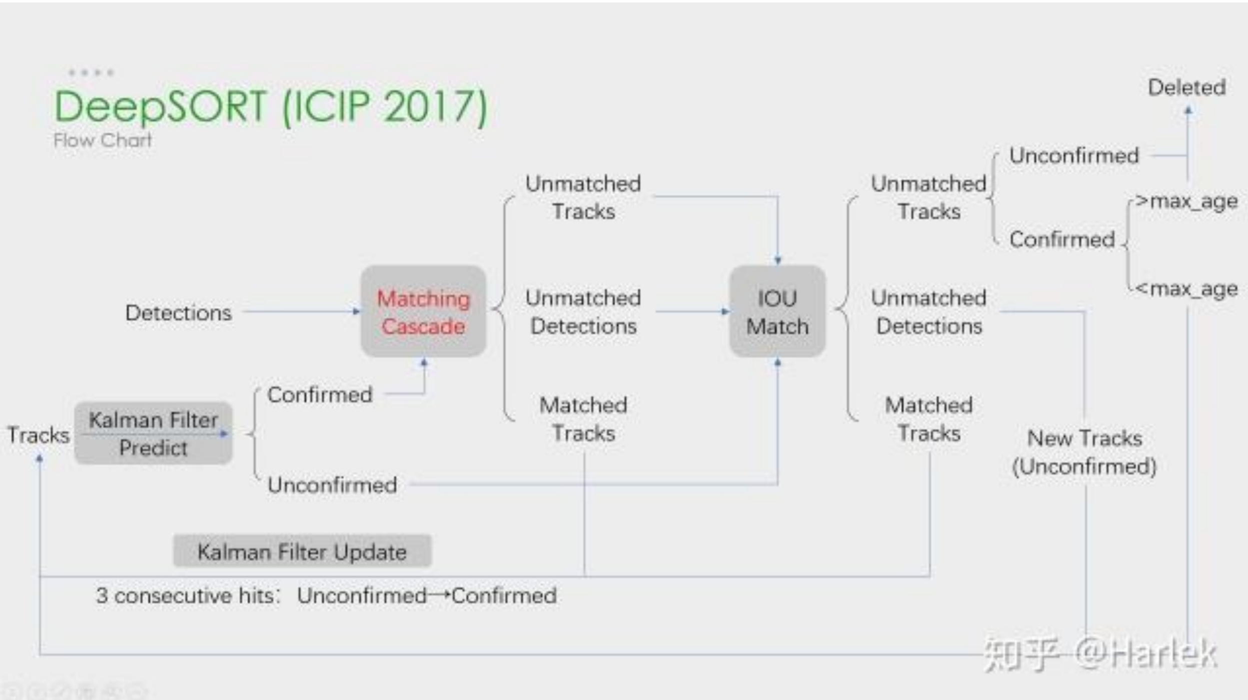


Fig 6.13 DeepSORT (ICIP2017)

It can be seen that the Deep SORT algorithm adds Matching Cascade + new trajectory confirmation (confirmed) to the SORT algorithm. The overall process is:

Kalman filter predicts trajectory Tracks

Use the Hungarian algorithm to match the predicted trajectory Tracks with the detections in the current frame (cascade matching and IOU matching)

Kalman filter update.

Match Detection and Track, so there are several situations

Detection and Track match, that is Matched Tracks. Ordinary continuous tracking targets belong to this situation. There are targets in the two frames before and after, which can be matched.

Detection did not find a matching track, that is Unmatched Detections. When a new target suddenly appears in the image, Detection cannot find a matching target in the previous track.

Track did not find a matching Detection, that is Unmatched Tracks. The continuously tracked target exceeds the image area, and Track cannot match any current Detection.

The above does not involve a special case, that is, the case where two targets are blocked. The track of the target that was just blocked cannot match the Detection, and the target temporarily disappears from the image. When the occluded target reappears later, you should try to keep the ID assigned by the occluded target unchanged and reduce the number of ID Switch appearances. This requires cascading matching.

6.5.4-On-screen results:

DeepSORT uses OpenCV 3 to draw 2D boxes around each tracked object in the 2D Space frame in which their 4 corners represent the edges of a car. It also distinguishes different objects using different colors in the RGB scale and for number of objects it gives an id number for every detection. It keeps incrementing the id for every object detected regardless of forgotten objects or objects that are detected a long time ago.

7-IMPLEMENTATION

Our project

We established our project based on Yolov5 (Darknet) and DeepSORT to detect vehicles and track them. Built in hosting language: Python. Method: Pytorch. Which gave us much customization and flexibility during our work.

7.1 Model and weights using KITTI dataset.

We did custom data learning using KITTI dataset [44] which consisted of labeled 7480 photos from random places that contains cars, trucks, pedestrians, cyclists, and more. But discovered that this is not the best idea.

First, the dataset labeling is not compatible with darknet which requires classes numerically in a txt file for each coordinate. But in KITTI, labels were in letters.

We have written a python script to replace every class name with an equivalent label number for the compatibility.

Another thing is photos in KITTI was sized to 1242 x 375 which was too wide for YOLOv5 input. We forced our model to learn wide photos, but it turned out to be not a good solution as we encountered errors during training phase and detection phase. We tried to narrow photos to be acceptable for the model, but this did not work greatly as it requires input to be narrowed and distorted in detection phase to get the most accuracy of it. It was trained in 20 epochs for batch size of 6 photos in one batch.

This large dataset took training time of 3 hours and 22 minutes with non-promising results on our i5 9300h, GTX 1650m, 16 GB of ram with 2666 MHz of speed. We did not expect those non-promising results to happen since the dataset is famous and very large and rich. Here is the generated confusion matrix from our training session.

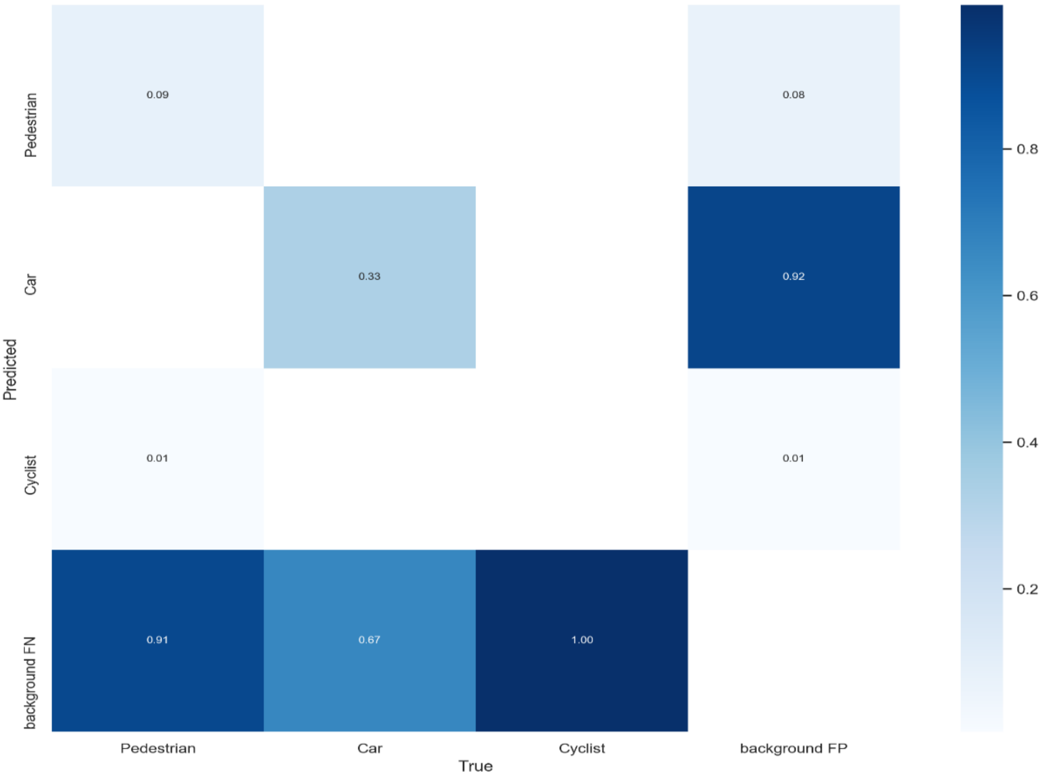


Fig 7.1

In our generated confusion matrix above, we see that there are a lot of false positives and negatives in car class due to problems we discussed previously.

Fig 7.2

We can see the false positives and negatives in the above screenshots from our evaluation session.

So, we decided to go for transfer learning. [45] For transfer learning we used a pre trained weights with the most perfect dataset from Microsoft which is the great MSCOCO.[46]

We downloaded a trained yolov5 medium model to use for detection.

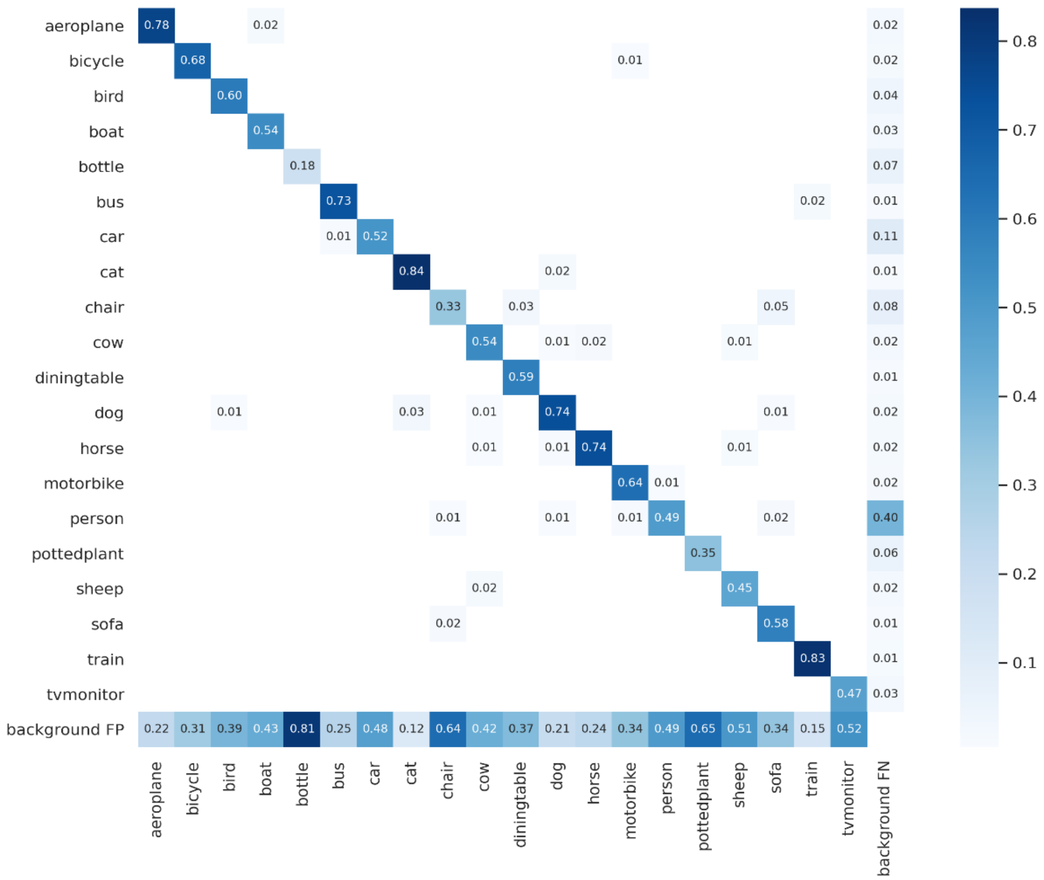


Fig 7.3

We see in above figure an official confusion matrix from “ultralyrics” the developer of YOLOv5 that cars are better detected using this dataset with better accuracy and now suitable for tracking.

We used random dashcam videos from YouTube for real evaluation as an input as we could not implement our proposed hardware system cameras due to challenges.

Fig 7.4

As we can see in the figure above (Fig 7.4) which is taken from our evaluation output there are still false negatives, but it is still much better accuracy than before and then we could elevate to the next level which is tracking.

7.2-Reverse direction violation

There is a statistic that states that the percentage of traffic accidents due to traffic in the reverse direction reaches to 87%. Most of these accidents lead to death. That is why we chose this violation to be one of the basics of this project. So, We use certain dataset to learn the machine how to catch the violator and It was in our ability to teach the machine so that it detects rear and front of the car by the camera where it's existing in front and back of the transportation public buses. We will show you every step what we did to learn and test the machine.

Frist: We choose dataset from Stanford the name of dataset is Cars Dataset and download it. Show in the figure 1.

Fig.7.5 The dataset

The Cars dataset contains 16,185 images of 196 classes of cars. The data is split into 8,144 training images and 8,041 testing images, where each class has been split roughly in a 50-50 split. Classes are typically at the level of Make, Model, Year, e.g., 2012 Tesla Model S or 2012 BMW M3 coupe.

The dataset of rear and front of the car is approximately 3700 images with 2 classes of cars.

This dataset is divided to training and testing. So, we did not need to divide it.

Second: Training the dataset through google COLAB, and we did 20 epochs only, and the results were in figure 2.

0/19	6.9G	0.04945	0.01414	0.01945	0.08304	14	640	0.7404	0.4176	0.3509	0.2008	0.02831	0.006706	0.01156
1/19	5.9G	0.03524	0.008822	0.007547	0.05161	13	640	0.4842	0.7307	0.4574	0.2564	0.02773	0.004932	0.007912
2/19	5.9G	0.03325	0.007732	0.006977	0.04795	14	640	0.3585	0.5038	0.331	0.1664	0.03257	0.005498	0.008464
3/19	5.9G	0.03191	0.008131	0.007546	0.04758	11	640	0.3899	0.6087	0.3634	0.2098	0.02949	0.005339	0.007155
4/19	5.9G	0.02877	0.007802	0.006321	0.0429	10	640	0.4614	0.7949	0.4673	0.3016	0.02559	0.004733	0.003523
5/19	5.9G	0.02676	0.007436	0.004966	0.03916	16	640	0.4765	0.8256	0.4727	0.2806	0.02381	0.004598	0.00364
6/19	5.9G	0.02519	0.007278	0.004338	0.03681	11	640	0.4939	0.8608	0.4954	0.3258	0.02204	0.004205	0.002048
7/19	5.9G	0.02407	0.007052	0.003954	0.03508	10	640	0.5027	0.9099	0.5142	0.3604	0.02064	0.004097	0.002215
8/19	5.9G	0.02318	0.00693	0.002909	0.03302	10	640	0.4933	0.9389	0.521	0.3656	0.01962	0.004054	0.001563
9/19	5.9G	0.02277	0.006778	0.002875	0.03243	9	640	0.5034	0.9181	0.5243	0.3657	0.01967	0.003964	0.001473
10/19	5.9G	0.02153	0.006691	0.002685	0.03091	11	640	0.5065	0.932	0.5301	0.3885	0.0182	0.003936	0.001192
11/19	5.9G	0.02127	0.006655	0.002406	0.03033	9	640	0.5004	0.9482	0.5272	0.3833	0.01873	0.003814	0.001318
12/19	5.9G	0.02068	0.006582	0.001955	0.02922	11	640	0.5028	0.9513	0.5314	0.3856	0.01798	0.00387	0.001291
13/19	5.9G	0.02009	0.006557	0.001958	0.0286	18	640	0.5025	0.9601	0.5295	0.39	0.01746	0.003783	0.0008312
14/19	5.9G	0.01966	0.006479	0.001847	0.02798	12	640	0.5064	0.9447	0.5306	0.3981	0.01676	0.003739	0.0007837
15/19	5.9G	0.01932	0.006436	0.001902	0.02766	19	640	0.5114	0.9361	0.5262	0.3933	0.01674	0.00378	0.000866
16/19	5.9G	0.019	0.00649	0.001813	0.02731	11	640	0.5029	0.9529	0.532	0.4026	0.01634	0.003724	0.000676
17/19	5.9G	0.01876	0.006367	0.001507	0.02664	19	640	0.514	0.9428	0.5393	0.409	0.01624	0.003742	0.0006641
18/19	5.9G	0.01814	0.006399	0.00171	0.02625	15	640	0.5096	0.9508	0.5432	0.4175	0.01605	0.00366	0.0007168
19/19	5.9G	0.01804	0.006267	0.001215	0.02553	12	640	0.5103	0.9601	0.5435	0.4174	0.01595	0.003654	0.0006064

Fig 7.6 Number of epochs

Third**:** Testing data, after the model trained, we are testing the model by the dataset of test.

Let show The Confusion Matrix of the experiments in figure 7.7.

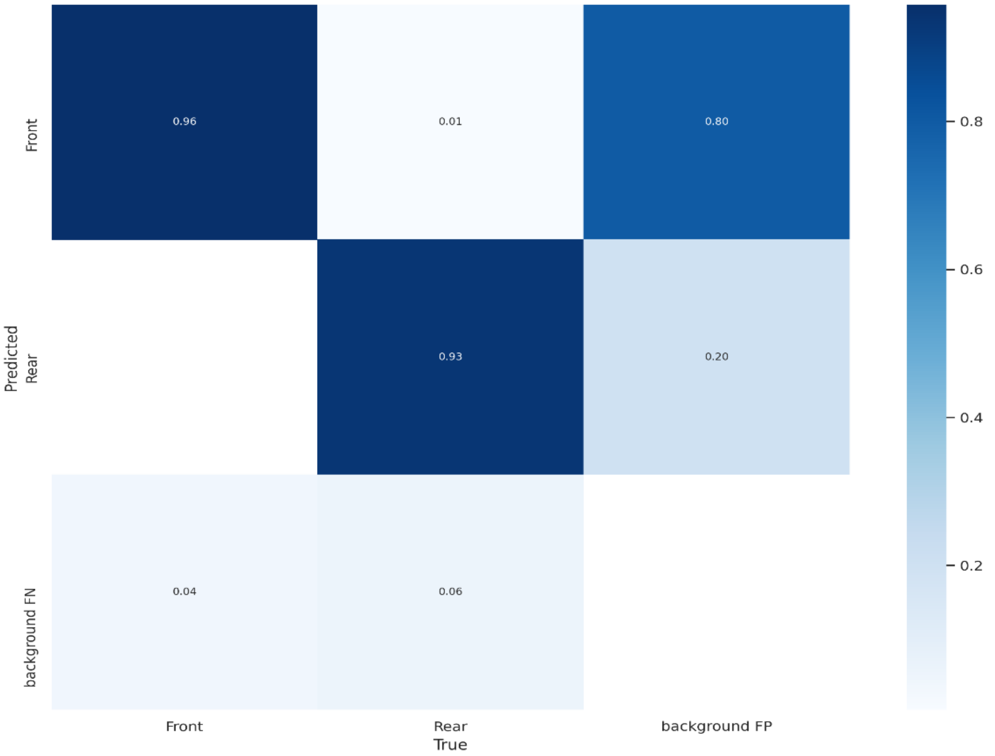


Fig 7.7 Confusion Matrix

- Our proposed model detects 96% of data as a car front which it is actual car's front.
- Our proposed model detects 93% of data as a car rear which it is actual car's rear.

Let us see the images that show detected in the figure 7.8, 7.9, and 7.10.

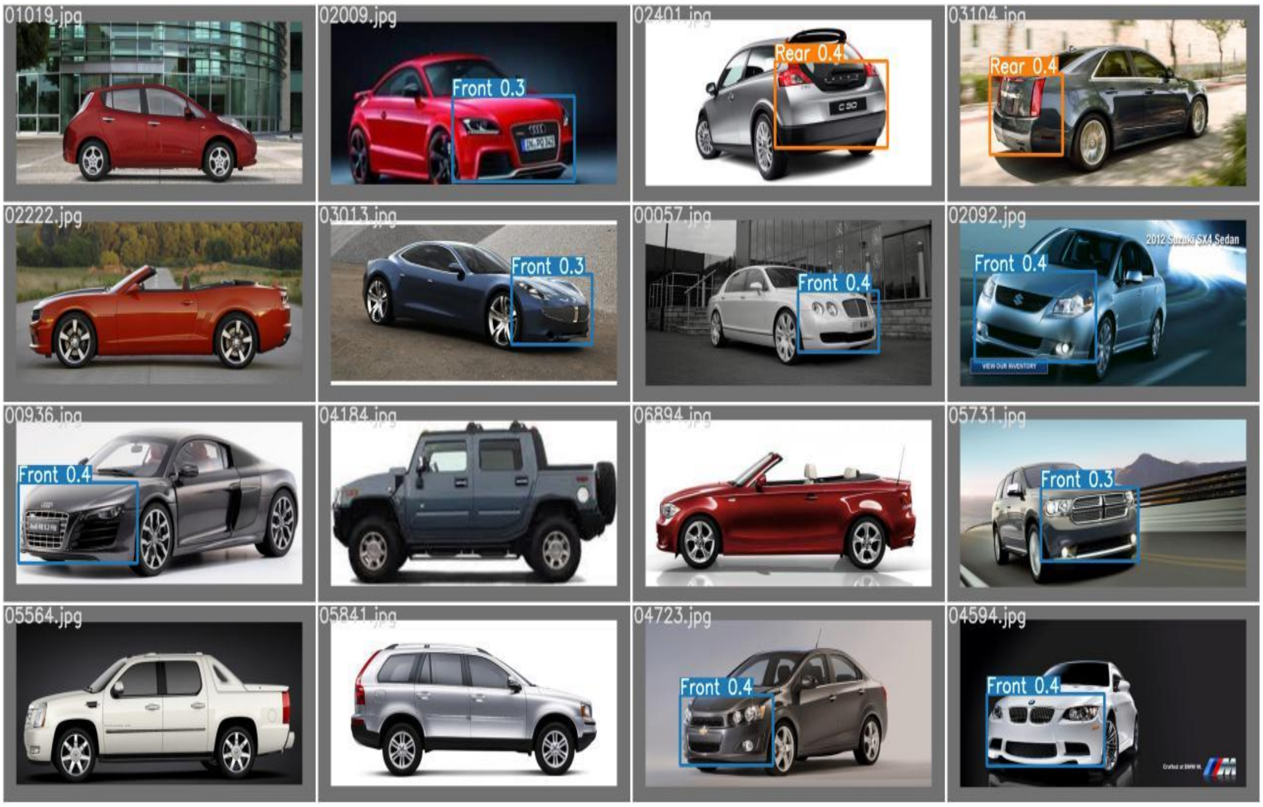


Fig 7.8

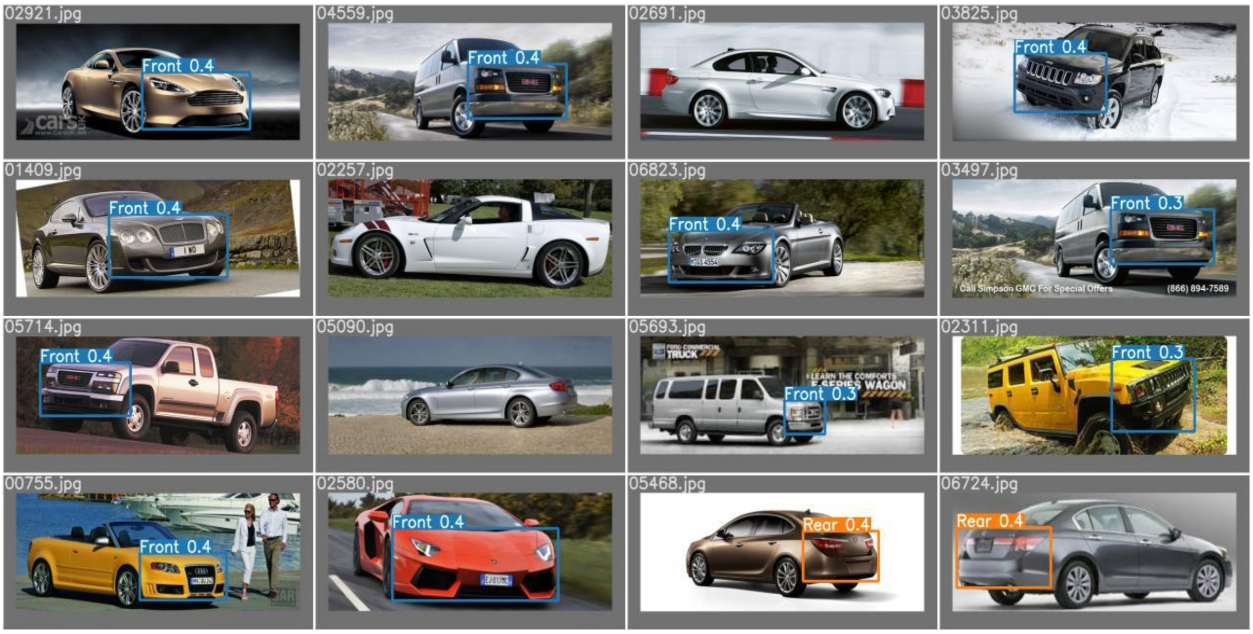


Fig 7.9

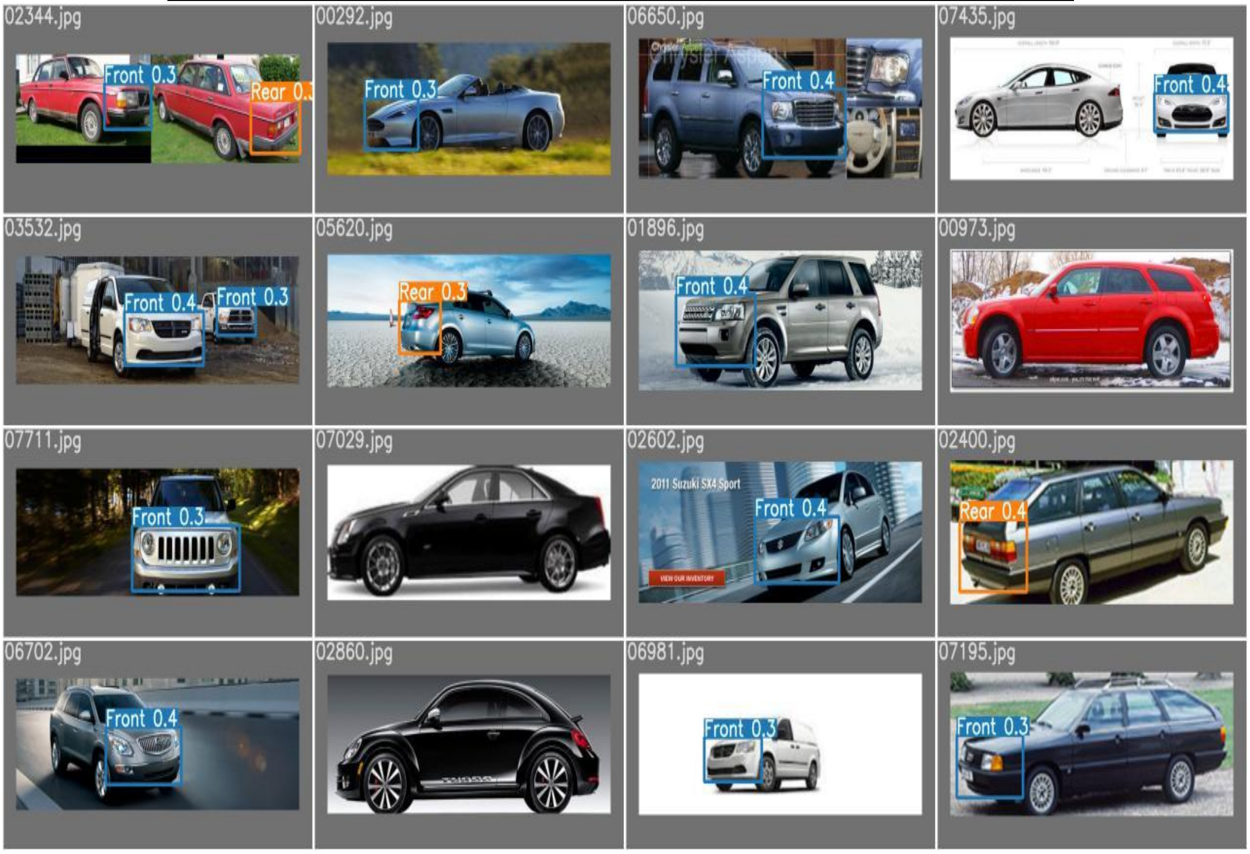


Fig 7.10

After we trained the model in this dataset and now model is to be able to determine the rear and the front of cars. We can record the violation on the driver who come in reverse direction. That it can be done by our system if the camera that is mounted on the front of the the transportation public buses detects the car come from the front (this means the car is going in reverse), the system detect violation and add it to the driver.

7.3-speed limit violation

We used OpenCV 3 to draw 2D boxes around each tracked object in the 2D Space frame in which their 4 corners represent the edges of a car. We also distinguish different cars using different colors in the RGB scale. We acquired the most accurate results using the parameters below:

MAX_DIST: 0.2

MIN_CONFIDENCE: 0.4

NMS_MAX_OVERLAP: 0.9

MAX_IOU_DISTANCE: 0.7

MAX_AGE: 70

N_INIT: 3

NN_BUDGET: 100

Where MAX_DIST: represent the max distance from an object from a frame to its previous version in the past frame.

MAX_AGE: represents the maximum seconds before forgetting the tracked object.

NMS_MAX_OVERLAP: this is responsible for keeping the only box that around the part that is visible when it is overlapped by another object.

In the figure below (Fig 7.11) we can see how accurate our model is with the specific parameters we used.

Fig 7.11 accuracy in our model

There are many ways to estimate relative speed of the moving objects in a 2D space frame. We depended on estimating variance of drawn box's size in X and Y to number of frames. To keep it simple, if the object is speeding and keeps distancing away from our mounted camera the box should get smaller in size in XY Plane. This is the main concept of our speed estimation system.

By storing the size of the box for every frame for 10 frames. Then subtracting final frame box's size from the most first frame in the range of 10 frames. The outcome value will deduce whether the object is moving away or moving closer.

After that we combined the result with the result of Euclidean distance between object and its history place through the frames. The result is promising for estimating the relative speed of the moving object.

Our speed estimation formula would look like this:

$S_1=Y_2-Y_1$

$S_2=Y_2-Y_1$

$S_3=Y_2-Y_1$

$S_n=Y_2-Y_1$

Where n is number of frames.
S is the size of the box.
Y_2 Represents Y value of the right lower corner coordinates of the box.
Y_1 Represents Y value of the left higher corner coordinates of the box.

We then determine the variance in box's size by subtracting box's size in n^{th} frame from 1^{st} frame.

Variance in box's size = $S_n - S_1$

We apply a factor that depends on the size of screen and pixel distribution and that varies depending on camera resolution and focal length to make sure the distance traveled on screen is equal to distance traveled in real life. Same way we take Euclidean distance between object in n^{th} and 1^{st} frame.

This is how the speed estimating formula looks:

V= Initial speed - Variance of box's size x reality factor - Euclidean distance x reality factor

Estimating initial speed using SIM5320E GPS 3G module connected to our system to estimate current speed and then sends it to the system as an input.

There are many disadvantages of using Euclidean distance **only** during estimation of speed. The major disadvantage was the stability of mounted camera. Speed would be affected any vibrations or road issues as it estimates the speed of movement of an object through XY plane no matter where the direction of movement was. To shorten, any speed bump or road issues in our lovely Egypt would affect speed estimation as the object is still tracked through the frames.

Now we have predicted speed of a moving object using a system that is immune to camera sudden moves and more accurate than expected.

Fig 7.12

7.4-Anomaly detection and collision prediction.

Next level is trying to estimate the distance between the camera and cars in every frame. Our method this time, determine the size of the box in pixels. As long as the car is very far away the box will look very small compared to a car which is so close to the camera its box shall be very big sized. We faced a little problem which the very far away objects may not be very accurate as the big variation happens in box size only happens when car is traveling from right front of the camera to further area, but further objects always look very close in box sizes. We fixed this by powering the pixel values to a positive reality factor to make sure it is a polynomial function. What does that mean? It means that as the box gets smaller in size the distance meter increments faster. By adjusting our reality factor, it gave accurate results.

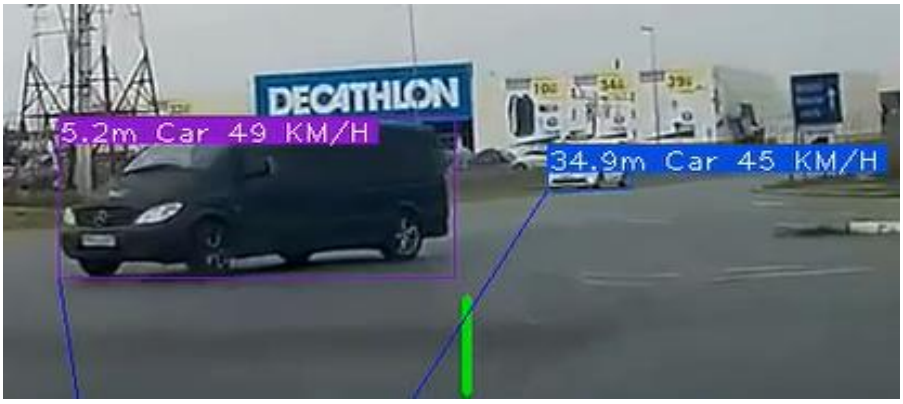


Fig 7.13

Obviously in the previous figure our model shows that there are 2 cars which are far away from each other correctly and with satisfying accuracy.

Our distance estimation formula looks like this:

Distance = (1 / (box’s size)) ^ (reality factor)

How could we use this to predict a collision?

Simply, if there is a collision that is going to happen to the driver, The camera should see a speeding vehicle towards it and very quick decrease in distance meter. That is how we managed to solve the challenge. See those screenshots from the output film from our evaluation and test.

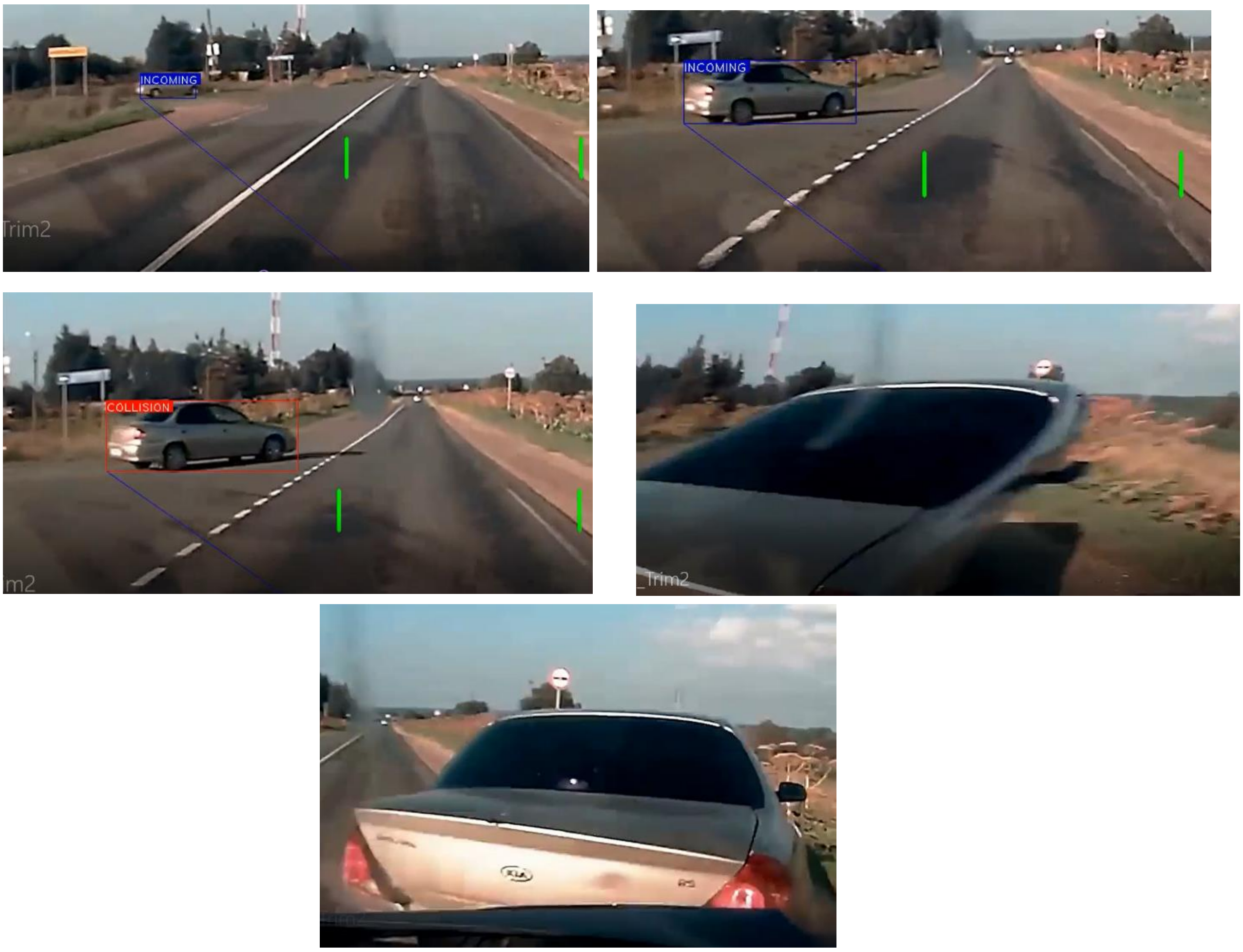


Fig 7.14

We can see that our system succeeded in informing and alerting our driver that there is a collision that is going to happen in enough time to act. We feel sorry for the driver. We wish this driver really had our system that could save him.

Fig 7.15

The above figure was another sample of situations that our model succeeded in. This sample was a raw video from the internet for random accidents.

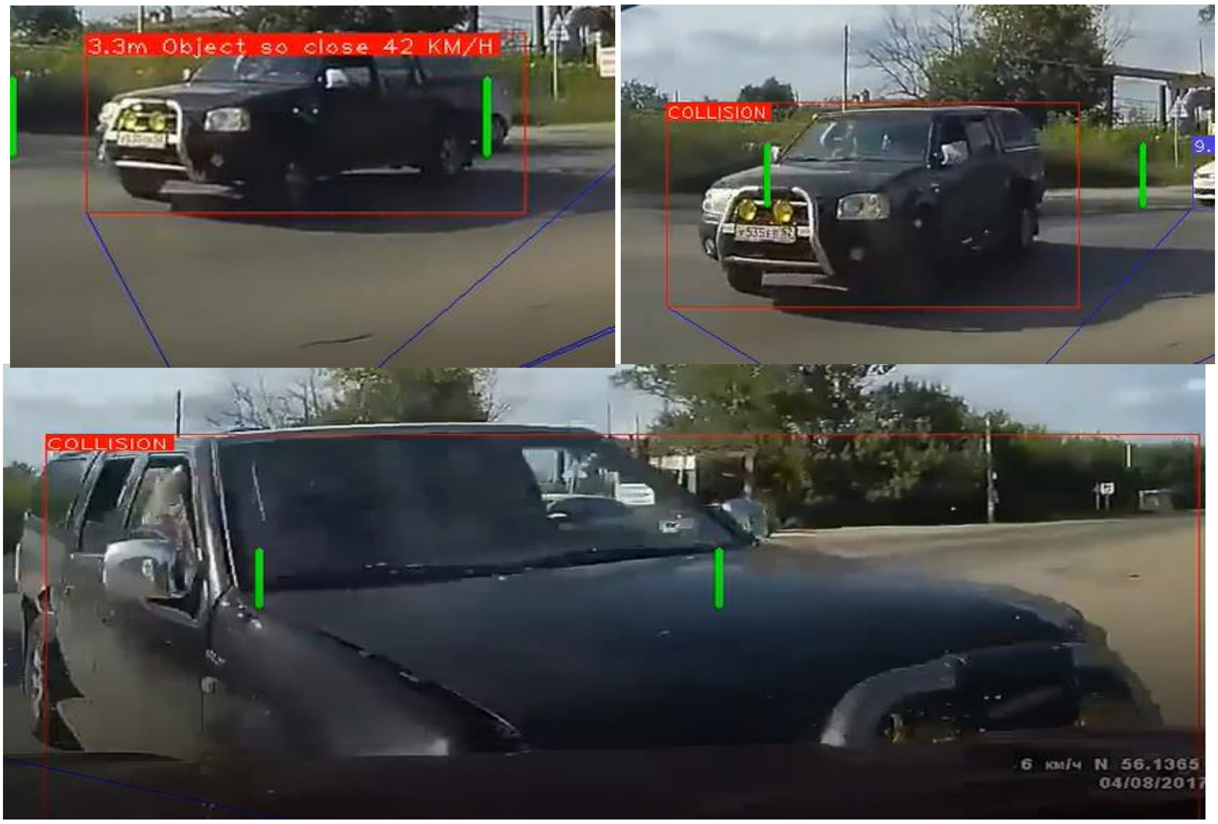


Fig 7.16

Also this figure 7.16 was created by our model which its input was a raw random accidents video from YouTube.

7.5-Licence Plate

A vehicle registration plate, also known as a number plate (British English), license plate (American English), or licence plate (Canadian English), is a metal or plastic plate attached to a motor vehicle or trailer for official identification purposes. All countries require registration plates for road vehicles such as cars, trucks, and motorcycles. The registration identifier is a numeric or alphanumeric ID that uniquely identifies the vehicle or vehicle owner within the issuing region's vehicle register. In some countries, the identifier is unique within the entire country, while in others it is unique within a state or province. Whether the identifier is associated with a vehicle, or a person also varies by issuing agency. There are also electronic license plates.

Fig7.17

In this project we use license plate to identify violation car and sending information about the violation and the owner of the vehicle to General Directorate of Traffic then Imposing a fine on the owner of the vehicle, this fine is estimated according to the type of violation committed by the driver of the vehicle.

We use certain dataset to learn the machine how to catch the violator and It was in our ability to teach the machine so that it detects only the numbers or letters found on the license plate, this system will be more accurate, but this system will be limited to use in a specific country or city. We will show you every step what we did to learn and test the machine.

Firstly: We choose dataset from Kaggle and download it, It turns out that the label of dataset was a xml file. So, we parsing it from xml to txt file in the next step.

```
# convert label from xml to txt
# -*- coding: utf-8 -*-
%cd /content/gdrive/MyDrive/project/licence-plate/licence_plate_data/
import xml.etree.ElementTree as ET
import os
from os import listdir
from os.path import join

classes = ["licence"] # own data sets which classes which
# category to write, in the order

def convert(size, box):
    dw = 1./(size[0])
    dh = 1./(size[1])
    x = (box[0] + box[1])/2.0 - 1
    y = (box[2] + box[3])/2.0 - 1
    w = box[1] - box[0]
    h = box[3] - box[2]
    x = x*dw
    w = w*dw
    y = y*dh
    h = h*dh
    return (x,y,w,h)

def convert_annotation(image_id):
    in_file = open('annotations/%s.xml'%(image_id), encoding = 'utf-8')
    out_file = open('labels/%s.txt'%(image_id), 'w')
    tree=ET.parse(in_file)
    root = tree.getroot()
    size = root.find('size')
    w = int(size.find('width').text)
    h = int(size.find('height').text)

    for obj in root.iter('object'):
        difficult = obj.find('difficult').text
        cls = obj.find('name').text
        if cls not in classes or int(difficult)==1:
            continue
        cls_id = classes.index(cls)
        xmlbox = obj.find('bndbox')
        b = (float(xmlbox.find('xmin').text), float(xmlbox.find('xmax').text),
             float(xmlbox.find('ymin').text), float(xmlbox.find('ymax').text))
        bb = convert((w,h), b)
        out_file.write(str(cls_id) + " " + " ".join([str(a) for a in bb]) + '\n')

val_percent = 0 # test set proportion of the total data set, the default 0.1, if
                # the test set and the training set have been demarcated, the
                # corresponding code is modified
data_path = '/content/gdrive/MyDrive/project/licence-plate/licence_plate_data/'
                # darknet relative path folder, see description github, and they
                # need to modify, according to note here the absolute path can
                # also be used
if not os.path.exists('labels/'):
    os.makedirs('labels/')
image_ids = [f for f in os.listdir (data_path + 'annotations')]
            # XML data storage folder
train_file = open('train.txt', 'w')
val_file = open('val.txt', 'w')
for i, image_id in enumerate(image_ids):
    if image_id [-3:] == "xml":
      # Sometimes jpg and xml files are placed in the same folder,
      # so to determine what suffixes
        if i < (len(image_ids) * val_percent):
            val_file.write(data_path + '%s\n'%(image_id[:-3] + 'png'))
        else:
            train_file.write(data_path + '%s\n'%(image_id[:-3] + 'png'))
```

Fig 7.18 Convert from xml to txt

Also this photo (fig 7.19) from inside xml file and txt file

```xml
<annotation>
    <folder>images</folder>
    <filename>Cars0.png</filename>
    <size>
        <width>500</width>
        <height>268</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>licence</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <occluded>0</occluded>
        <difficult>0</difficult>
        <bndbox>
            <xmin>226</xmin>
            <ymin>125</ymin>
            <xmax>419</xmax>
            <ymax>173</ymax>
        </bndbox>
    </object>
</annotation>
```

```
0 0.643 0.5522388059701493 0.386 0.1791044776119403
```

Fig 7.19

Secondly: Before training the dataset, we notice that dataset is not divided as training set and validation set. So, we divided it by python script in the figure 7.20.

```python
# Get all paths to your images files and text files
PATH = '/content/gdrive/MyDrive/project/licence-plate/licence_plate_data/'
img_paths = glob.glob(PATH+'new_data/images/*.png')
txt_paths = glob.glob(PATH+'labels/*.txt')

# Calculate number of files for training, validation
data_size = len(img_paths)
r = 0.8
train_size = int(data_size * 0.8)

# Shuffle two list
img_txt = list(zip(img_paths, txt_paths))
print(img_txt)
random.seed(43)
random.shuffle(img_txt)
img_paths, txt_paths = zip(*img_txt)

# Now split them
train_img_paths = img_paths[:train_size]
train_txt_paths = txt_paths[:train_size]

valid_img_paths = img_paths[train_size:]
valid_txt_paths = txt_paths[train_size:]

# Move them to train, valid folders
train_folder_img = PATH+'train/images/'
train_folder_lab = PATH+'train/labels/'
valid_folder_img = PATH+'valid/images/'
valid_folder_lab = PATH+'valid/labels/'
```

Fig 7.20

So, we split the dataset 80% as a training and 20% as a validation set.

Thirdly: Training the dataset through google colab, and we did 15 epoch only, and the results were in figure 7.10.

0/14	6.19G	0.1106	0.01965	0	0.1302	20	640	0.001313	0.03922	0.0001704	2.166e-05	0.1055	0.01574	0
1/14	8.9G	0.09666	0.01757	0	0.1142	12	640	0.0009782	0.0817	0.0001681	2.441e-05	0.09061	0.01424	0
2/14	8.9G	0.08545	0.01663	0	0.1021	12	640	0.002876	0.006536	0.0001561	2.444e-05	0.08201	0.01326	0
3/14	8.9G	0.07962	0.01594	0	0.09556	18	640	0.001088	0.006536	0.0001386	1.813e-05	0.07314	0.0126	0
4/14	8.9G	0.07122	0.01587	0	0.08709	16	640	0.01048	0.006536	0.0009027	0.0002143	0.06583	0.01199	0
5/14	8.9G	0.06681	0.01539	0	0.0822	9	640	0.01374	0.0817	0.005083	0.001472	0.06189	0.01133	0
6/14	8.9G	0.06063	0.01572	0	0.07635	15	640	0.3665	0.2288	0.1801	0.06093	0.05507	0.0108	0
7/14	8.9G	0.05751	0.01437	0	0.07189	17	640	0.5743	0.3235	0.2718	0.09395	0.05215	0.01016	0
8/14	8.9G	0.05463	0.01388	0	0.06851	11	640	0.5003	0.3922	0.3524	0.1474	0.05028	0.009774	0
9/14	8.9G	0.04573	0.0131	0	0.05882	16	640	0.6499	0.4118	0.4107	0.1794	0.04687	0.00934	0
10/14	8.9G	0.05017	0.01305	0	0.06322	14	640	0.6026	0.4955	0.4761	0.2061	0.04185	0.009074	0
11/14	8.9G	0.04492	0.01282	0	0.05774	19	640	0.6857	0.4804	0.5154	0.247	0.0388	0.008688	0
12/14	8.9G	0.04127	0.01217	0	0.05344	21	640	0.5914	0.5915	0.5645	0.2432	0.03749	0.008257	0
13/14	8.9G	0.03999	0.01215	0	0.05214	16	640	0.5967	0.6236	0.5991	0.2655	0.04037	0.007962	0
14/14	8.9G	0.04237	0.01198	0	0.05434	23	640	0.5484	0.6503	0.584	0.2734	0.03518	0.007635	0

Fig 7.21

But these results is unconvincing, So after some trying to get best result in train, we found the best number of epoch is 50 and the result in figure 7.22.

0/49	6.19G	0.1106	0.01965	0	0.1302	20	640	0.001312	0.03922	0.0001658	2.145e-05	0.1055	0.01574	0
1/49	8.9G	0.09679	0.01757	0	0.1144	12	640	0.0009932	0.08497	0.0001655	2.773e-05	0.0913	0.01424	0
2/49	8.9G	0.086	0.01663	0	0.1026	12	640	0.0008061	0.006536	0.0001121	2.053e-05	0.08022	0.01328	0
3/49	8.9G	0.07915	0.01593	0	0.09508	18	640	0.004388	0.006536	0.0002793	4.784e-05	0.07185	0.0125	0
4/49	8.9G	0.07083	0.01579	0	0.08662	16	640	0.002893	0.03268	0.001369	0.0003172	0.0636	0.01183	0
5/49	8.9G	0.06423	0.01526	0	0.07949	9	640	0.04445	0.08497	0.01273	0.004602	0.06141	0.01105	0
6/49	8.9G	0.0588	0.01542	0	0.07422	15	640	0.5022	0.2288	0.2265	0.09178	0.05651	0.01038	0
7/49	8.9G	0.05654	0.01371	0	0.07024	17	640	0.1569	0.1111	0.06145	0.01081	0.0581	0.00929	0
8/49	8.9G	0.05448	0.01339	0	0.06787	11	640	0.603	0.2451	0.2862	0.1258	0.0498	0.009429	0
9/49	8.9G	0.04853	0.01248	0	0.06101	16	640	0.4083	0.2778	0.2842	0.1122	0.05035	0.009002	0
10/49	8.9G	0.05158	0.01273	0	0.06431	14	640	0.6008	0.4869	0.4649	0.178	0.04219	0.008609	0
11/49	8.9G	0.04704	0.01225	0	0.05929	19	640	0.4905	0.5327	0.4497	0.1168	0.04523	0.007897	0
12/49	8.9G	0.04254	0.01176	0	0.0543	21	640	0.6237	0.6438	0.6073	0.2822	0.04437	0.00755	0
13/49	8.9G	0.04414	0.01154	0	0.05568	16	640	0.6013	0.6797	0.6436	0.2768	0.04613	0.007077	0
14/49	8.9G	0.04738	0.01116	0	0.05854	23	640	0.6703	0.7451	0.6844	0.3364	0.03835	0.006999	0
15/49	8.9G	0.04047	0.01035	0	0.05082	11	640	0.5364	0.6928	0.5163	0.1418	0.04115	0.006557	0
16/49	8.9G	0.04259	0.01034	0	0.05294	29	640	0.6569	0.8333	0.7088	0.3165	0.03729	0.006494	0
17/49	8.9G	0.03794	0.009603	0	0.04754	12	640	0.6797	0.8399	0.7446	0.3544	0.03506	0.00621	0
18/49	8.9G	0.04018	0.009644	0	0.04982	23	640	0.6964	0.8399	0.7255	0.3318	0.03611	0.005787	0
19/49	8.9G	0.03901	0.008707	0	0.04772	20	640	0.7128	0.8678	0.7464	0.4033	0.03922	0.005691	0
20/49	8.9G	0.03981	0.008284	0	0.0481	12	640	0.7007	0.8039	0.7222	0.3513	0.03787	0.005453	0
21/49	8.9G	0.0394	0.008466	0	0.04787	19	640	0.7208	0.8529	0.7482	0.3055	0.03742	0.00501	0
22/49	8.9G	0.0376	0.007988	0	0.04559	16	640	0.4997	0.6503	0.4875	0.09035	0.04436	0.00466	0
23/49	8.9G	0.0412	0.008571	0	0.04977	16	640	0.6905	0.8627	0.7499	0.3931	0.03643	0.005167	0
24/49	8.9G	0.04027	0.007782	0	0.04806	20	640	0.6453	0.856	0.7245	0.2614	0.0394	0.005167	0
25/49	8.9G	0.03946	0.007761	0	0.04722	17	640	0.6794	0.8529	0.7181	0.2542	0.03807	0.00475	0
26/49	8.9G	0.03977	0.007334	0	0.0471	22	640	0.7585	0.8627	0.7925	0.4374	0.03317	0.004766	0
27/49	8.9G	0.03545	0.007152	0	0.0426	23	640	0.734	0.9118	0.7762	0.4305	0.03251	0.0049	0
28/49	8.9G	0.03578	0.00699	0	0.04277	20	640	0.7581	0.8301	0.7593	0.3789	0.03664	0.004652	0
29/49	8.9G	0.03586	0.007036	0	0.0429	16	640	0.786	0.8529	0.8031	0.4012	0.03187	0.004525	0
30/49	8.9G	0.03759	0.007042	0	0.04463	20	640	0.7533	0.8784	0.766	0.437	0.03048	0.004448	0
31/49	8.9G	0.03428	0.007211	0	0.0415	13	640	0.74	0.9115	0.7917	0.414	0.03221	0.004258	0
32/49	8.9G	0.03247	0.006684	0	0.03915	16	640	0.7382	0.9314	0.802	0.472	0.02941	0.004395	0
33/49	8.9G	0.03233	0.006838	0	0.03917	28	640	0.7486	0.8953	0.7854	0.4568	0.02988	0.004207	0
34/49	8.9G	0.03211	0.006659	0	0.03877	14	640	0.7607	0.9248	0.8047	0.4775	0.02931	0.004166	0
35/49	8.9G	0.0308	0.006363	0	0.03716	14	640	0.7365	0.9641	0.7995	0.5009	0.02669	0.004124	0
36/49	8.9G	0.03011	0.006331	0	0.03644	17	640	0.748	0.9706	0.8155	0.4957	0.02633	0.004066	0
37/49	8.9G	0.02731	0.006042	0	0.03335	23	640	0.7504	0.9739	0.8125	0.4742	0.02815	0.003941	0
38/49	8.9G	0.02974	0.006282	0	0.03602	16	640	0.7465	0.9739	0.8208	0.4955	0.02758	0.00405	0
39/49	8.9G	0.03016	0.006616	0	0.03678	21	640	0.7659	0.9412	0.8209	0.4376	0.02797	0.003909	0
40/49	8.9G	0.0305	0.006307	0	0.0368	25	640	0.7441	0.9215	0.8025	0.4355	0.02867	0.003917	0
41/49	8.9G	0.02975	0.006348	0	0.0361	21	640	0.7597	0.951	0.8284	0.5214	0.02577	0.003904	0
42/49	8.9G	0.02917	0.00583	0	0.035	11	640	0.7522	0.9837	0.8396	0.5217	0.02721	0.003962	0
43/49	8.9G	0.02816	0.006431	0	0.03459	16	640	0.7687	0.9341	0.8294	0.4947	0.02595	0.003756	0
44/49	8.9G	0.02706	0.005684	0	0.03274	21	640	0.7601	0.9837	0.8372	0.5679	0.02536	0.003704	0
45/49	8.9G	0.02578	0.005768	0	0.03155	25	640	0.7575	0.9802	0.8319	0.5492	0.02367	0.003672	0
46/49	8.9G	0.02576	0.0061	0	0.03186	14	640	0.7627	0.9769	0.8515	0.5737	0.02352	0.003597	0
47/49	8.9G	0.02512	0.005525	0	0.03065	12	640	0.7639	0.9739	0.8547	0.5904	0.02436	0.003578	0
48/49	8.9G	0.02629	0.005677	0	0.03197	21	640	0.7712	0.9608	0.8646	0.5631	0.02446	0.003569	0
49/49	8.9G	0.02742	0.00596	0	0.03338	17	640	0.7853	0.9442	0.8637	0.5885	0.02256	0.003622	0

Fig 7.22

This is Confusion Matrix for first experiment and the final one in testing.

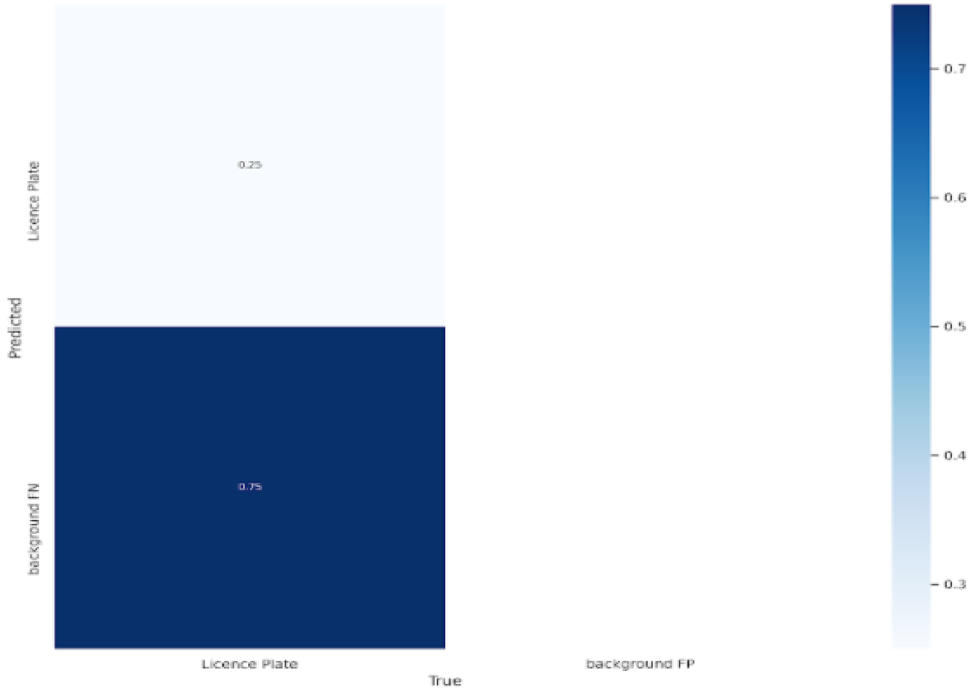


Fig 7.23 confusion matrix for first experiment

Fig 7.24 first experiment result Image

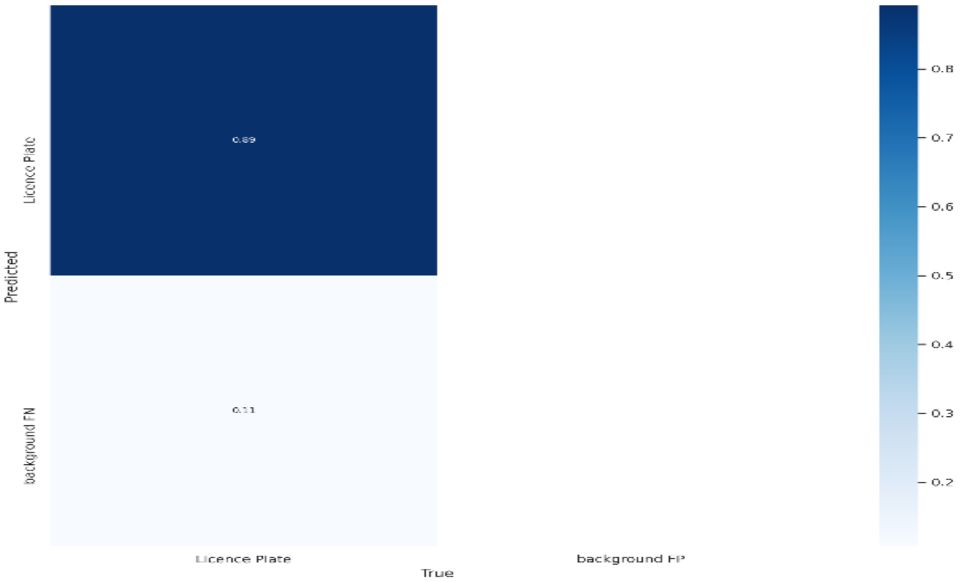


Fig 7.25 confusion matrix for final experiment

As we see from these figures 7.23 the result in first experiment is unconvincing. On the other as we see from figures 7.25 side the result is convincing.

Finally: this is some pictures from dataset after trained (tested data) in figures below.

Fig 7.26 Tested Image

8-CONCLUSION AND FUTUER WORK

8.1 Conclusion

In this project, we proposed a traffic violation detection system based on Convolutional Neural Network (CNN) method to detect traffic rule violations with more accuracy than other similar system because using CCTV cameras fixed on the public transport vehicles which allows us to cover places that are difficult for fixed cameras which makes it easy to implement this project. We use an object detection algorithm called YOLOv5 which is the state-of-the-art model. We apply that model on three use case speed limit violation, reverse direction violation and collision prediction , the proposed model detect violation in Real-Time so that we also using our proposed system to detect license plate to identifying violators automatically and uploads their details through 3G/4G-LTE module to General Directorate of Traffic servers to take a decision .

8.2 Future work:

With the increasing growth in traffic density all over the world, it possesses a great challenge to traffic management.in our system we start to automate the traffic rule violation detection system and make it easy for the traffic police department to monitor the traffic and take action against the violated vehicle by deducting the fee of the violation from his account automatically and send massage to them (figure 8.1).Our system can be improved in many different directions as we see in figure 8.2 . It can be improved to detect many different traffic violation types like a seatbelt, unsafe driving behavior,

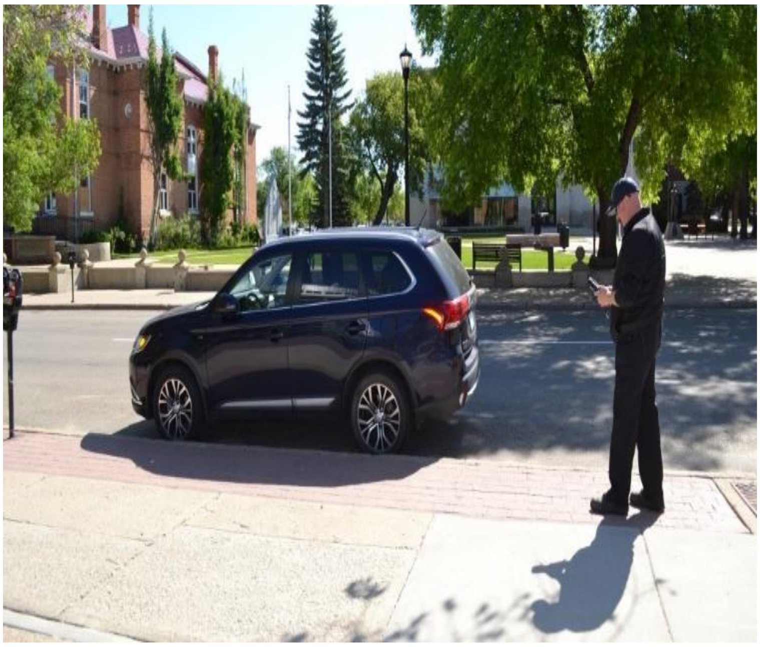


Figure 8.1

intersection violations, illegal parking and other type of traffic rule violation. Thanks to the unfixed cameras, our system can also improve in different majors like recognize a suspect vehicle and monitor it, also can be improved to detect accident and send signal to emergency to help save the injured as soon as possible.

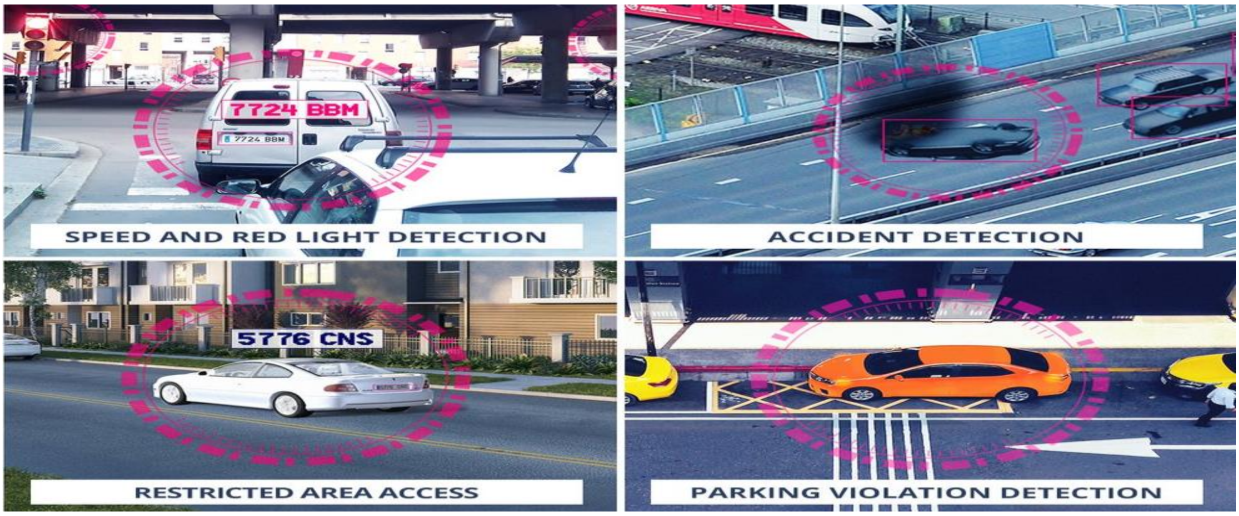


Figure 8.2

9-References

1) Bini SA. Artificial intelligence, machine learning, deep learning, and cognitive computing: what do these terms mean and how will they impact health care? J Arthroplast. 2018;33(8):2358–61. https://doi.org/10.1016/j.arth.2018.02.067
2) Maxmen JS. The post-physician era : medicine in the twenty-first century. Hoboken: Wiley; 1976
3) Topol EJ. High-performance medicine: the convergence of human and artificial intelligence. Nat Med. https://doi.org/10.1038/s41591-018-0300-7
4) Haeberle HS, Helm JM, Navarro SM, et al. Artificial intelligence and machine learning in lower extremity arthroplasty: a review. J Arthroplast. 2019. https://doi.org/10.1016/j.arth.2019.05.055
5) Definitions, methods, and applications in interpretable machine learning https://doi.org/10.1073/pnas.1900654116
6) https://www.ibm.com/cloud/learn/supervised-learning
7) https://www.ibm.com/cloud/learn/unsupervised-learning
8) https://www.ibm.com/cloud/learn/machine-learning#toc-reinforcem-oVSqcdnf
9) https://www.analyticsvidhya.com
10) https://www.ibm.com
11) https://aditi-mittal.medium.com
12) https://www.upgrad.com/blog/best-datasets-for-machine-learning-projects/
13) https://labelyourdata.com/articles/what-is-dataset-in-machine-learning/
14) https://machinelearningmastery.com/data-learning-and-modeling/
15) https://labelyourdata.com/articles/machine-learning-and-training-data/#splitting_your_data_set_training_data_vs_testing_data_in_machine_learning
16) https://blog.ldodds.com/2013/02/09/what-is-a-dataset/

17) "Traffic definition and meaning | Collins English Dictionary". www.collinsdictionary.com. Retrieved 3 January 2020.
18) "traffic". American Heritage Dictionary (Fifth ed.). 2013. Retrieved 23 March 2014.
19) Harper, Douglas (2001–2014). "traffic (n.)". Online Etymological Dictionary. Retrieved 23 March 2014.
20) "traffic, n.". OED Online. Oxford University Press. March 2014.

21) Jennifer V. Riverside, CA Comedy traffic school, https://www.comedytrafficschool.com/the-history-of-traffic/
22) The November 23, 1924, cover of the New York Times shows a common representation of cars during the era — as killing machines. (New York Times)
23) Government safety posters ridicule jaywalking in the 1920s and '30s. (National Safety Council/Library of Congress)
24) "Motoring firsts". National Motoring Museum.
25) "Ohio History Central". Ohio History. Retrieved 2008-08-27.
26) "data2.collectionscanada.gc.ca". collectionscanada.gc.ca. co
27) Pagan Kennedy Aug. 30, 2013, www.nytimes.com/2013/09/01/magazine/who-made-that-traffic-radar.html
28) ADAS and Automated/Autonomous Driving System Market in China 2019 No.2237
29) Alghnam S, Alkelya M, Alfraidy M, Al-Bedah K, Albabtain IT, Alshenqeety O. Outcomes of road traffic injuries before and after the implementation of a camera ticketing system: a retrospective study from a large trauma center in Saudi Arabia. Ann Saudi Med. 2017 Jan-Feb

30) J.YANG and J. LI, “APPLICATION OF DEEP CONVOLUTION NEURAL NETWORK”

31) Vibha, L., et al. “Moving Vehicle Identification Using Background Registration Technique for Traffic Surveillance.” Computer, vol. I, 2008, pp. 19–21.
32) Altundogan, T. G., & Karakose, M. (2019). Image Processing and Deep Neural Image Classification Based Physical Feature Determiner for Traffic Stakeholders. 2019 7th International Istanbul Smart Grids and Cities Congress and Fair (ICSG). doi:10.1109/sgcf.2019.8782371
33) Jahangiri, Arash, et al. “Adopting Machine Learning Methods to Predict Red-Light Running Violations.” IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, vol. 2015– October, 2015, pp. 650–55, doi:10.1109/ITSC.2015.112.

34) Zaheri, Donia, and Montasir Abbas. "An Algorithm for Identifying Red Light Runners from Radar Trajectory Data." IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, vol. 2015– October, 2015, pp. 2683–87, doi:10.1109/ITSC.2015.431.

35) Altundogan T. G., Karakose M., A Time Limited Park Violation Detection Approach Using Size Based Feature Extraction, International Conference on Advances and Innovations in Engineering, , 2017.

36) Wang, Jin-xiang. Research of Vehicle Speed Detection Algorithm in Video Surveillance. 2016, pp. 1–4.

37) https://blogs.sas.com

38) https://www.sas.com

39) J.YANG and J. LI, "APPLICATION OF DEEP CONVOLUTION NEURAL NETWORK"

40) https://analyticsindiamag.com

41) Thuan, DoCong. "Do Thuan EVOLUTION OF YOLO ALGORITHM AND YOLOV5: THE STATE-OF-THE-ART OBJECT DETECTION ALGORITHM EVOLUTION OF YOLO ALGORITHM AND YOLOV5: THE STATE-OF-THE-ART OBJECT DETECTION ALGORITHM." (2021).

42) https://medium.com/analytics-vidhya/all-about-yolos-part1-a-bit-of-history-a995bad5ac57

43) https://analyticsindiamag.com/yolov5

44) author = {Andreas Geiger and Philip Lenz and Raquel Urtasun},
title = {Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite},
book title = {Conference on Computer Vision and Pattern Recognition (CVPR)},
year = {2012}

45) Torrey transfer learning, by Lisa Torrey and Jude Shavlik},
{Transfer Learning}

46) Lin TY. et al. (2014) Microsoft COCO: Common Objects in Context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48