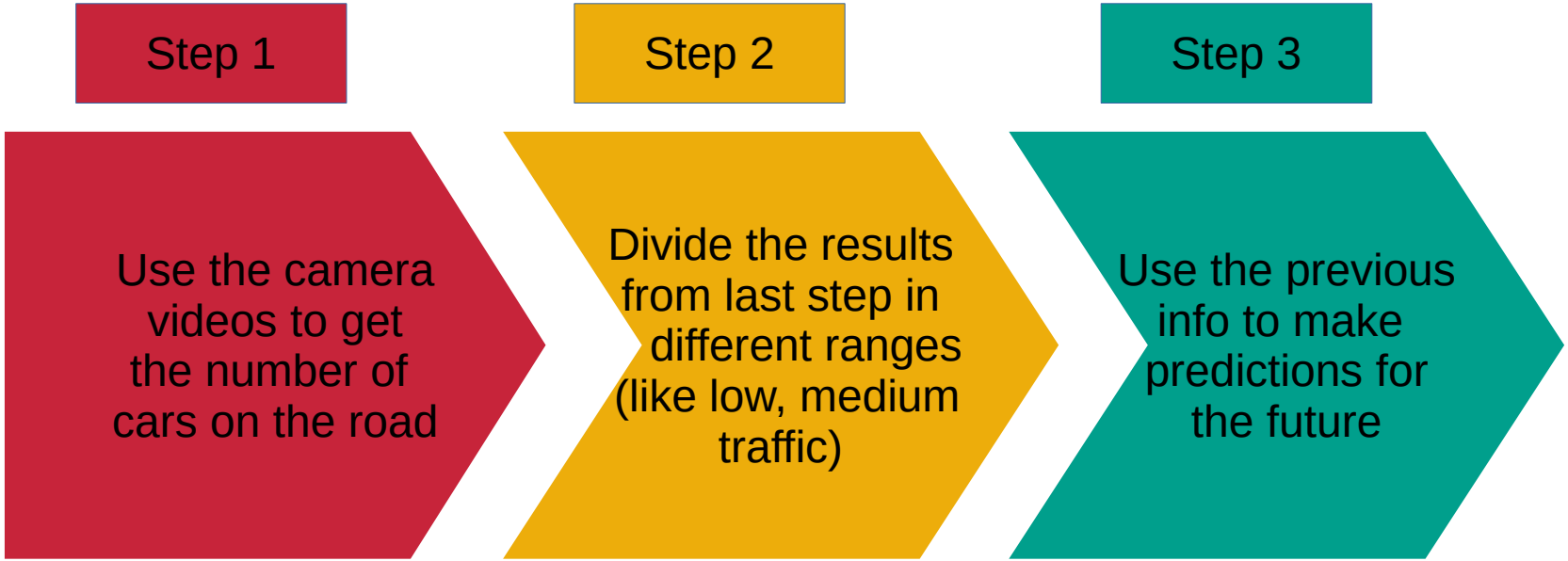
Predicting Traffic Using Real Time Data

Kushajveer Singh
Punjab Engineering College
Chandigarh

Problem

- By using real time video from cameras we can make inferences on the amount of traffic after certain amount of time(like traffic after 3 minutes).
- Now the technique I discuss to do the above can be extended to control traffic automatically.(more on that later)

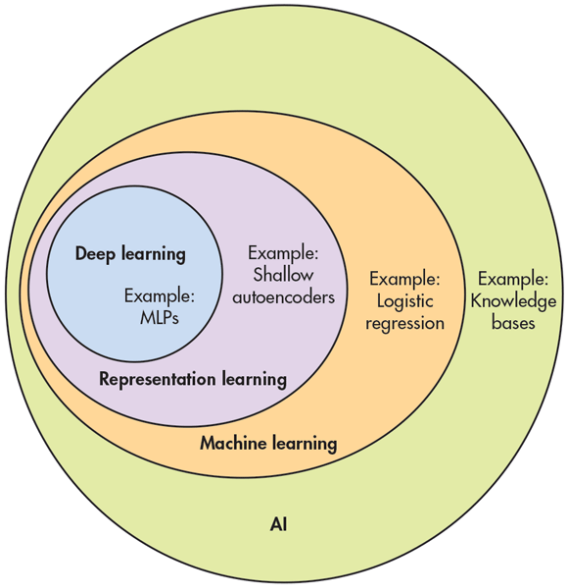
Solution


Step 1
Step 2
Step 3
Use the camera videos to get the number of cars on the road
Divide the results from last step in different ranges (like low, medium traffic)
Use the previous info to make predictions for the future

A Brief Intro to Deep Learning

Machine Learning is concerned with giving machines ability to learn. Now Deep Learning grew as a subfield of ML where we used algorithms that were modelled on our brain and which gave them their power.

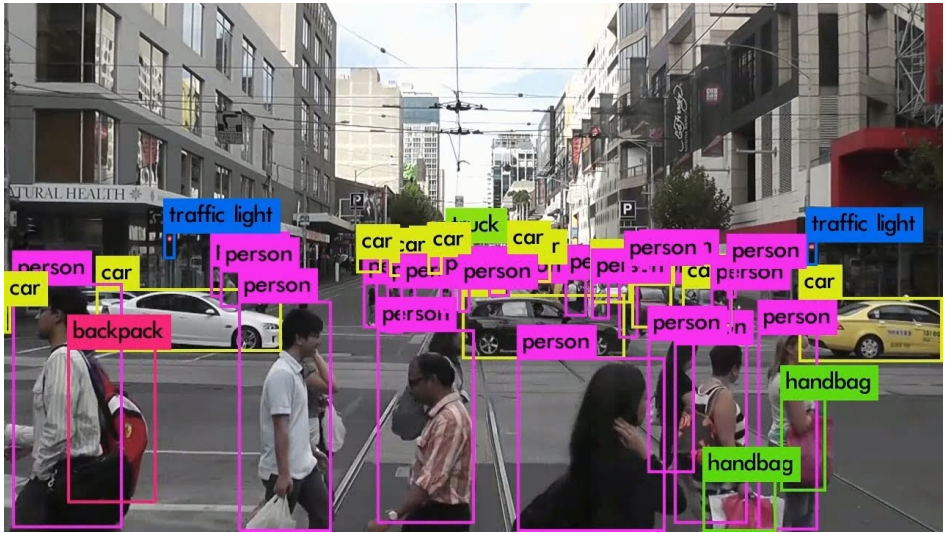
In simple terms DL tries to learn non-linear representations using the data that we give them.



Solution to Step 1

- I used a specific field of deep learning called object recognition. In this technique we give image to a the model and it finds the position of all the objects in it that it was trained to find.

The object detection model used in this project is YOLO.



What is YOLO?

- YOLO is a state of the art model for object detection. It is different versions where v3 is the latest.
- The model used in this project is tiny-yolo(which as name suggests is a very minimilistic part of the main algo)
- Just to get a idea
  - YOLOv3 has weights of size 256MB
  - Tiny-yolo has weights of size 50MB
  - Impling yolov3 has 20,000,000 has more parameters in best case.
- Naively more weights means more prediction power.

tiny-yolo without training

/home/kushaj/Desktop/Chandigarh Project/darkflow/video.avi

After training

Only cars were used after training

/home/kushaj/Desktop/Chandigarh Project/Trained Model/video.avi

Training Inferences

- Limitations :
  - Noise like time at top right corner
  - Simple algo used
  - It was trained using only 120 images where most state of the art models are trained using millions of images

- Solution :
  - Noise can be easily filtered out
  - Tiny-yolo was used due to compute limitations and time contraints to get the results
  - As far as training images go, it is clear that with even 120 images it started classifying most the images in that noise. So as far as training is concerned maybe 500 – 600 images would serve the purpose.

Solution to Step 2

- This step is simple. The bounding boxes produced ny the yolo algorithms are stored as a JSON file and the len of that json file gives the number of cars in that frame. Now you can define different ranges for the traffic.
- As this project was just a test run for the approach, and due to various limitations discusses the ranges chosen by me are
  - 0-2 -> small
  - 3-5 -> medium
  - 6-8 -> high
  - >8 ->very high
- They are them one-hot encoded like [1,0,0,0] for small, [0,1,0,0] for medium, [0,0,1,0] for high, [0,0,0,1] for very high

Solution to Step 3

- After getting the number of cars from step 1, we can use various time forecasting techniques to predict the state of the traffic after certain amount of time.
- State of the art methods for time series forecasting exists like LSTMs (Long Short Term Memory), seasonal ARIMA. But they were not used in this project due to amount of data they need to train. (solution to this problem discussed in next slide)

- So I trained a very simple neural network which is note even fit to do time seires forecasting, but the results after training look promising.

  ## Show get_inferences.py for the results

- To train the prediction algorithm we do not need any manual effort. The solution is that the prediction algortihm used the number of cars as input to output a range.
- Suppose a particular day, you run the object detection algorithm to get number of cars on a road and save the data over the entire day. Then, at night you can just simply use that data to train the algorithm. Thus requiring no manual effort and thus the prediction algorithm can also learn daily traffic patterns.
- LSTM would be used as a prediction algorithm, as in the current era deep learning recurrent networks (on which LSTM is based) is the closest that we have got to model the brain.

Further ideas

- By getting the number of cars at a cross we can change the duration of traffic lights to give more preference to the road with more cars. Solution I thought it is make a Reinforcement Leanring agent that would automatically learn it.
- By integrating it with google map or making an independent app(still thinking) it can be used by the people to get live traffic status and just plan their routes.
- There is a technique in machine leanrning called Anomaly detection which can be used to extend this model to get results that there is an accident on the road or the road is flooded or something abnormal.
- A dynamic routing algorithm could be developed which in real time can suggest new paths to take with least traffic. Amulance routing can benefit from this technique.

- Problems I have not thought about yet
  - As I am more on the side of deep learning research rather than on the model deployment, my knowledge about how to deploy a model is limited.
  - But I am aware of the common terms used when deploying deep learning models, so if given sufficient time I can become acquainted with the required knowledge

THANK YOU