International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

TRAFFIC PREDICTION FOR SMART TRANSPORTATION SYSTEM USING DEEPLEARNING

MS. S.KEERTHANA[1], MR. I.SUDHAKAR[2]

Student of 11 MSC(Computer Science),Department of Science with Computer Science,VLB Janakiammal College of Arts and Science, Kovaipudur, Coimbatore,India.

MCA.,M.Phill.,MBA.,Assistant Professor,Department of Science with Computer Science,VLB Janakiammal College of Arts and Science,Kovaipudur, Coimbatore,India

ABSTRACT :

This project entitled **"Traffic Prediction for Smart Transportation System using Deep Learning"** is developed by using Front end as a Python. Currently the traffic control systemin our country is non-flexible to the ever-growing number of vehicles on the road. Traffic light is the basic element in traffic flow control through specified waiting and going time, fixed traffic light time systems is a bad control way. Intelligent traffic systems include smart way to control traffic light time based on number of vehicles in each lane. Improving traffic signal control systemwill increase safety, reliability, and traffic flow speed and reduce average travelling and waiting time for passengers. The objective is to design an efficient automatic Traffic Time Saver system. The system is implemented on the traffic control. In this proposed application system first captures the vehicle image. Vehicle images are extracted using the image segmentation finally converting the images from RGB to gray scale. Next, the segmentation is applied on the prepared image and then for each segment the neural networks will predict a vehicle or not. A counter will count the positive segments. Finally, the suitable periods for each light color will display in GUI.

1. INTRODUCTION :

The traffic prediction system is to provide a user-friendly interface for vehicle user they can save time in Traffic signal more effectively. Because traffic signal is a bad control way. Because of traffic signal authority manually set the timer in traffic signal. The problem is will leads tome consuming process user need to wait for green color. Some lines don't have any vehicles, also sometime green color signals will show because signal timing will statically time control . Thisleads some problem to driving user they need to wait long time interval. Now a day's driving user facing lot of problem in Traffic because traffic control system is a bad control way. Sometime traffic signal authority manually set the timer in traffic The main objective of the proposed systemis to provide a user- friendly interface for time save in Traffic signal. Proposed system initially user can be able to upload four traffic signal images. After successfully image upload Vehicle image is extracted using the image segmentation finally converting the images from RGB to grayscale. Next, the segmentation is applied on the prepared image and then for each segment the neuralnetworks will predict a vehicle or not. Finally, the system automatically counts each uploaded image of a total vehicle count and control traffic signal automatically.

OBJECTIVES :

The main objective of the proposed system is to provide a user-friendly interface for time save in Traffic signal. Proposed system initially user can be able to upload four traffic signal images. After successfully image upload Vehicle image is extracted using the image segmentation finally converting the images from RGB to gray scale. Next, the segmentation is applied on the prepared image and then for each segment the neural networks will predict a vehicle or not. Finally, the systemautomatically counts each uploaded image total vehicle count and control traffic signal automatically .

SCOPE OF THE STUDY :

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements can be made to the application with IOT interface, camera will capture the traffic signal image and will count vehicle information and display the traffic signal in IOT interface.

MODULE DESCRIPTION :

- **Dataset Image Upload process**

This is a first module in our application .Deep learning application initially we need to collect data for training purpose **.**This is first module of this application initially user an able to get image dataset source from UCI repository .After that user can be able to upload dataset through browse option .Dataset image may be png or jpeg format.

- **Image segmentation**

Image segmentation is a commonly used technique in digital image processing and analysis to partition an image into multiple parts or regions, often based on the characteristics of the pixels in the image. Image segmentation is one of the important processes to identify vehicle count on traffic. Image Segmentation is the process by which a digital image is partitioned into various subgroups (of pixels) called Image Objects, which can reduce the complexity of the image, andthus analyzing the image becomes simpler.

- **Detect Object and Boundary Boxes**

A bounding box is an imaginary rectangle that serves as a point of reference for object detection and creates a collision box for that object. Data annotators draw these rectangles over images, outlining the object of interest within each image by defining its X and Y coordinates. This
makes it easier for machine learning algorithms to find what they're looking for, determine collision paths, and conserves valuable computing resources. Bounding boxes are one of the most popular image annotation techniques in deep learning. Compared to other image processing methods, this method can reduce costs and increase annotation efficiency

- **YOLO object Classification**

Object detection is a computer technology related to computer vision and image processing that deals with detecting objects of a certain class in digital images. You Only Look Once (YOLO) propose using an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once. It differs from the approach taken by previous object detection algorithms, which repurposed classifiers to perform detection. YOLO is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy.
It has been used in various applications to detect traffic vehicle count and classification purpose.

- **Vehicle count detection**

Each segment object analysis using neural networks will predict if it is a positive segment (a vehicle) or not. A counter will count the positive Segment. Finally positive count wills maintain separate table. Finally, system will display total number of vehicle count is in traffic.

INTRODUCTION TO BACK END AND FRONT END :

PYTHON

The primary programming language used for developing the software. Python is chosen for its simplicity, readability, and extensive support for web scraping and data manipulation tasks. Its robust ecosystem of libraries makes it an ideal choice for implementing the scraping and processing functionalities request for this project. In Open-source it's free and available for further improvements, like adding helpful features or fixing bugs.

LIBRARIES AND FRAMEWORKS

Image Processing Libraries

OpenCV: An open-source computer vision library used for real-time image and video processing. It is essential for tasks such as capturing live video feeds, applying image preprocessing techniques (like grayscale conversion), and performing image segmentation to detect vehicles.

DATA HANDLING AND ANALYSIS :

NumPy: A fundamental library for numerical computing in Python. It is essential for handling large arrays, matrices, and performing operations that are crucial in deep learning pipelines.
Pandas: A powerful data manipulation tool for handling structured data (like traffic statistics). It is used to manage time-series data, which can be leveraged for training predictive models.

VEHICLE DETECTION AND OBJECT TRACKING

YOLO (You Only Look Once): A real-time object detection algorithm that can be used to detect vehicles in the traffic system. YOLO is known for its speed and accuracy in detecting objects in images and videos.

PERFORMANCE MONITORING AND LOGGING

Tensor Board is an essential tool integrated with TensorFlow for visualizing deep learning model performance, including metrics like loss, accuracy, and predictions.

DATABASE MANAGEMENT

SQLite or MySQL Databases for storing traffic data such as vehicle counts, historical traffic patterns, and signal timings. These can also store training data for deep learning models.

API AND WEB INTEGRATION

Fast API is a modern framework for building fast web APIs, suitable for deploying machine learning models to provide real-time traffic predictions and control suggestions.

CLOUD AND DISTRIBUTED COMPUTING

Cloud platforms offer scalable infrastructure to deploy the traffic control system in a real-world environment, handling large datasets and real-time predictions.

DEEP LEARNING FRAMEWORKS

TensorFlow or PyTorch both of these deep learning libraries are powerful tools for building neural networks and performing image processing tasks. They offer robust capabilities for creating, training, and deploying deep learning models, especially for tasks like vehicle detection and traffic prediction.

REAL TIME VEHICLE DETECTION

Uses image segmentation techniques to detect and count vehicles in each lane. The system processes video streams or images from traffic cameras to identify vehicles, using deep learning algorithms for precise detection.

DYNAMIC TRAFFIC LIGHT CONTROL

Based on the detected vehicle count, the software adjusts traffic light timings. This dynamic control minimizes wait times for underutilized lanes and prioritizes heavier traffic, ensuring optimal flow.

DEEP LEARNING PREDICTION

Neural networks are employed to predict traffic patterns based on historical data, helping the system anticipate traffic surges or dips. This prediction capability allows for proactive traffic signal adjustments.

GRAYSCALE CONVERSION FOR IMAGE PROCESSING

To enhance efficiency, the system converts RGB images into grayscale for faster image segmentation. This reduces computational load while maintaining accuracy in vehicle detection.

SCALABILITY

Designed to be implemented in multiple intersections of urban areas, the system can handle traffic data from various locations, making it adaptable to different traffic volumes and patterns.

METHODLOGY :

ALGORITHM

The YOLO (You Only Look Once) algorithm is a popular object detection method in deep learning, primarily used for real-time object detection tasks. It can detect objects in images or video streams and classify them into various categories. YOLO is known for its speed and efficiency, making it ideal for applications where real-time processing is required.

Single Pass Detection:

Unlike traditional methods that apply a sliding window approach or region proposals, YOLO divides the image into a grid and processes the entire image in one pass. Each grid cell is responsible for predicting a bounding box and a class probability for objects within the cell.

Bounding Boxes:

For each grid cell, YOLO predicts a set of bounding boxes, each with a confidence score. This confidence score reflects how likely it is that a box contains an object and how accurate the box is in locating the object.
Class Prediction:
YOLO outputs the probability of an object belonging to each of the predefined classes. The final prediction combines the bounding box and class probability to identify objects in the image.

Real-Time Detection:

YOLO's main strength is its real-time detection capability. It processes images at a much faster rate compared to other algorithms like R-CNN or Faster R-CNN.

YOLO algorithm works using the following three techniques:

- Residual blocks
- Bounding box regression
- Intersection over Union (IOU)
- Residual blocks

First, the image is divided into various grids. Each grid has a dimension of S x S. The following image shows how an input image is divided into grids.

Bounding box regression

A bounding box is an outline that highlights an object in an image.
Every bounding box in the image consists of the following attributes :

- Width (bw)
- Height (bh)
- Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.
- Bounding box center (bx,by)

YOLO Algorithm Implementation

Step-1: Import the required libraries

Step-2: Create a function for filtering the boxes based on their probabilities and threshold:

```
def yolo_filter_boxes(box_confidence, boxes, box_class_probs, threshold = .6):
box_scores = box_confidence*box_class_probsbox_classes = K.argmax(box_scores,-1) box_class_scores =
K.max(box_scores,-1) filtering_mask = box_class_scores>threshold scores = tf.boolean_mask(box_class_scores,filtering_mask)
boxes = tf.boolean_mask(boxes,filtering_mask)  classes =
tf.boolean_mask(box_classes,filtering_mask)return scores, boxes, classes
```

Step-3: Define a function to calculate the IoU between two boxes:

```
def iou(box1, box2): xi1 = max(box1[0],box2[0]) yi1 = max(box1[1],box2[1]) xi2 = min(box1[2],box2[2]) yi2 =
min(box1[3],box2[3]) inter_area = (yi2-yi1)*(xi2-xi1) box1_area =(box1[3]-box1[1])*(box1[2]-box1[0]) box2_area =
(box2[3]- box2[1])*(box2[2]-box2[0]) union_area = box1_area+box2_area-inter_area iou = inter_area/union_area return
iou
```

Step-4: Define a function for Non-Max Suppression.

Step-5: Create a random volume of shape (19,19,5,85) and then predict the bounding boxes.

Step-6: Finally, we will define a function which will take the outputs of a CNN as input and return the suppressed boxes:

Step-7: Use the yolo_eval function to make predictions for a random volume: scores, boxes, classes = yolo_eval(yolo_outputs)

Step-8: Use a pretrained YOLO algorithm on new images and see how it works: sess =K.get_session() class_names = read_classes("model_data/coco_classes.txt")anchors = read_anchors("model_data/yolo_anchors.txt") yolo_model = load_model("model_data/yolo.h5")

Step-9: Define a function to predict the bounding boxes and save the images with these boundingboxes included.

Step-10: Read an image and make predictions using the predict function:

img = plt.imread('images/img.jpg')

image_shape = float(img.shape[0]), float(img.shape[1]) scores, boxes, classes = yolo_eval(yolo_outputs, image_shape)**tep-11:** Plot the predictions: out_scores, out_boxes, out_classes = predict(sess, "img.jpg")

CONCLUSION :

Implementation is the stage of the project when theoretical design is turned into a working system. Thus, it can be the most critical stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective. Proposed system successfully presented YOLO, a new approach to object detection, algorithm that can improve the detection performance based on limited training data and an effective database expansion method. In this proposed application system first captures the vehicle image. Vehicle image is extracted using the image segmentation finally converting the images from RGB to gray scale. Next, the segmentation is applied to the prepared image and then for each segment the neuralnetworks will predict a vehicle or not.

REFERENCE:

1. Python Tricks: A Buffet of Awesome Python Features by Dan Bader
2. Python Tricks: A Buffet of Awesome Python Features by Dan Bader
3. Python Tricks: A Buffet of Awesome Python Features by Dan Bader EPUB

Web Reference:

1. Mark Lutz ," core Python language with this hands-on book.",2013
2. https://creatorpdf.com/B0785Q7GSY
3. https://creatorpdf.com/B0785Q7GSY