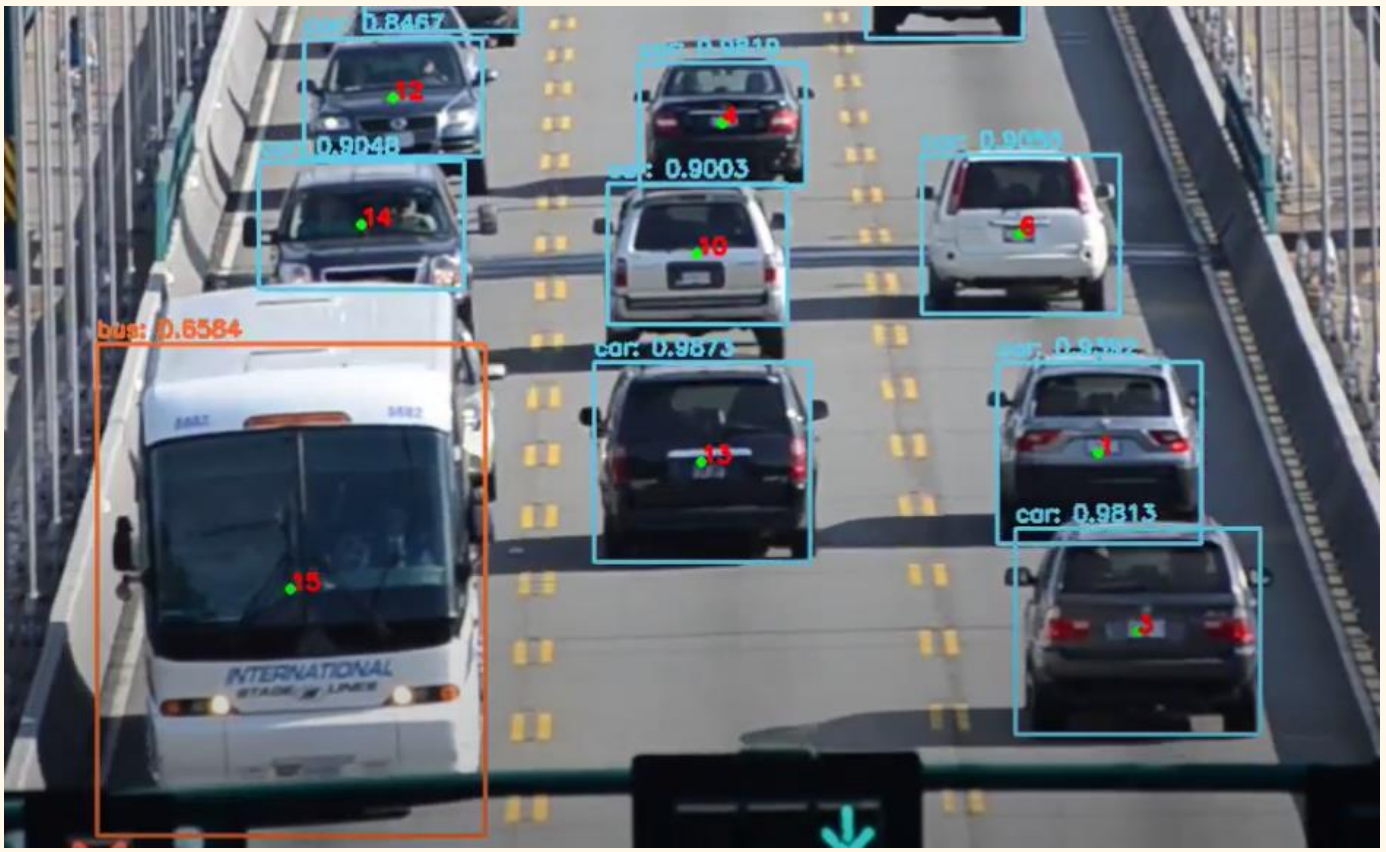
CV
PROJECT REPORT

Vehicle Counting & Classification

—

By GROUP - 7



Abstract:

Vehicle counting and classification typically involves analyzing images or video frames to detect and track vehicles, and then classify them based on their type or other characteristics. This requires different tools and techniques, such as computer vision libraries (e.g., OpenCV), object detection algorithms (e.g., YOLO, Faster R-CNN), and machine learning models (e.g., SVM, neural networks).

If we have images or video frames that one wants to analyze for vehicle counting and classification, we can use OpenCV to preprocess the images, detect and track vehicles, and then use machine learning models to classify them. There are also pre-trained models available that can be fine-tuned for specific use cases.

We incorporated the task of vehicle counting and classification into our project. This involved analyzing images or video frames to detect and track vehicles, and then classifying them based on their type or other characteristics. We utilized tools such as the OpenCV library, object detection algorithms (e.g., YOLO), and machine learning models (e.g., SVM) to accomplish this task. We acknowledge that vehicle counting, and classification is a complex task that requires a good understanding of computer vision and machine learning.

Main Objective:

- To detect vehicles in a video using YOLOv3.
- To count the number of vehicles detected in the video.
- To draw bounding boxes around the detected vehicles.
- To classify the detected vehicles based on their type or category.
- To generate an output video with the detected vehicles.

Methodology:

We made this project which detects vehicles in a video stream and counts them. The script uses YOLOv3 (You Only Look Once version 3) to detect vehicles in the video stream, and then counts the vehicles based on the number of bounding boxes around the vehicles. The script takes as input a video file and outputs a new video file with the detected vehicles and the vehicle count displayed in the top-left corner of the video.

Basic:

1. The necessary packages are imported, such as NumPy, OpenCV, and time.

2. A list of vehicle classes is defined. This list includes the classes "bicycle", "car", "motorbike", "bus", and "truck".

3. The command line arguments are parsed to extract the values required for the script. These include the labels, weights path, configuration path, input video path, output video path.

4. A list of colors is generated to represent each possible class label.

Functions:

5. A function is defined to display the vehicle count on the top-left corner of the frame.

6. A function is defined to display the frames per second (FPS) of the detected video.

7. A function is defined to draw all the detection boxes with a green dot at the center.

8. A function is defined to initialize the video writer with the output video path and the same number of frames per second, width, and height as the source video.

9. A function is defined to identify if the current box was present in the previous frames.

10. A function is defined to count the vehicles based on the number of bounding boxes around the vehicles.

Important Methodology:

11. The image is pre-processed using YOLOv3, and the detections are made using the pre-defined confidence and threshold values.

12. The previous frame detections are checked to identify if the current box was present in the previous frames.

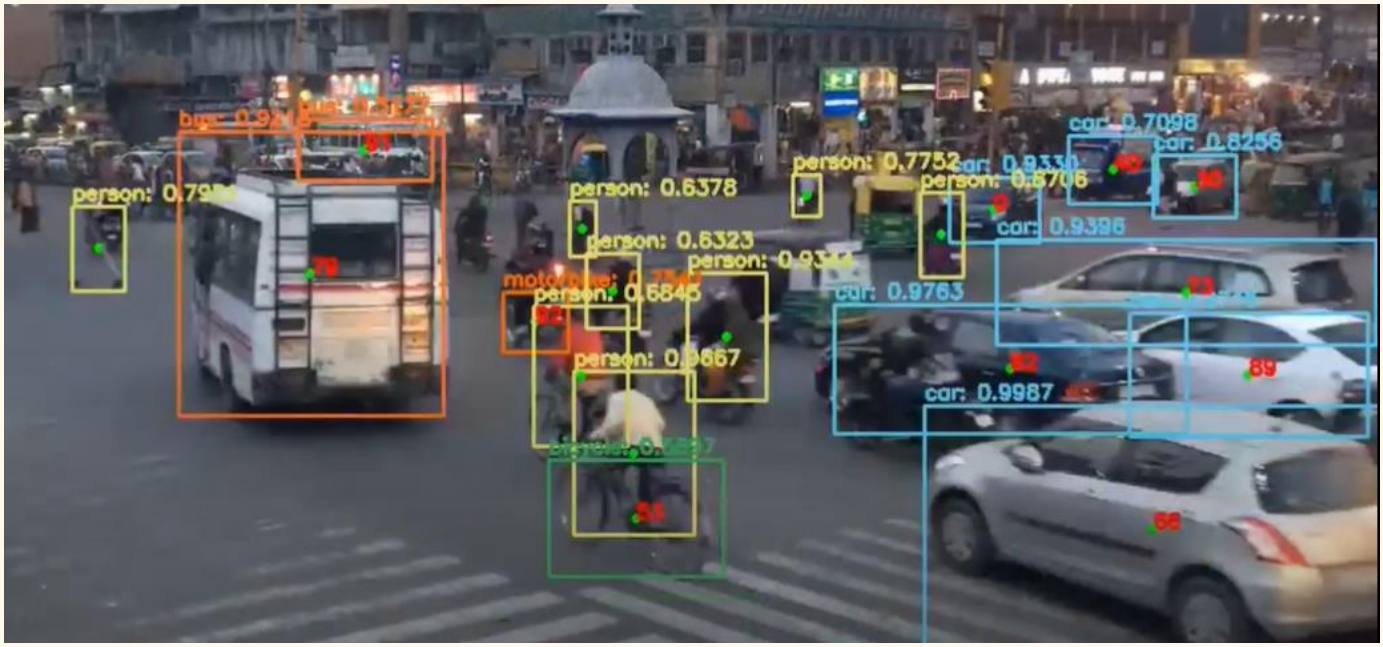
13. The detected boxes are counted based on the number of bounding boxes around the vehicles.

14. The vehicle count is displayed on the top-left corner of the frame, and the FPS is displayed in the terminal.

15. The detection boxes are drawn with a green dot at the center.

16. Generating an output video with the detected Vehicles.

Results:



Conclusion:

This project successfully detected vehicles in a video using YOLOv3 and met all its objectives. This implementation has potential applications in traffic management, surveillance systems, and autonomous driving. The project can be extended to perform other tasks such as license plate recognition and vehicle tracking.

Contributions:

Name	Contributions
Rahul Varma	boxInPreviousFrames, drawDetectionBoxes, main function
Aalhad Kate	count_vehicles, initializeVideoWriter, main function
Nitigya Joshi	displayVehicleCount, input_retrieval, main function
Chandra Shekhar	displayFPS, main function

Thank You

Group 7.