GROUP: 2

Madhav Institute of Technology and Science

Object Detection for Autonomous Vehicles

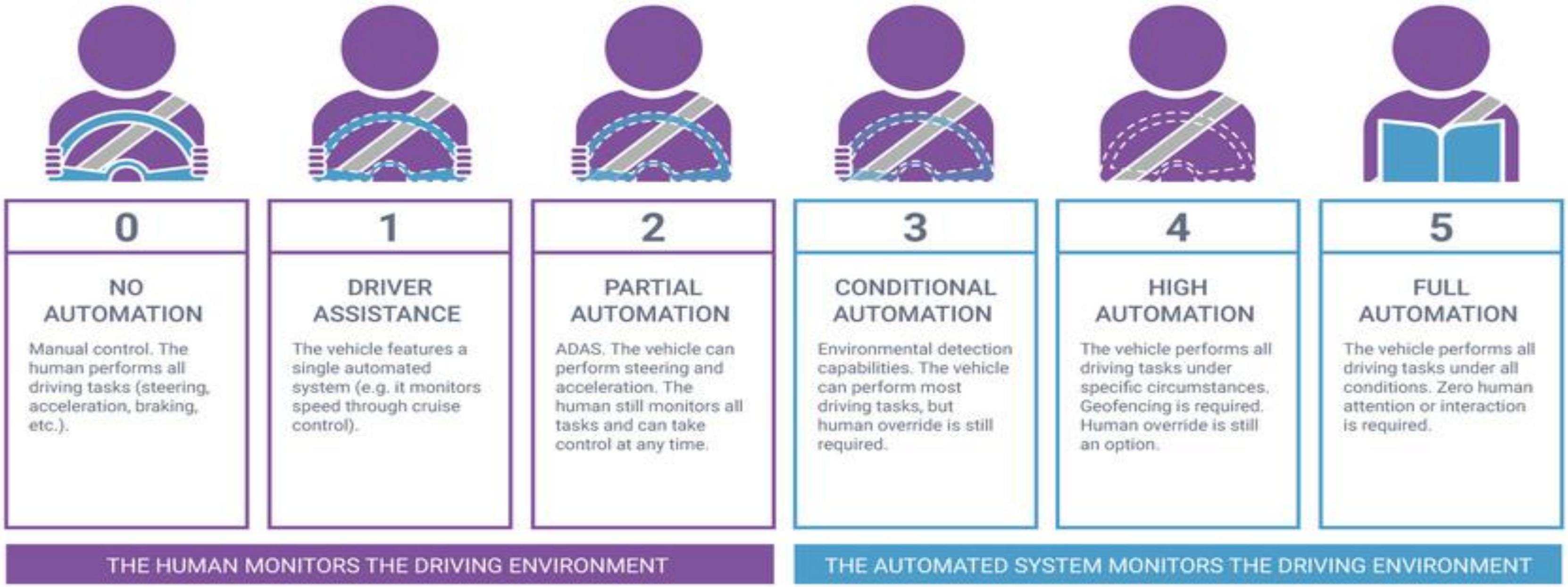
Kevin Shaju 49
Alok Kumar 11
Aman Gupta 12
Jatin Shrivastava 46

Under the Guidance of:
Dr. Karuna Markam
Prof. Rishabh Shukla

Idea

Creating an object detection and lane assist system for autonomous driving vehicles.By using these two technologies we can create a system that is Level 2 or partial autonomous vehicle. In this work, we have to examine an approach to deep object detection that makes bounding box predictions for an in real time.

LEVELS OF DRIVING AUTOMATION



0	1	2	3	4	5
NO AUTOMATION	DRIVER ASSISTANCE	PARTIAL AUTOMATION	CONDITIONAL AUTOMATION	HIGH AUTOMATION	FULL AUTOMATION
Manual control. The human performs all driving tasks (steering, acceleration, braking, etc.).	The vehicle features a single automated system (e.g. it monitors speed through cruise control).	ADAS. The vehicle can perform steering and acceleration. The human still monitors all tasks and can take control at any time.	Environmental detection capabilities. The vehicle can perform most driving tasks, but human override is still required.	The vehicle performs all driving tasks under specific circumstances. Geofencing is required. Human override is still an option.	The vehicle performs all driving tasks under all conditions. Zero human attention or interaction is required.
THE HUMAN MONITORS THE DRIVING ENVIRONMENT			THE AUTOMATED SYSTEM MONITORS THE DRIVING ENVIRONMENT		

Future Scope

Automation can help reduce the number of crashes on our roads. Government data identifies driver behavior or error as a factor in 94 percent of crashes, and self-driving vehicles can help reduce driver error. Higher levels of autonomy have the potential to reduce risky and dangerous driver behaviors.

Abstract

In this project, vehicle (car) detection takes a video as input and produces the bounding boxes as the output. We use TensorFlow Object Detection API, which is an open source framework built on top of TensorFlow to construct, train and deploy object detection models. The Object Detection API also comes with a collection of detection models pre-trained on the COCO dataset that are well suited for fast prototyping. Specifically, we use a lightweight model: centernet_resnet50_v2.

Tensorflow Object Detection API

The TensorFlow object detection API is the framework for creating a deep learning network that solves object detection problems. There are already pretrained models in their framework which they refer to as Model Zoo. This includes a collection of pretrained models trained on the COCO dataset, the KITTI dataset, and the Open Images Dataset. These models can be used for inference if we are interested in categories only in this dataset.

Dependencies

1. TensorFlow 2.4.1
2. python version 3.8.8
3. Cuda toolkit 11.2
4. cuDNN version 8.1.1
5. google protocol buffer(protobuf) protoc-3.15.5-win64
6. coco api
7. Visual studio community - desktop developer tools
8. Setting up the environment variables

Model Used

CenterNet Object detection model with the ResNet-v2-50 backbone, trained on COCO 2017 dataset with training images scaled to 512x512.

The MS COCO dataset contains 330K images containing:

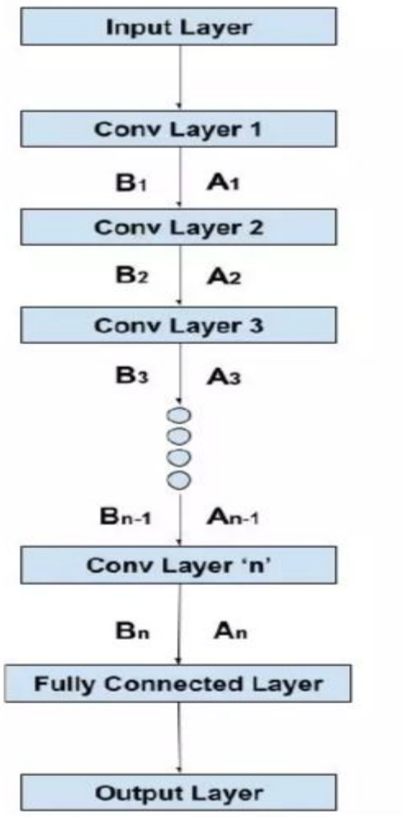
1.5 million object instances

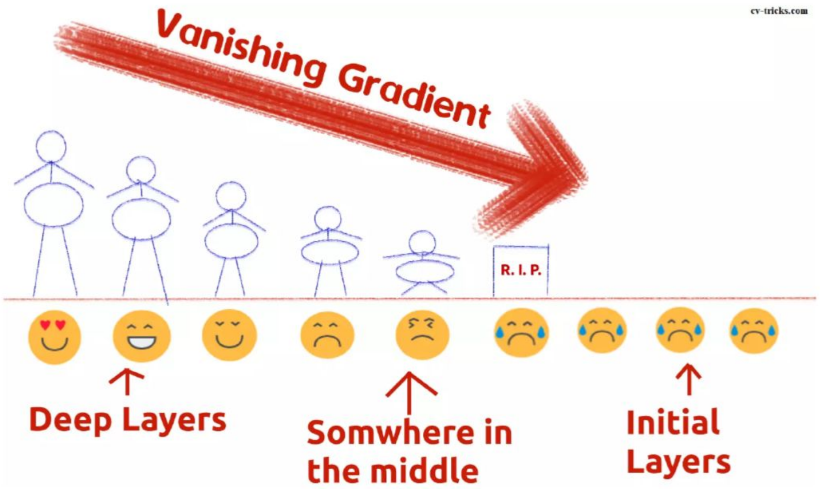
80 object categories

91 stuff categories

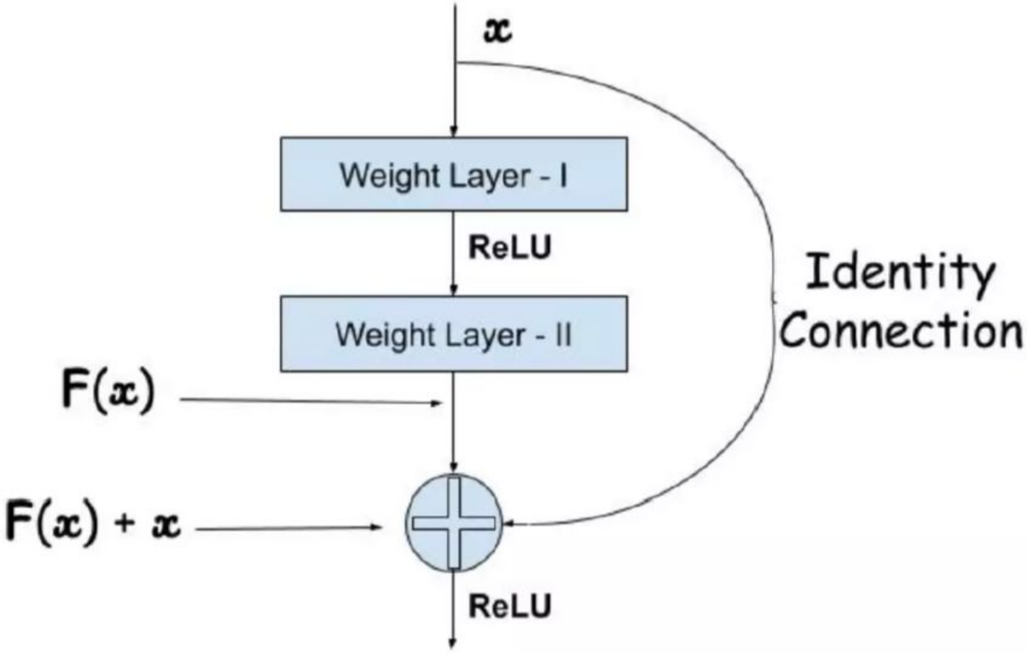
5 captions per image

250,000 people with key points





Vanishing Gradient Intuition

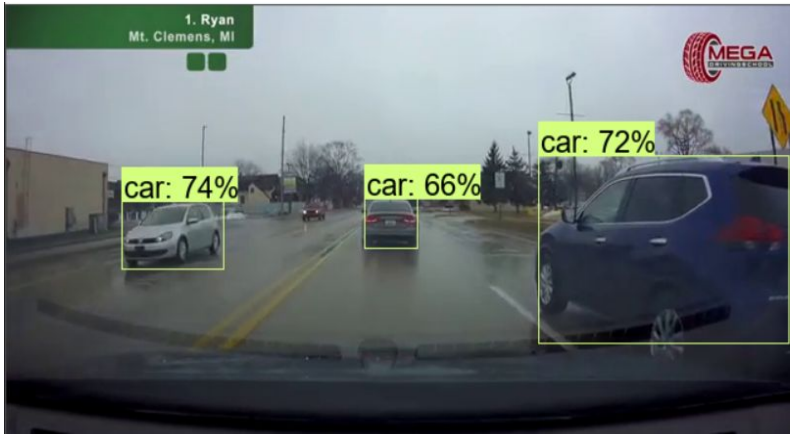


$$H(x) = \mathrm{ReLU}(F(x) + x)$$

Real time working

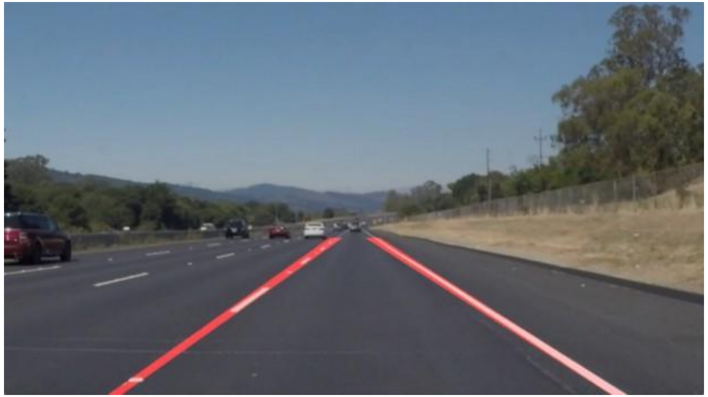
Input Frame

Output Frame



Lane Detection

We have developed a simple pipeline using OpenCV and Python For finding lane lines in an image, then apply this pipeline to a full video feed.



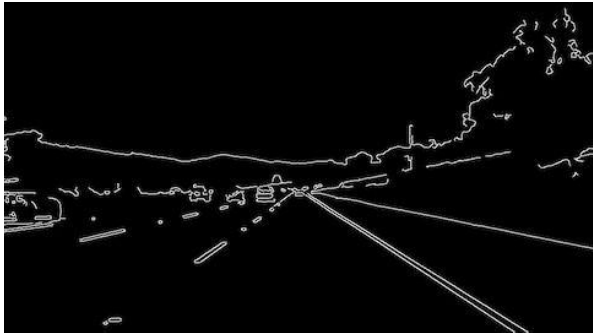
Working of Lane Detection

1. Removing Noise
2. Convert Image to Grayscale
3. Detect Edges
4. Determine Region of Interest
5. Draw lines
6. Hough Transform
7. Sort Lines
8. Reject outliers
9. Linear Regression
10. Draw lines and return Final Image

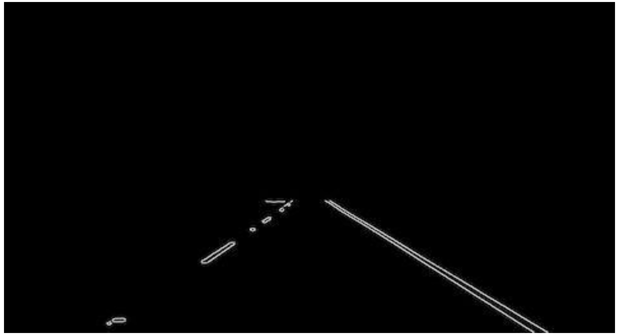
Noise Removed

Grayscale

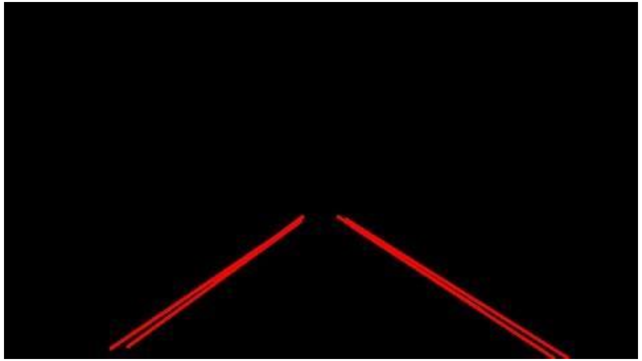
Edge Detection



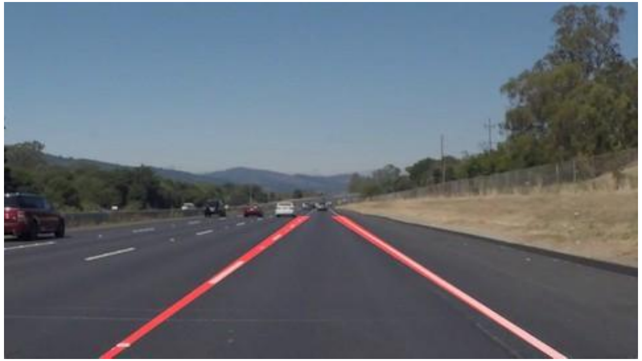
Region Of Interest



Hough Transform



Output Image



Thank You