



AUTONOMOUS DRIVING VEHICLE

Konark Verma
Aditya Mishra

SELF DRIVING CAR

Under the bonnet

How a self-driving car works

Signals from **GPS (global positioning system)** satellites are combined with readings from tachometers, altimeters and gyroscopes to provide more accurate positioning than is possible with GPS alone

Lidar (light detection and ranging) sensors bounce pulses of light off the surroundings. These are analysed to identify lane markings and the edges of roads

Video cameras detect traffic lights, read road signs, keep track of the position of other vehicles and look out for pedestrians and obstacles on the road

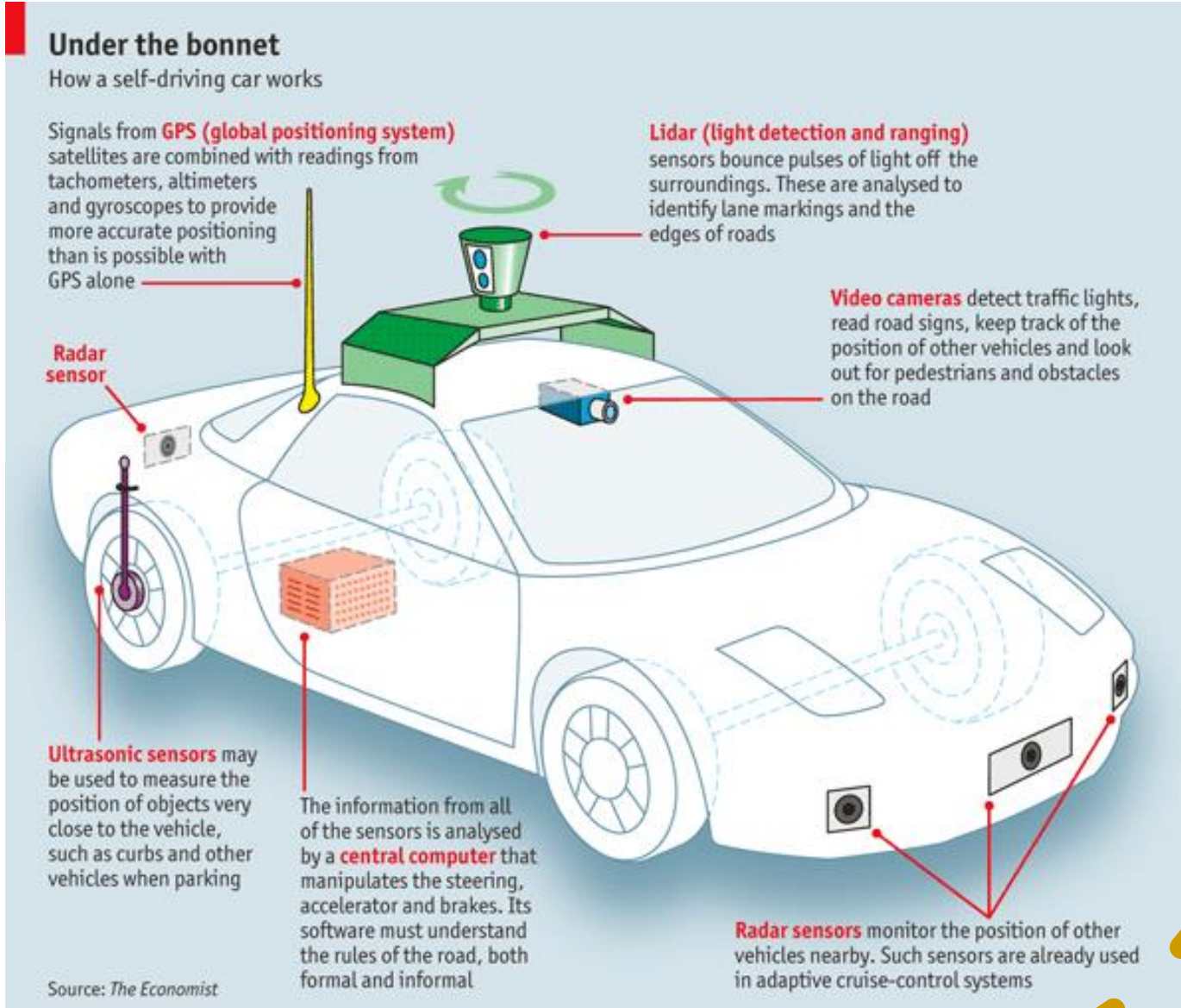
Radar sensor

Ultrasonic sensors may be used to measure the position of objects very close to the vehicle, such as curbs and other vehicles when parking

The information from all of the sensors is analysed by a **central computer** that manipulates the steering, accelerator and brakes. Its software must understand the rules of the road, both formal and informal

Radar sensors monitor the position of other vehicles nearby. Such sensors are already used in adaptive cruise-control systems

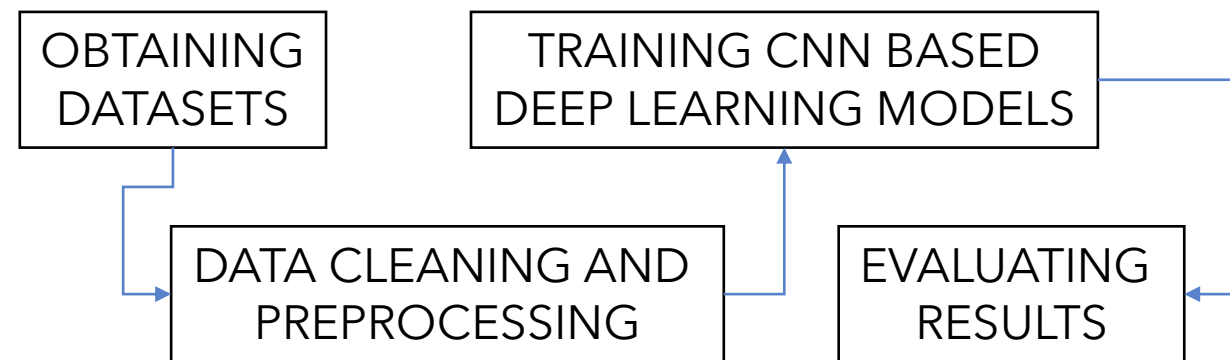
Source: *The Economist*



PROBLEM STATEMENT

- Problem is to implement a self driving car's steering wheel component with front board video as sensory input.
- We must predict the angle of rotation of the steering wheel according to the curvature of the road.
- Various CNN based deep learning methods are used to solve this problem. Our main goal here is to reduce the training MSE loss.

PIPELINE



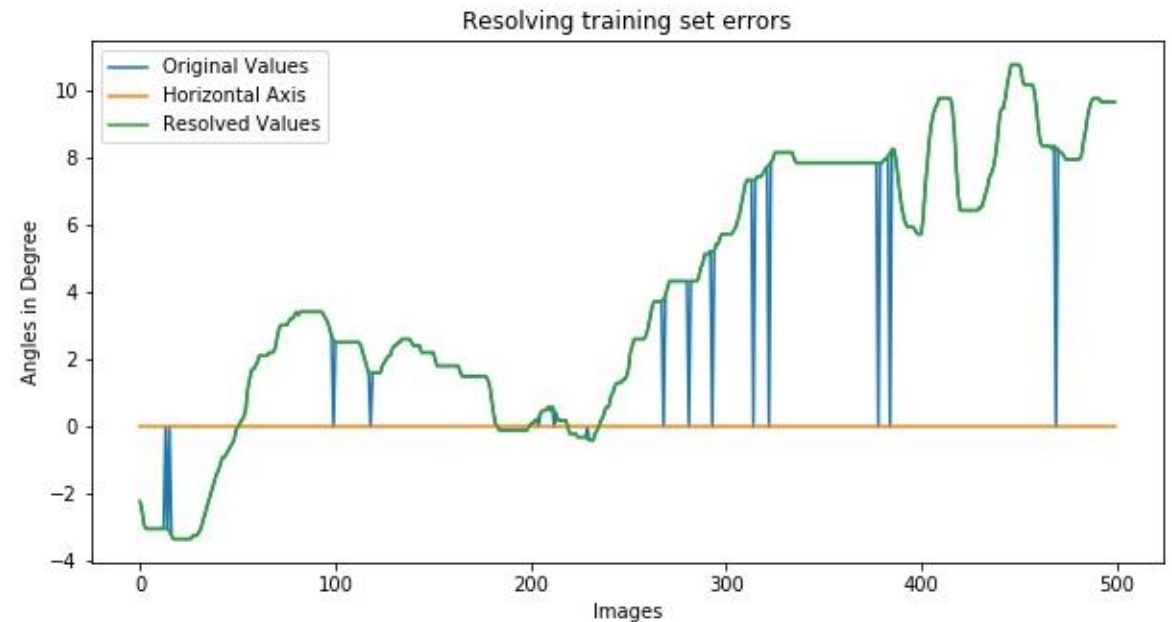
DATASET & REFERENCES

- Our Dataset:
 - Recorded Video of length 25 minutes
 - Size = 2.3 GB
 - It contains $25 \times 60 \times 30 = 45,000$ images
- Obtained from:
 - <https://github.com/SullyChen/Autopilot-TensorFlow>
 - <https://drive.google.com/file/d/0B-KJCaaF7elleG1RbzVPZWV4Tlk/view>
- Research Papers read:
 - End to End Learning for Self-Driving Cars by Nvidia.
 - <https://arxiv.org/pdf/1604.07316.pdf>



DATA CLEANING & PREPROCESSING

- Video file is first split into a sequence of 45405 images.
- Each image is then converted to a 200×200 pixel image from a 256×455 pixel image.
- The RGB values between 0-255 are converted to values between 0 and 1.
- The angles file contained some errors, which are resolved by smoothening the curve.
- The angles are also converted to radians from degrees.



CNN BASED DEEP LEARNING MODEL

Model: "sequential"

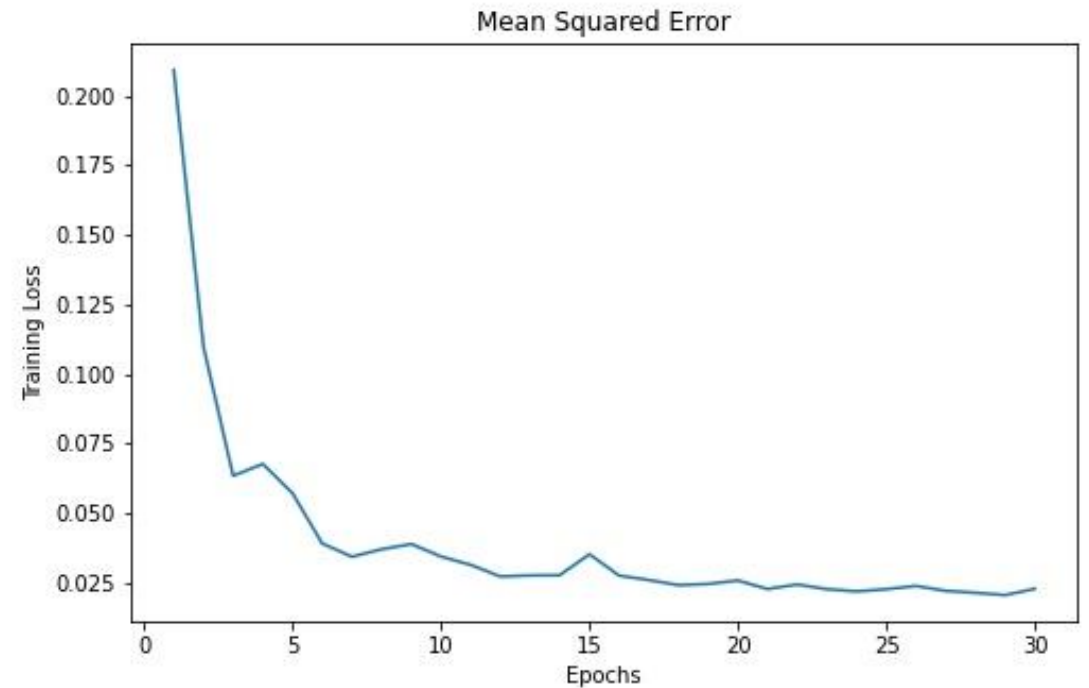
Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 198, 198, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 99, 99, 64)	0
conv2d_1 (Conv2D)	(None, 97, 97, 64)	36928
max_pooling2d_1 (MaxPooling2	(None, 48, 48, 64)	0
conv2d_2 (Conv2D)	(None, 46, 46, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 23, 23, 128)	0
conv2d_3 (Conv2D)	(None, 21, 21, 128)	147584
max_pooling2d_3 (MaxPooling2	(None, 10, 10, 128)	0
flatten (Flatten)	(None, 12800)	0
dense (Dense)	(None, 512)	6554112
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 64)	32832
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65
=====		

Total params: 6,847,169
Trainable params: 6,847,169
Non-trainable params: 0

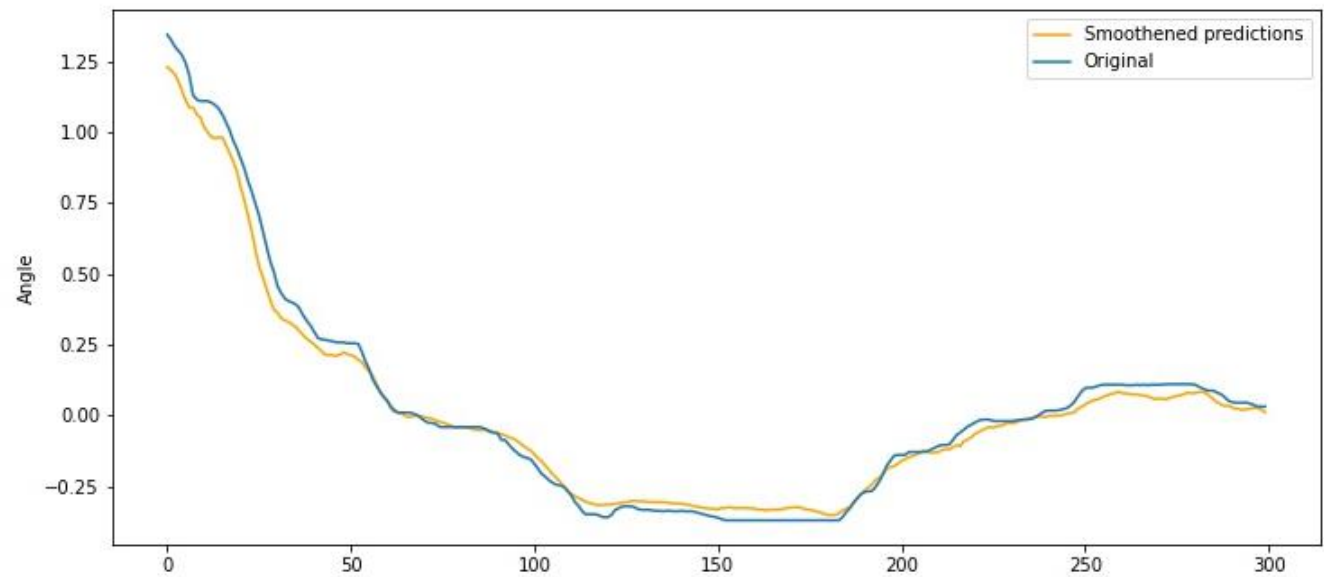
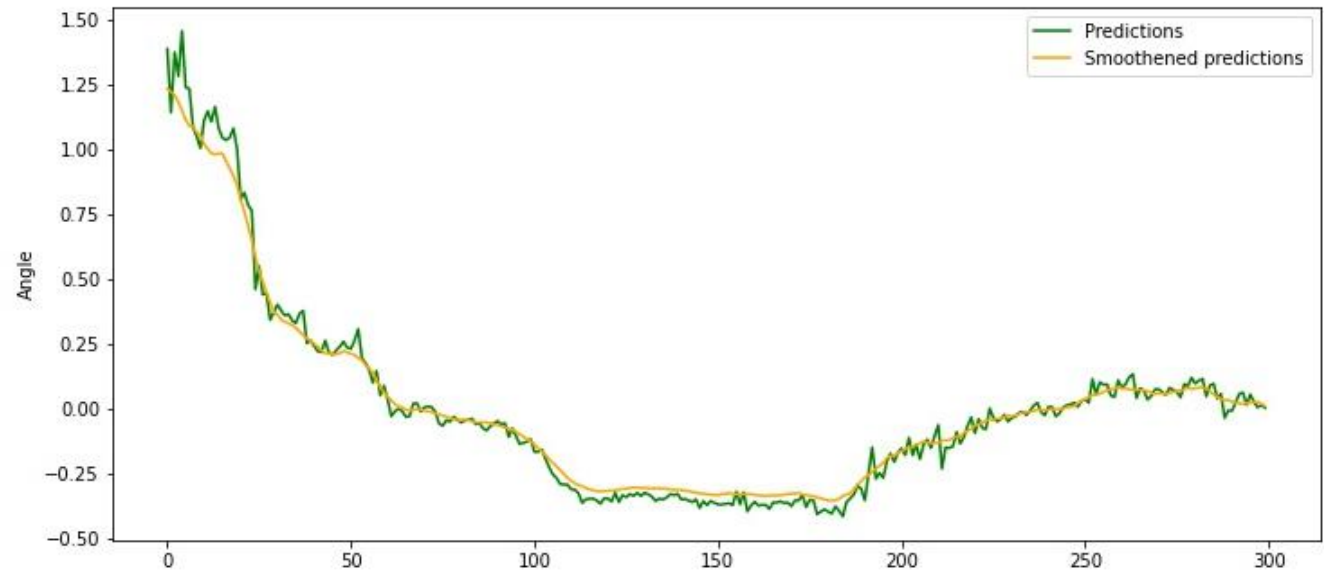
TRAINING THE MODEL

Parameters:

- Optimizer = Adam
- Learning Rate = 0.001
- Decaying learning rate
- Loss = Mean Squared Error (MSE)
- Epochs = 30



PREDICTION PLOTS





DEMO

The image features a large white circle centered on a solid orange background. A dashed orange line, composed of several short segments, curves around the upper-left portion of the white circle. A solid orange dot is positioned on the lower-right edge of the white circle.

THANK YOU.