

# **AIML**

---

## CAPSTONE PROJECT

# COMPUTER VISION

OBJECT DETECTION - CAR

# PROBLEM STATEMENT

- **DOMAIN:** Automotive. Surveillance.

- **CONTEXT:**

Computer vision can be used to automate supervision and generate action appropriate action trigger if the event is predicted from the image of interest. For example a car moving on the road can be easily identified by a camera as make of the car, type, colour, number plates etc.

- **DATA DESCRIPTION:**

The Cars dataset contains 16,185 images of 196 classes of cars. The data is split into 8,144 training images and 8,041 testing images, where each class has been split roughly in a 50-50 split. Classes are typically at the level of Make, Model, Year, e.g. 2012 Tesla Model S or 2012 BMW M3 coupe.

**Data description:**

- **Train Images:** Consists of real images of cars as per the make and year of the car.
- **Test Images:** Consists of real images of cars as per the make and year of the car.
- **Train Annotation:** Consists of bounding box region for training images.
- **Test Annotation:** Consists of bounding box region for testing images.

Dataset has been attached along with this project. Please use the same for this capstone project.

**Dataset:** <https://drive.google.com/drive/folders/1y6JWx2CpsOuka00uePe72jNgr7F9sK45?usp=sharing>

Original dataset link for your reference only: <https://www.kaggle.com/jutrera/stanford-car-dataset-by-classes-folder>

**Reference:** 3D Object Representations for Fine-Grained Categorisation, Jonathan Krause, Michael Stark, Jia Deng, Li Fei-Fei 4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13). Sydney, Australia. Dec. 8, 2013.

- **PROJECT OBJECTIVE:** Design a DL based car identification model.

- **PROJECT TASK:** [ Duration: 6 weeks, Score: 100 points]

1. **Milestone 1:** [ Duration: 2 weeks, Score: 20 points]

- **Input:** Interim report
- **Process:**
  - Step 1: Import the data
  - Step 2: Map training and testing images to its classes.
  - Step 3: Map training and testing images to its annotations.
  - Step 4: Display images with bounding box
- **Output:** Images mapped to its class and annotation ready to be used for deep learning

2. **Milestone 2:** [ Duration: 2 weeks, Score: 20 points]

- **Input:** Output of milestone 1
- **Process:**
  - Step 1: Design, train and test CNN models to classify the car.
  - Step 2: Design, train and test RCNN & its hybrids based object detection models to impose the bounding box or mask over the area of interest.
- **Output:** Pickled model to be used for future prediction
- **Submission:** Interim report

3. **Milestone 3:** [ Duration: 2 weeks, Score: 60 points]

- **Input:** Pickled model from milestone 2
- **Process:** [ 15 points ]
  - Step 1: Design a clickable UI which can automate tasks performed under milestone 1 [ 5 points ]
  - Step 2: Design a clickable UI which can automate tasks performed under milestone 2 [ 5 points ]
  - Step 3: Design a clickable UI based interface which can allow the user to browse & input the image, output the class and the bounding box or mask [ highlight area of interest ] of the input image [ 5 points ]
- **Output:** Clickable UI based interface
- **Submission:** Final report [ 45 points ]

- References and hints:

- Please refer to the blog to understand the basics of object detection: <https://www.mygreatlearning.com/blog/object-detection-using-tensorflow/>
- YOLO object detection using OpenCV: <https://www.mygreatlearning.com/blog/yolo-object-detection-using-opencv/?highlight=detection>
- Face detection: <https://www.mygreatlearning.com/blog/face-recognition/?highlight=detection>
- To make GUI as a desk app you can use TKINTER library. You are free to use any other library.
- To make web service GUI you can use FLASK or DJANGO library.