|  |
| --- |
| aiml Capstone project |
|  |
| Computer Vison Car Detection |
|  |
|  |
|  |

Project Member

|  |  |  |
| --- | --- | --- |
| Sr No | Name | Email ID |
| 1 | Rupali Botre | botrerupalid@gmail.com |
| 2 | Bama Charan Kundu | bamachrn@gmail.com |
| 3 | Karthikeyan C |  |
| 4 | Chanchal Singh Adhikari |  |

**TABLE OF CONTENTS**

[1. Project Description 4](#_Toc64840461)

[2. Milestone 1 5](#_Toc64840462)

[2.1 Summary of Problem Statement 5](#_Toc64840463)

[2.2 Approach to EDA and Pre-processing 5](#_Toc64840464)

[2.3 Data and Findings 6](#_Toc64840465)

[3. Milestone 2 7](#_Toc64840466)

[3.1 RESNET 7](#_Toc64840467)

[3.2 Mobile Net 8](#_Toc64840468)

# Project Description

**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.

‣ **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. Original link to the dataset: <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.

# Milestone 1

## **Summary of Problem Statement**

The Stanford Car Dataset will be utilized to build a vehicle recognition predictive

model. The goal of the model is to classify a car’s year, make and model given an

input image.

## **Approach to EDA and Pre-processing**

* The Kaggle Stanford Cars Dataset contains total 16,185 images of cars. There are a

total of 196 classes of cars in this dataset. The data is split in half to be used as training and testing sets. The data also comes with class labels and bounding boxes for all images.

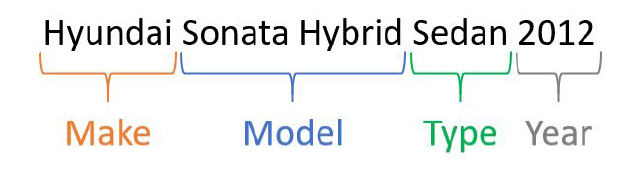
* The classes are typically at the level of Year, Make and Model (e.g. 2012 Tesla Model S or 2012 BMW M3 coupe). The sizes of each image are different. Utilization of the bounding boxes is essential in the pre-processing phase to first obtain images that focus on the objects of interest, which in this case are the vehicles. The actual images are in JPG format, but the data comes zipped in TGZ/TAR format.
* The Stanford Car Dataset will be utilized to build a vehicle recognition predictive model.
* The goal of the model is to classify a car’s year, make and model given an input image.
* The dataset contained no missing values, so no imputations or data removal was

required due to the nature of image data. In terms of Exploratory Data Analysis, the class labels were split to explore the individual Make, Model, Type and Year levels of the labels.

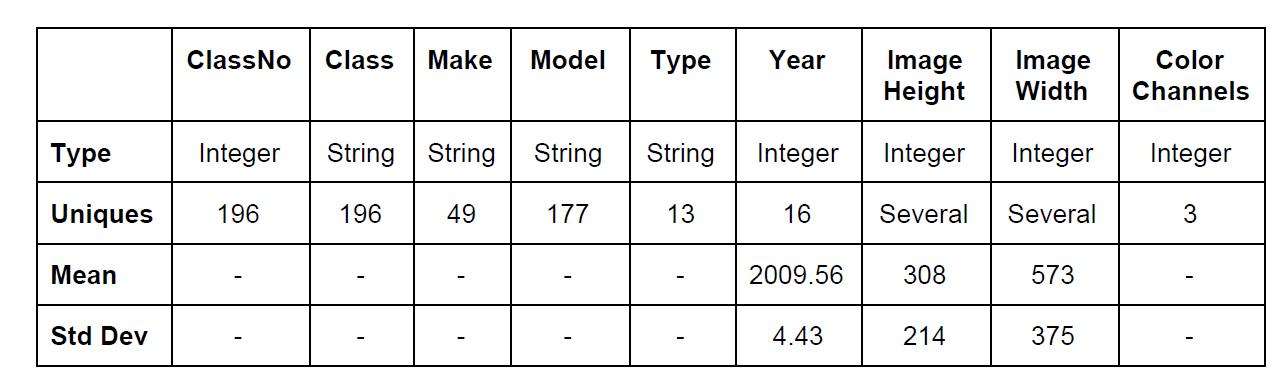
* The string-formatted labels were split by a space, then the output of that were categorized into the Make, Model, Type and Year levels. This was tricky, since some the Make and Model levels had different lengths (for instance, Aston Martin vs. BMW in the Make level and Sonata vs. F-450 Super Duty Crew in the Model level). This extraction of class label levels was performed to the best of our abilities.
* There were 196 classes originally. Because of this high total class number, the levels of class labels were analysed with the hopes of reducing the total class number. Initially the class labels were analyzed by human eyes.
* While the Stanford dataset contained pre-split training and testing data, 50:50 data.

## **Data and Findings**

* The original dataset defined a ‘class’ as the combination of make, model, and year.
* This yields 196 individual and unique classes. An example of one of these classes is shown in Figure. The class levels were parsed into the components also shown in Figure. It may be possible to extract more useful information by separating these characteristics.



* The following table provides specific descriptive statistics from the entire original dataset.
* The image dimensions (height, width, and channels) were added to support future modelling decisions. Due to the way that the image of this dataset was created, a thorough Exploratory Data Analysis of the original class distributions was highly desired.



# Milestone 2

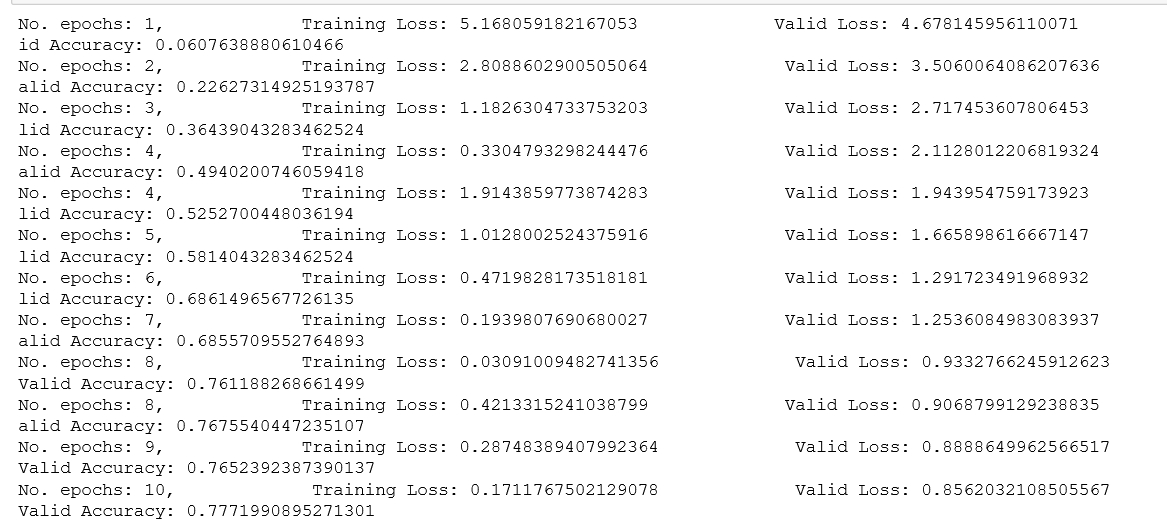
## **RESNET**

* The initial custom convolutional neural networks with 1~3 ConvNet layers (conv-conv-pool) with relu activations fully connected layers performed poorly. These custom CNNs was trained after tuning the learning rate according to the One-Cycle-Policy by Leslie and fit on regular Stanford Cars images (under-sampled). The 3rd model (3 ConvNet layers) had the highest validation accuracy of about 5% post learning rate tuning.
* These initial set of models illustrated that simple CNN models will not be able to perform well on this complex and non-structured car image data. Since these initial models did not perform well.
* Custom convolutional neural networks (CNNs) were focused on predicting the Vehicle Make level. The following plot shows the training and validation accuracies (red and

blue) along with the Top-5 accuracies (green and black). The top-5 accuracy scores show that the network has more success at proposing the five most-likely manufacturers instead of picking just one.

It may be possible to leverage this model as a type of autoencoder for a separate downstream model to make better classifications.

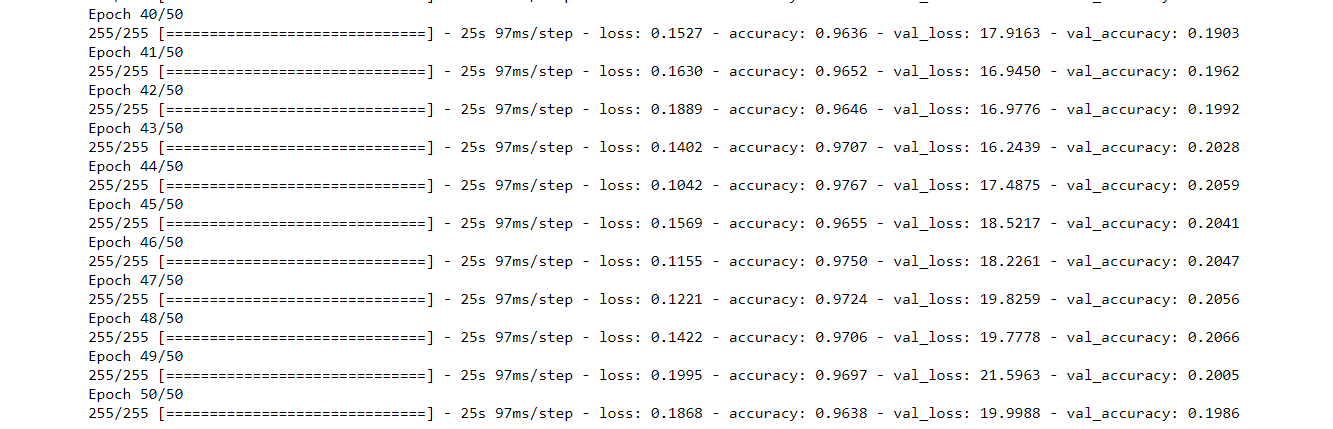
* Some model-specific image pre-processing is done prior to training. The model is trained in batches using augmented images since it is believed that there may not be enough data for the large number of target classes proposed. Image augmentation has greatly improved the rate at which models converge on a solution and the maximum accuracy overall.
* These models are trained on large visual databases, which are often used in object recognition. Since these pre-trained models are trained on large visual databases, the first step of implementing these models are adding custom fully connected layers. Once the top layers (FC layer) are added, the model has to be frozen (setting the convolution layers’ trainable status to False) in order to only train the top layers. This process makes these pre-trained models suitable for any image data.
* The initial setup for the ResNet34 model was done utilizing module. Which had multiple BatchNorms, Dropouts and Dense layer with linear and relu activations.



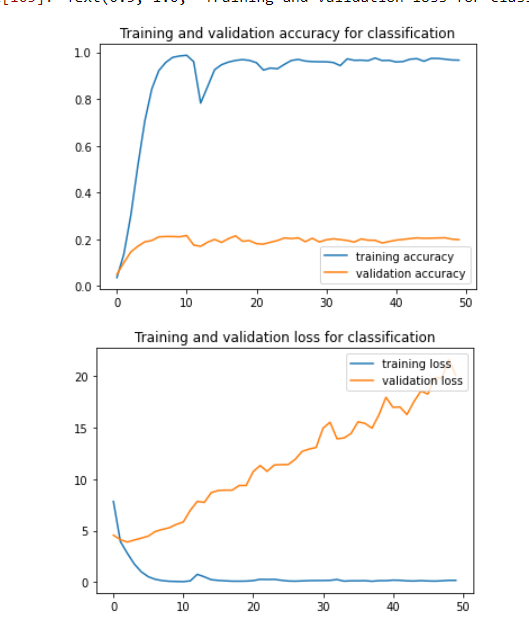
## **Mobile Net**

We shall be using Mobilenet as it is lightweight in its architecture. It uses depthwise separable convolutions which basically means it performs a single convolution on each colour channel rather than combining all three and flattening it.

* We have used the MobileNetV2 model with weights - ‘imagenet’ and Activation Function as Softmax.
* Used two Dense Layers 1024 Relu and 196 Sofmax.
* Created Bounding Box model.
* Created IOU function for finding out the difference between provided bounding box and predicted bounding box.
* For Bounding Box Model, we achieved 19% accuracy with 50 Epoch.



* We have achieved following Training Validation Accuracy and Loss as shown in following graph:



* We have used the MobileNetV2 model with weights from - ‘imagenet’ and Activation Function as Softmax.
* For bounding box we have taken the output of top layer from mobilenet
* Added two Layers on top of it
  + 2Dconv layer with kernel size 7X7 naming coords. This layers gives the output as (1,1,4) array
  + Reshape layer to generate co-ords for the bounding box.
* We have trained classification model with 10 epoch resulting maximum of 80% validation IoU and 92% training IoU.
* Created IOU function for finding out the difference between provided bounding box and predicted bounding box.
* We have got the following matrix for the IoU and loss