CAPSTONE PROJECT – COMPUTER VISION

BOUNDING BOX REGRESSION AND MULTICLASS CLASSIFICATION OF STANFORD CAR IMAGES

Submitted by - AIML Batch-B Jan, 21

Summary of problem statement, data and findings

The **Stanford Cars** dataset consists of 196 classes of cars with a total of 16,185 images. The data is divided into almost a 50-50 train/test split with 8,144 training images and 8,041 testing images. Categories are typically at the level of Make, Model, Year. The images are of varying dimensions

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.

Car names and make: CSV file consisting of Car Names and associated Class Numbers

Test and Train Annotations: CSV files consisting of Image Names, Bounding box coordinates, and associated Class Numbers

Data loading activity:

1. Read and unzip the image files – There are 196 folders in each Train and Test datasets, each class has 30-50 images which is fairly well balanced dataset.
2. Annotations CSV – read and uploaded into pandas dataframe object.
3. Dataframe object columns for bounding boxes are renamed to StartX, StartY, EndX, EndY
4. Car Name and Make CSV – Read and uploaded into pandas dictionary object
5. Dataframe object is added with new columns with Classname from dictionary object. 1 more column added with the images file path information.
6. 4 more columns are added to the Dataframe, these are the scaled values of the bounding box coordinates relative to the spatial dimensions of the input image
7. Load the images and preprocess it to 224,224 size to prepare it for model's training input dimensions
8. Using numpy list to update image data, class labels, bounding boxes, and image paths
9. Using LabelBinarizer – to perform one-hot encoding on the labels
10. Then we convert the data, class labels, bounding boxes, and image paths to NumPy arrays, scaling the input pixel intensities from the range [0, 255] to [0, 1]

Summary of the Approach to EDA and Pre-processing

Deciding Models and Model Building

How to improve your model performance?