Machine Learning Capstone Project

Smart Car

Capstone Proposal Overview

Creating accurate machine learning models capable of localizing and identifying multiple objects in a single image or video remains a core challenge in computer vision.

Domain Background

college research for detect objects on videos or images inside the college to use it to get information about the place in real time detection with smart car embedded system that will detect.

Problem Statement

learning smart car how to detect and define objects when it's move in the real time inside the college and tell us about it.

Data Sets and Inputs

we will download the model which is trained on the **COCO dataset**. COCO stands for **Common Objects in Context,** this dataset contains around 330K labeled images.
The input cases:
- images from the college.
- real time capturing inside college.
- videos from the college.

Solution Statement

The Tensor Flow Object Detection API is an
open source framework built on top of Tensor Flow that makes it easy to construct, train and
deploy object detection models.

Object Detection can be done via multiple ways:

- Feature-Based Object Detection
- Viola Jones Object Detection
- SVM Classifications with HOG Features
- Deep Learning Object Detection

In this Object Detection project, we'll focus on **Deep Learning Object Detection** as Tensor flow uses Deep Learning for computation.

```
MODEL_NAME =
'ssd_mobilenet_v1_coco_11_06_2017'
```

Benchmark Model

smart car that can detect objects and label them with accuracy up to 95%.

- This code runs the inference for a single image, where it detects the objects, make boxes and provide the class and the class score of that particular object.

Evaluation Metrics

tf.metrics.precision

Computes the precision of the predictions with respect to the labels.

The `precision` function creates two local variables, `true_positives` and `false_positives`, that are used to compute the precision. This value is ultimately returned as `precision`, an idempotent operation that simply divides `true_positives` by the sum of `true_positives` and `false_positives`.

For estimation of the metric over a stream of data, the function creates an `update_op` operation that updates these variables and returns the `precision`. `update_op` weights each prediction by the corresponding value in `weights`.

Project Design

The general sequence of steps are as follows:

a. **Data Visualization:** Visual representation of data to find the degree of correlations between predictors and target variable and find out correlated predictors. Additionally, we can see ranges and visible patterns of the predictors and target variable.

b. **Data Preprocessing:** Scaling and Normalization operations on data and splitting the data in training, validation and testing sets.

c. **Feature Engineering:** Finding relevant features, engineer new features using methods like PCA if feasible.

d. **Model Selection:** Experiment with various algorithms to find out the best algorithm for this use case.

e. **Model Tuning:** Fine tune the selected algorithm to increase performance without overfitting.

f. **Testing:** Test the model on testing dataset.